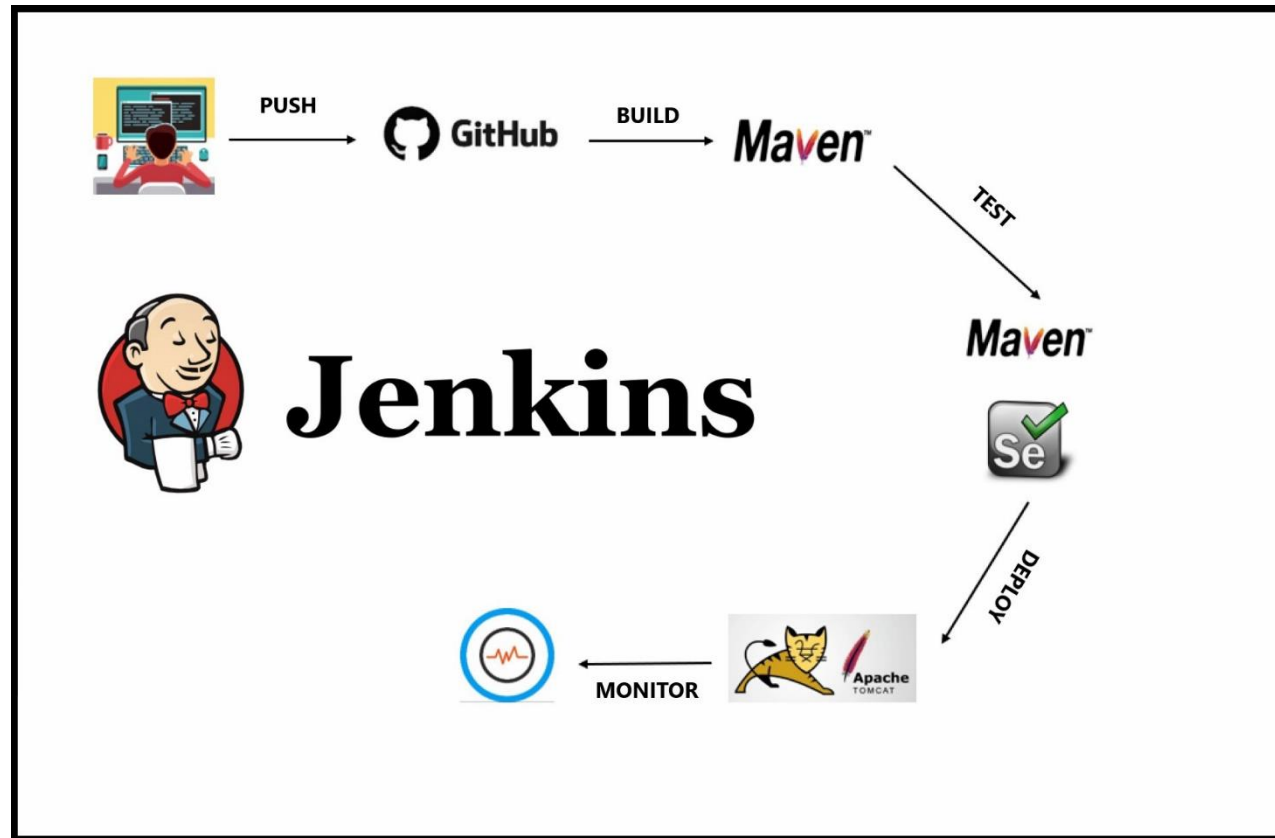


젠킨스(Jenkins)

젠킨스(Jenkins)

- 자바로 작성된 지속적인 통합을 위한 오픈소스 소프트웨어
- 지속적인 통합과 지속적인 전달을 포함한 모든 자동화에 중점을 둔 소프트웨어

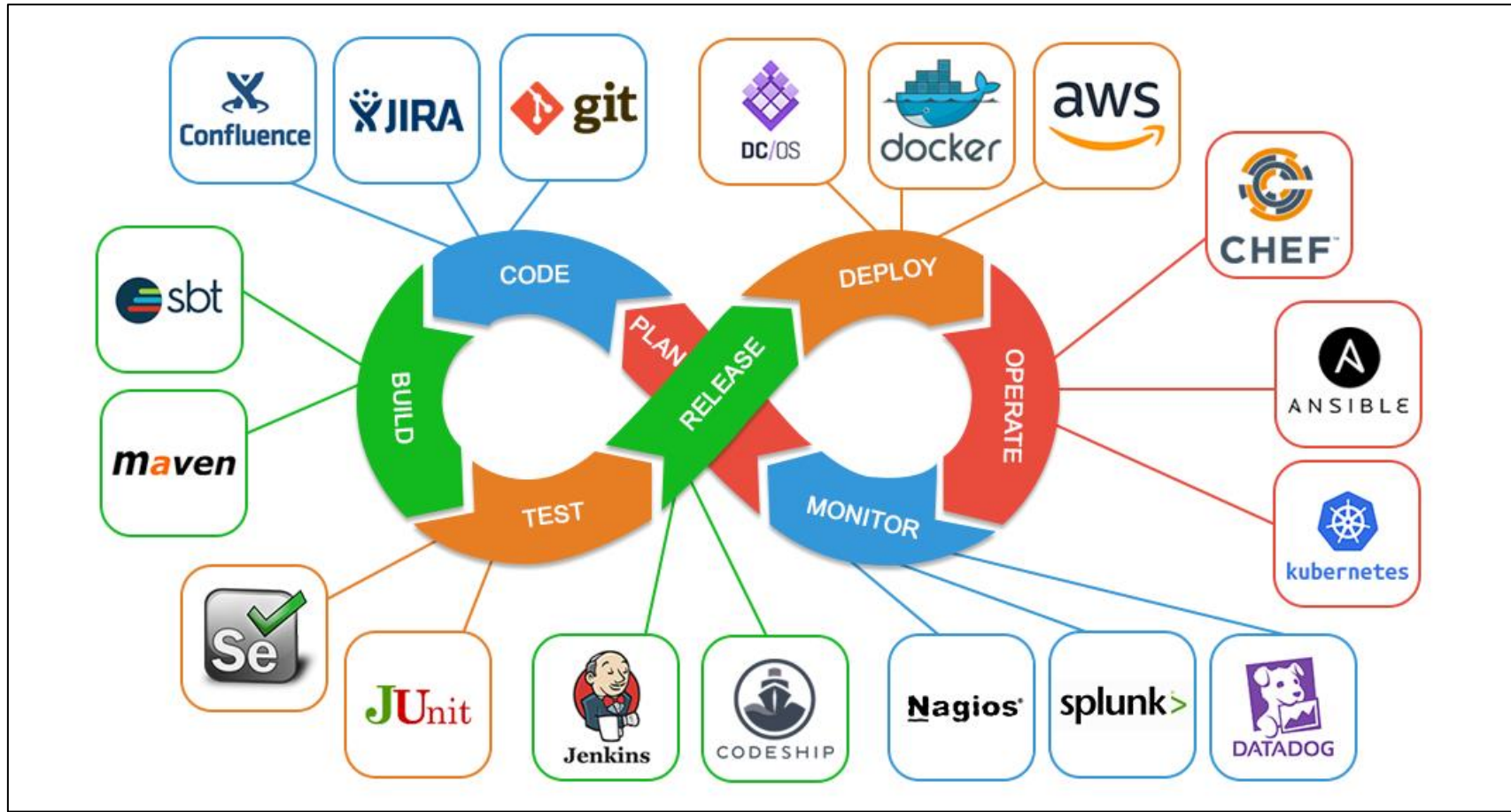


젠킨스(Jenkins)

- 다양한 플랫폼에서 사용 가능
- 애자일 환경에서 지속적인 통합 서비스로 활용
- 배치 명령, 셸 스크립트를 지원함.
- 다양한 플러그인 제공
 - 소스코드 관리 : Git, CVS
 - 빌드 트리거 : Accelerated Build Now, Build Flow
 - 빌드 보고서: Code Scanner, Disk Usage
 - 인증, 사용자 : Active Directory, GitHub OAuth
 - 클러스터 관리, 분산 빌드 : Amazon EC2, Azure Slave

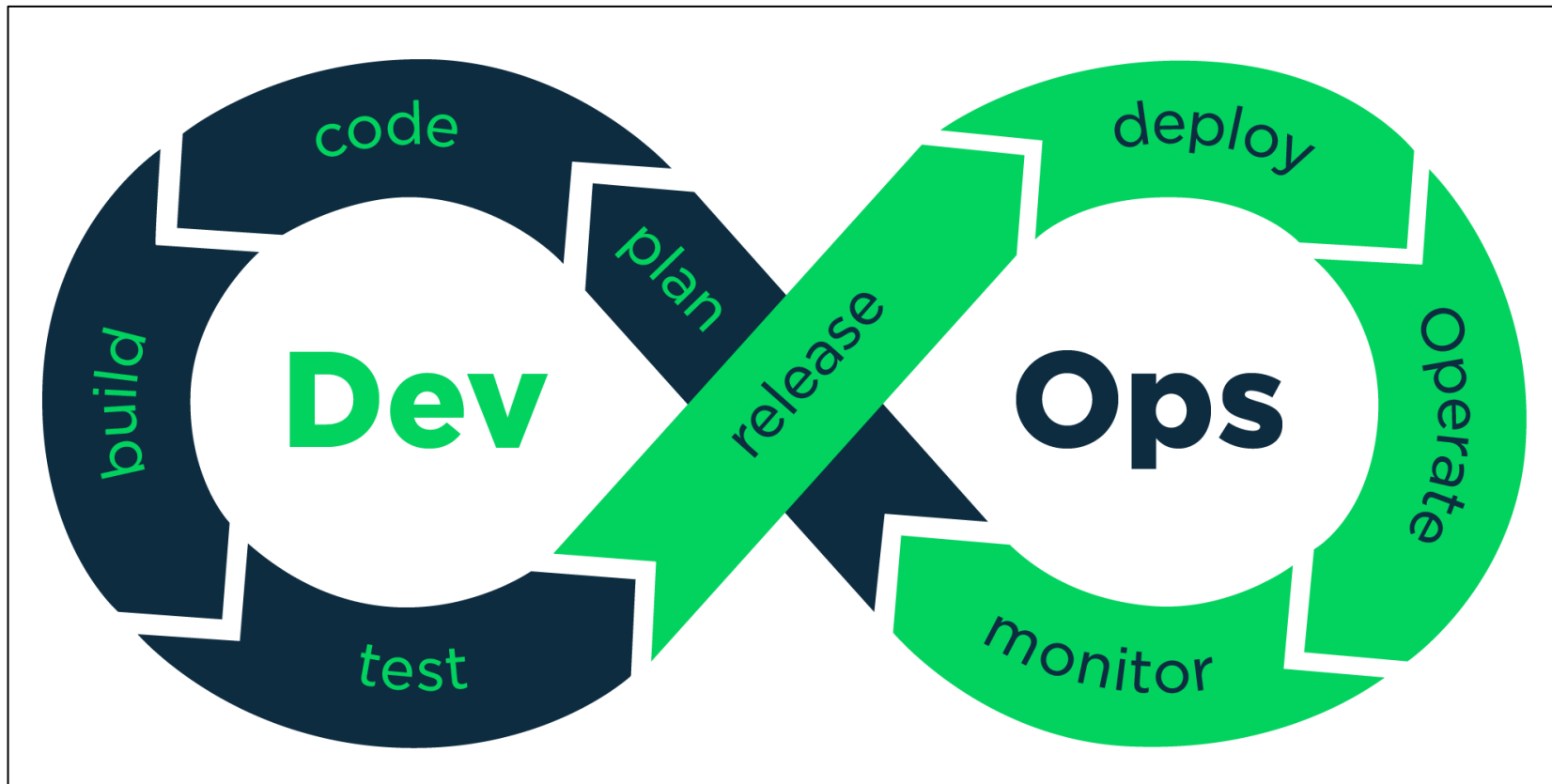
젠킨스(Jenkins)

- 각 개발 프로세스 별 사용 도구



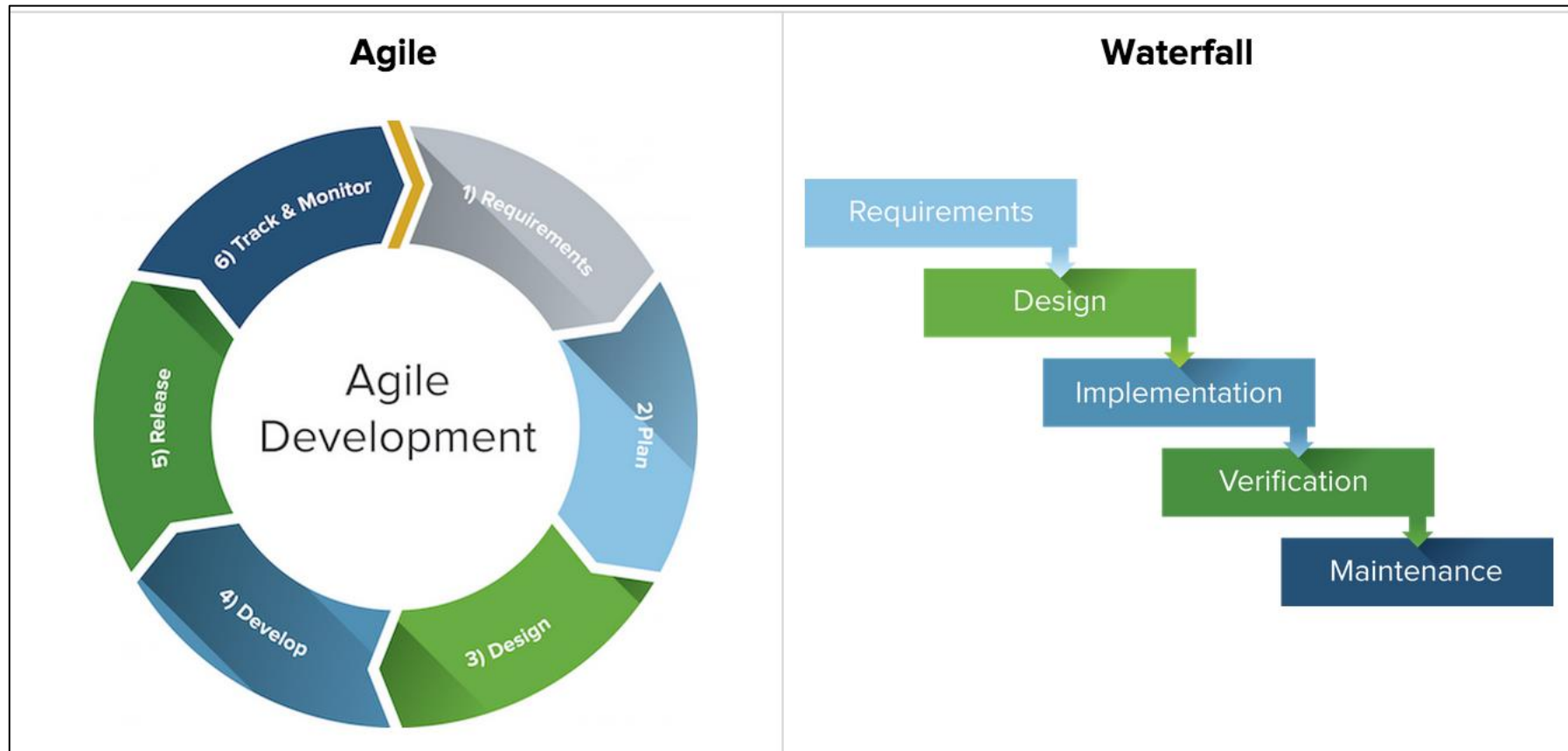
데브옵스(DevOps)

- 개발팀과 운영팀의 통합
- 변화 속도에 맞게 관리 및 자동화 구축
- 빠른 배포를 요구하는 환경



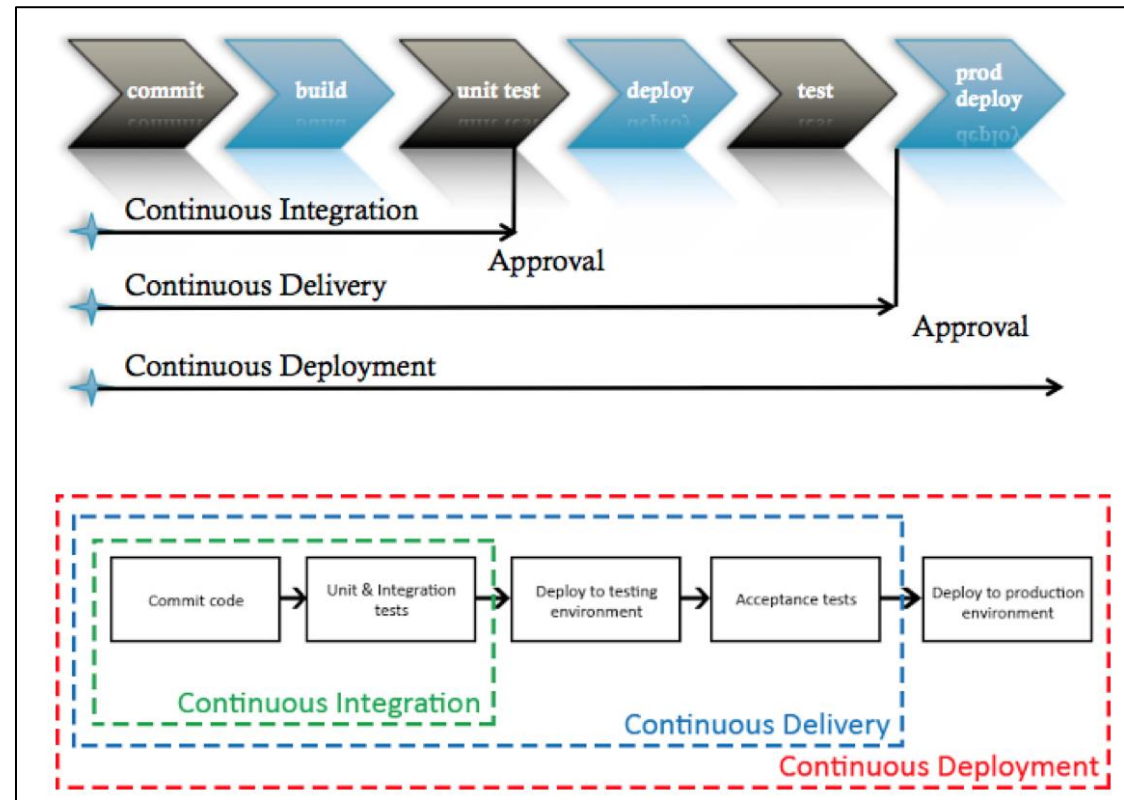
애자일과 폭포수

- 폭포수 모델
 - 순차적인 프로세스
 - 통제 및 관리



지속적인 통합/지속적인 전달(CI/CD)

- CI
 - 빌드 자동화, 단위테스트, 패키징 프로세스
 - 코드 퀄리티
- CD
 - 짧은 주기로 소프트웨어 개발에 배포 자동화
 - 빠른 문제 해결
 - 소모적인 시간 절약



실습

- 깃허브와 젠킨스 구성하기.
- 젠킨스와 Tomcat 서버를 같은 환경에 구성하기.
- 풀 메커니즘 : 예정된 시간에 자동 빌드하기.
- 젠킨스와 Tomcat서버를 서로 다른 환경에 구성하기.
- 푸시 메커니즘 : 변경사항이 저장소에 저장되는 경우 자동 빌드하기.

환경구성

- JDK 1.8 설정 및 Git 설치
 - 플러그인 지원 버전이 1.8버전
 - 11버전 이상 추천 플러그인 설치 미지원

```
C:\Windows\system32\cmd.exe
Microsoft Windows [Version 10.0.19044.1706]
(c) Microsoft Corporation. All rights reserved.

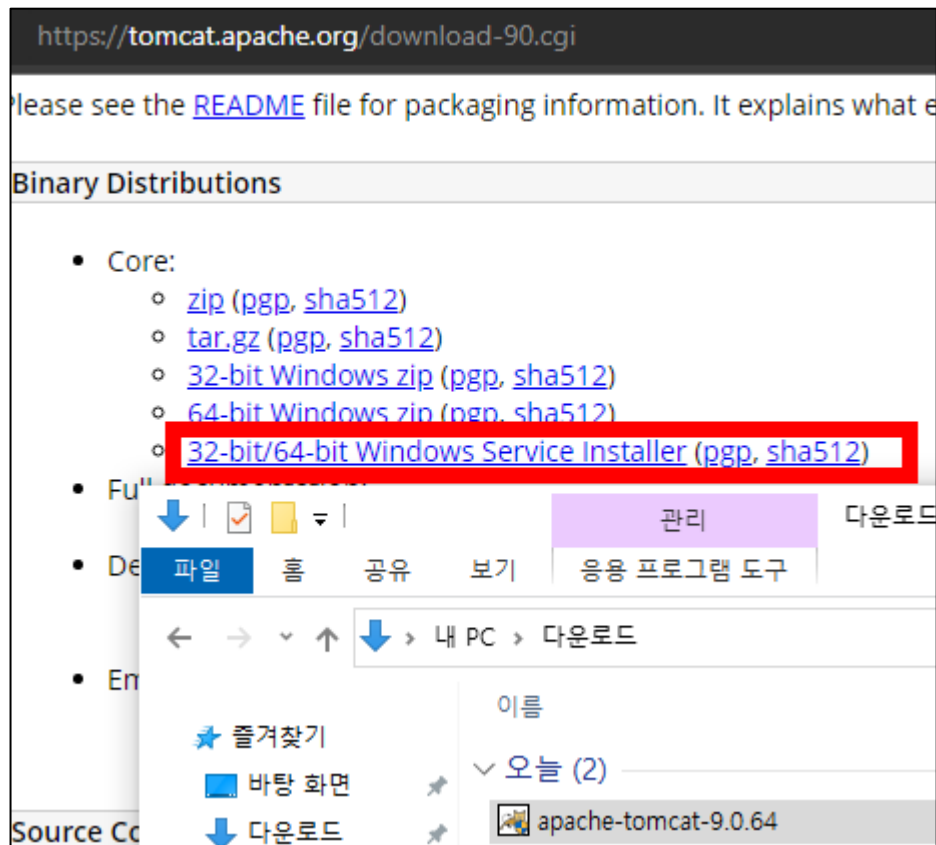
C:\Users\User>java -version
java version "1.8.0_311"
Java(TM) SE Runtime Environment (build 1.8.0_311-b11)
Java HotSpot(TM) 64-Bit Server VM (build 25.311-b11, mixed mode)
```

```
C:\Windows\system32\cmd.exe
Microsoft Windows [Version 10.0.19044.1706]
(c) Microsoft Corporation. All rights reserved.

C:\Users\User>git --version
git version 2.36.1.windows.1
```

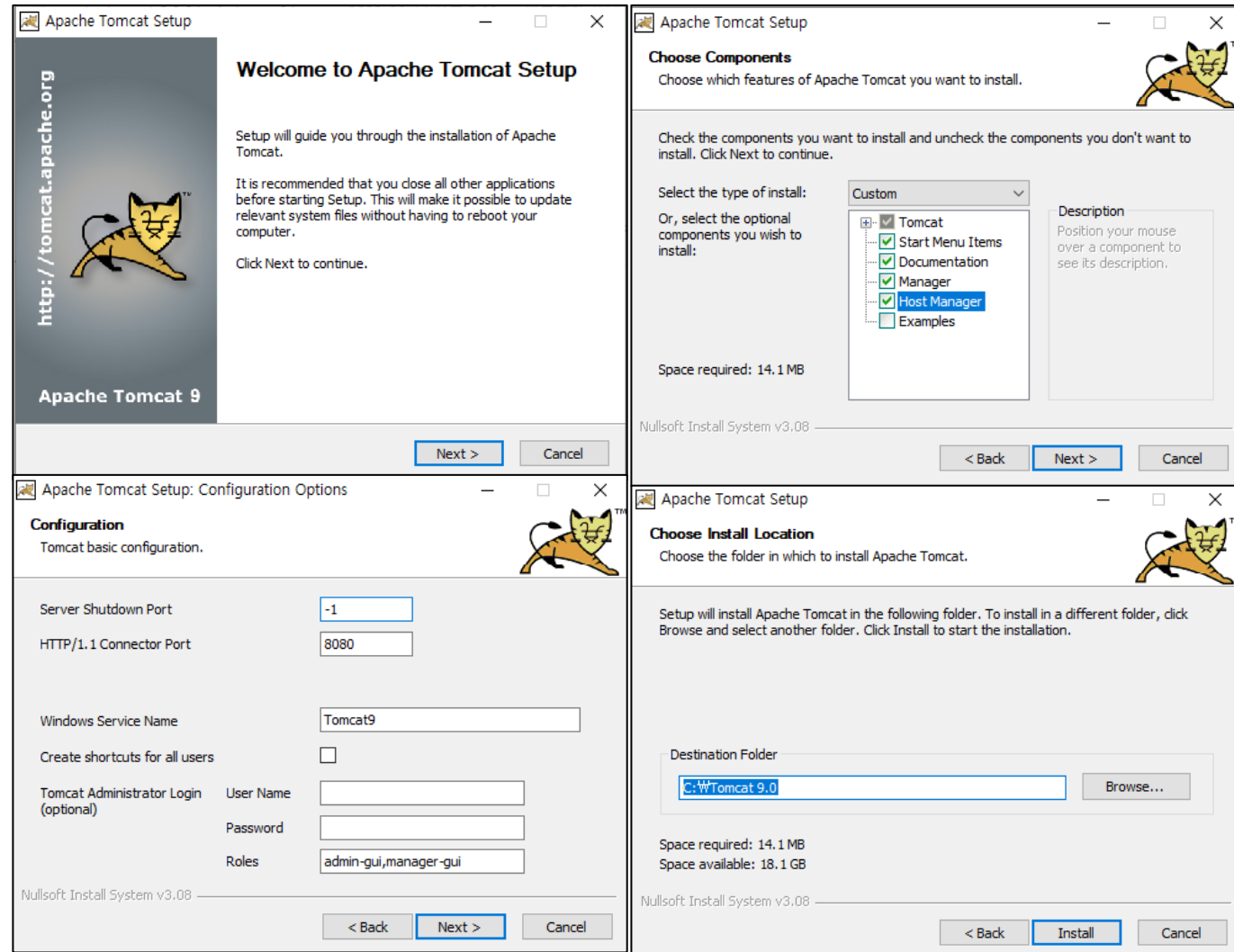
환경구성

- 톰캣 (로컬 배포)
 - <https://tomcat.apache.org/download-90.cgi>



환경구성

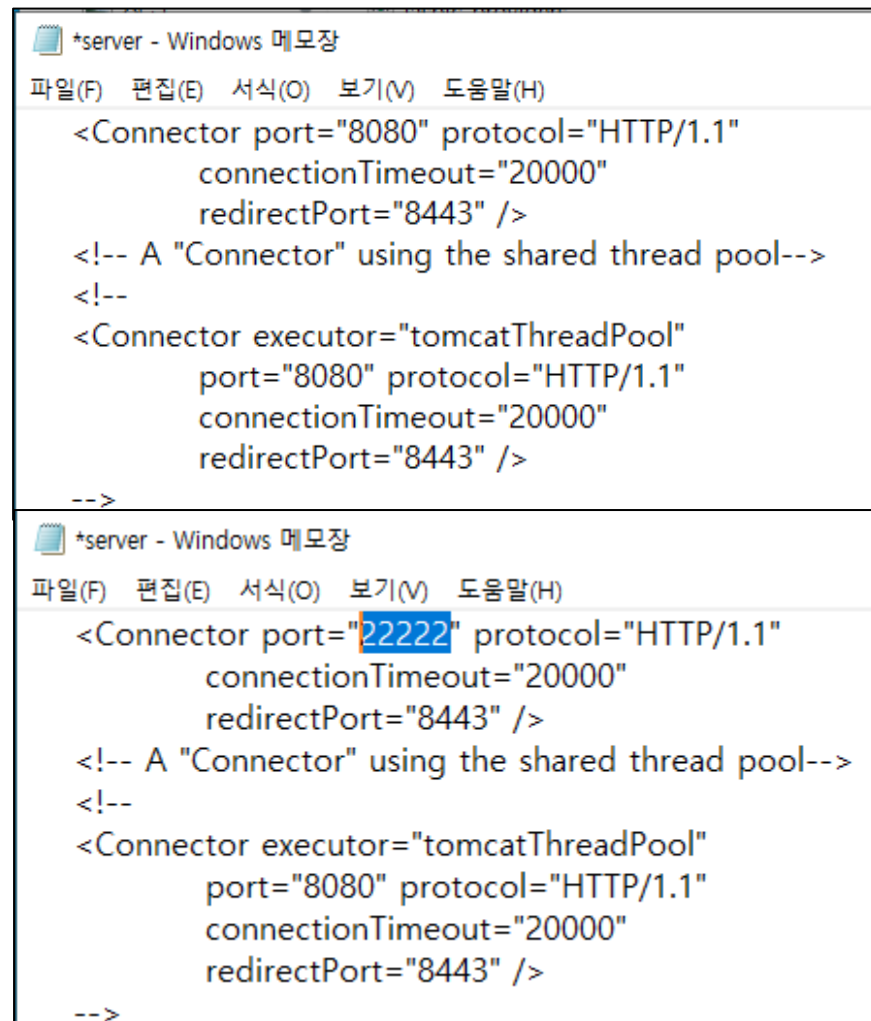
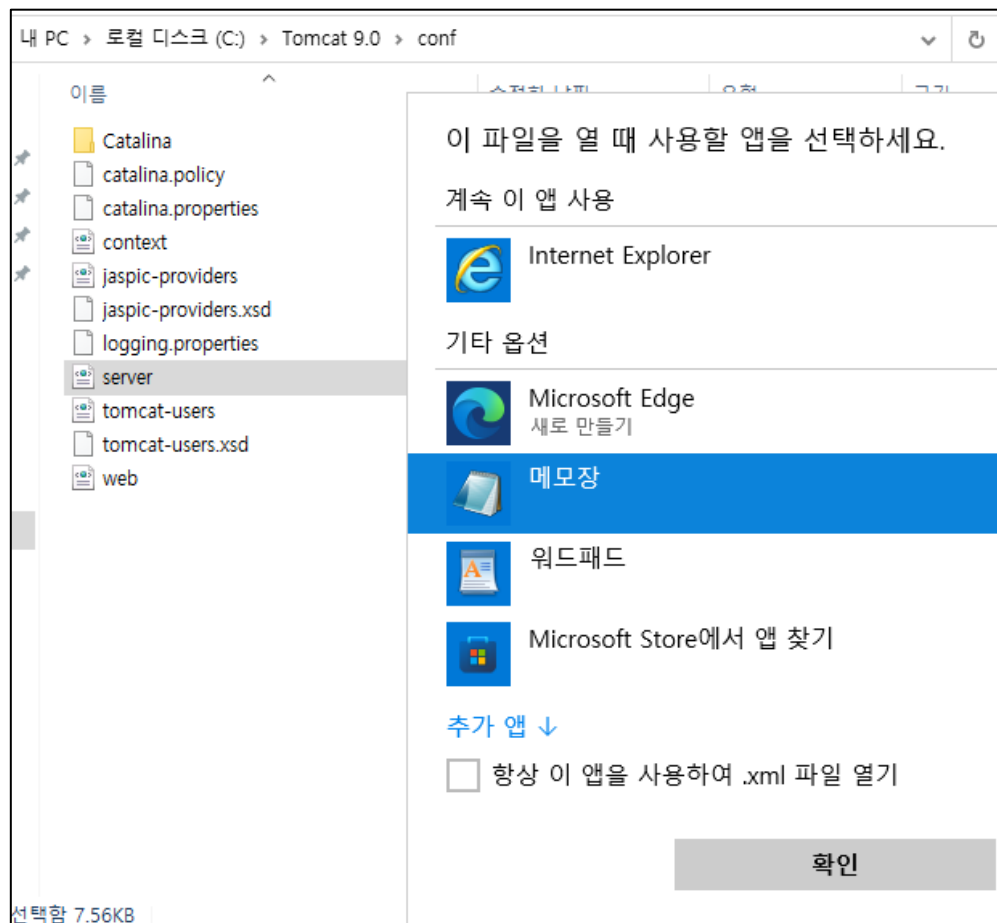
- 톰캣 설치



환경구성

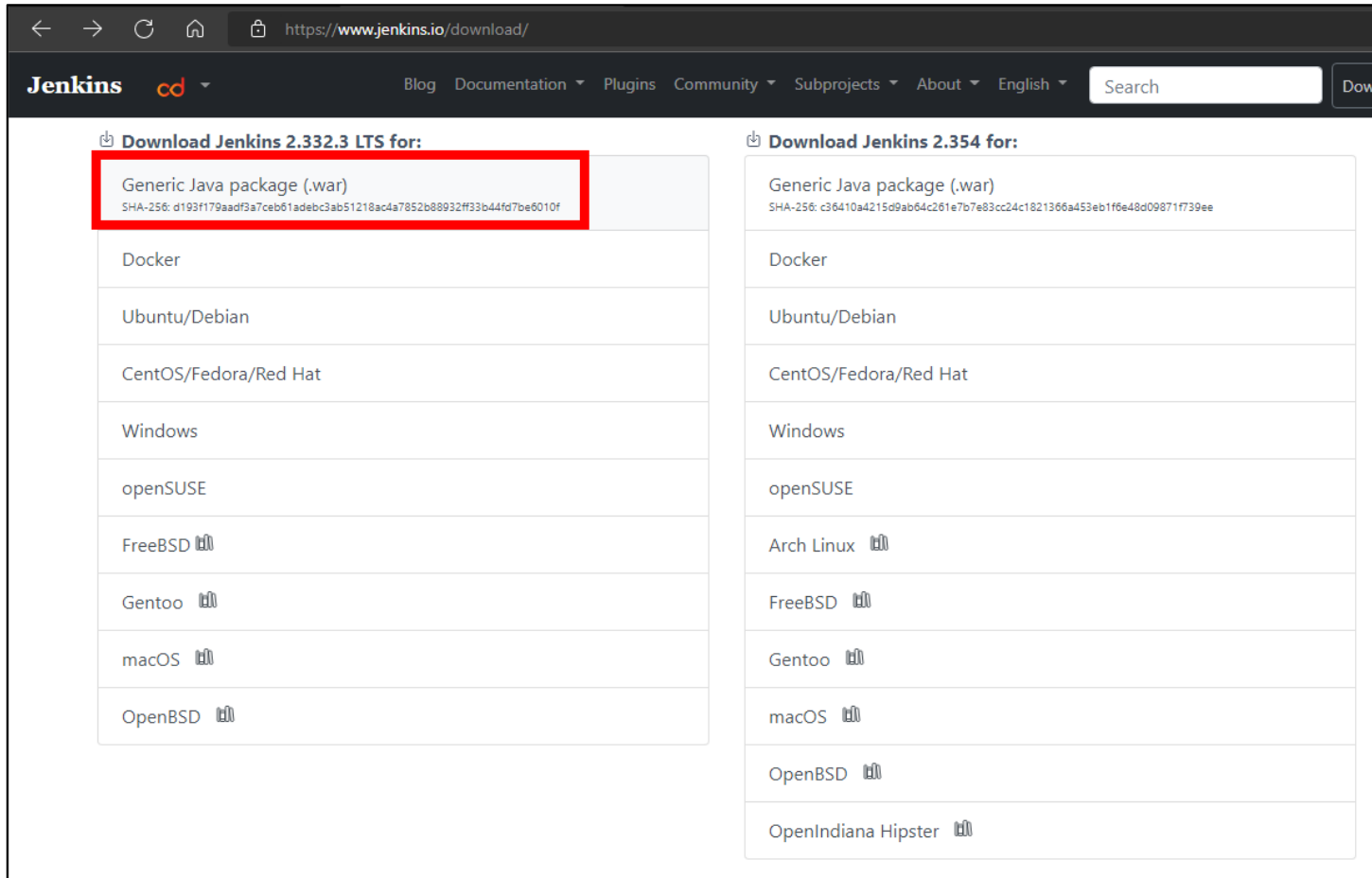
• 톰캣 설정

- conf 폴더 내 server.xml 파일을 메모장으로 실행.
- port="8080" 정보를 port="설정하고 싶은 포트" 로 변경.



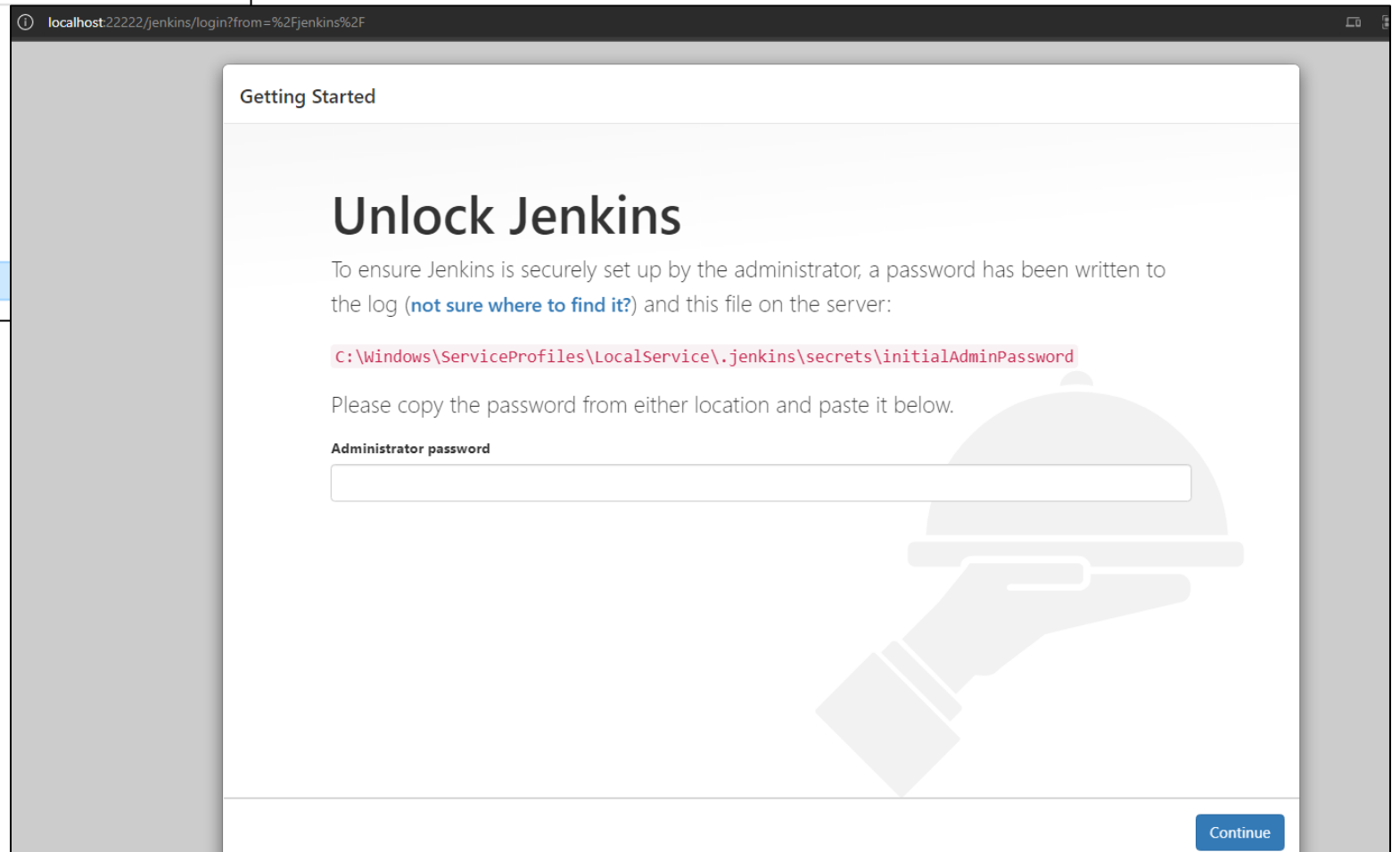
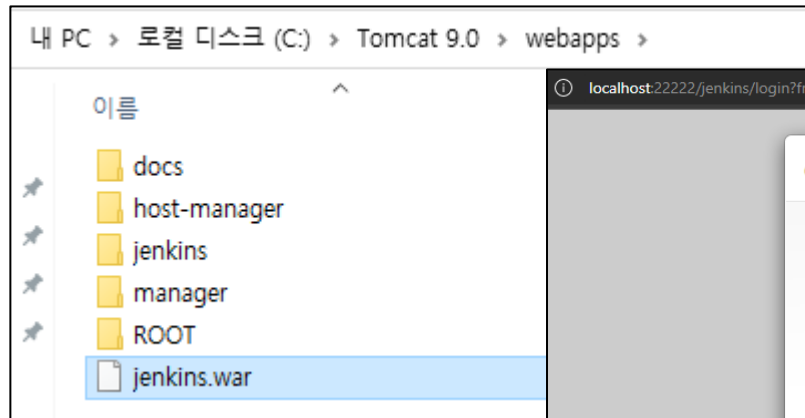
환경구성

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



환경구성

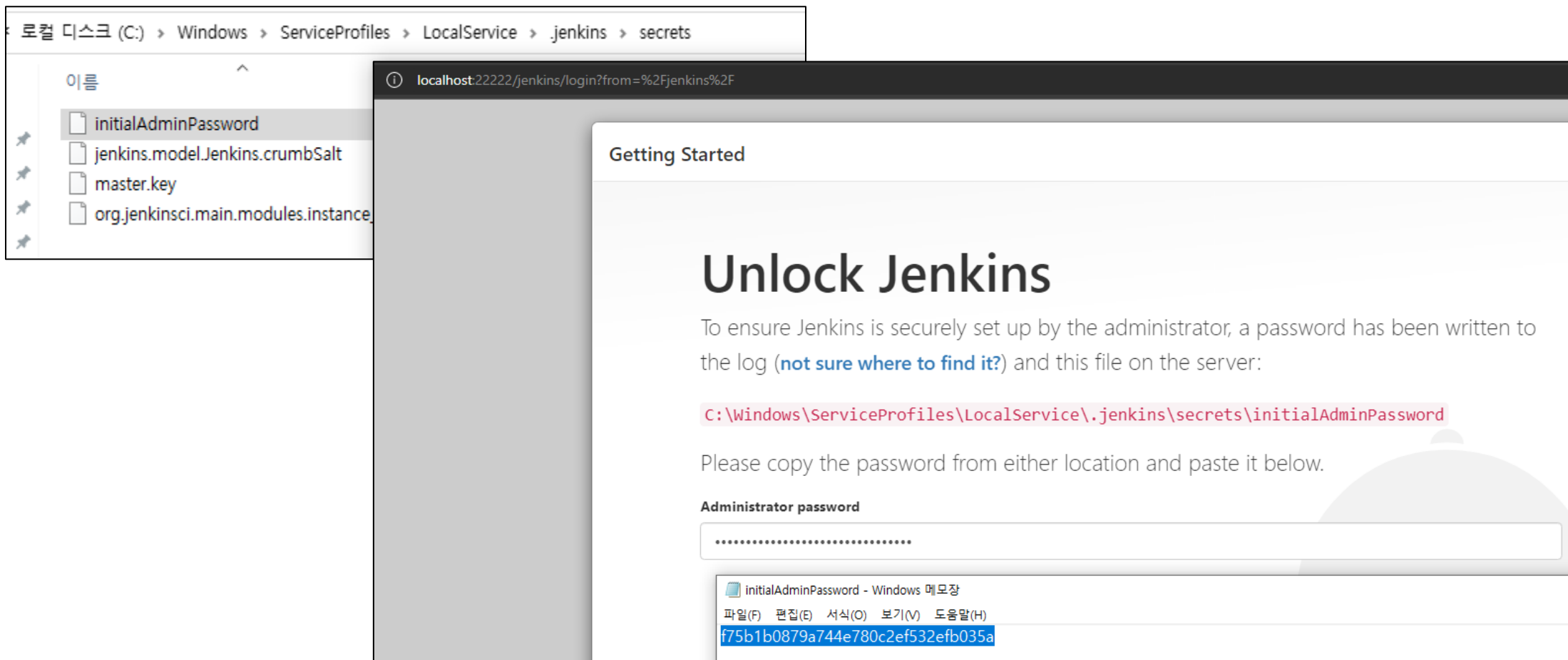
- 젠킨스 설치
 - 톰캣 폴더 내 webapps로 이동.
 - <http://localhost:22222/jenkins/> 접근



환경구성

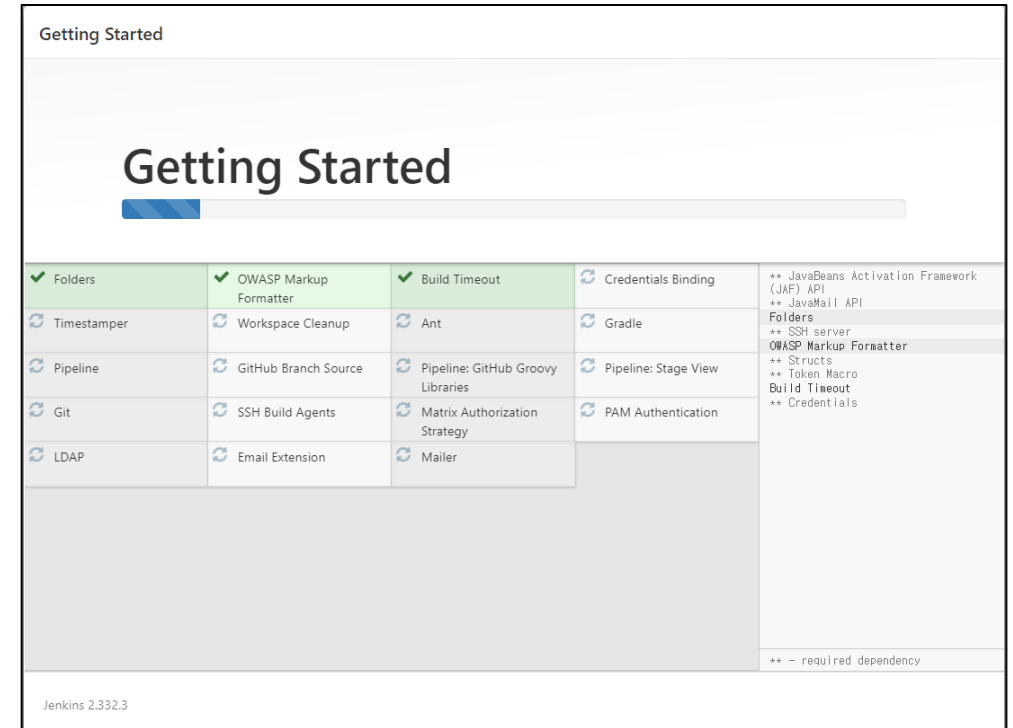
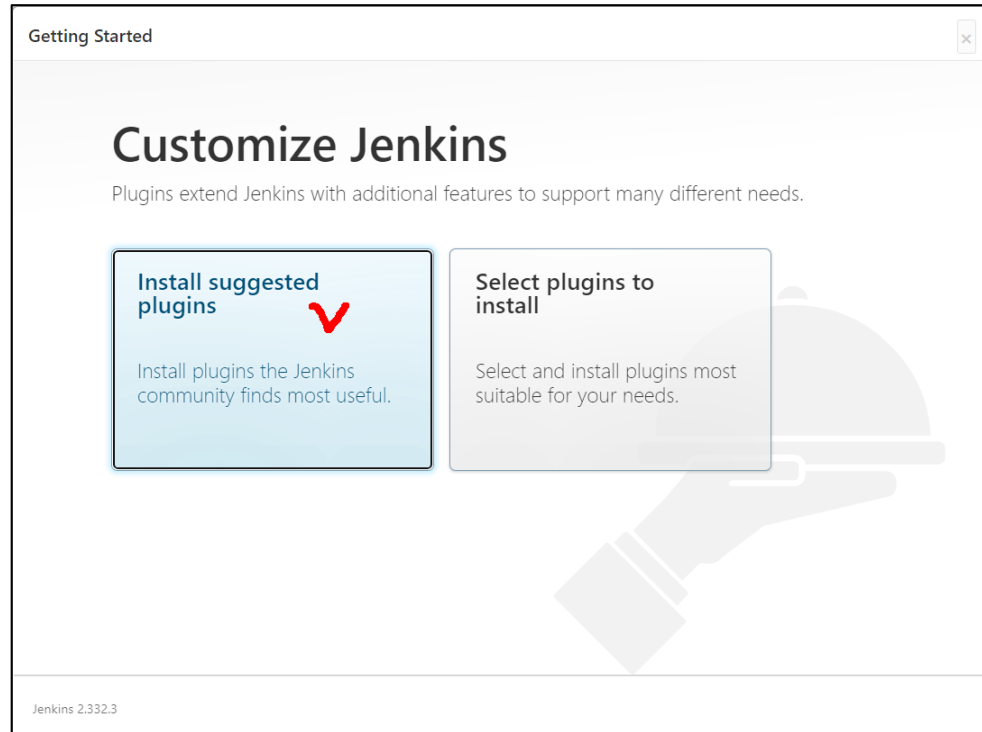
- 젠킨스 설치

- C:\Windows\ServiceProfiles\LocalService\.jenkins\secrets 폴더로 이동
- initialAdminPassword 파일을 메모장으로 실행 후 비밀번호 입력.



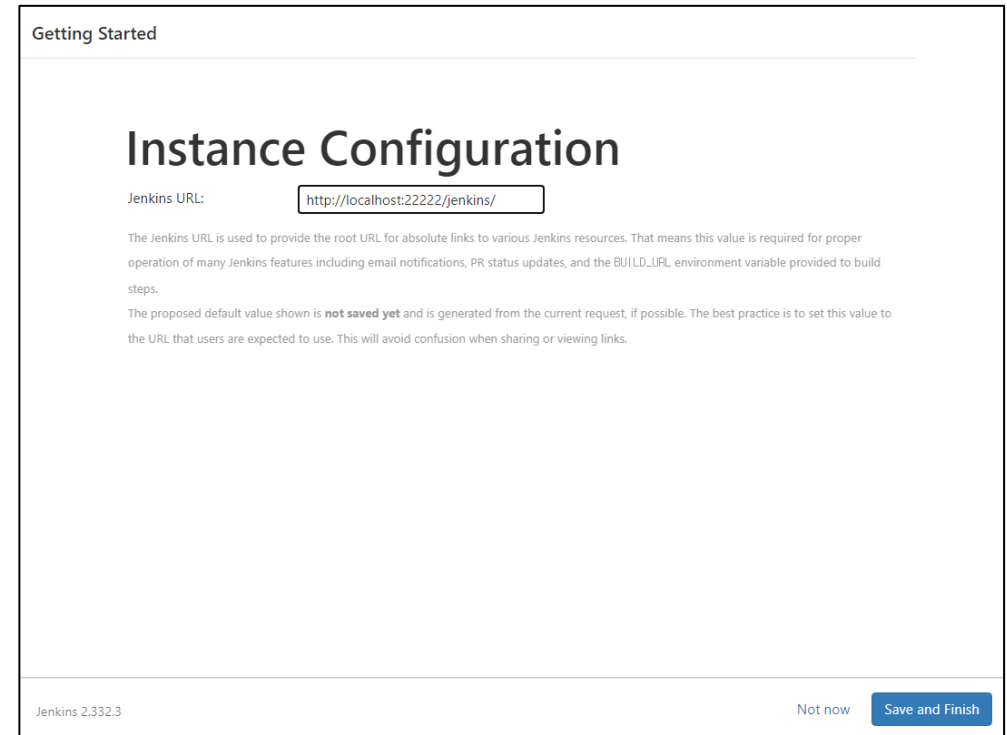
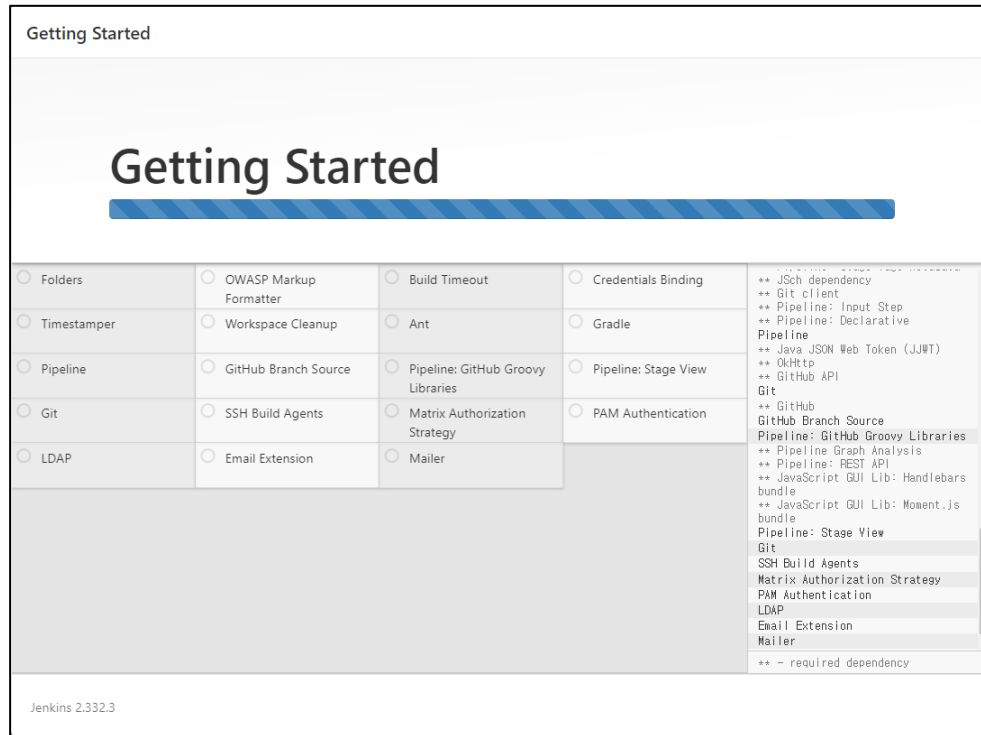
환경구성

- 젠킨스 설치
 - 추천하는 플러그인 설치(Install suggested plugins) 선택



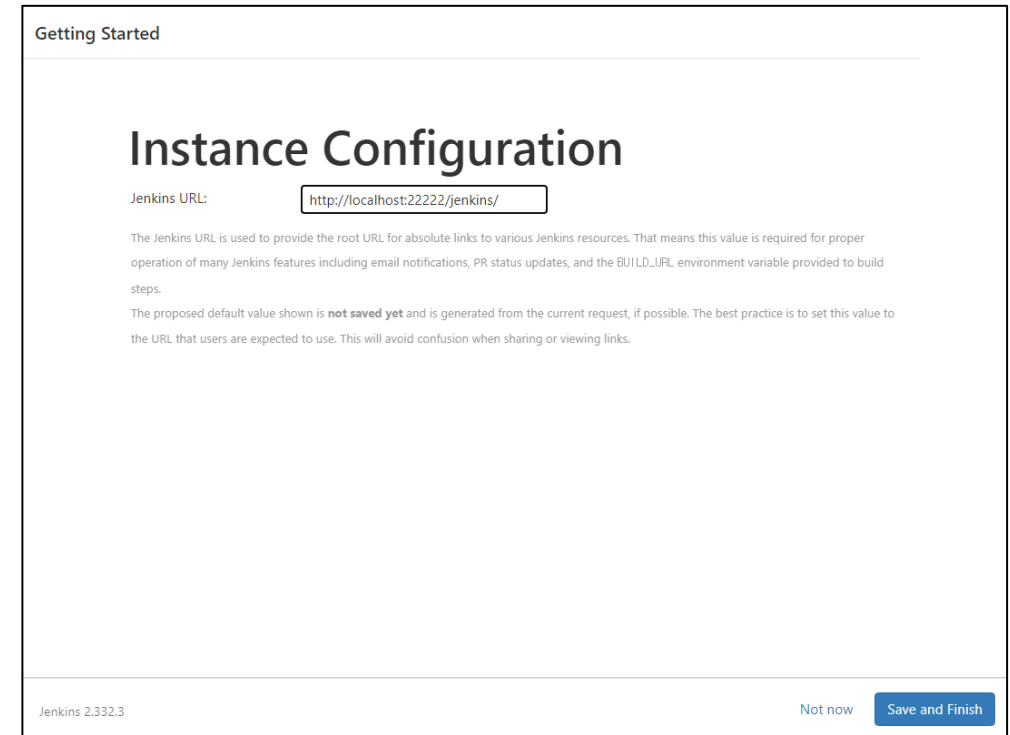
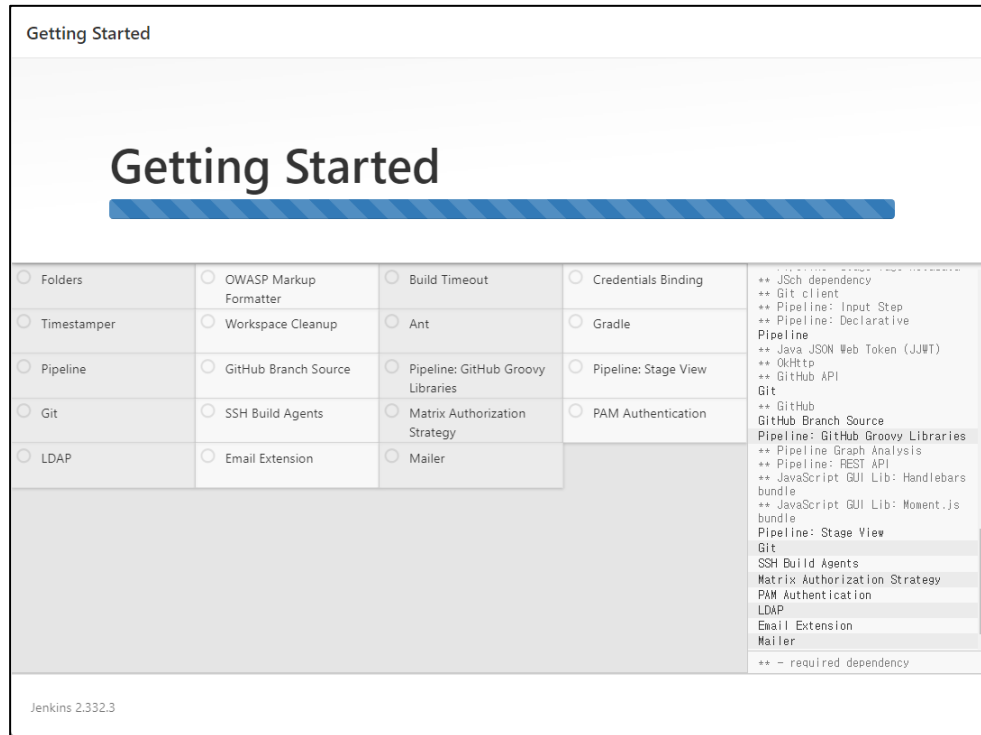
환경구성

- 젠킨스 설치
 - Save and Finish



환경구성

- 젠킨스 설치
 - Save and Finish



환경구성

- 젠킨스 설치
 - 계정명 : 로그인 시 사용

Getting Started

Create First Admin User

계정명:

암호:

암호 확인:

이름:

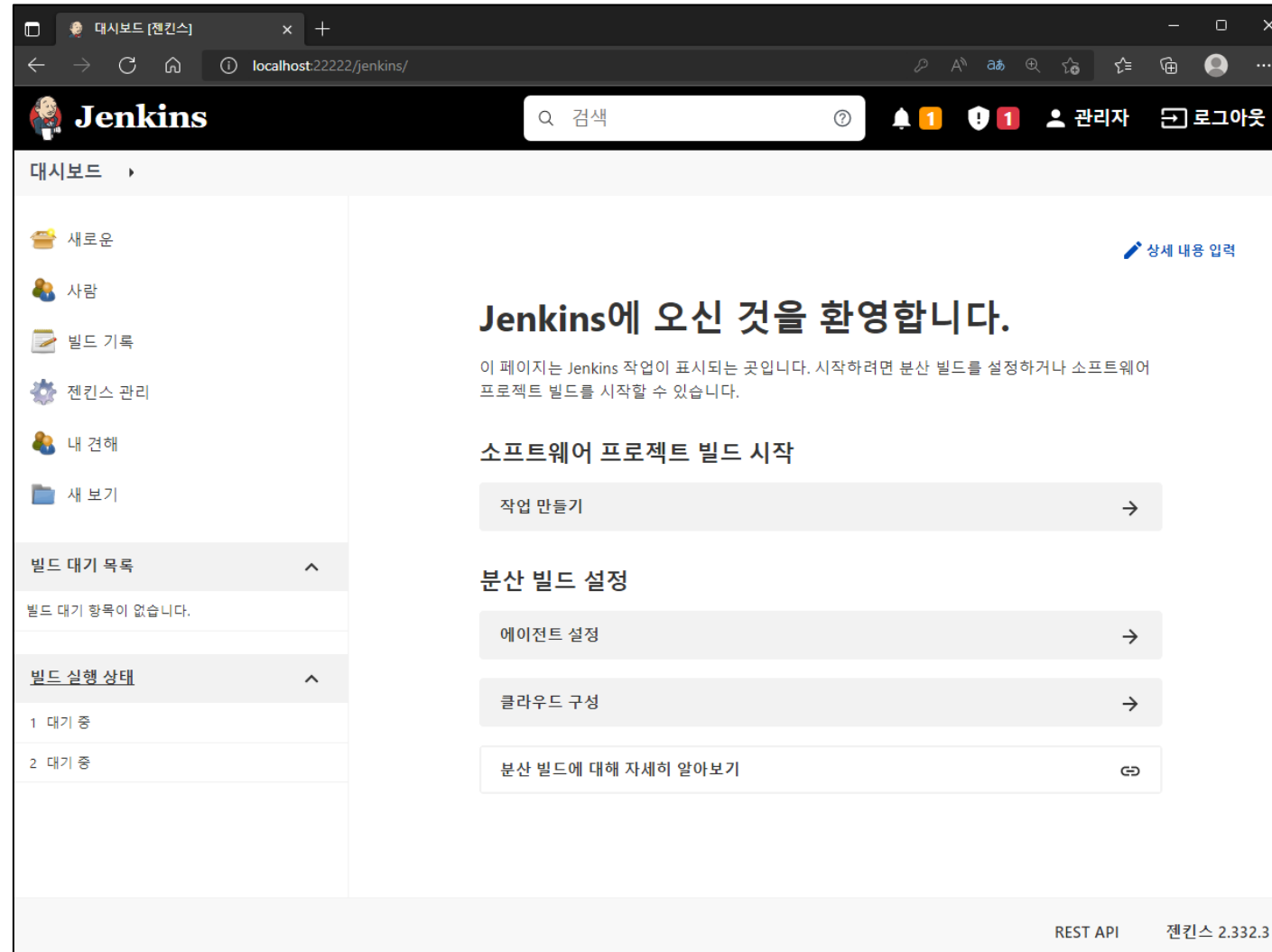
이메일 주소:

Jenkins 2.346.1

[Skip and continue as admin](#) [Save and Continue](#)


환경구성

- 젠킨스 설치
 - 설치 완료



젠킨스

- 젠킨스 관리
 - Global Tool Configuration
 - JDK 구성
 - 메이븐 구성

 **Jenkins**

Dashboard > Global Tool Configuration

[↑ Back to Dashboard](#)
[Manage Jenkins](#)

Global Tool Configuration

Maven Configuration

Default settings provider

Use default maven settings

Default global settings provider

Use default maven global settings

JDK

JDK installations

List of JDK installations on this system

Add JDK

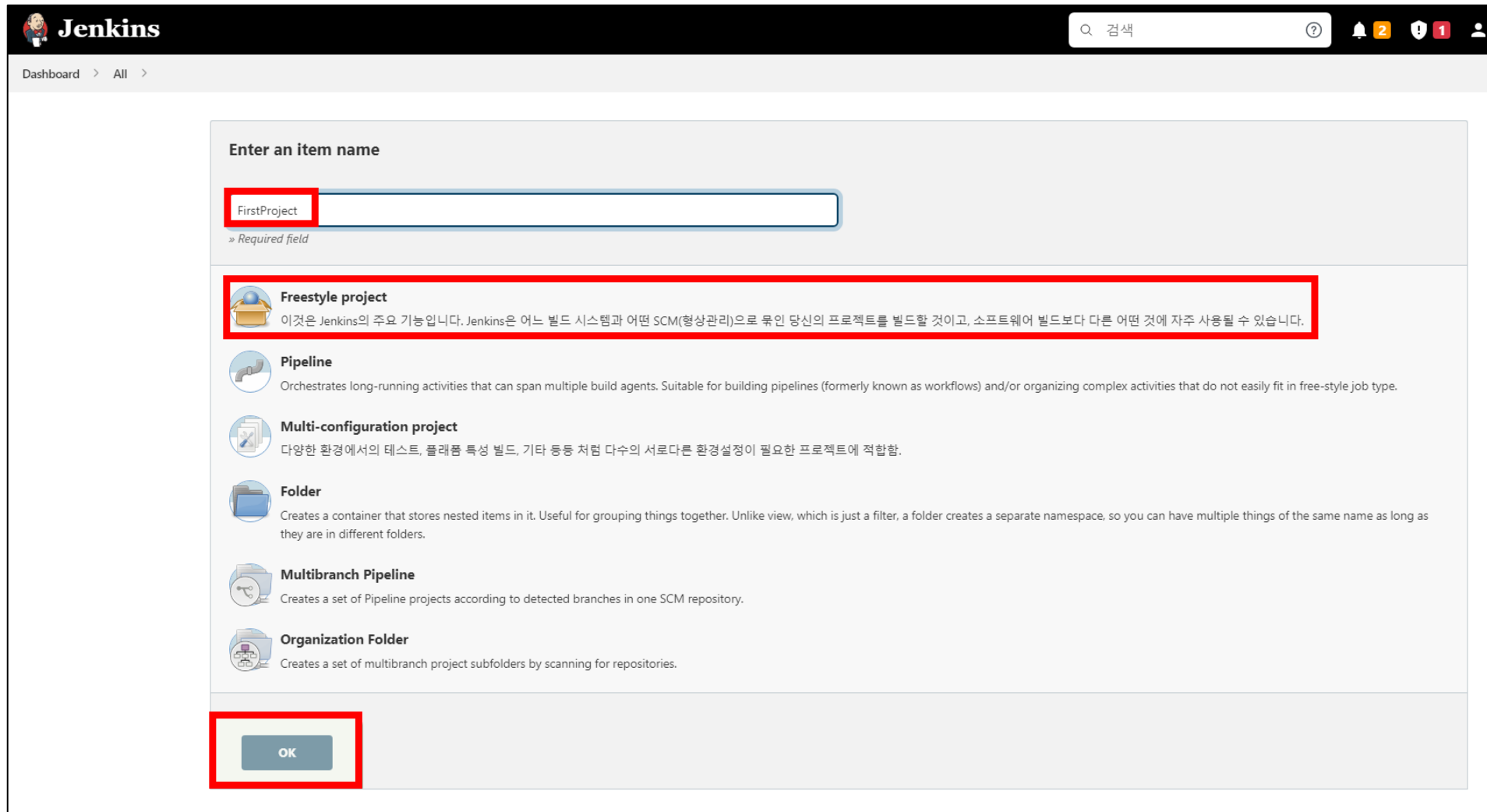
Maven

Maven installations

[Save](#) [Apply](#)

젠킨스

- 프로젝트 생성
 - New Item



The image shows the Jenkins 'New Item' dialog box. At the top, the Jenkins logo and name are on the left, and a search bar and notification icons are on the right. Below the header, the breadcrumb 'Dashboard > All >' is visible. The main section is titled 'Enter an item name' and contains a text input field with 'FirstProject' entered. A red box highlights this input field. Below the input field, the text '» Required field' is displayed. A list of project types follows, each with an icon and a description. A red box highlights the 'Freestyle project' option. The 'OK' button at the bottom is also highlighted with a red box.

Enter an item name

FirstProject

» Required field

Freestyle project
이것은 Jenkins의 주요 기능입니다. Jenkins은 어느 빌드 시스템과 어떤 SCM(형상관리)으로 묶인 당신의 프로젝트를 빌드할 것이고, 소프트웨어 빌드보다 다른 어떤 것에 자주 사용될 수 있습니다.

Pipeline
Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.

Multi-configuration project
다양한 환경에서의 테스트, 플래폼 특성 빌드, 기타 등등 처럼 다수의 서로다른 환경설정이 필요한 프로젝트에 적합함.

Folder
Creates a container that stores nested items in it. Useful for grouping things together. Unlike view, which is just a filter, a folder creates a separate namespace, so you can have multiple things of the same name as long as they are in different folders.

Multibranch Pipeline
Creates a set of Pipeline projects according to detected branches in one SCM repository.

Organization Folder
Creates a set of multibranch project subfolders by scanning for repositories.

OK

젠킨스

- 프로젝트 - 구성
 - 프로젝트 설정

Dashboard > FirstProject >

General소스 코드 관리빌드 유발빌드 환경Build빌드 후 조치

설명

[Plain text] [미리보기](#)

☐ GitHub project

☐ Throttle builds ?

☐ 오래된 빌드 삭제 ?

☐ 이 빌드는 매개변수가 있습니다 ?

☐ 빌드 안함 ?

☐ 필요한 경우 concurrent 빌드 실행 ?

고급...

소스 코드 관리

☒ None


☐ Git ?


저장


Apply


젠킨스


- 대시보드
 - 프로젝트 생성 완료

 Jenkins

 2


 1


 관리자


 로그아웃


Dashboard >


+ 새로운 Item

 사람

 빌드 기록

 Jenkins 관리

 My Views

 새로운 뷰

빌드 대기 목록



빌드 대기 항목이 없습니다.

빌드 실행 상태

1 대기 중

2 대기 중

All +

S	W	Name ↓	최근 성공	최근 실패	최근 소요 시간
		FirstProject	—	—	—

아이콘: S M L

Icon legend

Atom feed 모두

Atom feed 실패

Atom feed 최근 빌드

상세 내용 입력

젠킨스

- 파이프라인 구성
 - 메이븐 빌드 도구 설정
 - 대시보드 – Jenkin관리 – Global Tool Configuration

Dashboard > Global Tool Configuration

List of Ant installations on this system

Add Ant

Maven

Maven installations

List of Maven installations on this system

Add Maven

≡ Maven

Name

maven3.8.6

✓ Install automatically ?

≡ Install from Apache

Version

3.8.6

Add Installer ▾

Add Maven

Save

Apply

젠킨스

- 파이프라인 구성
 - GitHub 레포 설정
 - 프로젝트 – 구성관리 – 소스코드 관리

The screenshot shows the Jenkins web interface for configuring a project named 'FirstProject'. The breadcrumb navigation at the top reads 'Dashboard > FirstProject >'. Below this, a series of tabs are visible: 'General', '소스 코드 관리' (Source Code Management), '빌드 유발' (Build Triggers), '빌드 환경' (Build Environment), 'Build', and '빌드 후 조치' (Post-build Actions). The '소스 코드 관리' tab is currently selected.

Under the '소스 코드 관리' tab, there are two radio buttons: 'None' and 'Git'. The 'Git' option is selected and highlighted with a blue dot. Below these buttons is a section titled 'Repositories' with a help icon (?). This section is enclosed in a dashed border and contains the following fields:

- Repository URL** (with a help icon ?): A text input field containing the URL 'https://github.com/kyesCare/test.git'. A red 'X' icon is visible in the top right corner of the dashed box.
- Credentials** (with a help icon ?): A dropdown menu showing 'kyes.care@gmail.com/*****' with a downward arrow icon.

Below the 'Credentials' field, there is a '+ Add' button. At the bottom of the dashed box, there is a '고급...' (Advanced...) button. Below the dashed box, there is an 'Add Repository' button.

젠킨스

- 파이프라인 구성
 - GitHub 레포 설정
 - 프로젝트 – 구성관리 – 소스코드 관리

Global Credentials (System)

Kind

Username with password

Scope ?

Global (Jenkins, nodes, items, all child items, etc)

Username ?

깃허브 아이디

☐ Treat username as secret ?

Password ?

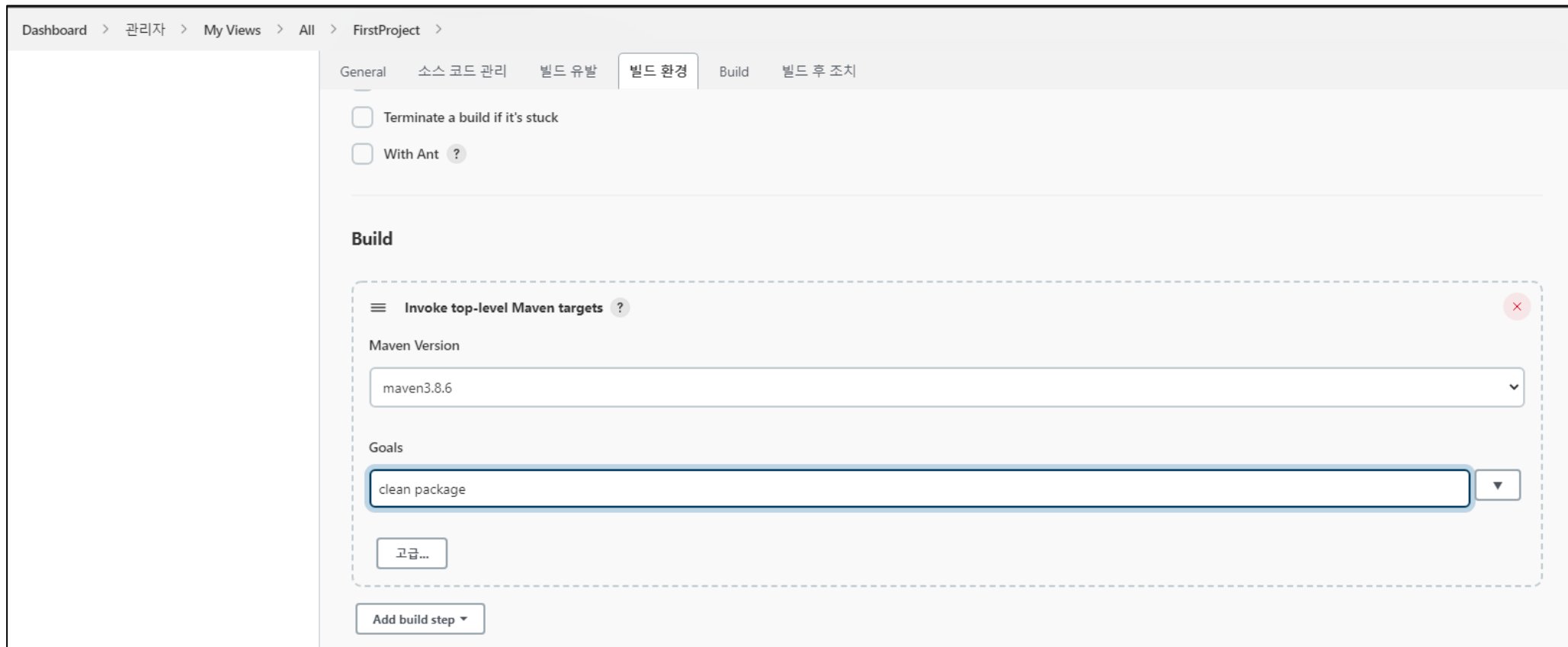
ID ?

Description ?

Add Cancel

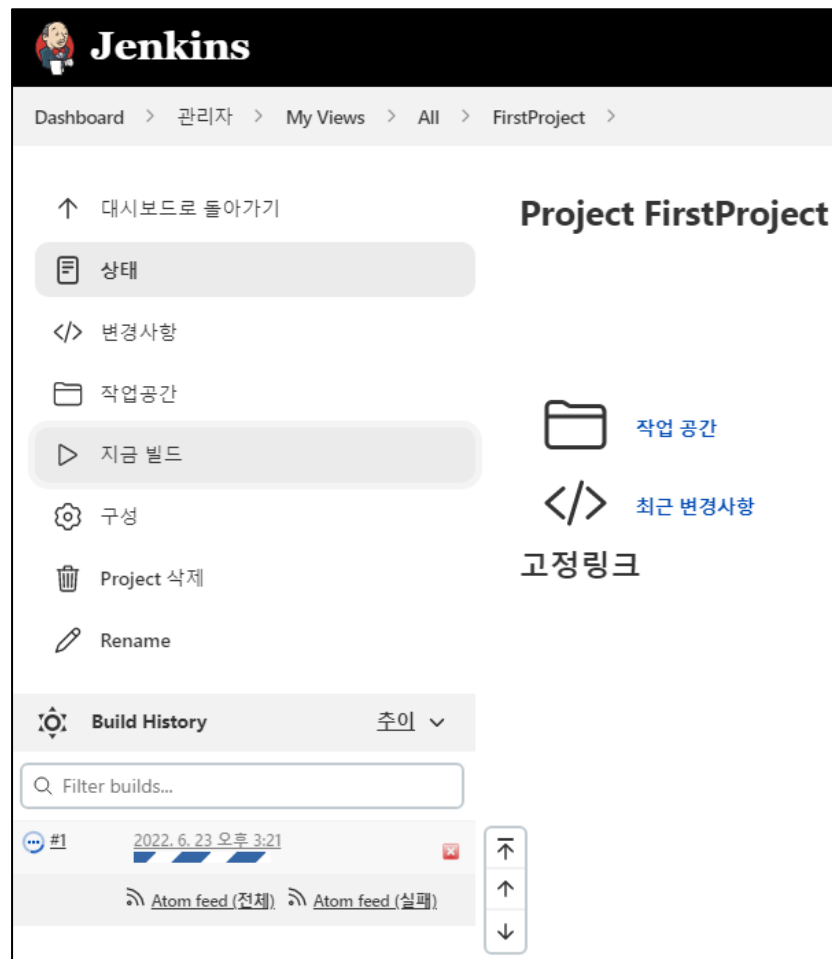
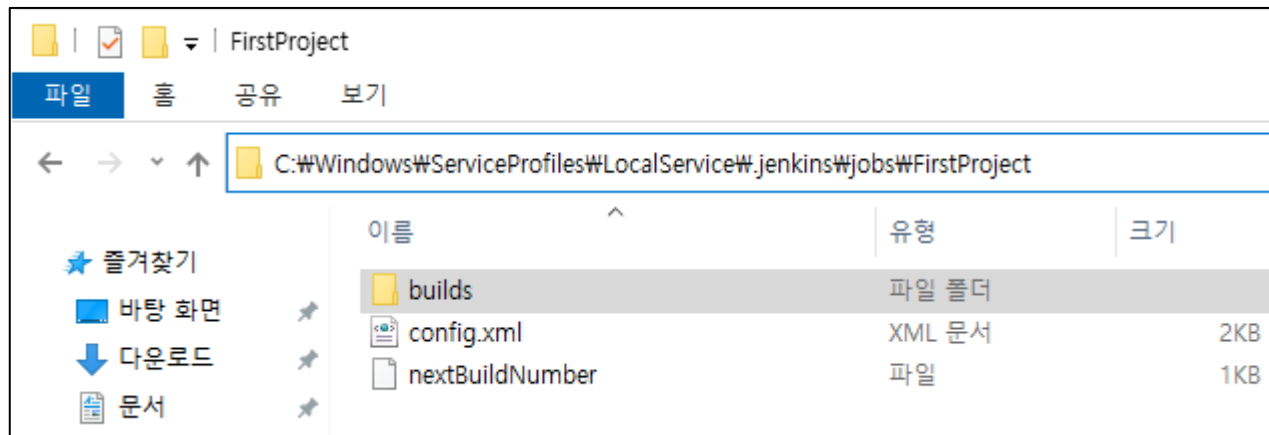
젠킨스

- 파이프라인 구성
 - GitHub 레포 설정
 - 프로젝트 – 구성관리 – 빌드 환경



젠킨스

- 파이프라인 구성
 - 지금 빌드 선택
 - 작업 폴더에서 확인 가능



젠킨스

- 로컬 서버 배포
 - war 패키징 구성

```
1 package com.example.demo;
2
3 import org.springframework.boot.SpringApplication;
4
5
6
7
8 @SpringBootApplication
9 public class FirstApplication extends SpringBootServletInitializer{
10
11     @Override
12     protected SpringApplicationBuilder createSpringApplicationBuilder() {
13         // TODO Auto-generated method stub
14         return super.createSpringApplicationBuilder();
15     }
16
17     public static void main(String[] args) {
18         SpringApplication.run(FirstApplication.class, args);
19     }
20
21 }
22
```

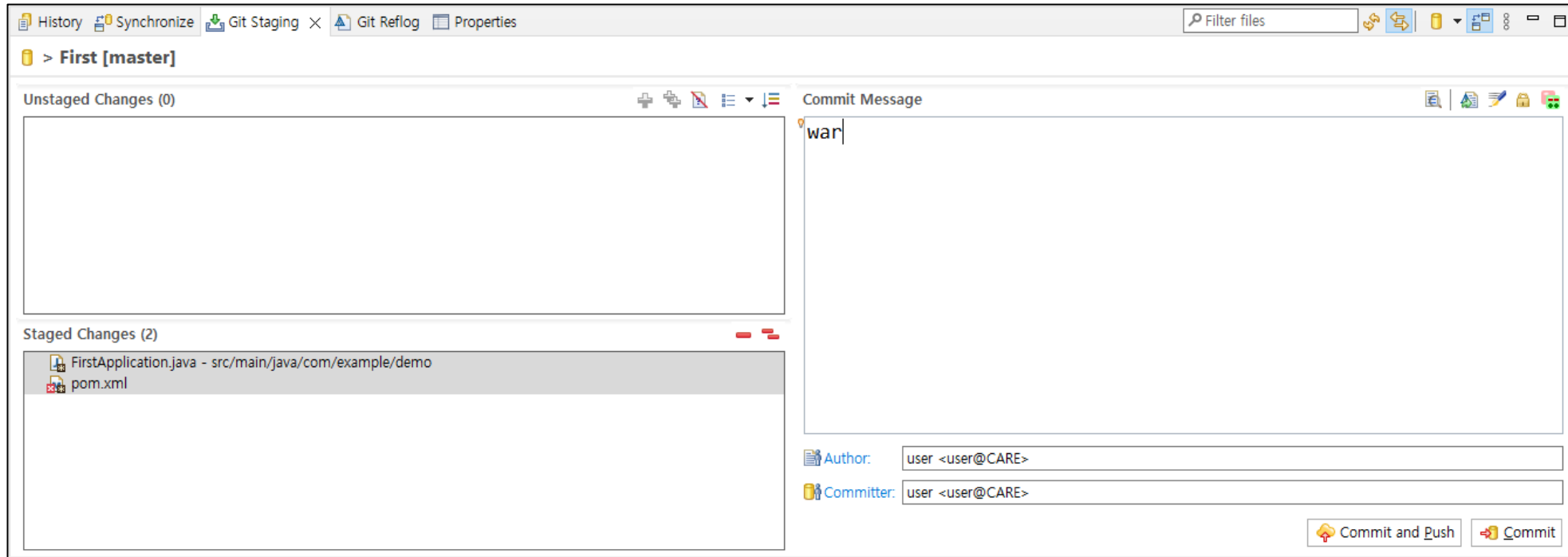
젠킨스

- 로컬 서버 배포
 - war 패키징 구성

```
8      <artifactId>spring-boot-starter-parent</artifactId>
9      <version>2.7.0</version>
10     <relativePath /> <!-- lookup parent from repository -->
11   </parent>
12   <groupId>com.example</groupId>
13   <artifactId>First</artifactId>
14   <version>0.0.1-SNAPSHOT</version>
15   <name>First</name>
16   <description>Demo project for Spring Boot</description>
17   <packaging>war</packaging>
18 <properties>
19   <java.version>1.8</java.version>
20 </properties>
21 <dependencies>
22   <dependency>
23     <groupId>org.springframework.boot</groupId>
24     <artifactId>spring-boot-starter-tomcat</artifactId>
25     <scope>provided</scope>
26   </dependency>
27   <dependency>
28     <groupId>org.springframework.boot</groupId>
29     <artifactId>spring-boot-starter-web</artifactId>
30   </dependency>
```

젠킨스

- 로컬 서버 배포
 - github push



젠킨스

- 로컬 서버 배포
 - server.xml

```
<Service name="Catalina2">
  <Connector port="8888" protocol="HTTP/1.1"
    connectionTimeout="20000"
    redirectPort="9443" URIEncoding="UTF-8" maxThreads="250" maxHttpHeaderSize="8192" />
  <Connector port="9009" protocol="AJP/1.3" redirectPort="9443" />

  <Engine name="Catalina2" defaultHost="localhost">

    <Realm className="org.apache.catalina.realm.LockOutRealm">
      <Realm className="org.apache.catalina.realm.UserDatabaseRealm"
        resourceName="UserDatabase"/>
    </Realm>

    <Host name="localhost" appBase="webapps" unpackWARs="true" autoDeploy="true">
      <Valve className="org.apache.catalina.valves.AccessLogValve" directory="logs"
        prefix="localhost_access_log" suffix=".txt"
        pattern="%h %l %u %t &quot;%r&quot; %s %b" />
    </Host>

  </Engine>
</Service>
```

젠킨스

- 로컬 서버 배포
 - tomcat-users.xml

```
*tomcat-users.xml - Windows 메모장
파일(F)  편집(E)  서식(O)  보기(V)  도움말(H)

<role rolename="role1"/>
<user username="tomcat" password="<must-be-changed>" roles="tomcat"/>
<user username="both" password="<must-be-changed>" roles="tomcat,role1"/>
<user username="role1" password="<must-be-changed>" roles="role1"/>
-->
<role rolename="manager-gui"> </role>
  <role rolename="manager-script"> </role>
  <role rolename="manager-status"> </role>
  <user username="jenkins" password="1234" roles="manager-gui,manager-script,manager-status"/>

</tomcat-users>
```

젠킨스

- 로컬 서버 배포
 - Download now and install after restart

The screenshot shows the Jenkins web interface. At the top is the Jenkins logo and name. Below it is a breadcrumb trail: Dashboard > Plugin Manager. On the left sidebar, there are two links: '대시보드로 돌아가기' (Return to Dashboard) with an upward arrow icon, and 'Jenkins 관리' (Manage Jenkins) with a gear icon. The main content area is titled 'Plugin Manager'. It features four tabs: '업데이트된 플러그인 목록' (Updated Plugins List), '설치 가능' (Installable) which is currently selected, '설치된 플러그인 목록' (Installed Plugins List), and '고급' (Advanced). Below the tabs is a search bar containing the text 'deploy'. Under the search bar, there is a table with the following content:

Install	Name ↓
<input type="checkbox"/>	Deploy to container 1.16
<input type="checkbox"/>	Artifact Uploaders

Below the 'Artifact Uploaders' entry, there is a description: 'This plugin allows you to deploy a war to a container after a successful build. Glassfish 3.x remote deployment'.

젠킨스

- 로컬 서버 배포

Dashboard > 관리자 > My Views > All > FirstProject >

General 소스 코드 관리 빌드 유발 빌드 환경 Build 빌드 후 조치

Deploy war/ear to a container

WAR/EAR files ?

**/*.war

Context path ?

/First

Containers

Tomcat 9.x Remote

Credentials

jenkins/*****

+ Add

Tomcat URL ?

http://localhost:8888

고급...

Add Container ▾

☐ Deploy on failure

젠킨스

- 서로 다른 환경 구성하기
 - 공인아이피를 지정하여 사용할 서버.



AWS에 가입

Support 플랜 선택

비즈니스 또는 개인 계정에 대한 Support 플랜을 선택합니다. [플랜 및 요금 예시를 비교](#) [여기](#) 보세요.
언제든지 AWS Management Console에서 플랜을 변경할 수 있습니다.

☒ 기본 지원 - 무료

- AWS를 처음 시작하는 신규 사용자에게 권장
- AWS 리소스에 대한 연중무휴 24시간 셀프 서비스 액세스
- 계정 및 청구 문제 전용
- Personal Health Dashboard 및 Trusted Advisor에 대한 액세스



☐ 개발자 지원 - 시작가는 29 USD/월

- AWS를 체험해보는 개발자에게 권장
- 업무 시간 중 AWS Support에 대한 이메일 액세스
- 12시간(업무 시간 기준) 이내의 응답 시간



☐ 비즈니스 지원 - 시작가는 100 USD/월

- AWS 기반 프로덕션 워크로드 실행에 추천
- 이메일, 전화 및 채팅을 통한 연중무휴 24시간 기술 지원
- 1시간 이내의 응답 시간
- Trusted Advisor 모범 사례 권장 사항 전체 세트





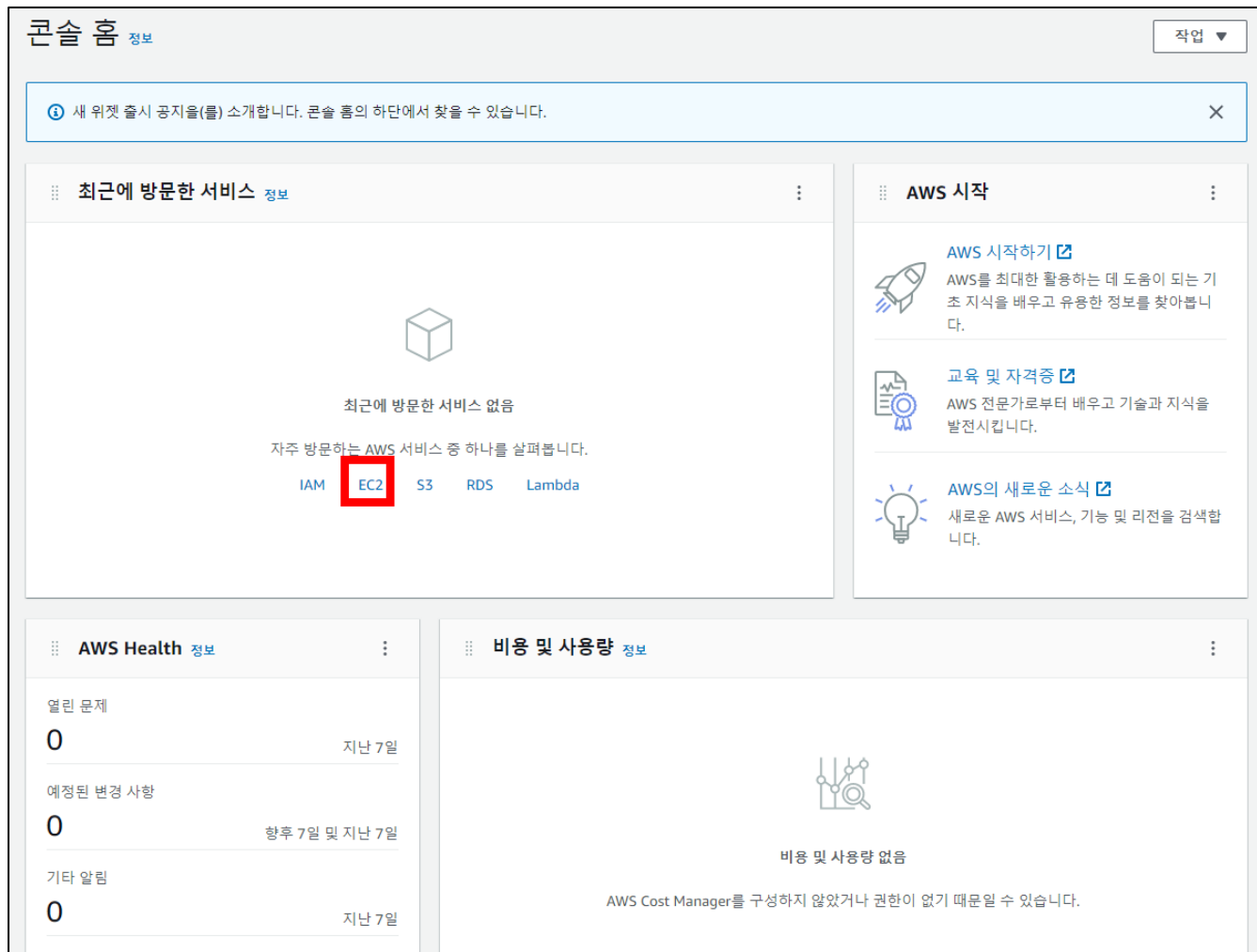
엔터프라이즈 수준의 지원이 필요하신가요?

최저 월 15,000 USD로 15분 이내에 응답을 받을 수 있으며 기술 지원 관리자가 배정된 컨시어지 스타일의 서비스를 이용할 수 있습니다. [자세히 알아보기](#) [여기](#)

[가입 완료](#)

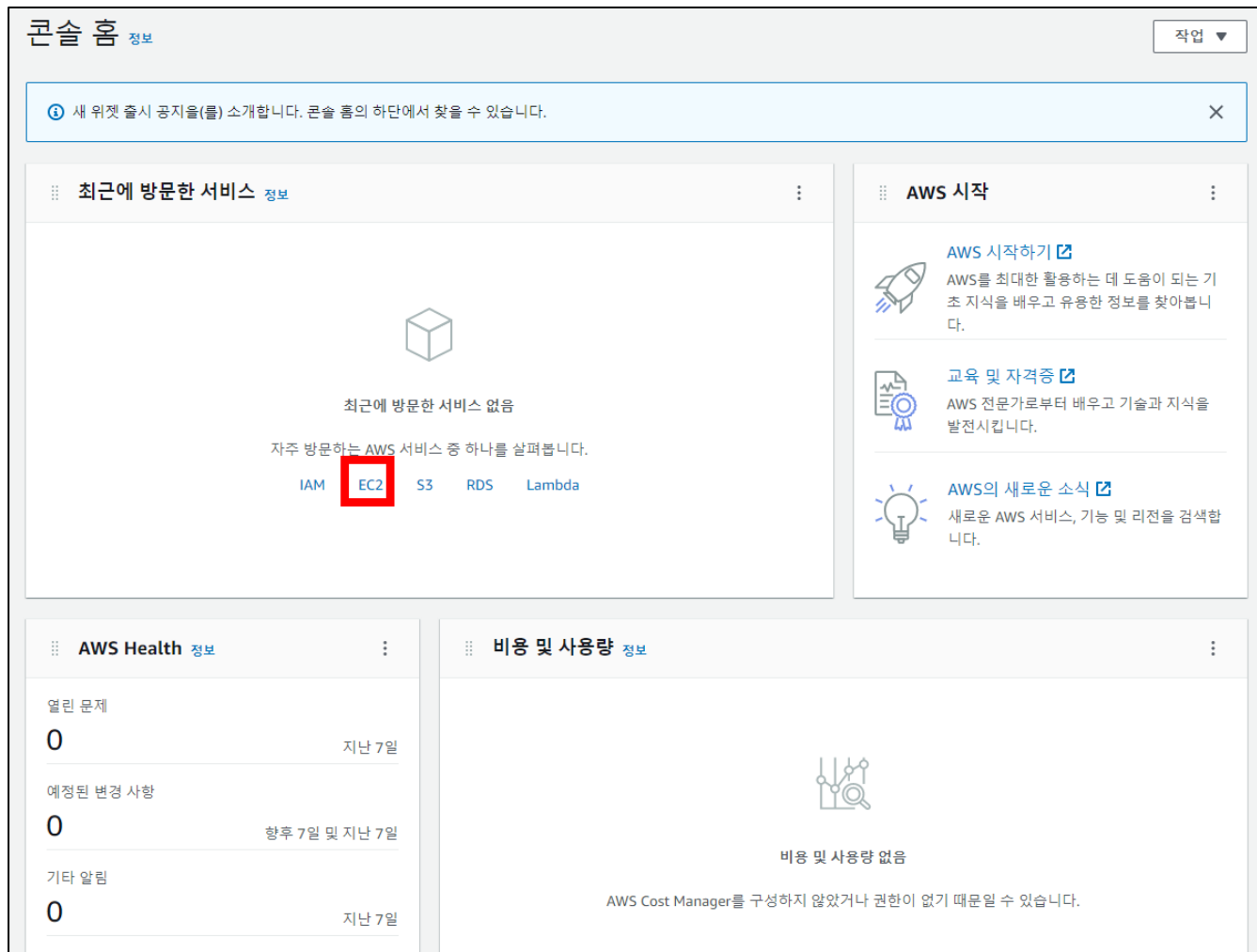
젠킨스

- 서로 다른 환경 구성하기
 - EC2 서버 구성하기.



젠킨스

- 서로 다른 환경 구성하기
 - EC2 서버 구성하기.



젠킨스

- 서로 다른 환경 구성하기
 - EC2 서버 구성하기.

The screenshot displays the AWS Management Console interface for the EC2 service. The left sidebar contains navigation links for various AWS services, including EC2, IAM, and S3. The main content area is divided into several sections:

- 리소스 (Resources):** A table showing the usage of various Amazon EC2 resources in the '미국 서부 (오레곤)' region. The resources and their counts are: 인스턴스(실행 중) 0, 로드 밸런서 0, 배치 그룹 0, 보안 그룹 1, 볼륨 0, 스냅샷 0, 인스턴스 0, 전용 호스트 0, 키 페어 0, and 탄력적 IP 0.
- 인스턴스 시작 (Start Instance):** A section with a red box highlighting the '인스턴스 시작' button. Below it, there is a link to '서버 마이그레이션' and a note about starting instances in the '미국 서부 (오레곤)' region.
- 서비스 상태 (Service Status):** A section showing the status of the EC2 service in the '미국 서부 (오레곤)' region. The status is '이 서비스가 정상적으로 작동 중입니다.' (This service is operating normally).
- 영역 (Regions):** A table listing the available regions and their IDs.

영역 이름	영역 ID
us-west-2a	usw2-az2
us-west-2b	usw2-az1
us-west-2c	usw2-az3
us-west-2d	usw2-az4

젠킨스

- 서로 다른 환경 구성하기
 - 프리티어 환경으로 서버 구성하기.
 - 이름 : 구분할 서버 이름
 - 운영체제 : Ubuntu 18.04

EC2 > 인스턴스 > 인스턴스 시작

인스턴스 시작 정보

Amazon EC2를 사용하면 AWS 클라우드에서 실행되는 가상 머신 또는 인스턴스를 생성할 수 있습니다. 아래의 간단한 단계에 따라 빠르게 시작할 수 있습니다.

이름 및 태그 정보

이름

외부서버

[추가 태그 추가](#)

▼ 애플리케이션 및 OS 이미지(Amazon Machine Image) 정보

AMI는 인스턴스를 시작하는 데 필요한 소프트웨어 구성(운영 체제, 애플리케이션 서버 및 애플리케이션)이 포함된 템플릿입니다. 아래에서 찾고 있는 항목이 보이지 않으면 AMI를 검색하거나 찾아보십시오.

🔍 수천 개의 애플리케이션 및 OS 이미지를 포함하는 전체 카탈로그 검색

최근 사용

Quick Start

Amazon Linux

aws

Ubuntu

ubuntu®

Windows

Microsoft

Red Hat

Red Hat

SUSE Linux

SUSE

🔍

더 많은 AMI 찾아보기

AWS, Marketplace 및 커뮤니티의 AMI 포함

Amazon Machine Image(AMI)

Ubuntu Server 18.04 LTS (HVM), SSD Volume Type
ami-0b152cfd354c4c7a4 (64비트(x86)) / ami-0fcbcd64ee75d9d12 (64비트(Arm))
가상화: hvm ENA 활성화됨: true 루트 디바이스 유형: ebs

프리 티어 사용 가능 ▼

젠킨스

- 서로 다른 환경 구성하기
 - 프리티어 환경으로 서버 구성하기.
 - 인스턴스 유형 : 프리티어 사용 가능 중 선택.
 - 새 키 페어 생성 : 이름과 .pem 선택 된 상태에서 키 페어 생성.

▼ 인스턴스 유형 정보

인스턴스 유형

t2.micro
패밀리: t2 1 vCPU 1 GiB 메모리
온디맨드 Linux 요금: 0.0116 USD 시간당
온디맨드 Windows 요금: 0.0162 USD 시간당

프리 티어 사용 가능

인스턴스 유형 비교

▼ 키 페어(로그인) 정보

키 페어를 사용하여 인스턴스에 안전하게 연결할 수 있습니다. 인스턴스를 시작하기 전에 선택한 키 페어에 대한 액세스 권한이 있는지 확인하세요.

키 페어 이름 - 필수

jenkinsTest

새 키 페어 생성

키 페어 생성

키 페어를 사용하면 인스턴스에 안전하게 연결할 수 있습니다.

아래에 키 페어의 이름을 입력합니다. 메시지가 표시되면 프라이빗 키를 사용자 컴퓨터의 안전하고 액세스 가능한 위치에 저장합니다. 나중에 인스턴스에 연결할 때 필요합니다. [자세히 알아보기](#)

키 페어 이름

jenkinsTest

이름은 최대 255개의 ASCII 문자를 포함할 수 있습니다. 선행 또는 후행 공백은 포함할 수 없습니다.

프라이빗 키 파일 형식

☒ .pem
OpenSSH와 함께 사용

☐ .ppk
PuTTY와 함께 사용

취소

키 페어 생성

젠킨스

- 서로 다른 환경 구성하기
 - 보안 그룹 생성
 - SSH, HTTP, HTTPS 트래픽 허용
 - 인스턴스 시작으로 서버 구성 완료.

▼ 네트워크 설정

편집

네트워크
vpc-0fe37d44e1bc65dec

서브넷
기본 설정 없음(가용 영역의 기본 서브넷)

퍼블릭 IP 자동 할당
활성화

방화벽(보안 그룹) 정보
A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

☒ 보안 그룹 생성 ☐ 기존 보안 그룹 선택

다음 규칙을 사용하여 'launch-wizard-2'(이)라는 새 보안 그룹을 생성합니다.

☒ 에서 SSH 트래픽 허용
인스턴스 연결에 도움 위치 무관
0.0.0.0/0

☒ 인터넷에서 HTTPS 트래픽 허용
예를 들어 웹 서버를 생성할 때 엔드포인트를 설정하려면

☒ 인터넷에서 HTTP 트래픽 허용
예를 들어 웹 서버를 생성할 때 엔드포인트를 설정하려면

⚠ 소스가 0.0.0.0/0인 규칙은 모든 IP 주소에서 인스턴스에 액세스하도록 허용합니다. 알려진 IP 주소의 액세스만 허용하도록 보안 그룹을 설정하는 것이 좋습니다.

×

▼ 요약

인스턴스 개수 정보
1

소프트웨어 이미지(AMI)
Canonical, Ubuntu, 18.04 LTS, ...더 보기
ami-0b152cfd354c7a4

가상 서버 유형(인스턴스 유형)
t2.micro

방화벽(보안 그룹)
새 보안 그룹

스토리지(볼륨)
1개의 볼륨 – 8GiB

❗ 프리 티어: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 30 GiB of EBS storage, 2 million IOs, 1 GB of snapshots, and 100 GB of bandwidth to the internet.

×

취소

인스턴스 시작

젠킨스

- 서로 다른 환경 구성하기
 - 서비스 포트 8888번 추가

EC2 > 보안 그룹 > sg-000ac491628763da3 - launch-wizard-2 > 인바운드 규칙 편집

인바운드 규칙 편집 [정보](#)

인바운드 규칙은 인스턴스에 도달하도록 허용된 수신 트래픽을 제어합니다.

인바운드 규칙 [정보](#)

보안 그룹 규칙 ID	유형 정보	프로토콜 정보	포트 범위 정보	소스 정보	설명 - 선택 사항 정보	
sgr-03337a0d56edc21de	SSH ▼	TCP	22	사용자 지정 ▼ Q 0.0.0.0/0 ✕		삭제
sgr-06fb1abb7866d6905	HTTPS ▼	TCP	443	사용자 지정 ▼ Q 0.0.0.0/0 ✕		삭제
sgr-0cea11e27875af5cc	HTTP ▼	TCP	80	사용자 지정 ▼ Q 0.0.0.0/0 ✕		삭제
-	사용자 지정 TCP ▼	TCP	8888	Anywhere-I... ▼ Q 0.0.0.0/0 ✕		삭제

규칙 추가

취소

변경 사항 미리 보기

규칙 저장

젠킨스

- 서로 다른 환경 구성하기
 - 서비스 포트 8888번 추가

인스턴스 (1/3) 정보

🔍 검색

Name	인스턴스 ID	인스턴스 상태	인스턴스 유형	상태 검사
외부서버	i-05aee5461fdc18d2f	실행 중	t2.micro	초기화

🔄 연결

인스턴스 상태 ▼

작업 ▼

인스턴스 시작 ▼

EC2 > 인스턴스 > i-05aee5461fdc18d2f > 인스턴스에 연결

인스턴스에 연결 정보

다음 옵션 중 하나를 사용하여 인스턴스 i-05aee5461fdc18d2f (외부서버)에 연결

EC2 인스턴스 연결

Session Manager

SSH 클라이언트

EC2 직렬 콘솔

인스턴스 ID
📄 i-05aee5461fdc18d2f (외부서버)

1. SSH 클라이언트를 엽니다.

2. 프라이빗 키 파일을 찾습니다. 이 인스턴스를 시작하는 데 사용되는 키는 jenkinsTest.pem입니다.

3. 필요한 경우 이 명령을 실행하여 키를 공개적으로 볼 수 없도록 합니다.
📄 `chmod 400 jenkinsTest.pem`

4. 퍼블릭 DNS을(를) 사용하여 인스턴스에 연결:
📄 `ec2-18-236-119-105.us-west-2.compute.amazonaws.com`

예:
📄 `ssh -i "jenkinsTest.pem" ubuntu@ec2-18-236-119-105.us-west-2.compute.amazonaws.com`

참고: 대부분의 경우 추정된 사용자 이름은 정확합니다. 하지만 AMI 사용 지침을 읽고 AMI 소유자가 기본 AMI 사용자 이름을 변경했는지 확인하십시오.

취소

젠킨스

- 서로 다른 환경 구성하기
 - SSH로 접근 후 명령입력.

```
C:\Windows\system32\cmd.exe - ssh -i "jenkinsTest.pem" ubuntu@ec2-18-236-119-105.us-west-2.compute.amazonaws.com
C:\Users\user>cd

C:\>ssh -i "jenkinsTest.pem" ubuntu@ec2-18-236-119-105.us-west-2.compute.amazonaws.com
The authenticity of host 'ec2-18-236-119-105.us-west-2.compute.amazonaws.com (18.236.119.105)' can't be established.
ECDSA key fingerprint is SHA256:1LVexsSf3qN55WfceE1pgNQKR9l3iGeQOpFSwSBv40E.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
```

```
선택 ubuntu@ip-172-31-9-165: ~
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-172-31-9-165:~$ sudo apt update
Hit:1 http://us-west-2.ec2.archive.ubuntu.com/ubuntu bionic InRelease
Get:2 http://us-west-2.ec2.archive.ubuntu.com/ubuntu bionic-updates InRelease [88.7 kB]
Get:3 http://us-west-2.ec2.archive.ubuntu.com/ubuntu bionic-backports InRelease [74.6 kB]
Get:4 http://us-west-2.ec2.archive.ubuntu.com/ubuntu bionic/universe amd64 Packages [8570 kB]
Get:5 http://security.ubuntu.com/ubuntu bionic-security InRelease [88.7 kB]
Get:6 http://us-west-2.ec2.archive.ubuntu.com/ubuntu bionic/universe Translation-en [4941 kB]
```

젠킨스

- 서로 다른 환경 구성하기
 - JDK 다운로드 및 구성

```
Processing triggers for mime-support (3.60ubuntu1) ...
Processing triggers for ureadahead (0.100.0-21) ...
Processing triggers for libc-bin (2.27-3ubuntu1.6) ...
Processing triggers for systemd (237-3ubuntu10.53) ...
Processing triggers for man-db (2.8.3-2ubuntu0.1) ...
ubuntu@ip-172-31-9-165:~$ sudo apt install openjdk-11-jdk
```

```
ubuntu@ip-172-31-4-50: ~
ubuntu@ip-172-31-4-50:~$ sudo vim /etc/profile
ubuntu@ip-172-31-4-50:~$ tail -1 /etc/profile
export JAVA_HOME=/usr/lib/jvm/java-11-openjdk-amd64
ubuntu@ip-172-31-4-50:~$ source /etc/profile
ubuntu@ip-172-31-4-50:~$
```

젠킨스

- 서로 다른 환경 구성하기
 - tomcat 계정 및 그룹 생성
 - `sudo useradd -r -m -U -d /opt/tomcat -s /bin/false tomcat`
 - `sudo tail -5 /etc/shadow`
 - `sudo tail -5 /etc/group`

```
ubuntu@ip-172-31-4-50: ~  
ubuntu@ip-172-31-4-50:~$ sudo useradd -r -m -U -d /opt/tomcat -s /bin/false tomcat  
ubuntu@ip-172-31-4-50:~$ sudo tail -5 /etc/shadow  
landscape:*:19153:0:99999:7:::  
sshd:*:19153:0:99999:7:::  
pollinate:*:19153:0:99999:7:::  
ubuntu:!:19166:0:99999:7:::  
tomcat:!:19166:0:99999:7:::  
ubuntu@ip-172-31-4-50:~$ sudo tail -5 /etc/group  
landscape:x:112:  
admin:x:113:  
netdev:x:114:ubuntu  
ubuntu:x:1000:  
tomcat:x:999:  
ubuntu@ip-172-31-4-50:~$
```


젠킨스

- 서로 다른 환경 구성하기
 - 톰캣 파일 다운로드
 - `wget http://mirror.apache-kr.org/tomcat/tomcat-9/v9.0.64/bin/apache-tomcat-9.0.64.tar.gz`

```
ubuntu@ip-172-31-4-50: ~  
ubuntu@ip-172-31-4-50:~$ wget http://mirror.apache-kr.org/tomcat/tomcat-9/v9.0.64/bin/apache-tomcat-9.0.64.tar.gz  
--2022-06-23 12:07:10-- http://mirror.apache-kr.org/tomcat/tomcat-9/v9.0.64/bin/apache-tomcat-9.0.64.tar.gz  
Resolving mirror.apache-kr.org (mirror.apache-kr.org)... 125.209.216.167  
Connecting to mirror.apache-kr.org (mirror.apache-kr.org)|125.209.216.167|:80... connected.  
HTTP request sent, awaiting response... 200 OK  
Length: 11579290 (11M) [application/octet-stream]  
Saving to: 'apache-tomcat-9.0.64.tar.gz'  
  
apache-tomcat-9.0.64.tar.gz 100%[=====>] 11.04M 3.29MB/s in 6.4s  
  
2022-06-23 12:07:17 (1.71 MB/s) - 'apache-tomcat-9.0.64.tar.gz' saved [11579290/11579290]  
ubuntu@ip-172-31-4-50:~$ _
```

젠킨스

- 서로 다른 환경 구성하기
 - 톰캣 파일 압축 후 디렉토리 명 변경
 - `sudo tar -xf /home/ubuntu/apache-tomcat-9.0.64.tar.gz -C /opt/tomcat`
 - `sudo mv apache-tomcat-9.0.64/ tomcat-9`

```
ubuntu@ip-172-31-4-50: ~  
ubuntu@ip-172-31-4-50:~$ sudo tar -xf /home/ubuntu/apache-tomcat-9.0.64.tar.gz -C /opt/tomcat  
ubuntu@ip-172-31-4-50:~$ cd /opt/tomcat/  
ubuntu@ip-172-31-4-50:/opt/tomcat$ ls  
apache-tomcat-9.0.64  
ubuntu@ip-172-31-4-50:/opt/tomcat$ sudo mv apache-tomcat-9.0.64/ tomcat-9  
ubuntu@ip-172-31-4-50:/opt/tomcat$ ls  
tomcat-9
```

젠킨스

- 서로 다른 환경 구성하기
 - 톰캣 디렉토리 권한 설정
 - `sudo chown -RH tomcat: /opt/tomcat/tomcat-9/`

```
ubuntu@ip-172-31-4-50: ~  
ubuntu@ip-172-31-4-50:/opt/tomcat$ sudo chown -RH tomcat: /opt/tomcat/tomcat-9/  
ubuntu@ip-172-31-4-50:/opt/tomcat$ ls -al tomcat-9/  
total 156  
drwxr-xr-x 9 tomcat tomcat 4096 Jun 23 12:08 .  
drwxr-xr-x 3 tomcat tomcat 4096 Jun 23 12:09 ..  
-rw-r----- 1 tomcat tomcat 18986 Jun  2 19:08 BUILDING.txt  
-rw-r----- 1 tomcat tomcat  6210 Jun  2 19:08 CONTRIBUTING.md  
-rw-r----- 1 tomcat tomcat 57092 Jun  2 19:08 LICENSE  
-rw-r----- 1 tomcat tomcat  2333 Jun  2 19:08 NOTICE  
-rw-r----- 1 tomcat tomcat  3398 Jun  2 19:08 README.md  
-rw-r----- 1 tomcat tomcat  6901 Jun  2 19:08 RELEASE-NOTES  
-rw-r----- 1 tomcat tomcat 16505 Jun  2 19:08 RUNNING.txt  
drwxr-x--- 2 tomcat tomcat  4096 Jun 23 12:08 bin  
drwx----- 2 tomcat tomcat  4096 Jun  2 19:08 conf  
drwxr-x--- 2 tomcat tomcat  4096 Jun 23 12:08 lib  
drwxr-x--- 2 tomcat tomcat  4096 Jun  2 19:08 logs  
drwxr-x--- 2 tomcat tomcat  4096 Jun 23 12:08 temp  
drwxr-x--- 7 tomcat tomcat  4096 Jun  2 19:08 webapps  
drwxr-x--- 2 tomcat tomcat  4096 Jun  2 19:08 work
```

젠킨스

- 서로 다른 환경 구성하기
 - 톰캣 shell script 파일 실행 권한 부여
 - `sudo sh -c 'chmod +x /opt/tomcat/tomcat-9/bin/*.sh'`

```
선택 ubuntu@ip-172-31-4-50: ~
ubuntu@ip-172-31-4-50:/opt/tomcat$ sudo sh -c 'chmod +x /opt/tomcat/tomcat-9/bin/*.sh'
ubuntu@ip-172-31-4-50:/opt/tomcat$ sudo ls -al tomcat-9/bin
total 896
drwxr-x--- 2 tomcat tomcat 4096 Jun 23 12:08 .
drwxr-xr-x 9 tomcat tomcat 4096 Jun 23 12:08 ..
-rw-r----- 1 tomcat tomcat 34699 Jun  2 19:08 bootstrap.jar
-rw-r----- 1 tomcat tomcat 1664 Jun  2 19:08 catalina-tasks.xml
-rw-r----- 1 tomcat tomcat 16840 Jun  2 19:08 catalina.bat
-rwxr-x--x 1 tomcat tomcat 25294 Jun  2 19:08 catalina.sh
-rw-r----- 1 tomcat tomcat 2123 Jun  2 19:08 ciphers.bat
-rwxr-x--x 1 tomcat tomcat 1997 Jun  2 19:08 ciphers.sh
-rw-r----- 1 tomcat tomcat 211777 Jun  2 19:08 commons-daemon-native.tar.gz
-rw-r----- 1 tomcat tomcat 25410 Jun  2 19:08 commons-daemon.jar
-rw-r----- 1 tomcat tomcat 2040 Jun  2 19:08 configtest.bat
-rwxr-x--x 1 tomcat tomcat 1922 Jun  2 19:08 configtest.sh
```

젠킨스

- 서로 다른 환경 구성하기
 - 톰캣 서비스 포트 변경
 - `sudo vim /opt/tomcat/tomcat-9/conf/server.xml`

```
선택 ubuntu@ip-172-31-4-50: ~  
<Connector port="8888" protocol="HTTP/1.1"  
        connectionTimeout="20000"  
        redirectPort="8443" />  
<!-- A "Connector" using the shared thread pool-->
```

젠킨스

- 서로 다른 환경 구성하기
 - 톰캣 서비스 등록 파일 구성.
 - `sudo vim /etc/systemd/system/tomcat.service`

```
ubuntu@ip-172-31-4-50: ~  
[Unit]  
Description=Tomcat 9 servlet container  
After=network.target  
  
[Service]  
Type=forking  
  
User=tomcat  
Group=tomcat  
  
Environment="JAVA_HOME=/usr/lib/jvm/java-11-openjdk-amd64/"  
Environment="JAVA_OPTS=-Djava.security.egd=file:///dev/urandom -Djava.awt.headless=true"  
  
Environment="CATALINA_BASE=/opt/tomcat/tomcat-9"  
Environment="CATALINA_HOME=/opt/tomcat/tomcat-9"  
Environment="CATALINA_PID=/opt/tomcat/tomcat-9/temp/tomcat.pid"  
Environment="CATALINA_OPTS=-Xms512M -Xmx1024M -server -XX:+UseParallelGC"  
  
ExecStart=/opt/tomcat/tomcat-9/bin/startup.sh  
ExecStop=/opt/tomcat/tomcat-9/bin/shutdown.sh  
  
[Install]  
WantedBy=multi-user.target_
```

젠킨스

- 서로 다른 환경 구성하기
 - 톰캣 서비스 등록 및 서비스 시작
 - `sudo systemctl daemon-reload`
 - `sudo systemctl start tomcat`
 - `sudo systemctl status tomcat`

```
선택 ubuntu@ip-172-31-4-50: ~
ubuntu@ip-172-31-4-50:/opt/tomcat$ sudo systemctl status tomcat
● tomcat.service - Tomcat 9 servlet container
   Loaded: loaded (/etc/systemd/system/tomcat.service; disabled; vendor preset: enabled)
   Active: inactive (dead)
ubuntu@ip-172-31-4-50:/opt/tomcat$ sudo systemctl daemon-reload
ubuntu@ip-172-31-4-50:/opt/tomcat$ sudo systemctl start tomcat
ubuntu@ip-172-31-4-50:/opt/tomcat$ sudo systemctl status tomcat
● tomcat.service - Tomcat 9 servlet container
   Loaded: loaded (/etc/systemd/system/tomcat.service; disabled; vendor preset: enabled)
   Active: active (running) since Thu 2022-06-23 12:20:24 UTC; 3s ago
     Process: 18409 ExecStart=/opt/tomcat/tomcat-9/bin/startup.sh (code=exited, status=0/SUCCESS)
    Main PID: 18426 (java)
       Tasks: 15 (limit: 1134)
      CGroup: /system.slice/tomcat.service
              └─18426 /usr/lib/jvm/java-11-openjdk-amd64/bin/java -Djava.util.logging.config.file:

Jun 23 12:20:23 ip-172-31-4-50 systemd[1]: Starting Tomcat 9 servlet container...
Jun 23 12:20:24 ip-172-31-4-50 startup.sh[18409]: Tomcat started.
Jun 23 12:20:24 ip-172-31-4-50 systemd[1]: Started Tomcat 9 servlet container.
```

젠킨스

- 서로 다른 환경 구성하기
 - 플러그인 설치
 - Jenkins 관리 – Plugin Manager – Publish Over SSH – Install without restart

The screenshot shows the Jenkins web interface. At the top is the Jenkins logo and a search bar. Below the logo, the breadcrumb 'Dashboard > Plugin Manager' is visible. On the left sidebar, there are links for '대시보드로 돌아가기' (Return to Dashboard) and 'Jenkins 관리' (Manage Jenkins). The main area is titled 'Plugin Manager' and contains tabs for '업데이트된 플러그인 목록' (Updated Plugins), '설치 가능' (Installable), '설치된 플러그인 목록' (Installed Plugins), and '고급' (Advanced). The '설치 가능' tab is selected. A search bar contains the text 'publish over ssh'. Below the search bar is a table with the following data:

Install	Name	Released
<input checked="" type="checkbox"/>	Publish Over SSH 1.24 Artifact Uploaders Build Tools Send build artifacts over SSH	4 mo 1 day ago

At the bottom of the table, there are two buttons: 'Install without restart' and 'Download now and install after restart'. To the right of these buttons, it says 'Update information obtained: 2 hr 7 min ago' and a '지금 확인' (Check Now) button.

젠킨스

- 서로 다른 환경 구성하기
 - 플러그인 설정
 - AWS에서 배포해준 xxx.pem 파일 내용을 key에 입력

Dashboard > Configure System >

Publish over SSH

Jenkins SSH Key ?

Passphrase ?

Path to key ?

Key ?

NnaOJFhxTacNiJzdxjkKjwfsg6qur8Ya6R3SEWbxMoVDmDEkXHc0/hNARfSU+J7j
OAJJoa+a+UrID2CMn5VbjbD/8WstBzgJDGPfv4Zlbyhs7BW5o4Rzz8V1xaaPEzEd
rAZ3AoGBANe1ArhcnHCuv7h5DnZwlgUbQuokrErnWSBFNzKHDfpf6bNyZYt4Ruy
ILKuh2do2gxm8YMXvbMZLHdy34IFRtvBIA659mJSBqDnR2iBBtD/2qy8YJaPOmX6
ZpnpmnDTMLLuQ9hRkVw5STLA+xwtUAiX6j9e8DrGhG8aTxt9bsTt
-----END RSA PRIVATE KEY-----

젠킨스

- 서로 다른 환경 구성하기
 - 플러그인 설정
 - AWS서버 퍼블릭 아이피, 서버 계정명, 서버 홈디렉토리 설정.

Dashboard > Configure System >

SSH Servers

SSH Server

Name ?
ec2

Hostname ?
18.236.119.105

Username ?
ubuntu

Remote Directory ?
/home/ubuntu

고급...

Test Configuration

젠킨스

- 서로 다른 환경 구성하기
 - 플러그인 설정
 - 전에 설정한 SSH Server 이름 확인



젠킨스

- 서로 다른 환경 구성하기
 - 플러그인 설정
 - Source files : Jenkins Workspace\프로젝트명\
 - Exec command : Home Directory 배치 후 Tomcat webapps Directory로 이동

The screenshot shows the Jenkins web interface for configuring a job named 'first'. The breadcrumb navigation at the top reads 'Dashboard > first >'. Below this, a series of tabs are visible: 'General', '소스 코드 관리', '빌드 유발', '빌드 환경', 'Build', and '빌드 후 조치'. The '빌드 후 조치' (Build After) tab is currently selected. The main content area is titled 'Transfers' and contains a dashed border box with a red 'X' icon in the top right corner. Inside this box, there are four configuration sections:

- Transfer Set**: A section header with a red 'X' icon.
- Source files**: A text input field containing 'target/First*.war'.
- Remove prefix**: A text input field containing 'target'.
- Remote directory**: An empty text input field.
- Exec command**: A text input field containing the command: `sudo mv /home/ubuntu/First*.war /opt/tomcat/tomcat-9/webapps/First.war >> First.log &`

젠킨스

- 서로 다른 환경 구성하기
 - 플러그인 설정
 - 퍼블릭 IPv4 DNS:포트번호/ContextPath



젠킨스

- 푸시 메커니즘

- 외부 서버에서 젠킨스를 구성.
- github로 push를 하게 되면 hook 설정을 통해 자동빌드 수행
- hook 설정은 공인IP만 가능.
- 레포지토리에서 다운로드 받기 위해 경로 설정
- `wget -q -O - https://pkg.jenkins.io/debian-stable/jenkins.io.key | sudo apt-key add -`
- `sudo sh -c 'echo deb https://pkg.jenkins.io/debian-stable binary/ > \`
- `/etc/apt/sources.list.d/jenkins.list'`

```
ubuntu@ip-172-31-35-7: ~  
ubuntu@ip-172-31-35-7:~$ wget -q -O - https://pkg.jenkins.io/debian-stable/jenkins.io.key | sudo apt-key add -  
OK  
ubuntu@ip-172-31-35-7:~$ sudo sh -c 'echo deb https://pkg.jenkins.io/debian-stable binary/ > \  
> /etc/apt/sources.list.d/jenkins.list'  
ubuntu@ip-172-31-35-7:~$ sudo apt-get update  
Hit:1 http://ap-northeast-2.ec2.archive.ubuntu.com/ubuntu bionic InRelease  
Get:2 http://ap-northeast-2.ec2.archive.ubuntu.com/ubuntu bionic-updates InRelease [88.7 kB]  
Get:3 http://ap-northeast-2.ec2.archive.ubuntu.com/ubuntu bionic-backports InRelease [74.6 kB]  
Ign:4 https://pkg.jenkins.io/debian-stable binary/ InRelease  
Get:5 https://pkg.jenkins.io/debian-stable binary/ Release [2044 B]  
Get:6 https://pkg.jenkins.io/debian-stable binary/ Release.gpg [833 B]
```

젠킨스

- 푸시 메커니즘
 - 외부 서버에서 젠킨스를 구성.
 - jdk 구성

```
ubuntu@ip-172-31-35-7: ~
```

```
ubuntu@ip-172-31-35-7:~$ sudo apt-get install openjdk-8-jdk
```

```
ubuntu@ip-172-31-35-7: ~
```

```
ubuntu@ip-172-31-35-7:~$ which javac
```

```
/usr/bin/javac
```

```
ubuntu@ip-172-31-35-7:~$ ls -al /usr/bin/javac
```

```
lrwxrwxrwx 1 root root 23 Jun 24 05:34 /usr/bin/javac -> /etc/alternatives/javac
```

```
ubuntu@ip-172-31-35-7:~$ ls -al /etc/alternatives/javac
```

```
lrwxrwxrwx 1 root root 43 Jun 24 05:34 /etc/alternatives/javac -> /usr/lib/jvm/java-8-openjdk-amd64/bin/javac
```

```
ubuntu@ip-172-31-35-7:~$ ls -al /usr/lib/jvm/java-8-openjdk-amd64/bin/javac
```

```
-rwxr-xr-x 1 root root 6312 Nov  3 2021 /usr/lib/jvm/java-8-openjdk-amd64/bin/javac
```

```
ubuntu@ip-172-31-35-7:~$ vim /etc/profile
```

```
ubuntu@ip-172-31-35-7:~$ ubuntu@ip-172-31-35-7:~$
```

```
ubuntu@ip-172-31-35-7:~$ tail -1 /etc/profile
```

```
export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64/
```

```
ubuntu@ip-172-31-35-7:~$ source /etc/profile
```

젠킨스

- 푸시 메커니즘
 - 외부 서버에서 젠킨스를 구성.
 - `sudo apt-get install jenkins`

```
ubuntu@ip-172-31-35-7: ~  
ubuntu@ip-172-31-35-7:~$ sudo apt-get install jenkins  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
The following NEW packages will be installed:  
  jenkins  
0 upgraded, 1 newly installed, 0 to remove and 7 not upgraded.  
Need to get 87.7 MB of archives.  
After this operation, 91.3 MB of additional disk space will be used.  
Get:1 https://pkg.jenkins.io/debian-stable binary/ jenkins 2.346.1 [87.7 MB]  
19% [1 jenkins 21.0 MB/87.7 MB 24%]
```


젠킨스

- 푸시 메커니즘
 - 외부 서버에서 젠킨스를 구성.
 - `sudo vi /etc/default/jenkins`
 - `HTTP_PORT=9999`
 - `sudo vim /lib/systemd/system/jenkins.service`
 - `Environment="JENKINS_PORT=9999"`

```
선택 ubuntu@ip-172-31-35-7: ~  
ubuntu@ip-172-31-35-7:~$ sudo vim /etc/default/jenkins  
# port for HTTP connector (default 8080; disable with -1)  
HTTP_PORT=9999
```

```
ubuntu@ip-172-31-35-7: ~  
ubuntu@ip-172-31-35-7:~$ sudo vim /lib/systemd/system/jenkins.service  
# directive below.  
Environment="JENKINS_PORT=9999"
```

젠킨스

- 푸시 메커니즘
 - 외부 서버에서 젠킨스를 구성.
 - `sudo systemctl stop jenkins`
 - `sudo systemctl daemon-reload`
 - `sudo systemctl start jenkins`
 - `sudo systemctl status jenkins`

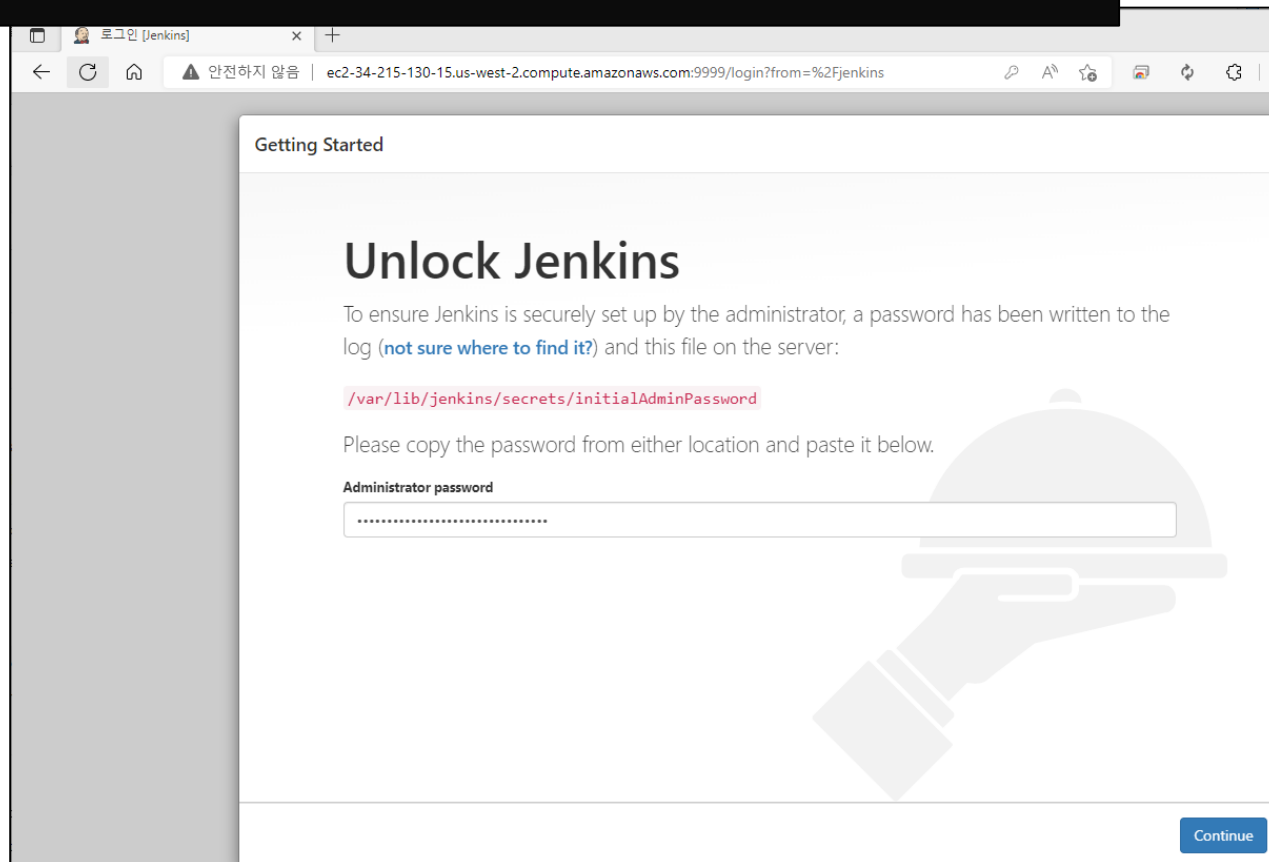
선택 ubuntu@ip-172-31-35-7: ~

```
ubuntu@ip-172-31-35-7:~$ sudo systemctl daemon-reload
ubuntu@ip-172-31-35-7:~$ sudo systemctl start jenkins
ubuntu@ip-172-31-35-7:~$ sudo systemctl status jenkins
● jenkins.service - Jenkins Continuous Integration Server
```

젠킨스

- 푸시 메커니즘
 - 외부 서버에서 젠킨스를 구성.
 - 젠킨스 접근하여 기본설정 및 서로 다른 환경 구성하기(Publish Over SSH)

```
ubuntu@ip-172-31-4-50: ~  
ubuntu@ip-172-31-4-50:~$ sudo cat /var/lib/jenkins/secrets/initialAdminPassword  
904d54e1cec84a3e8c45a9258e16eb64
```

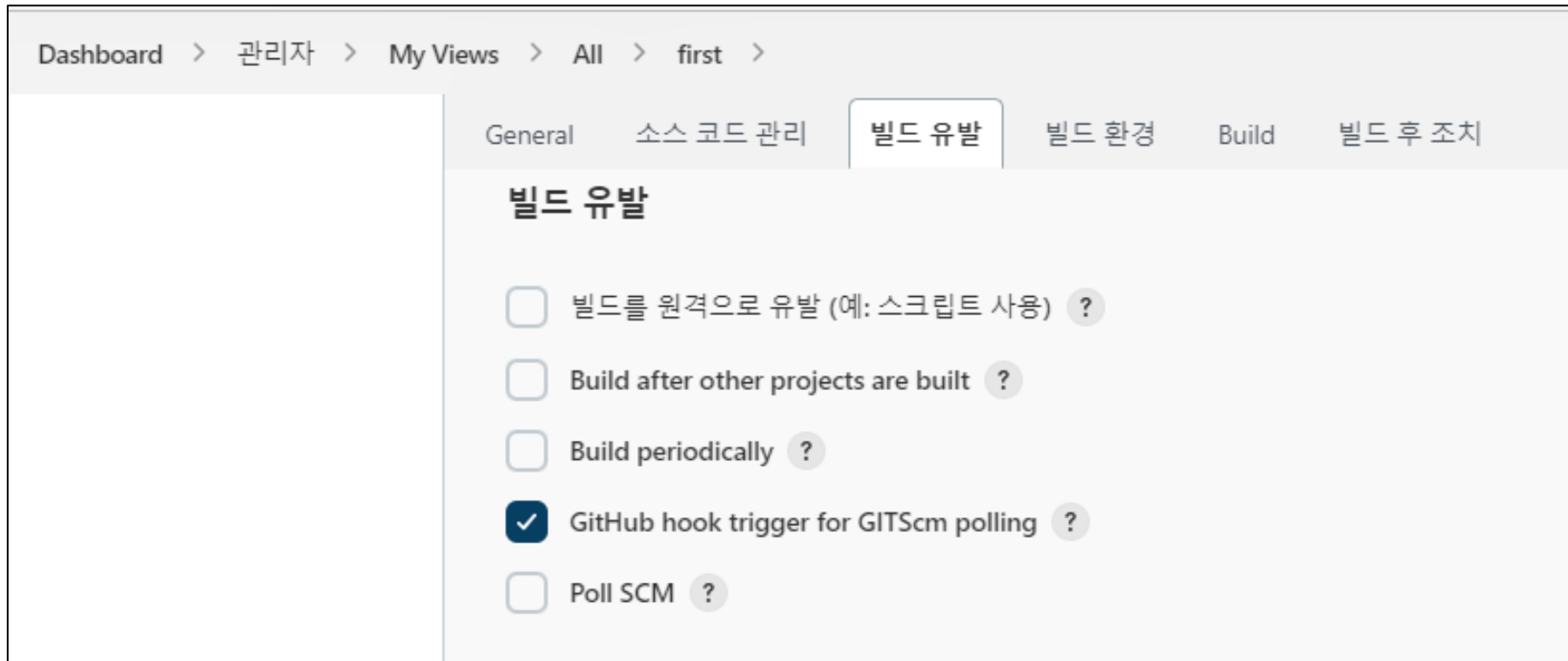


젠킨스

- 실습
 - 전역 도구 설정 – 메이븐 또는 그레들 설정
 - 플러그인 관리 – Publish Over SSH 설치
 - 전역 도구 설정 – SSH Servers 설정
 - New Item - 프로젝트 생성
 - 프로젝트 구성 – 소스 코드 관리 – 깃허브 설정
 - 프로젝트 구성 – 빌드 – 메이븐 또는 그레들 설정
 - 프로젝트 구성 – 빌드 후 조치 – Send Build artifacts over SSH 설정
 - 수동으로 프로젝트 빌드
 - AWS EC2로 배포 확인 및 웹 브라우저로 접근 확인

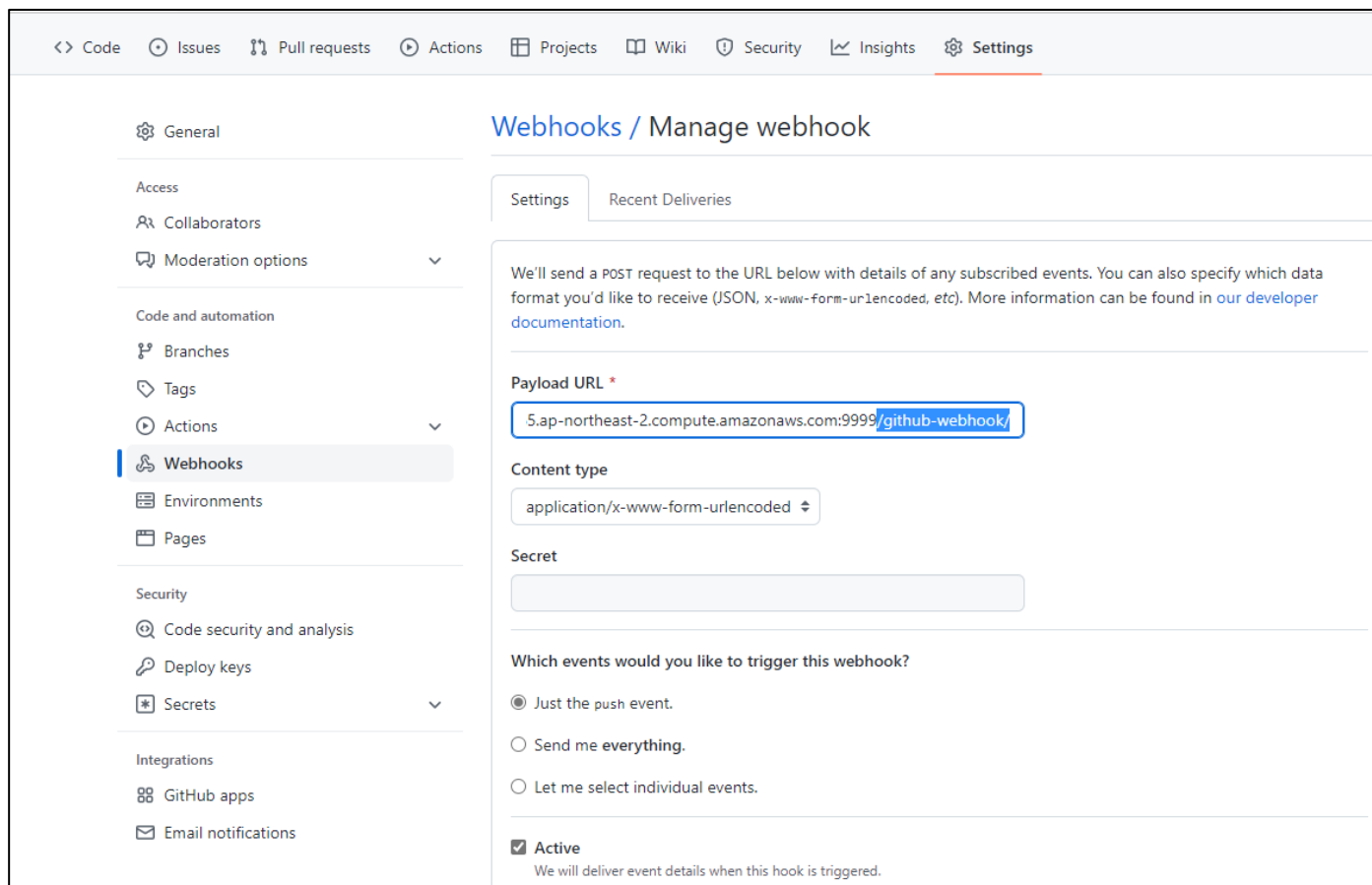
젠킨스

- 푸시 메커니즘
 - 프로젝트 – 구성 – 빌드 유발 – GitHub hook trigger for GITScm polling 체크



젠킨스

- 푸시 메커니즘
 - 깃허브 – Settings – Webhooks – Payload URL
 - `http://ec2-15-164-251-165.ap-northeast-2.compute.amazonaws.com:9999/github-webhook/`



젠킨스

- 푸시 메커니즘
 - 깃허브로 Push 후 프로젝트의 GitHub Logs에서 확인 가능.

Last GitHub Push

```
Started on Jun 24, 2022 6:21:07 AM
Started by event from 140.82.115.117 => http://ec2-15-164-251-165.ap-northeast-2.compute.amazonaws.com:9999/github-webhook/ on Fri Jun
24 06:21:06 UTC 2022
Using strategy: Default
[poll] Last Built Revision: Revision fa3873eba9623c554c95342a7813211d3edecf41 (refs/remotes/origin/master)
The recommended git tool is: NONE
No credentials specified
> git --version # timeout=10
> git --version # 'git version 2.17.1'
> git ls-remote -h -- https://github.com/kyesCare/test.git # timeout=10
Found 1 remote heads on https://github.com/kyesCare/test.git
[poll] Latest remote head revision on refs/heads/master is: 9912ec5f72ef9592862baf8a39bda0916821d683
Done. Took 0.36 sec
Changes found
```

젠킨스

- 푸시 메커니즘
 - 웹서버 접근 시 변경 확인 가능.

