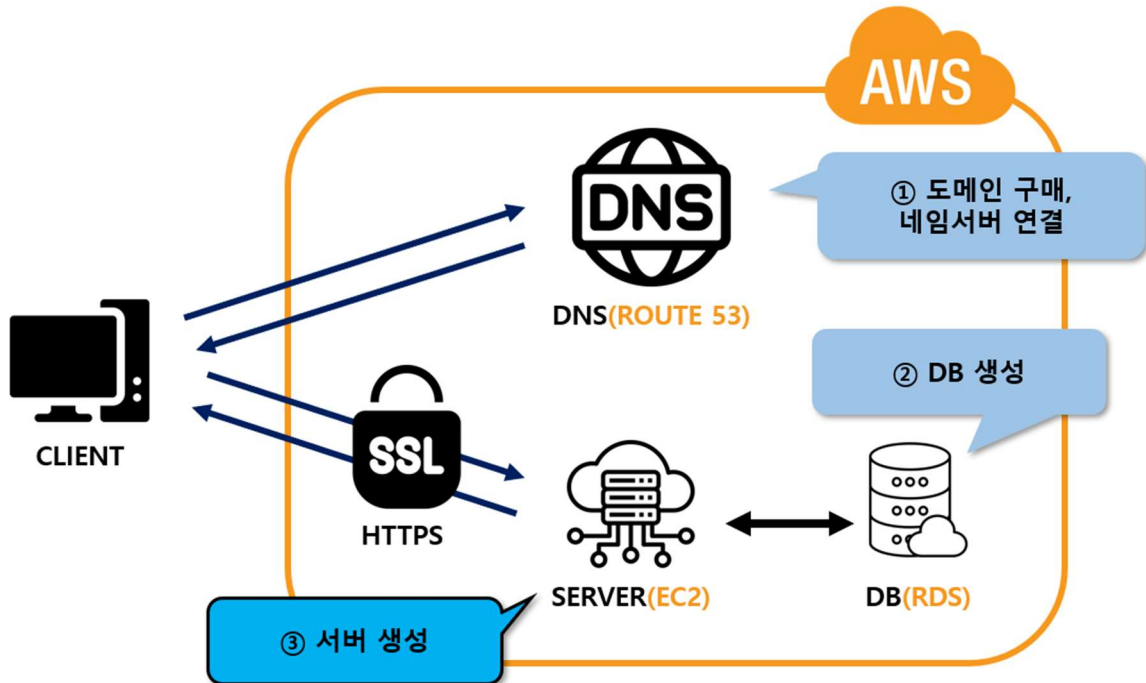
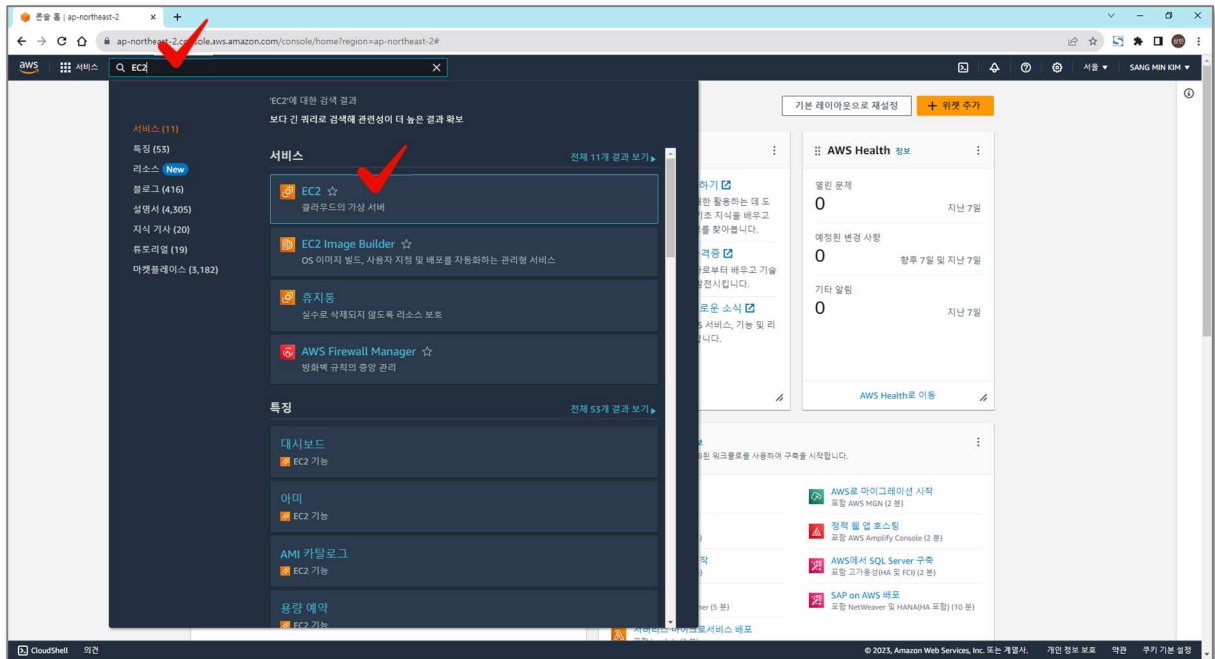


## EC2로 서버 구축하기

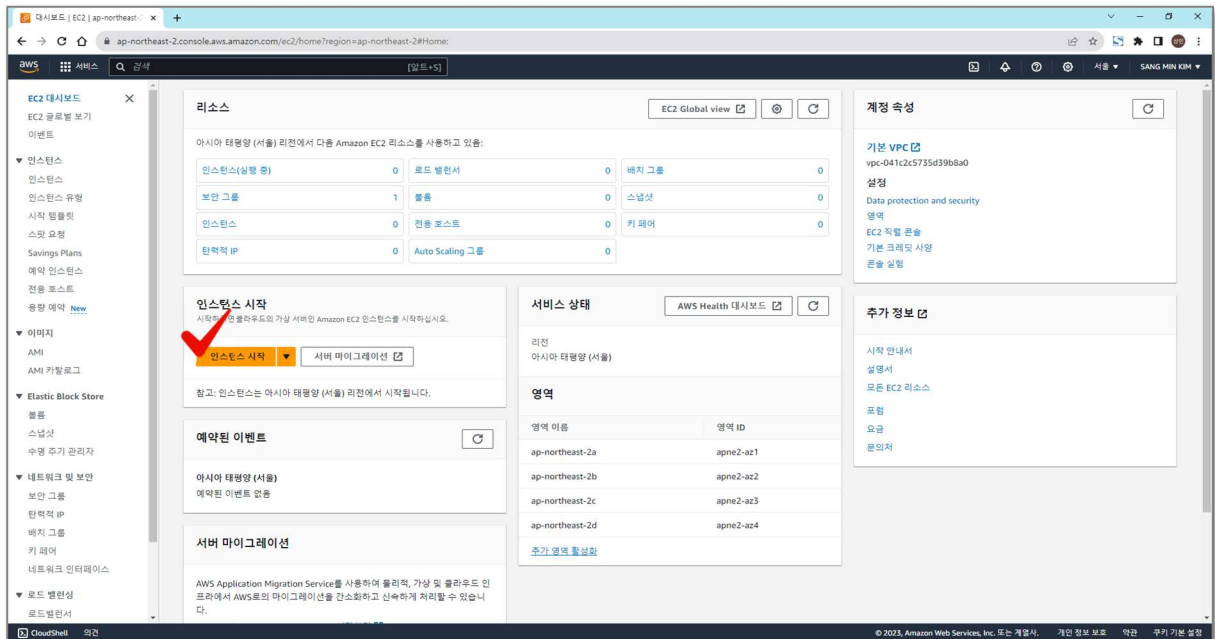


|                  |    |
|------------------|----|
| EC2 인스턴스 생성..... | 2  |
| 탄력적 IP 할당.....   | 8  |
| EC2 연결 테스트.....  | 10 |
| Tomcat 설치.....   | 13 |

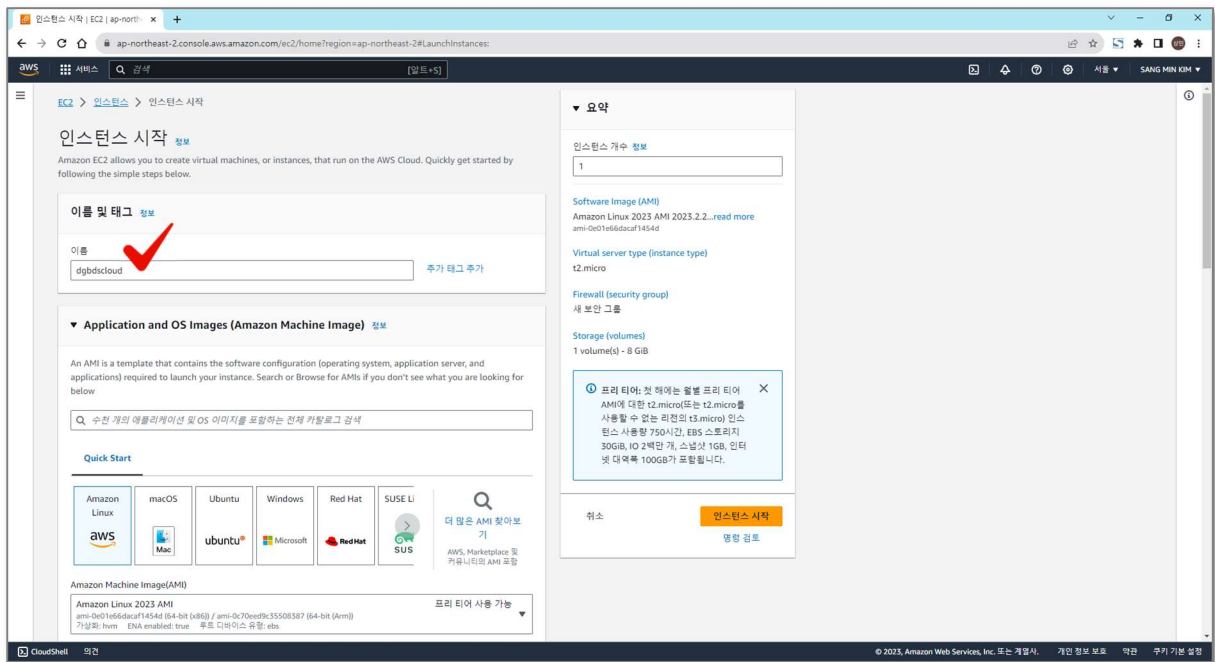
## EC2 인스턴스 생성



1. AWS접속
2. EC2 검색 – [EC2] 클릭



1. [인스턴스 시작] 클릭



## 1. [이름 및 태그] - [이름] 입력



## 1. [키 페어(로그인)] - [새 키 페어 생성] 클릭

키 페어 생성

키 페어 이름

키 페어를 사용하면 인스턴스에 안전하게 연결할 수 있습니다.

dgbdscloud

이름에는 최대 255개의 ASCII 문자를 포함할 수 있습니다. 앞 또는 뒤에 공백을 포함할 수 없습니다.

키 페어 유형

☒ RSA  
RSA 암호화된 프라이빗 및 퍼블릭 키 페어

☐ ED25519  
ED25519 암호화된 프라이빗 및 퍼블릭 키 페어

프라이빗 키 파일 형식

☐ .pem  
OpenSSH와 함께 사용

☒ .ppk  
PuTTY와 함께 사용

메시지가 표시되면 프라이빗 키를 사용하여 컴퓨터의 안전하고 액세스 가능한 위치에 저장합니다. 나중에 인스턴스에 연결할 때 필요합니다. [자세히 알아보기](#)

취소

키 페어 생성

1. [키 페어 이름] 입력
2. [프라이빗 키 파일 형식] - [.ppk] 선택
3. [키 페어 생성] 클릭

인스턴스 시작 | EC2 | ap-north-1

ap-northeast-2.console.aws.amazon.com/ec2/home?region=ap-northeast-2#LaunchInstances:

인스턴스 시작

인스턴스 시작

인스턴스 시작

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 - 풀수

dgbdscloud

새 키 페어 생성

네트워크 설정 정보

네트워크 정보

vpc-041c2c5735d39b8a0

서브넷 정보

기본 설정 옵션(가용 영역의 기본 서브넷)

퍼블릭 IP 자동 할당 정보

활성화

Firewall (security groups) 정보

보안 그룹은 인스턴스에 대한 트래픽을 제어하는 방화벽 규칙 세트입니다. 해당 프라이빗이 인스턴스에 도달하도록 허용하는 규칙을 추가합니다.

☒ Create security group
☐ Select existing security group

We'll create a new security group called 'launch-wizard-1' with the following rules:

☒ Allow SSH traffic from  
Helps you connect to your instance

내 IP

118.235.33.207/32

☐ 인터넷에서 HTTPS 트래픽 허용  
여름 동안 웹 서버를 생성할 때 인스턴트를 설정하려면

☐ 인터넷에서 HTTP 트래픽 허용  
여름 동안 웹 서버를 생성할 때 인스턴트를 설정하려면

요약

인스턴스 개수 정보

1

Software Image (AMI)

Amazon Linux 2023 AMI 2023.2.2...read more

ami-0d0166dcafc1454d

Virtual server type (instance type)

t2.micro

Firewall (security group)

새 보안 그룹

Storage (volumes)

1 volume(s) - 8 GiB

프리 티어: 첫 해에는 월별 프리 티어 AMI에 대한 t2.micro(또는 t2.micro를 사용할 수 있는 기간의 t3.micro) 인스턴스 사용량 750시간, EBS 스토리지 30GiB, IO 2백만 개, 스냅샷 1GB, 인터넷 대역폭 100GB가 포함됩니다.

취소

인스턴스 시작

명령 검토

- ppk파일이 자동으로 다운로드 되니 잘 보관할 것!

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

dgbdscloud

새 키 페어 생성

네트워크 설정 정보

네트워크 정보

vpc-041c2c5735d39b8a0

서브넷 정보

기본 설정 없음(가용 영역의 기본 서브넷)

퍼블릭 IP 자동 할당 정보

활성화

Firewall (security groups) 정보

보안 그룹은 인스턴스에 대한 트래픽을 제어하는 방화벽 규칙 세트입니다. 특정 트래픽이 인스턴스에 도달하도록 허용하는 규칙을 추가합니다.

Create security group

Select existing security group

일반 보안 그룹 정보

보안 그룹 선택

default sg-0bc79fd6b80fd9a2a

보안 그룹 규칙 비교

VPC: vpc-041c2c5735d39b8a0

여기서 추가 또는 제거하는 보안 그룹은 모든 네트워크 인터페이스에서 추가 또는 제거됩니다.

요약

인스턴스 개수 정보

1

Software Image (AMI)

Amazon Linux 2023 AMI 2023.2.2...read more

ami-0e01e66dacf1454d

Virtual server type (instance type)

t2.micro

Firewall (security group)

default

Storage (volumes)

1 volume(s) - 8 GiB

프리 티어: 첫 해에는 월별 프리 티어 AMI에 대한 t2.micro(또는 t2.micro를 사용할 수 없는 리전의 t3.micro) 인스턴스 사용량 750시간, EBS 스토리지 30GiB, IO 2백만 개, 스냅샷 1GB, 인터넷 대역폭 100GB가 포함됩니다.

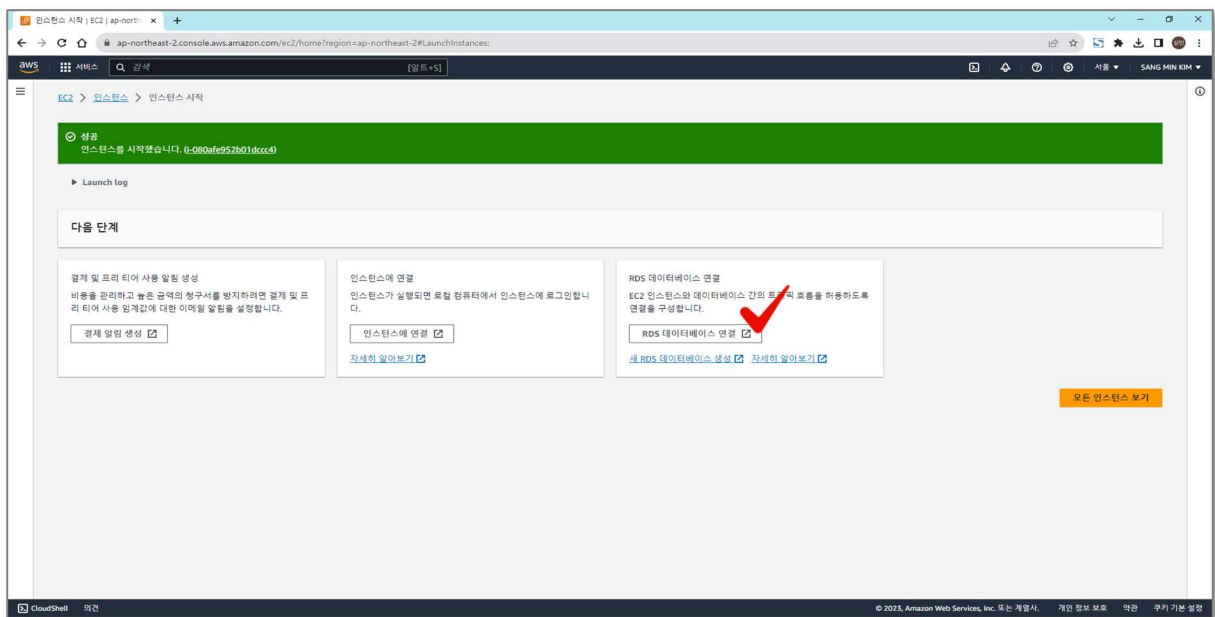
취소

인스턴스 시작

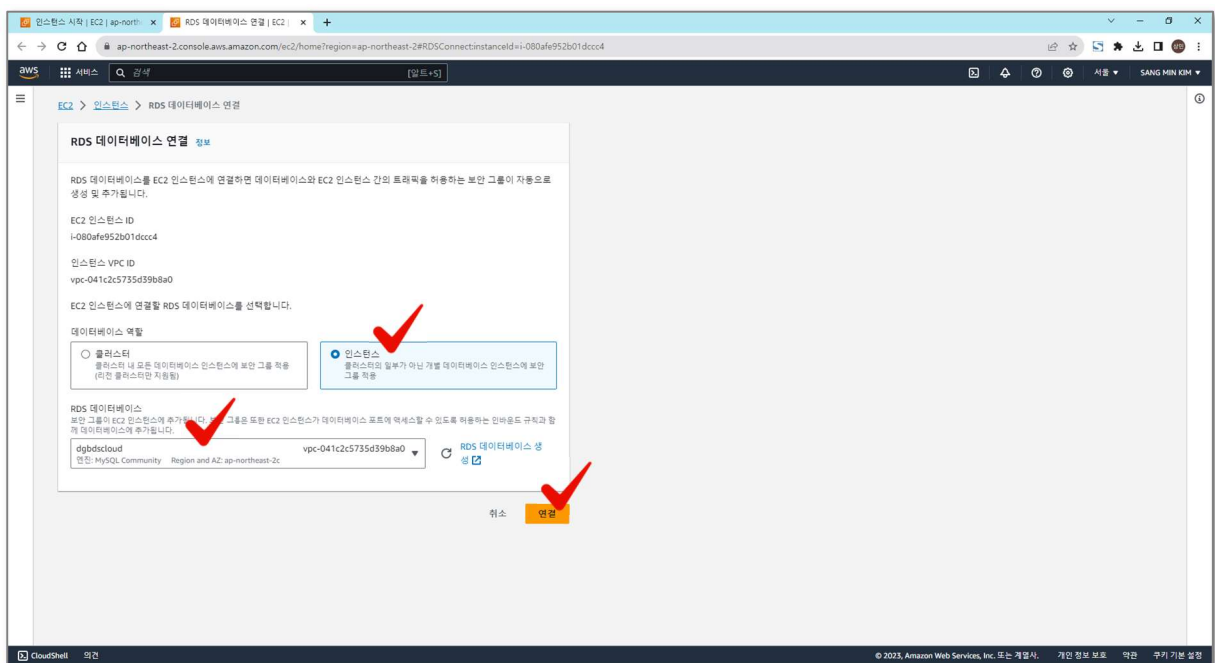
명령 검토

1. [네트워크 설정] – [Firewall] – [Select existing security group] 선택
2. 보안 그룹 선택
3. [인스턴스 시작] 클릭

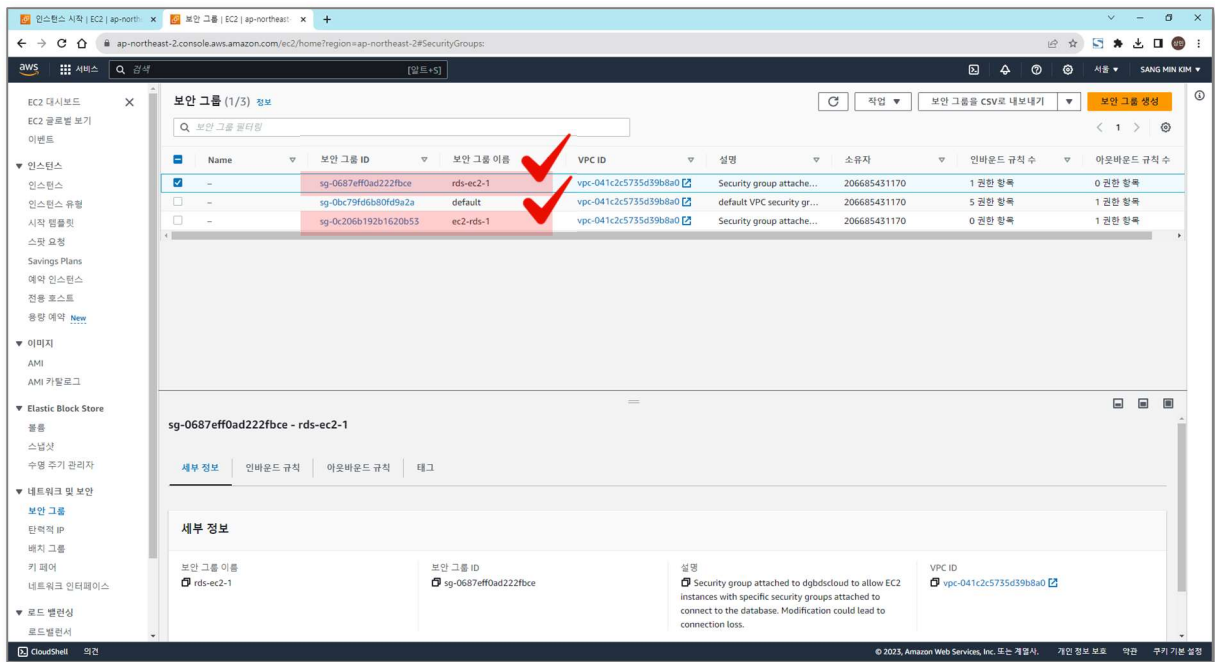
페이지 5 / 15



## 1. [RDS 데이터베이스 연결] 클릭



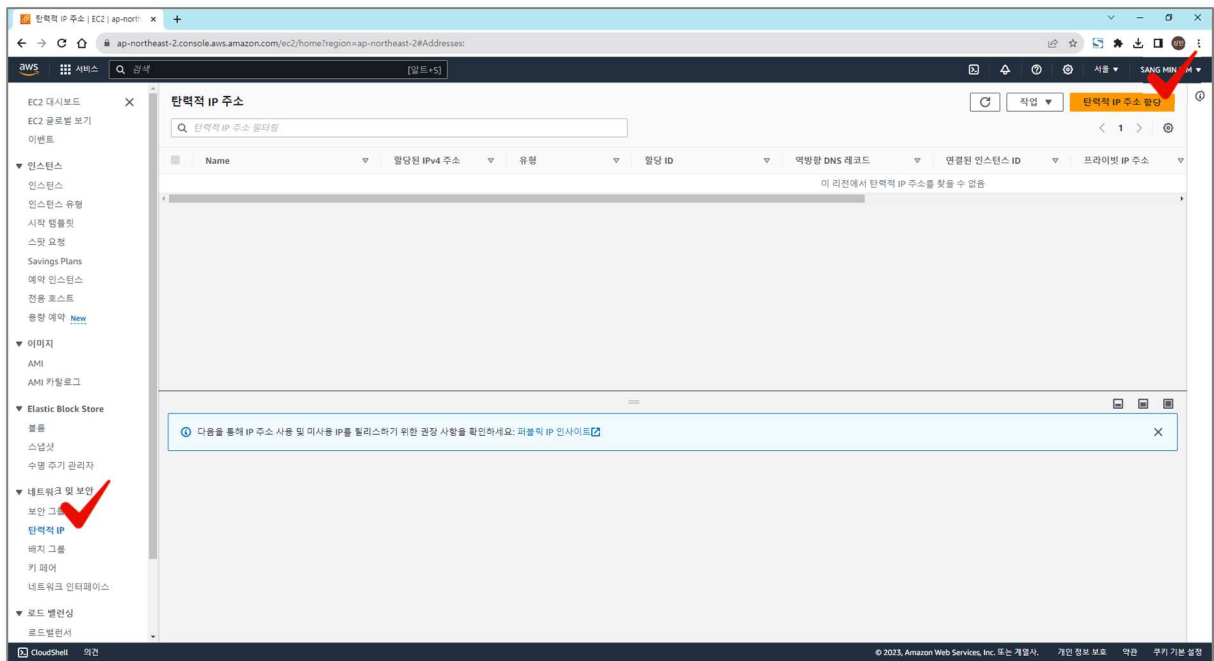
1. [데이터베이스 역할] - [인스턴스] 선택
2. [RDS 데이터베이스] - 생성한 RDS 인스턴스 선택
3. [연결] 클릭



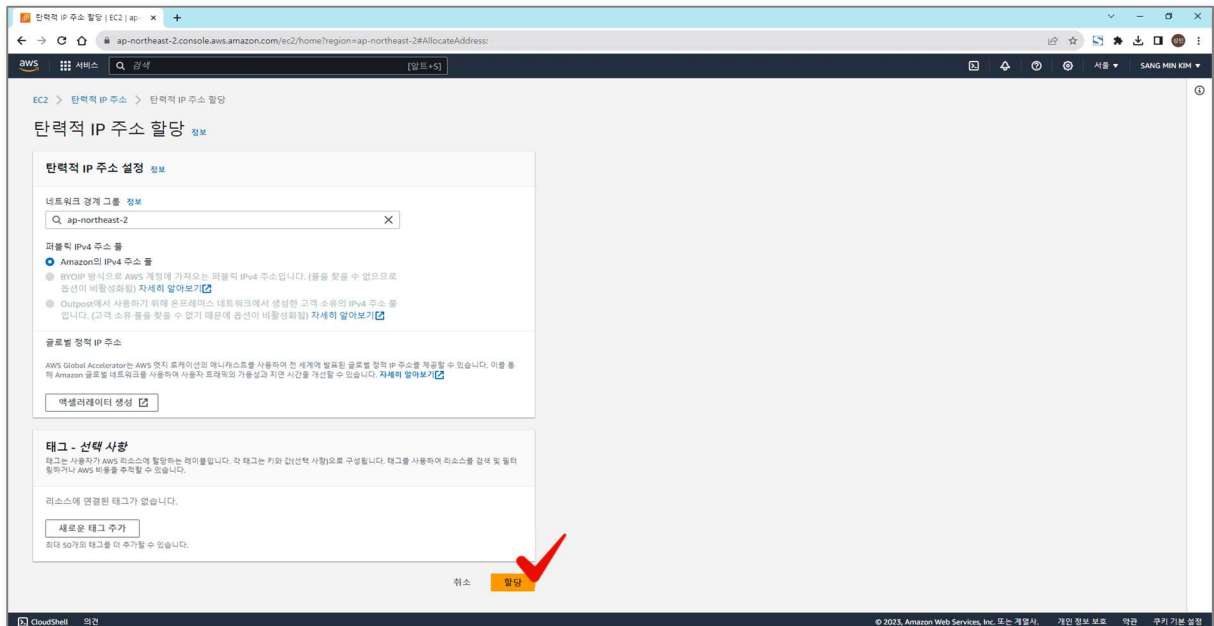
- [EC2 > 네트워크 및 보안 > 보안 그룹] 경로로 접속하면 RDS와 EC2를 연결하는 보안그룹이 생성되어 있다.

EC2 인스턴스 생성 완료!

## 탄력적 IP 할당

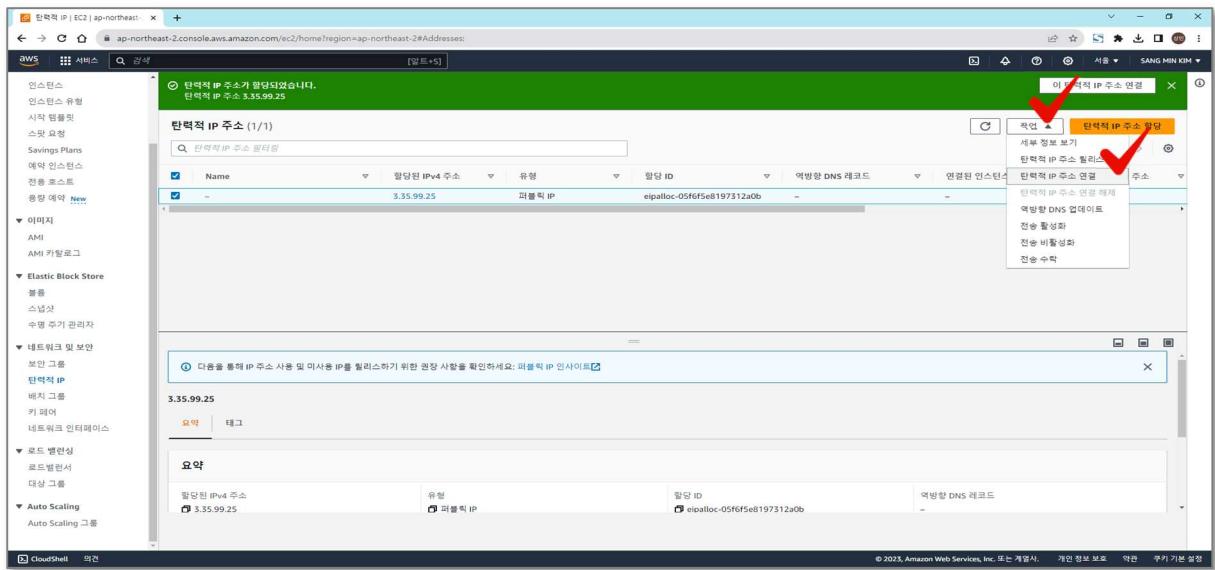


1. [EC2 > 네트워크 및 보안 > 탄력적 IP] 경로로 접속
2. [탄력적 IP 주소 할당] 클릭

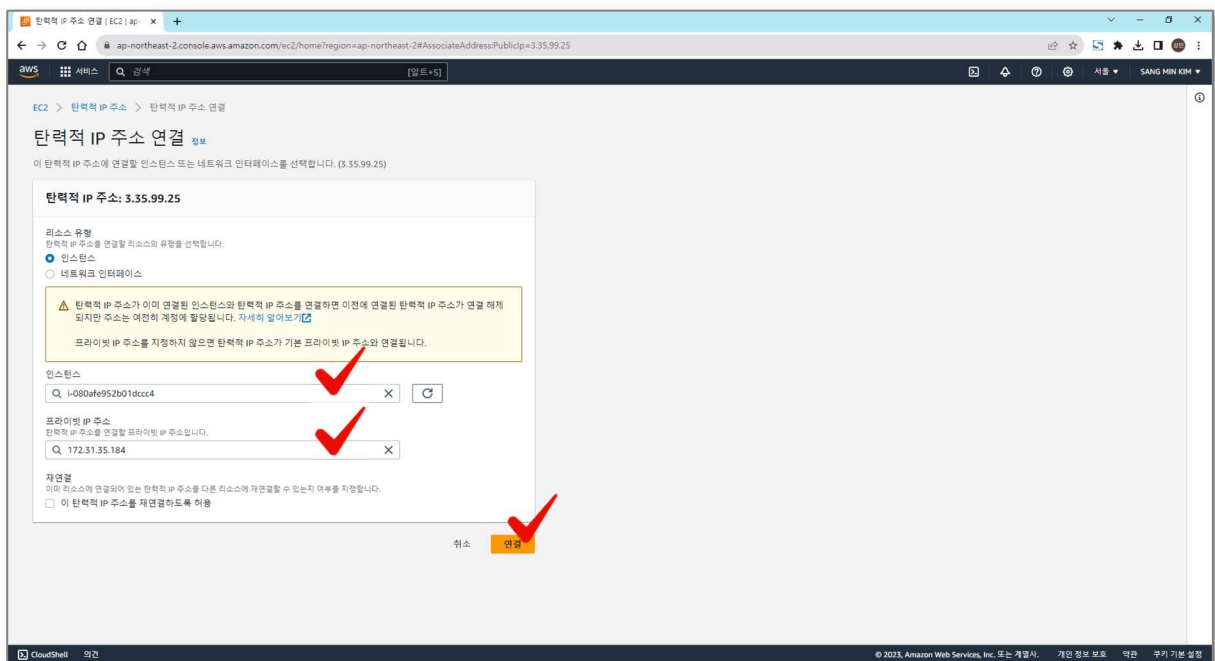


1. [할당] 클릭





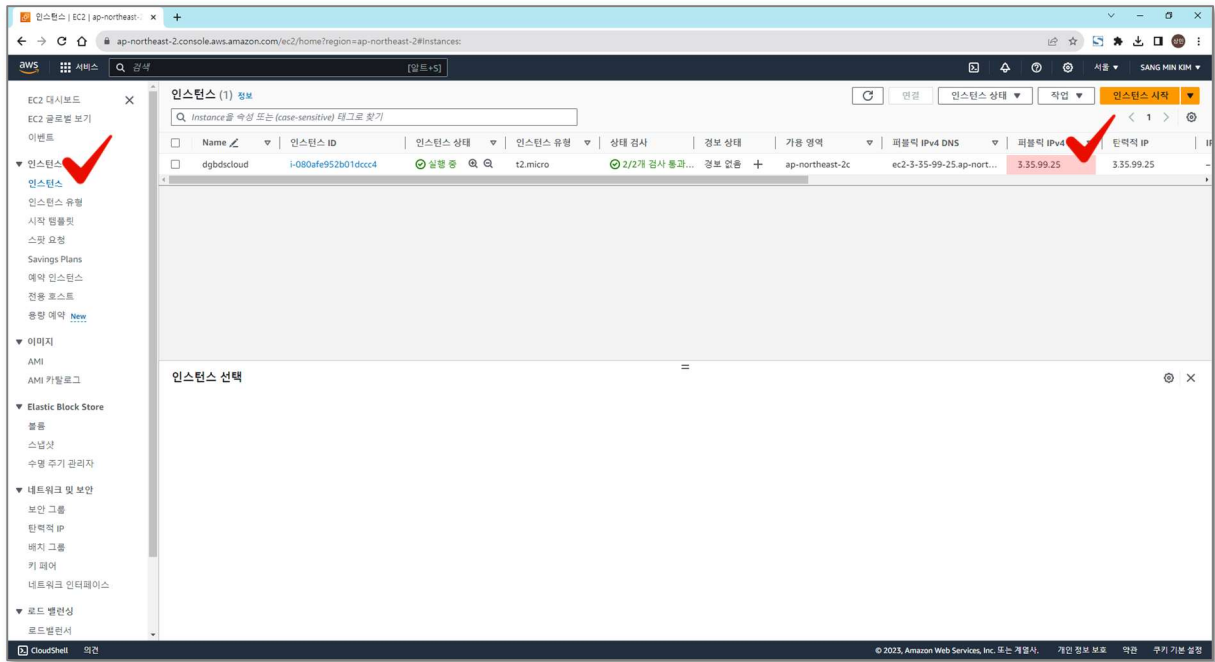
1. [작업] 클릭 - [탄력적 IP 주소 연결] 클릭



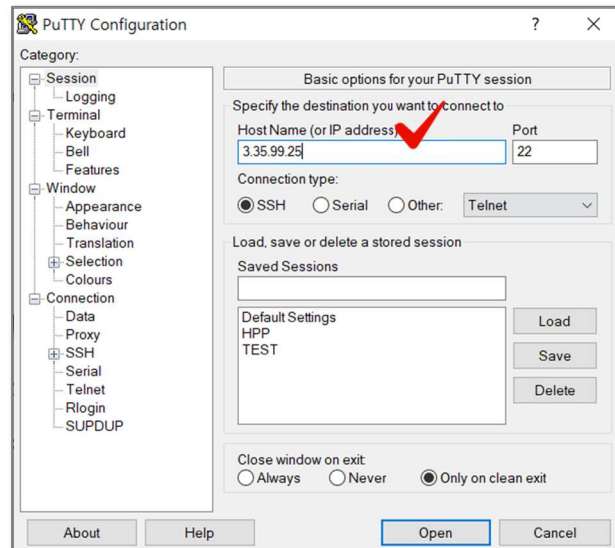
1. 인스턴스 선택
2. 프라이빗 IP 주소 선택
3. [연결] 클릭

**탄력적 IP 할당 완료!**

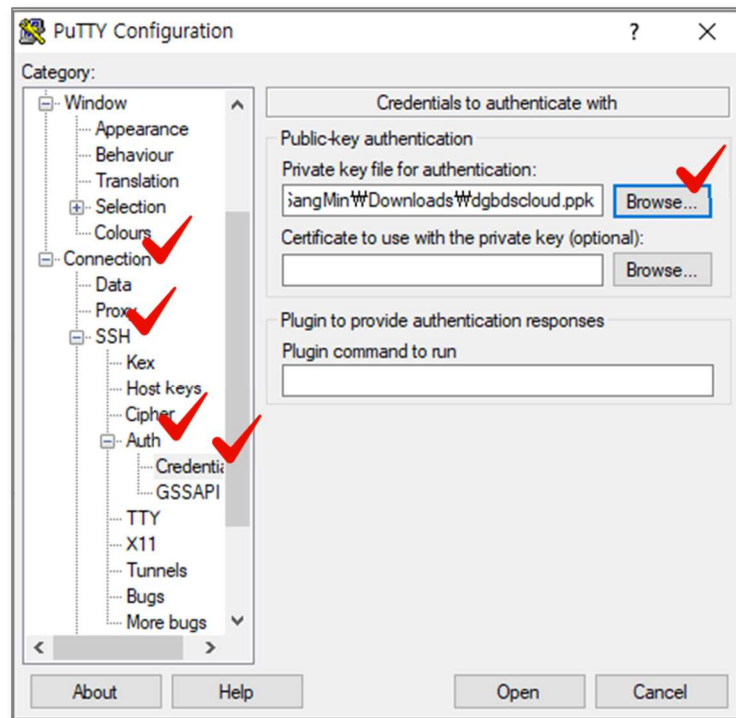
## EC2 연결 테스트



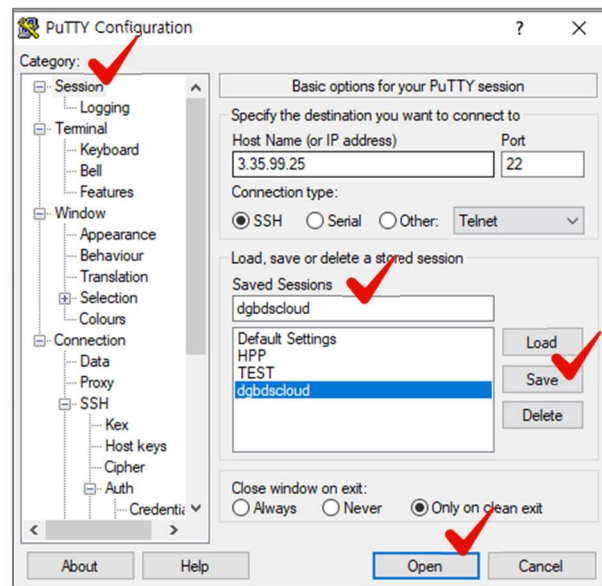
1. [EC2 > 인스턴스 > 인스턴스] 경로로 접속
2. 인스턴스 IP주소 복사



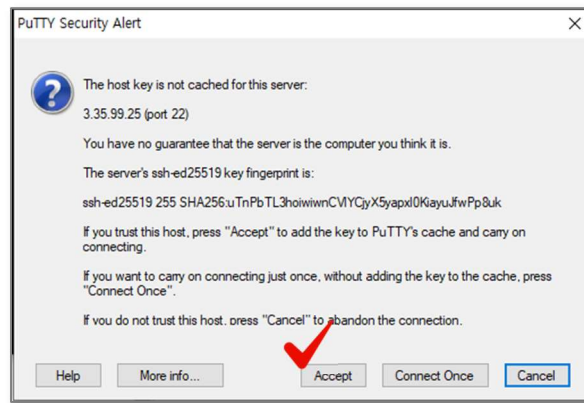
1. [Host Name]에 복사한 IP 입력



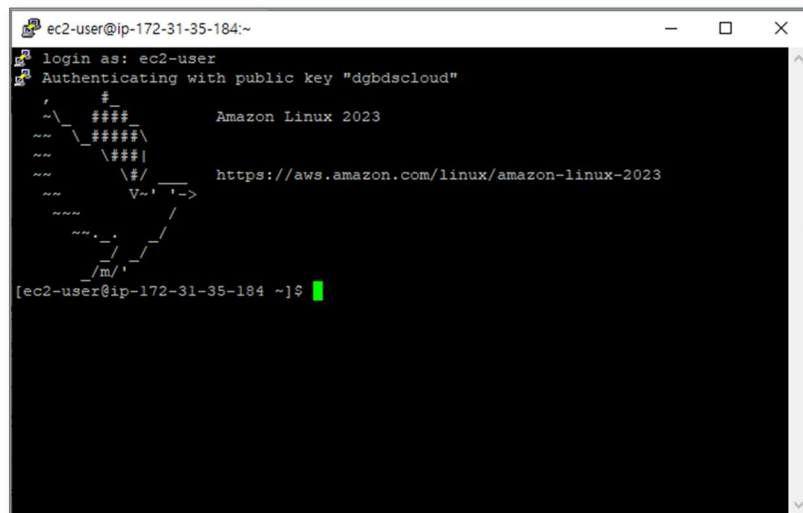
1. [Connection > SSH > Auth > Credential] 경로 열기
2. 인스턴스 생성할 때 다운받은 ppt 키 선택



1. [Session] 경로 이동 – [Saved Sessions]에 세션 이름 입력 – [Save] 클릭
2. [Open] 클릭



1. [Accept] 클릭



1. [ec2-user] 입력해서 로그인

- Amazon Linux의 기본 사용자는 ec2-user임

**EC2 연결 테스트 완료!**

## Tomcat 설치

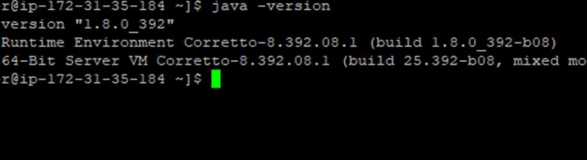
- java 11 설치

```
sudo yum install java-11-amazon-corretto.x86_64
```

[illegible]

- java 버전 확인

```
java -version
```



A terminal window titled "ec2-user@ip-172-31-35-184:~" with standard window controls. The terminal displays the output of the command `java -version`. The output consists of four lines: `openjdk version "1.8.0_392"`, `OpenJDK Runtime Environment Corretto-8.392.08.1 (build 1.8.0_392-b08)`, `OpenJDK 64-Bit Server VM Corretto-8.392.08.1 (build 25.392-b08, mixed mode)`, and a prompt `[ec2-user@ip-172-31-35-184 ~]$` with a green cursor.

```
ec2-user@ip-172-31-35-184:~  
[ec2-user@ip-172-31-35-184 ~]$ java -version  
openjdk version "1.8.0_392"  
OpenJDK Runtime Environment Corretto-8.392.08.1 (build 1.8.0_392-b08)  
OpenJDK 64-Bit Server VM Corretto-8.392.08.1 (build 25.392-b08, mixed mode)  
[ec2-user@ip-172-31-35-184 ~]$
```

- 톰캣 설치

mkdir server

cd server/

wget <https://dlcdn.apache.org/tomcat/tomcat-8/v8.5.95/bin/apache-tomcat-8.5.95.tar.gz>

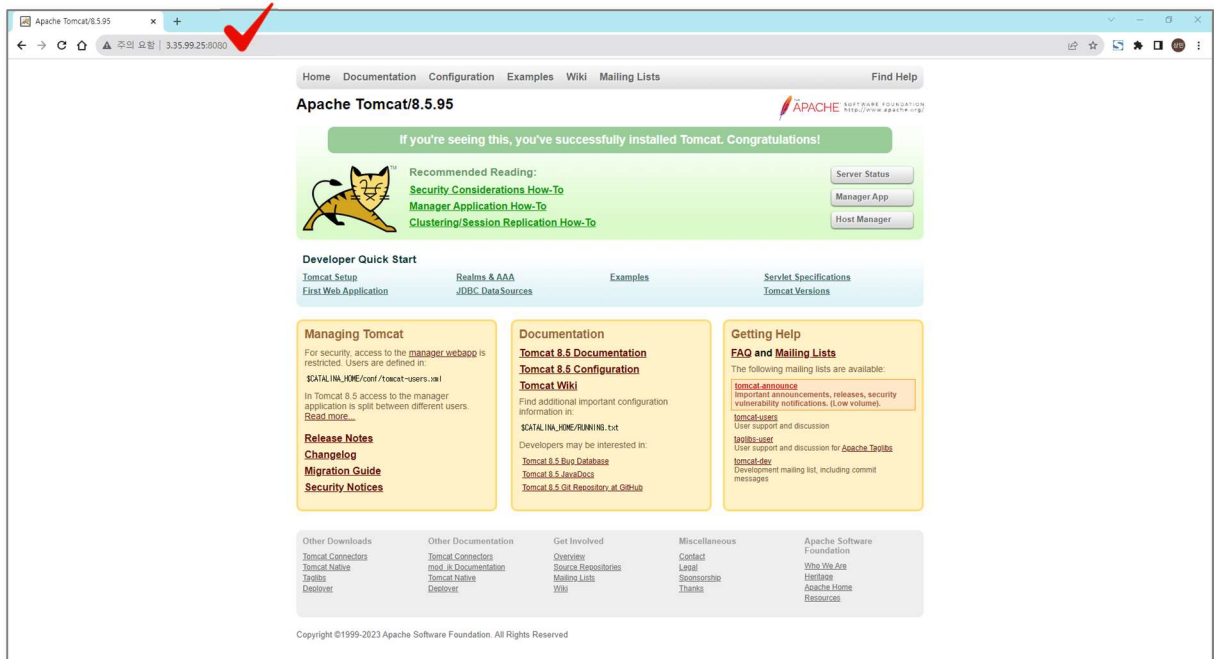
tar -zxvf apache-tomcat-8.5.95.tar.gz

```
ec2-user@ip-172-31-35-184:~/server
apache-tomcat-8.5.95/webapps/examples/jsp/tagplugin/foreach.jsp
apache-tomcat-8.5.95/webapps/examples/jsp/tagplugin/foreach2.jsp.html
apache-tomcat-8.5.95/webapps/examples/jsp/tagplugin/howto.html
apache-tomcat-8.5.95/webapps/examples/jsp/tagplugin/it.jsp
apache-tomcat-8.5.95/webapps/examples/jsp/tagplugin/it.jsp.html
apache-tomcat-8.5.95/webapps/examples/jsp/tagplugin/notes.html
apache-tomcat-8.5.95/webapps/examples/jsp/xml/xml.html
apache-tomcat-8.5.95/webapps/examples/jsp/xml/xml.jsp.html
apache-tomcat-8.5.95/webapps/examples/servlets/cookies.html
apache-tomcat-8.5.95/webapps/examples/servlets/helloworld.html
apache-tomcat-8.5.95/webapps/examples/servlets/images/cookie.gif
apache-tomcat-8.5.95/webapps/examples/servlets/images/execute.gif
apache-tomcat-8.5.95/webapps/examples/servlets/images/return.gif
apache-tomcat-8.5.95/webapps/examples/servlets/index.html
apache-tomcat-8.5.95/webapps/examples/servlets/nonblocking/bytecounter.html
apache-tomcat-8.5.95/webapps/examples/servlets/requesters.html
apache-tomcat-8.5.95/webapps/examples/servlets/reqinfo.html
apache-tomcat-8.5.95/webapps/examples/servlets/requester.html
apache-tomcat-8.5.95/webapps/examples/servlets/sessions.html
apache-tomcat-8.5.95/webapps/examples/websocket/chat.html
apache-tomcat-8.5.95/webapps/examples/websocket/forward.html
apache-tomcat-8.5.95/webapps/examples/websocket/echo.html
apache-tomcat-8.5.95/webapps/examples/websocket/index.html
apache-tomcat-8.5.95/webapps/examples/websocket/notes.html
apache-tomcat-8.5.95/webapps/host-manager/WEB-INF/context.xml
apache-tomcat-8.5.95/webapps/host-manager/WEB-INF/jsp/401.jsp
apache-tomcat-8.5.95/webapps/host-manager/WEB-INF/jsp/403.jsp
apache-tomcat-8.5.95/webapps/host-manager/WEB-INF/jsp/404.jsp
apache-tomcat-8.5.95/webapps/host-manager/WEB-INF/manager.xml
apache-tomcat-8.5.95/webapps/host-manager/WEB-INF/web.xml
apache-tomcat-8.5.95/webapps/host-manager/cas/manager.cas
apache-tomcat-8.5.95/webapps/host-manager/images/cas/login.svg
apache-tomcat-8.5.95/webapps/host-manager/images/tomcat.svg
apache-tomcat-8.5.95/webapps/manager/WEB-INF/context.xml
apache-tomcat-8.5.95/webapps/manager/WEB-INF/jsp/401.jsp
apache-tomcat-8.5.95/webapps/manager/WEB-INF/jsp/403.jsp
apache-tomcat-8.5.95/webapps/manager/WEB-INF/jsp/404.jsp
apache-tomcat-8.5.95/webapps/manager/WEB-INF/jsp/connectorCerts.jsp
apache-tomcat-8.5.95/webapps/manager/WEB-INF/jsp/connectorCipher.jsp
apache-tomcat-8.5.95/webapps/manager/WEB-INF/jsp/sessionDetail.jsp
apache-tomcat-8.5.95/webapps/manager/WEB-INF/jsp/sessionList.jsp
apache-tomcat-8.5.95/webapps/manager/WEB-INF/web.xml
apache-tomcat-8.5.95/webapps/manager/cas/manager.cas
apache-tomcat-8.5.95/webapps/manager/images/cas/login.svg
apache-tomcat-8.5.95/webapps/manager/images/tomcat.svg
apache-tomcat-8.5.95/webapps/manager/index.jsp
apache-tomcat-8.5.95/webapps/manager/status.xml
apache-tomcat-8.5.95/bin/catalina.sh
apache-tomcat-8.5.95/bin/catalina.xml
apache-tomcat-8.5.95/bin/ciphers.sh
apache-tomcat-8.5.95/bin/configtest.sh
apache-tomcat-8.5.95/bin/dsmon.sh
apache-tomcat-8.5.95/bin/jkpsv.sh
apache-tomcat-8.5.95/bin/setclasspath.sh
apache-tomcat-8.5.95/bin/shutdown.sh
apache-tomcat-8.5.95/bin/startup.sh
apache-tomcat-8.5.95/bin/cool-httpd.sh
apache-tomcat-8.5.95/bin/version.sh
[ec2-user@ip-172-31-35-184 server]$
```

- 톰캣 실행

./apache-tomcat-8.5.95/bin/startup.sh

```
ec2-user@ip-172-31-35-184:~/server
[ec2-user@ip-172-31-35-184 server]$ ./apache-tomcat-8.5.95/bin/startup.sh
Using CATALINA_BASE:   /home/ec2-user/server/apache-tomcat-8.5.95
Using CATALINA_HOME:   /home/ec2-user/server/apache-tomcat-8.5.95
Using CATALINA_TMPDIR: /home/ec2-user/server/apache-tomcat-8.5.95/temp
Using JRE_HOME:         /usr
Using CLASSPATH:        /home/ec2-user/server/apache-tomcat-8.5.95/bin/bootstrap.jar
Using CATALINA_OPTS:
Tomcat started.
[ec2-user@ip-172-31-35-184 server]$
```



1. 인터넷 브라우저에서 아래 URL로 접속

- 접속 URL: [http://\[인스턴스 ip\]:8080](http://[인스턴스 ip]:8080)

EC2 로 서버 구축하기 완료!