



## 5. AWS EC2 Spring Boot

*AWS EC2 Instance ( Cloud Server )에 실행환경 설정 및  
Spring Boot Project를 Build/Deploy 한다*








## EC2 Instance ( Cloud Server ) 실행환경 설정 및 Spring Boot Project Deploy

1. RDP를 이용해 EC2 Instance ( Server ) 에 접속
2. 방화벽 인바운드 80 port 추가
3. JDK 설치 / 설정
4. Local의 Spring Boot Project 설정 : application.properties
5. Maven build 를 통한 실행파일( jar ) 생성
6. EC2 Server에 Deploy ( Deploy 란 프로그램을 서버에 설치하여 서비스를 제공하는 것을 의미 )
7. Browser 로 public IP or DNS 로 접근해 서비스 확인

## RDP를 이용해 EC2 Instance ( Server ) 에 접속

크 (C:) > kosta238 > aws-ec2-cloud

이름

-  1-AWS-Cloud-Computing-FreeTier-가입.pdf
-  2-AWS-EC2-Instance.pdf
-  3-AWS-EC2-Instance-Server-설정.pdf
-  FeelGoodWebService.pem
-  FeelGoodWebService.rdp
-  MySpringFeelGoodServer.pem
-  MySpringFeelGoodServer.rdp
-  클라우드-AWS-개념.txt

# EC2 Server 의 80 port 방화벽 인바운드 설정

← → ▾ ↑ > 제어판 > 시스템 및 보안 > Windows Defender 방화벽

제어판 홈

Windows Defender 방화벽을  
통해 앱 또는 기능 허용

알림 설정 변경

Windows Defender 방화벽 설  
정 또는 해제

기본값 복원

고급 설정


네트워크 문제 해결


## Windows Defender 방화벽을 사용하여 PC 보호

Windows Defender 방화벽은 해커나 악성 소프트웨어가 인터넷 또는 네트워크를 통해 PC에 액세스하는 것  
을 방지해 줍니다.

 **개인 네트워크(R)** 연결됨 

네트워크의 사용자 및 장치를 알고 있으며 신뢰할 수 있는 홈 네트워크 또는 회사 네트워크입니다.

Windows Defender 방화벽 상태:	컴
들어오는 연결:	허용되는 앱 목록에 없는 모든 앱 연결 차단
활성 개인 네트워크:	 mj
알림 상태:	Windows Defender 방화벽이 새 앱을 차단할 때 알 림

 **게스트 또는 공용 네트워크(P)** 연결 안 됨 

# EC2 Server 의 80 port 방화벽 인바운드 설정

고급 보안이 포함된 Windows Defender 방화벽

파일(F) 동작(A) 보기(V) 도움말(H)



로컬 컴퓨터의 고급 보안이 포함된 Windows Defender 방화벽

인바운드 규칙

- 인바운드 규칙
- 아웃바운드 규칙
- 연결 보안 규칙
- 모니터링

새 인바운드 규칙 마법사

## 규칙 종류

만들려는 방화벽 규칙 종류를 선택합니다.

### 단계:

- 규칙 종류
- 프로토콜 및 포트
- 작업
- 프로필
- 이름

만들려는 규칙 종류는 무엇입니까?

- ☐ 프로그램(P)  
프로그램의 연결을 제어하는 규칙
- ☒ 포트(O)  
TCP 또는 UDP 포트의 연결을 제어하는 규칙
- ☐ 미리 정의됨(E):  
@FirewallAPI.dll,-80200  
Windows 환경의 연결을 제어하는 규칙
- ☐ 사용자 지정(C)  
사용자 지정 규칙

# EC2 Server 의 80 port 방화벽 인바운드 설정

고급 보안이 포함된 Windows Defender 방화벽

파일(F) 동작(A) 보기(V) 도움말(H)



로컬 컴퓨터의 고급 보안이 포함된 Windows Defender 방화벽

인바운드 규칙

- 인바운드 규칙
- 아웃바운드 규칙
- 연결 보안 규칙
- 모니터링

새 인바운드 규칙 마법사

프로토콜 및 포트

이 규칙을 적용할 프로토콜과 포트를 지정하십시오.

단계:

- 규칙 종류
- 프로토콜 및 포트
- 작업
- 프로필
- 이름

이 규칙은 TCP에 적용됩니까, UDP에 적용됩니까?

- ☒ TCP(T)
- ☐ UDP(U)

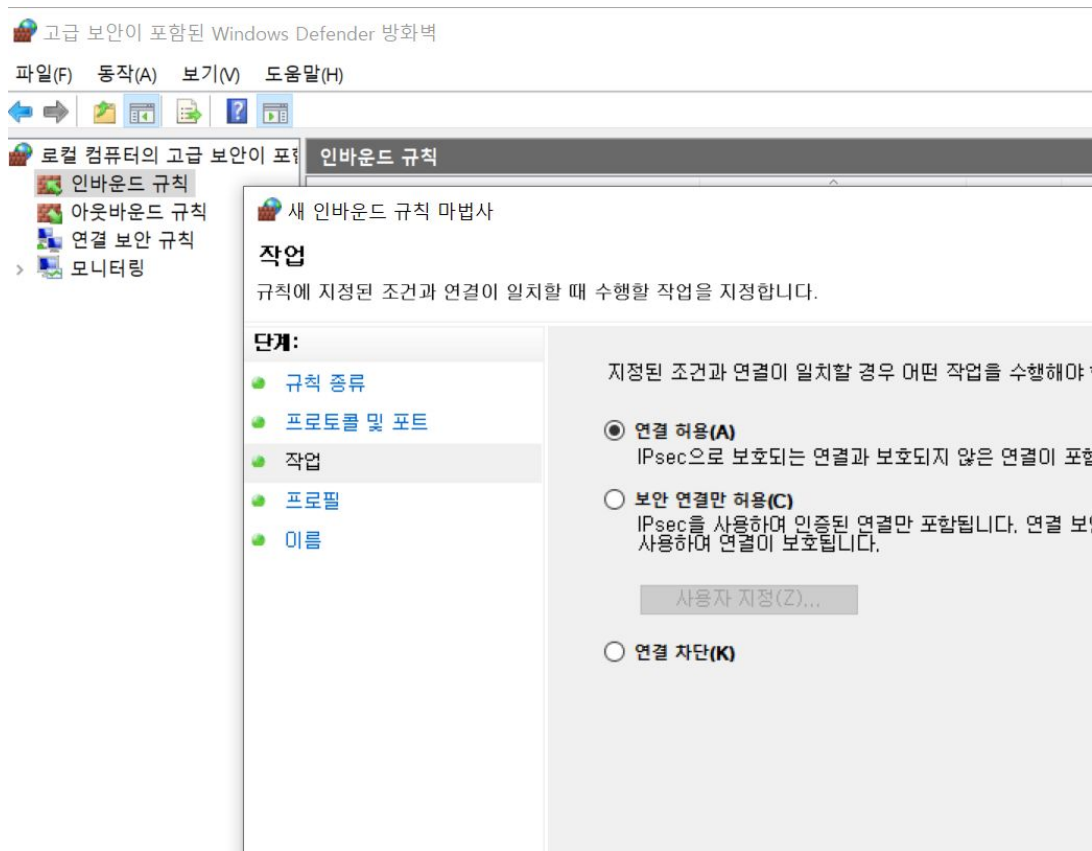
이 규칙은 모든 로컬 포트에 적용됩니까, 특정 로컬 포트에 적용됩니까?

- ☐ 모든 로컬 포트(A)
- ☒ 특정 로컬 포트(S):

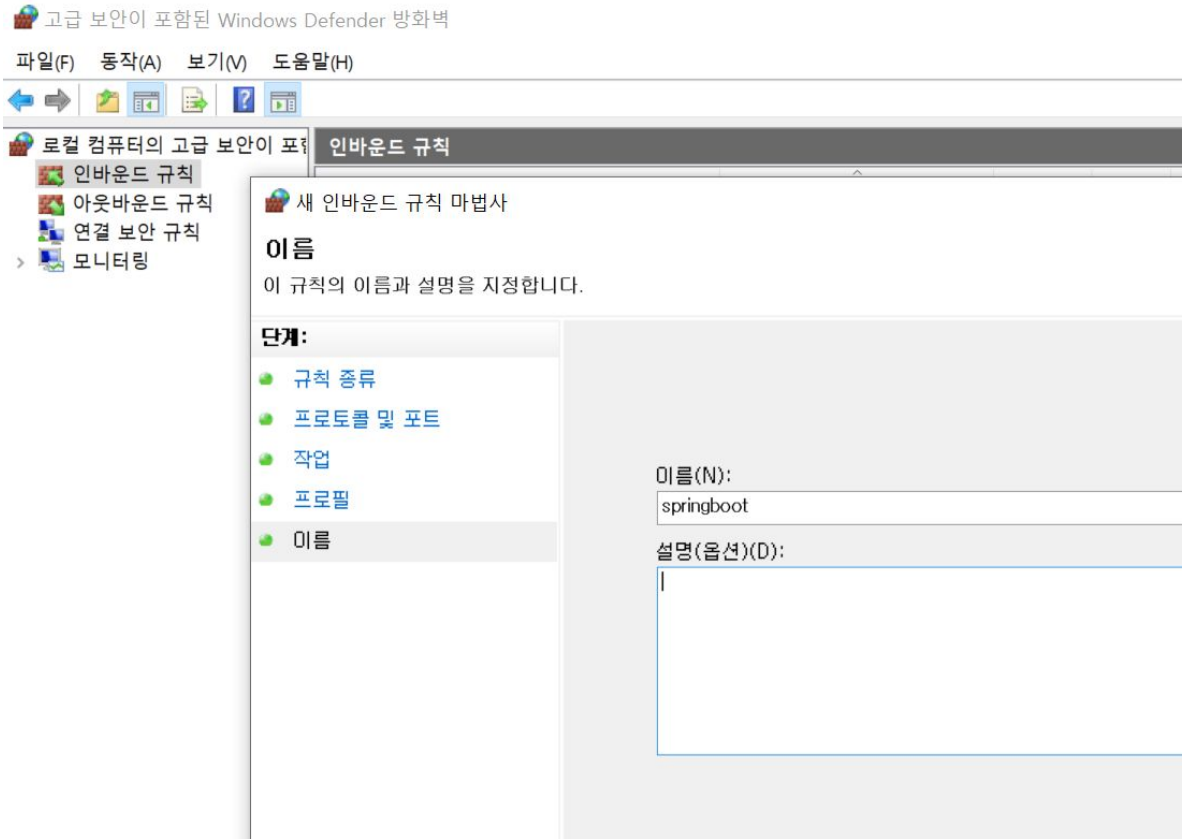
80

예: 80, 443, 5000

# EC2 Server 의 80 port 방화벽 인바운드 설정



# EC2 Server 의 80 port 방화벽 인바운드 설정





# EC2 Server 의 80 port 방화벽 인바운드 설정

FeelGoodWebService - ec2-52-79-240-99.ap-northeast-2.compute.amazonaws.com: 원격 데스크톱 연결

고급 보안이 포함된 Windows Defender 방화벽

파일(F) 동작(A) 보기(V) 도움말(H)

휴지통

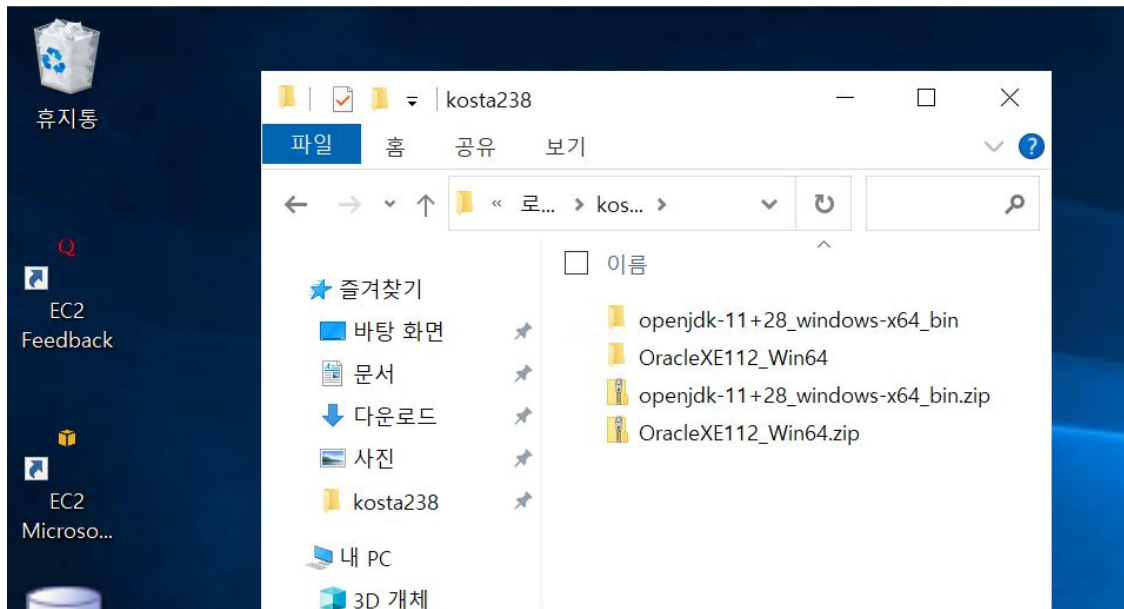
로컬 컴퓨터의 고급 보안이 포함된 Windows Defender 방화벽

- 인바운드 규칙
- 아웃바운드 규칙
- 연결 보안 규칙
- 모니터링

이름	그룹	프로필	사용	작업	
oracle		모두	예	허용	C
springboot		모두	예	허용	C
AllJoyn 라우터(TCP-In)	AllJoyn 라우터	도메인,...	예	허용	C
AllJoyn 라우터(UDP-In)	AllJoyn 라우터	도메인,...	예	허용	C
BranchCache 콘텐츠 검색(HTTP-In)	BranchCache - 콘텐츠 검색(...	모두	아니요	허용	C
BranchCache 피어 검색(WSD-In)	BranchCache - 피어 검색(W...	모두	아니요	허용	C
BranchCache 호스트 캐시 서버(HTTP-In)	BranchCache - 호스트 캐시 ...	모두	아니요	허용	C
COM+ 네트워크 액세스(DCOM-In)	COM+ 네트워크 액세스	모두	아니요	허용	C
COM+ 원격 관리(DCOM-In)	COM+ 원격 관리	모두	아니요	허용	C
Delivery Optimization (TCP-In)	Delivery Optimization	모두	예	허용	C
Delivery Optimization (UDP-In)	Delivery Optimization	모두	예	허용	C

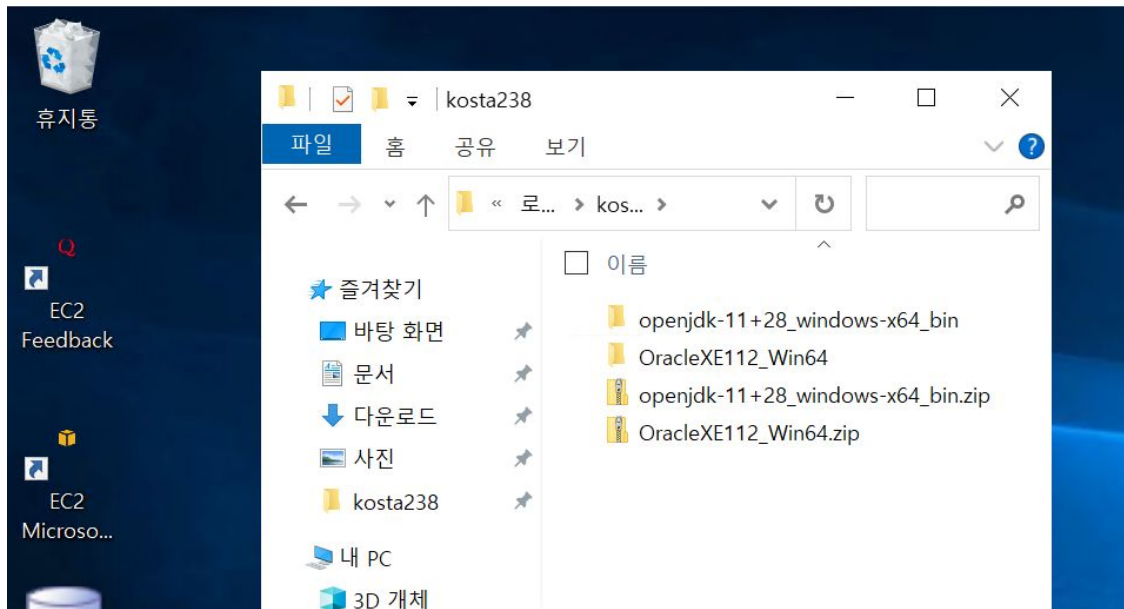
# AWS EC2 Server 에 jdk 를 배치, 압축해제

FeelGoodWebService - ec2-52-79-240-99.ap-northeast-2.compute.amazonaws.com: 원격 데스크톱 연결



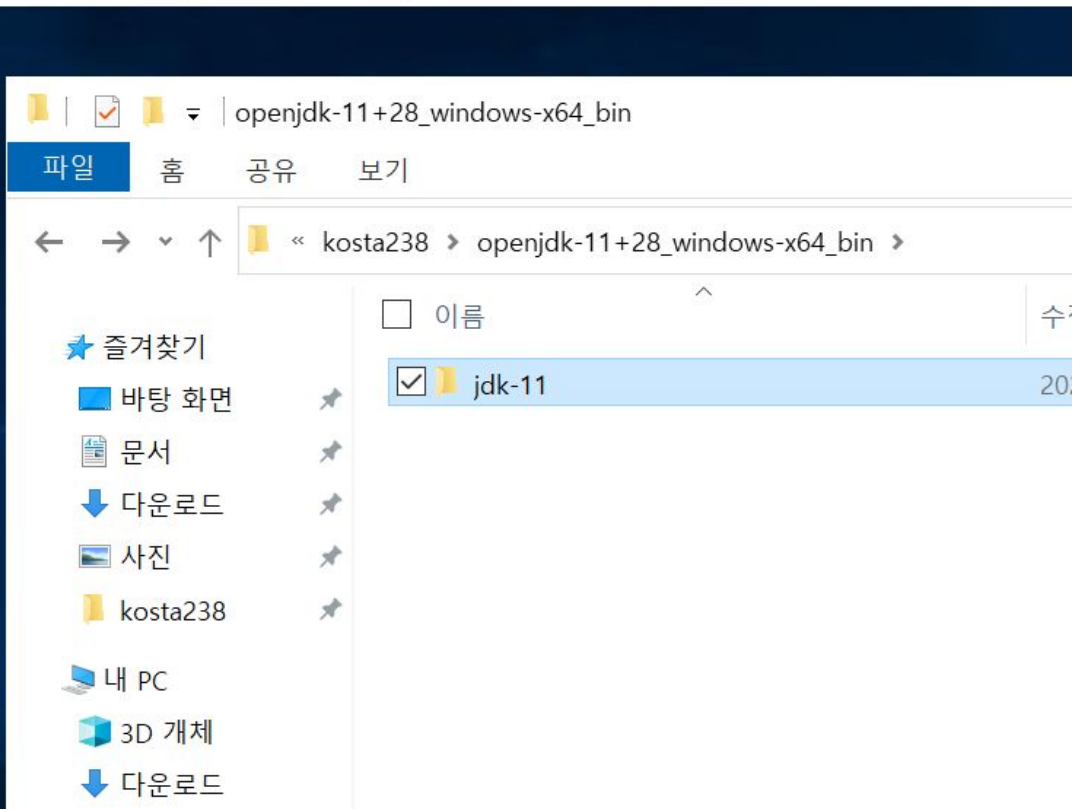
# AWS EC2 Server 에 jdk 를 배치, 압축해제

FeelGoodWebService - ec2-52-79-240-99.ap-northeast-2.compute.amazonaws.com: 원격 데스크톱 연결



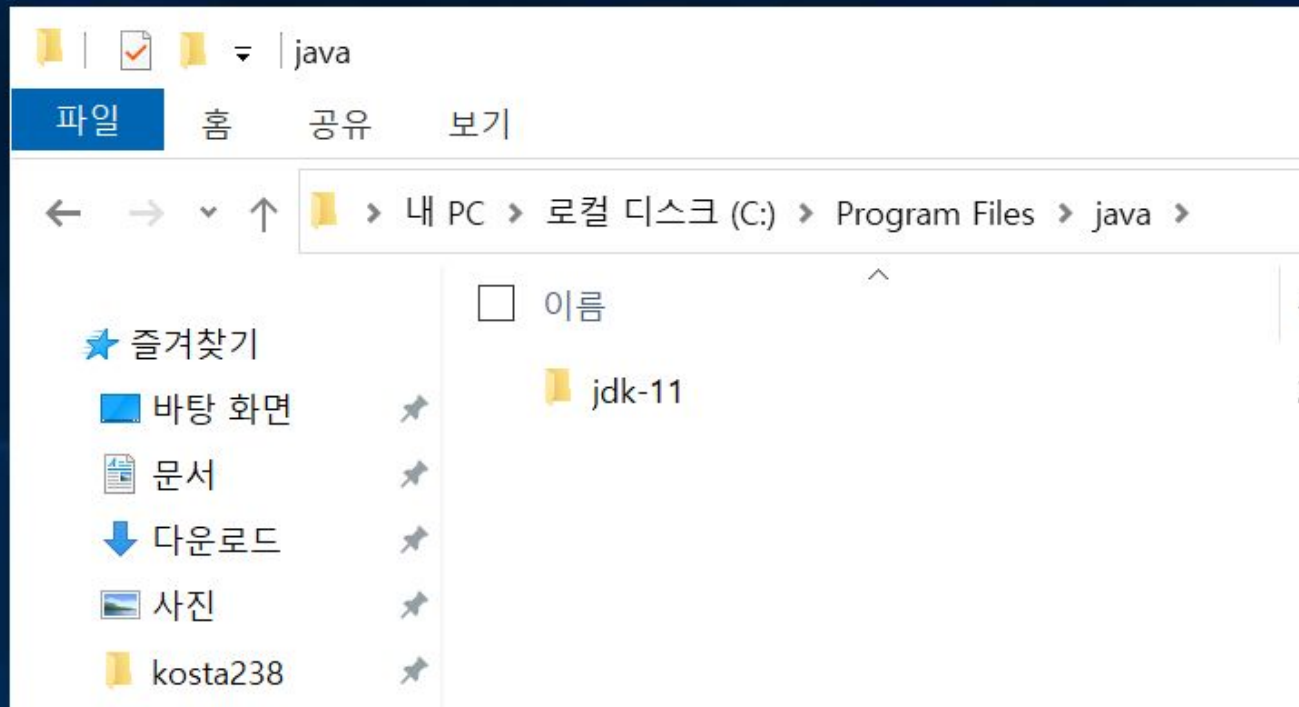
## jdk 압축해제한 후 jdk 11 잘라내기해서

service - ec2-52-79-240-99.ap-northeast-2.compute.amazonaws.com: 원격 데스크톱 연결



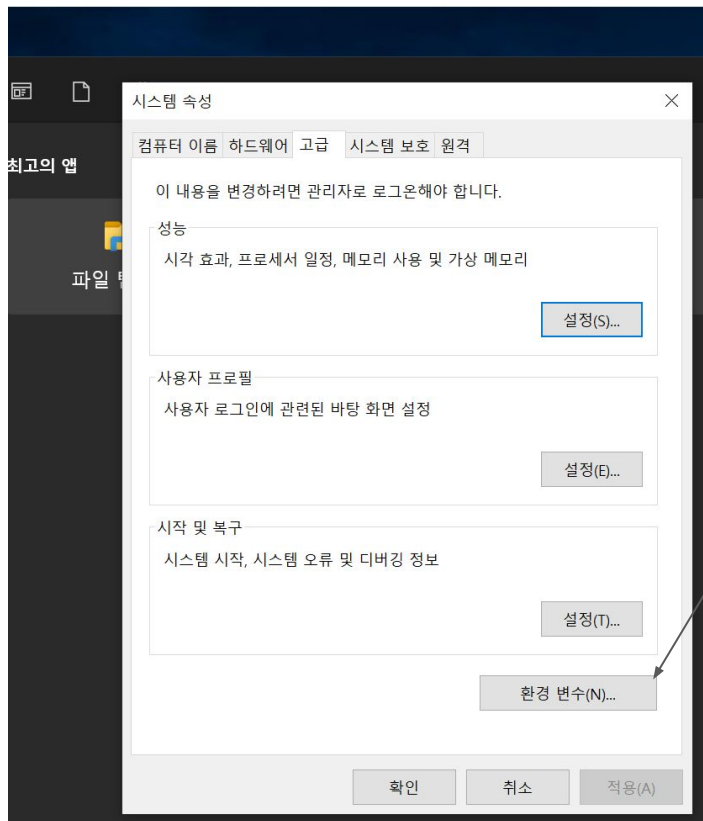
jdk 11 잘라내기해서 아래 경로에 배치한다

service - ec2-52-79-240-99.ap-northeast-2.compute.amazonaws.com: 원격 데스크톱 연결

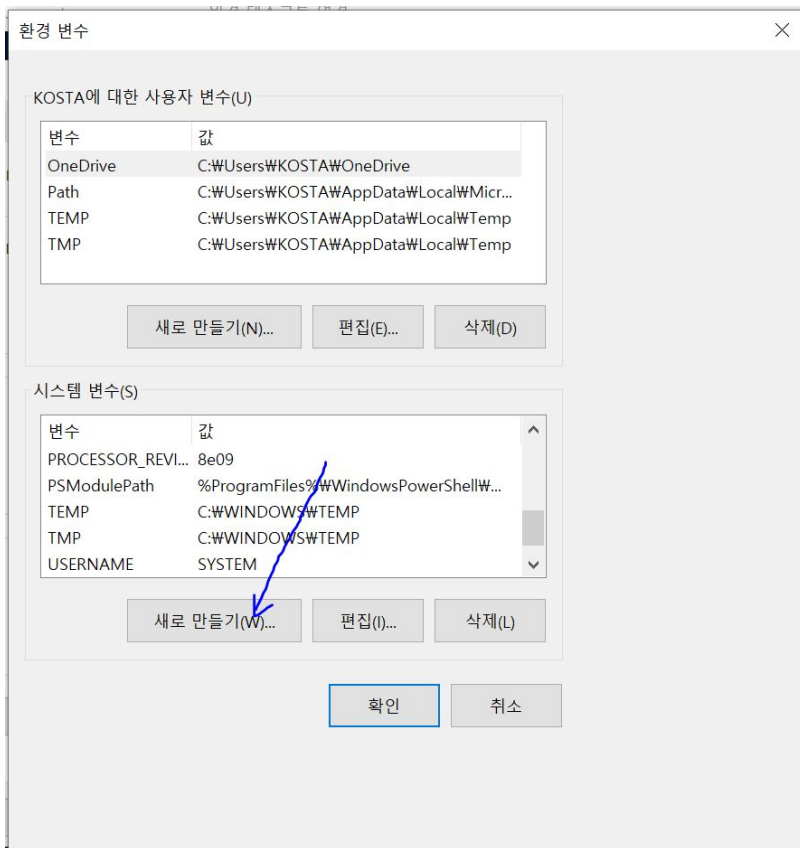


## 왼쪽 하단에 윈도우 검색창에 시스템 환경 변수 편집을 검색해서

odWebService - ec2-52-79-240-99.ap-northeast-2.compute.amazonaws.com: 원격 데스크톱 연결



아래편 시스템 변수 -> 새로 만들기 클릭



## JAVA\_HOME 추가

시스템 변수 편집

변수 이름(N): JAVA\_HOME

변수 값(V): C:\Program Files\Java\jdk-11\

디렉터리 찾아보기(D)...    파일 찾아보기(F)...    **확인**    취소



## Path 에 %JAVA\_HOME%\bin 추가 후 상단으로 이동시킨다

WebService - ec2-52-79-240-99.ap-northeast-2.compute.amazonaws.com: 원격 데스크톱 연결

시스템 속성

컴퓨터 이름 하드웨어 고급 원격

이 내용을 변경하려면 관리자로 로그인해야 합니다.

성능

시각 효과, 프로세서 일정, 메모리 사용 및 기타 설정

사용자 프로필

사용자 로그인에 관련된 바탕 화면 설정

환경 변수

환경 변수 편집

%JAVA\_HOME%\bin  
C:\Oracle\Java\jdk1.8.0\_101\bin  
%SystemRoot%\system32  
%SystemRoot%\system32\wbem  
%SYSTEMROOT%\System32\WindowsPowerShell\v1.0\  
%SYSTEMROOT%\System32\OpenSSH  
C:\Program Files\Amazon\cfn-bootstrap\awscli.exe

새로 만들기(N)  
편집(E)  
찾아보기(B)...  
삭제(D)  
위로 이동(U)  
아래로 이동(O)  
테스트 편집(T)

Hostname : EC2-52-79-240-99.ap-northeast-2.compute.amazonaws.com  
Instance ID : i-05...  
Private IP Address : ...  
Public IP Address : ...  
Instance Size : t2...  
Availability Zone : ap-northeast-2a  
Architecture : ARM64  
Total Memory : 16 GB  
Network : Low to medium utilization

## 명령 프롬프트 창에서 JDK 설정 확인

WebService - ec2-52-79-240-99.ap-northeast-2.compute.amazonaws.com: 원격 데스크톱 연결

C:\> Administrator: 명령 프롬프트

Microsoft Windows [Version 10.0.20348.709]

(c) Microsoft Corporation. All rights reserved.

C:\Users\Administrator>java -version

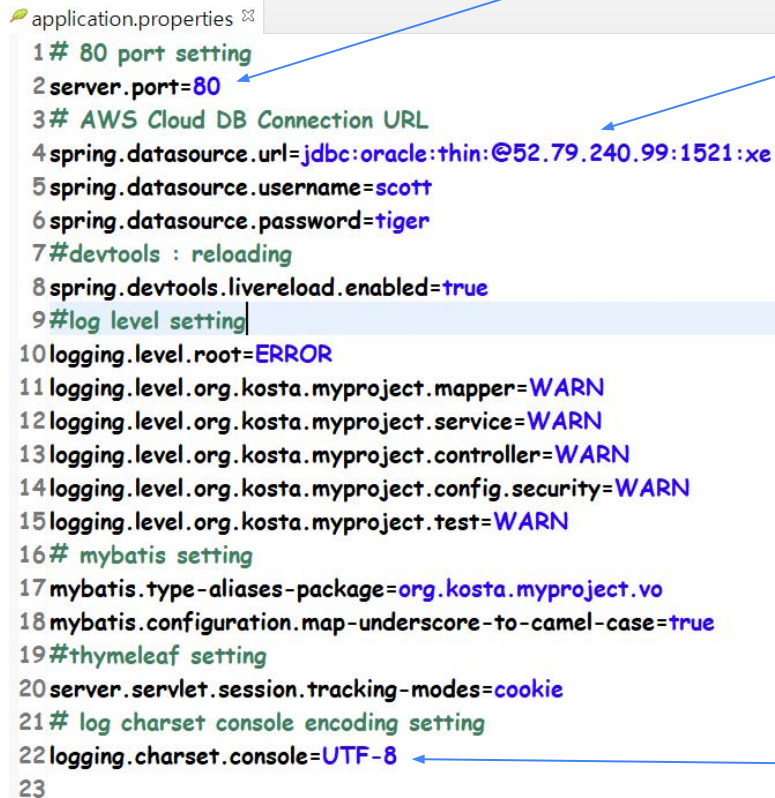
openjdk version "11" 2018-09-25

OpenJDK Runtime Environment 18.9 (build 11+28)

OpenJDK 64-Bit Server VM 18.9 (build 11+28, mixed mode)

C:\Users\Administrator>\_

# Spring Boot Project 설정



The image shows a code editor window with the file name 'application.properties'. The code contains various configuration properties for a Spring Boot application. There are three blue arrows pointing to specific lines: one to line 2 (server.port=80), one to line 3 (comment about AWS Cloud DB), and one to line 22 (logging.charset.console=UTF-8). The code is as follows:

```
application.properties
1 # 80 port setting
2 server.port=80
3 # AWS Cloud DB Connection URL
4 spring.datasource.url=jdbc:oracle:thin:@52.79.240.99:1521:xe
5 spring.datasource.username=scott
6 spring.datasource.password=tiger
7 #devtools : reloading
8 spring.devtools.livereload.enabled=true
9 #log level setting
10 logging.level.root=ERROR
11 logging.level.org.kosta.myproject.mapper=WARN
12 logging.level.org.kosta.myproject.service=WARN
13 logging.level.org.kosta.myproject.controller=WARN
14 logging.level.org.kosta.myproject.config.security=WARN
15 logging.level.org.kosta.myproject.test=WARN
16 # mybatis setting
17 mybatis.type-aliases-package=org.kosta.myproject.vo
18 mybatis.configuration.map-underscore-to-camel-case=true
19 #thymeleaf setting
20 server.servlet.session.tracking-modes=cookie
21 # log charset console encoding setting
22 logging.charset.console=UTF-8
23
```

## Spring Boot Project Executable Jar build ( 단독으로 실행 가능한 Jar file 을 생성 )

(참고 : JAR => Java Archive , WAR => Web Application Archive )

Spring Boot Project Maven pom.xml의 build 부분을 확인한다.

이 부분이 실행가능한 Jar를 build 하는 설정임

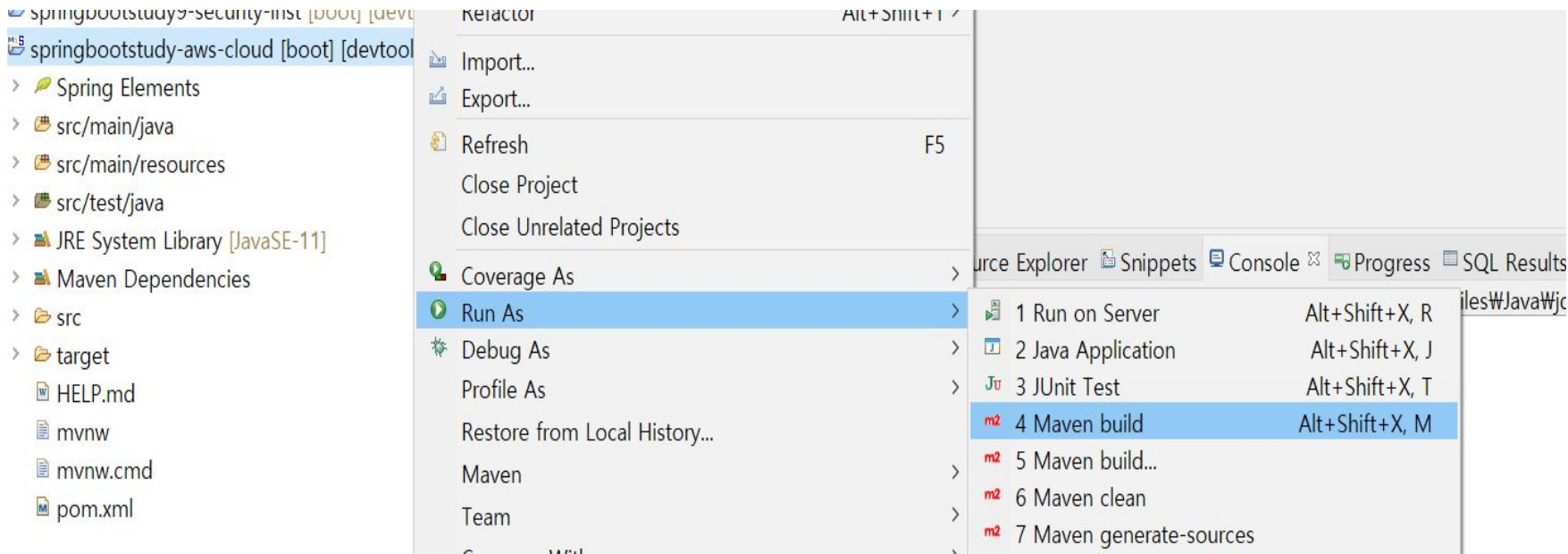
Build(빌드)란 소스코드를 실행가능한  
산출물로 만드는 과정을 말함  
Maven은 자바 프로젝트의 build를  
자동화해주는 Tool 중의 하나

```
springbootstudy-aws-cloud/pom.xml
72     </dependency>
73 </dependencies>
74
75 <build>
76   <plugins>
77     <plugin>
78       <groupId>org.springframework.boot</groupId>
79       <artifactId>spring-boot-maven-plugin</artifactId>
80     <configuration>
81       <excludes>
82         <exclude>
83           <groupId>org.projectlombok</groupId>
84           <artifactId>lombok</artifactId>
85         </exclude>
86       </excludes>
87     </configuration>
88   </plugin>
89 </plugins>
90 </build>
91
92 </project>
93
```

## Spring Boot Project Executable Jar build ( 단독으로 실행 가능한 Jar file 을 생성 )

( 참고 : JAR => Java Archive , WAR => Web Application Archive )

Spring Boot Project 선택-> 마.오 -> Run AS -> Maven build

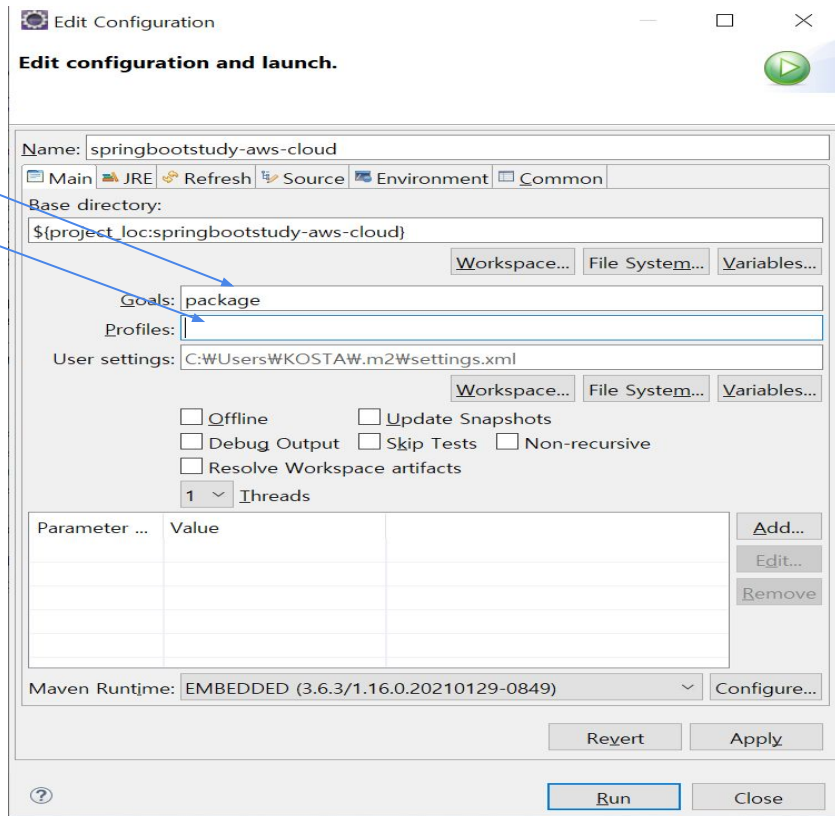


## Spring Boot Project Executable Jar build ( 단독으로 실행 가능한 Jar file 을 생성 )

Goals : package 입력

Profiles : pom.xml 삭제

Apply 후 Run



## Spring Boot Project Executable Jar build ( 단독으로 실행 가능한 Jar file 을 생성 )

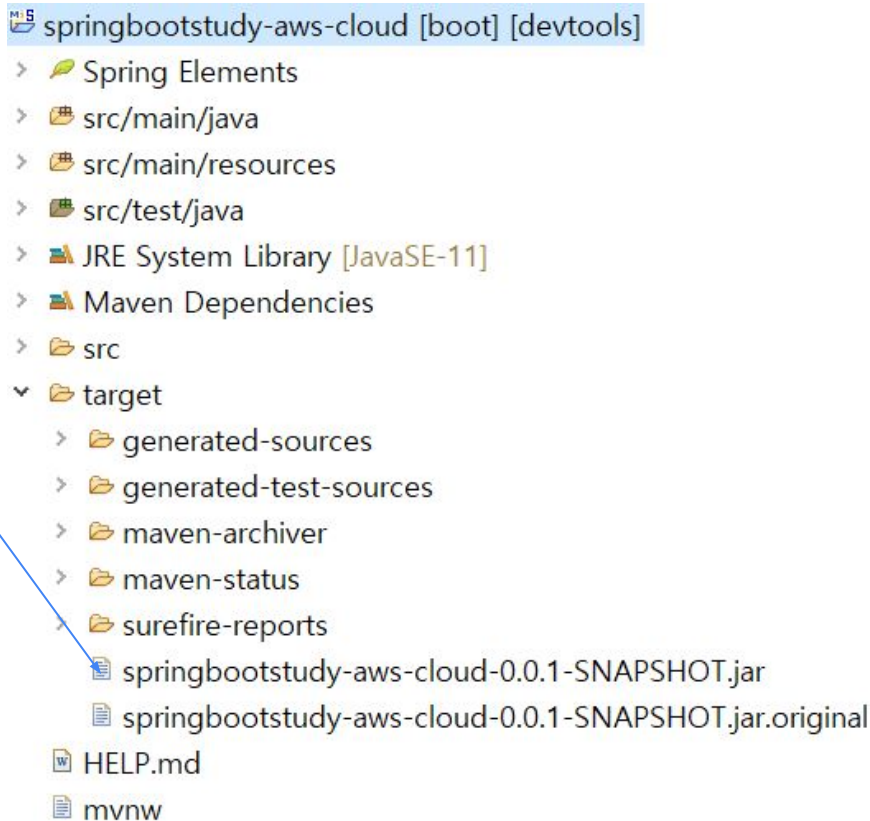
Console에 아래와 같이 build success 메시지가 나옴

```
[INFO]
[INFO] --- maven-jar-plugin:3.2.2:jar (default-jar) @ springbootstudy-aws-cloud ---
[INFO] Building jar: C:\kosta238\spring-workspace2\springbootstudy-aws-cloud\target\springbootstudy-aws-cloud-0.0.1-SNAPSHOT.jar
[INFO]
[INFO] --- spring-boot-maven-plugin:2.7.0:repackage (repackage) @ springbootstudy-aws-cloud ---
[INFO] Replacing main artifact with repackaged archive
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 18.368 s
[INFO] Finished at: 2022-06-21T17:48:52+09:00
[INFO] -----
```



## Spring Boot Project Executable Jar build ( 단독으로 실행 가능한 Jar file 을 생성 )

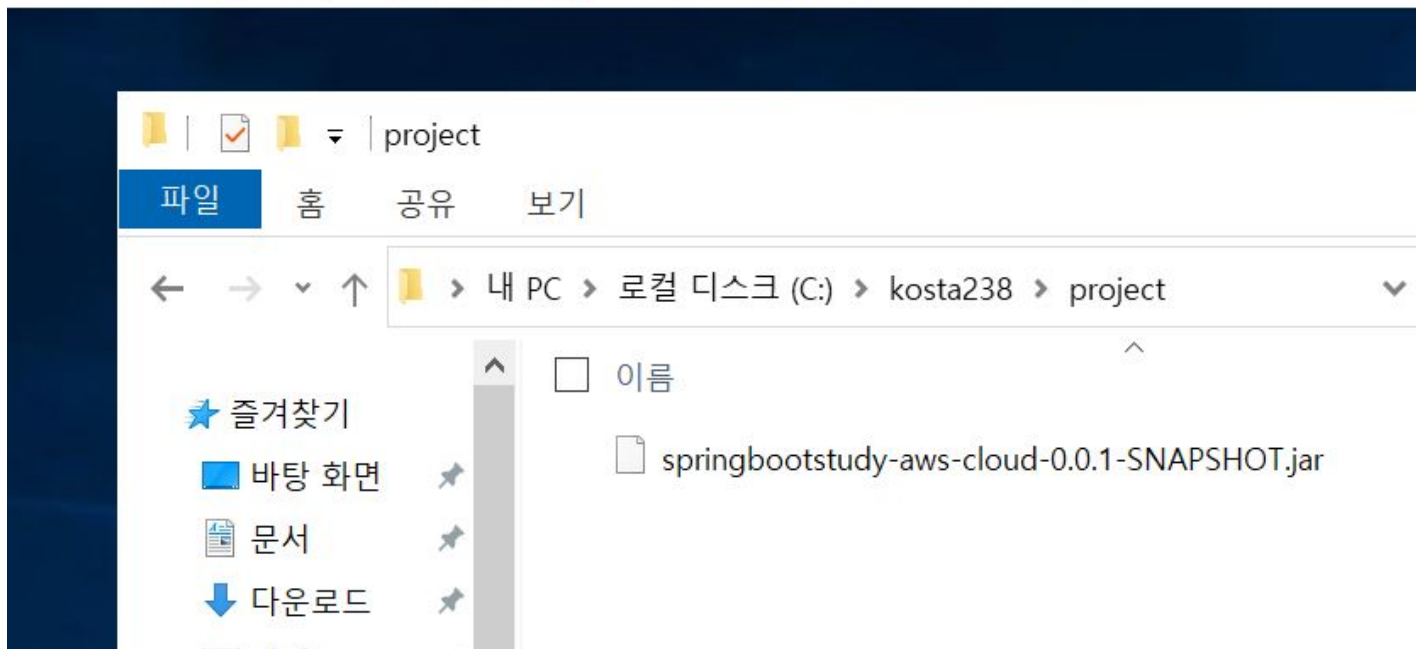
Build 완료 후 프로젝트에서  
마오 => refresh 를 하면  
Target에 jar가 생성되어 있음





Spring Boot Project Executable Jar 를 AWS EC2에서 실행하기 위해  
RDP 를 이용해 build 한 Spring Boot Jar를 배치한다

e - ec2-52-79-240-99.ap-northeast-2.compute.amazonaws.com: 원격 데스크톱 연결



Spring Boot Project Executable Jar 를 AWS EC2에서 실행하기 위해

RDP 를 이용해 EC2 서버에 접속해서 명령프롬프트 창을 실행한다

Spring Boot Application Jar 실행과 중지 명령은 아래와 같다

To run and stop the spring boot executable jar follows the following steps:-

Step 1:- Open "**Command Prompt**" and use "**java -jar**" as shown below

Step 2:- \$ java -jar myApp-0.0.1-SNAPSHOT.jar

Step 3:- To stop please use **ctrl+C**

## Spring Boot Project Executable Jar 를 AWS EC2에서 실행

· ec2-52-79-240-99.ap-northeast-2.compute.amazonaws.com: 원격 데스크톱 연결

Administrator: 명령 프롬프트 - java -jar springbootstudy-aws-cloud-0.0.1-SNAPSHOT.jar

Microsoft Windows [Version 10.0.20348.709]

(c) Microsoft Corporation. All rights reserved.

C:\Users\Administrator>cd C:\kosta238\project

C:\kosta238\project>java -jar springbootstudy-aws-cloud-0.0.1-SNAPSHOT.jar

```

      .
     /\ /___|
    ( ( )\___|
     \V /___|
      ' |___|
=====|_|=====
:: Spring Boot ::
              (v2.7.0)

```

Spring Boot Project Executable Jar 를 AWS EC2에서 실행 후 브라우저에서 테스트하기 위해  
AWS EC2 Console에서 public IP 또는 public Domain Name을 복사한다

EC2 > 인스턴스 > i-057e05cae1a59c8fb

### i-057e05cae1a59c8fb (FeelGoodWebService)에 대한 인스턴스 요약 정보

less than a minute 전에 업데이트됨

인스턴스 ID i-057e05cae1a59c8fb (FeelGoodWebService)	퍼블릭 IPv4 주소 52.79.240.99   <a href="#">개방 주소법</a>	퍼블릭 IPv4 주소 ec2-52-79-240-99.ap-northeast-2.compute.amazonaws.com   <a href="#">개방 주소법</a>
IPv6 주소 -	인스턴스 상태 실행 중	퍼블릭 IPv6 주소 -
호스트 이름 유형 IP 이름: ip-172-31-36-86.ap-northeast-2.compute.internal	프라이빗 IP DNS 이름(IPv4만 해당) ip-172-31-36-86.ap-northeast-2.compute.internal	프라이빗 IPv4 주소 -
프라이빗 리소스 DNS 이름 응답	인스턴스 유형	탄력적 IP 주소

퍼블릭 IPv4 DNS 복사됨

## Client Browser 실행화면

← → ↻ ⚠ 주의 요함 | ec2-52-79-240-99.ap-northeast-2.compute.amazonaws.com

AWS EC2 SpringBoot Home About Projects Contact

인증,인가테스트

회원가입

Hello AWS SpringBoot Project

# Spring Version Convention

SNAPSHOT : release 되기 전 버전, 테스트 개발용도

M (Milestone) : 정해진 주기마다 배포, M3 는 마일스톤 3번째를 의미

RC (Release Candidate) : M에서 향상된 최종 릴리즈 이전 단계

GA (General Availability) : 최종 Release단계의 릴리즈된 안정적 버전

참고 <https://spring.io/projects/spring-boot#learn>

2.7.0 **CURRENT** **GA**

3.0.0-SNAPSHOT **SNAPSHOT**

3.0.0-M3 **PRE**

2.7.1-SNAPSHOT **SNAPSHOT**

2.6.9-SNAPSHOT **SNAPSHOT**

2.6.8 **GA**

2.5.14 **GA**

2.4.13 **GA**

2.3.12.RELEASE **GA**

# Spring Version Convention

springbootstudy-aws-cloud/pom.xml

```
7     <groupId>org.springframework.boot</groupId>
8     <artifactId>spring-boot-starter-parent</artifactId>
9     <version>2.7.0</version>
10    <relativePath /> <!-- lookup parent from repository -->
11  </parent>
12  <groupId>org.kosta</groupId>
13  <artifactId>springbootstudy-aws-cloud</artifactId>
14  <version>1.0.0.RELEASE</version>
15  <name>springbootstudy-aws-cloud</name>
16  <description>Demo project for Spring Boot</description>
17  <properties>
18    <java.version>11</java.version>
19  </properties>
```

target

- > generated-sources
- > generated-test-sources
- > maven-archiver
- > maven-status
- > surefire-reports
- springbootstudy-aws-cloud-1.0.0.RELEASE.jar
- springbootstudy-aws-cloud-1.0.0.RELEASE.jar.original