

김인철



프로젝트



주제

세미 프로젝트 *GIVENGEL*

김인철 | 김민주 | 편이슬 | 김승철

AWS 리눅스 웹 배포



소개



목차



내용



배운 점



반성할 점

1. 시스템 설명



AWS 시스템 개요와 전체 흐름

AWS가 제공해주는 서비스와 구성
리눅스 서버(AWS EC2)와 데이터베이스(AWS RDS) 구조

2. JDK 설치



Linux JDK 설치와 테스트

JDK를 설치한 목적과 설치 후 정상 작동 여부를 테스트

3. Tomcat 설치



Apache Tomcat 설치

Tomcat version 소개와 설치 과정
Tomcat 설치 목적과 설치 후 구동 테스트

4. MySQL (RDS)



DataBase 구성과 작동

AWS가 제공하는 RDS
RDS (MySQL) 작동과 테스트

5. War 배포



War 파일 Export & 배포

Semi Project 'Givengel' 의 war 파일
War 파일 배포 방법과 리눅스 서버에 war파일 배포

6. 정상 실행 확인

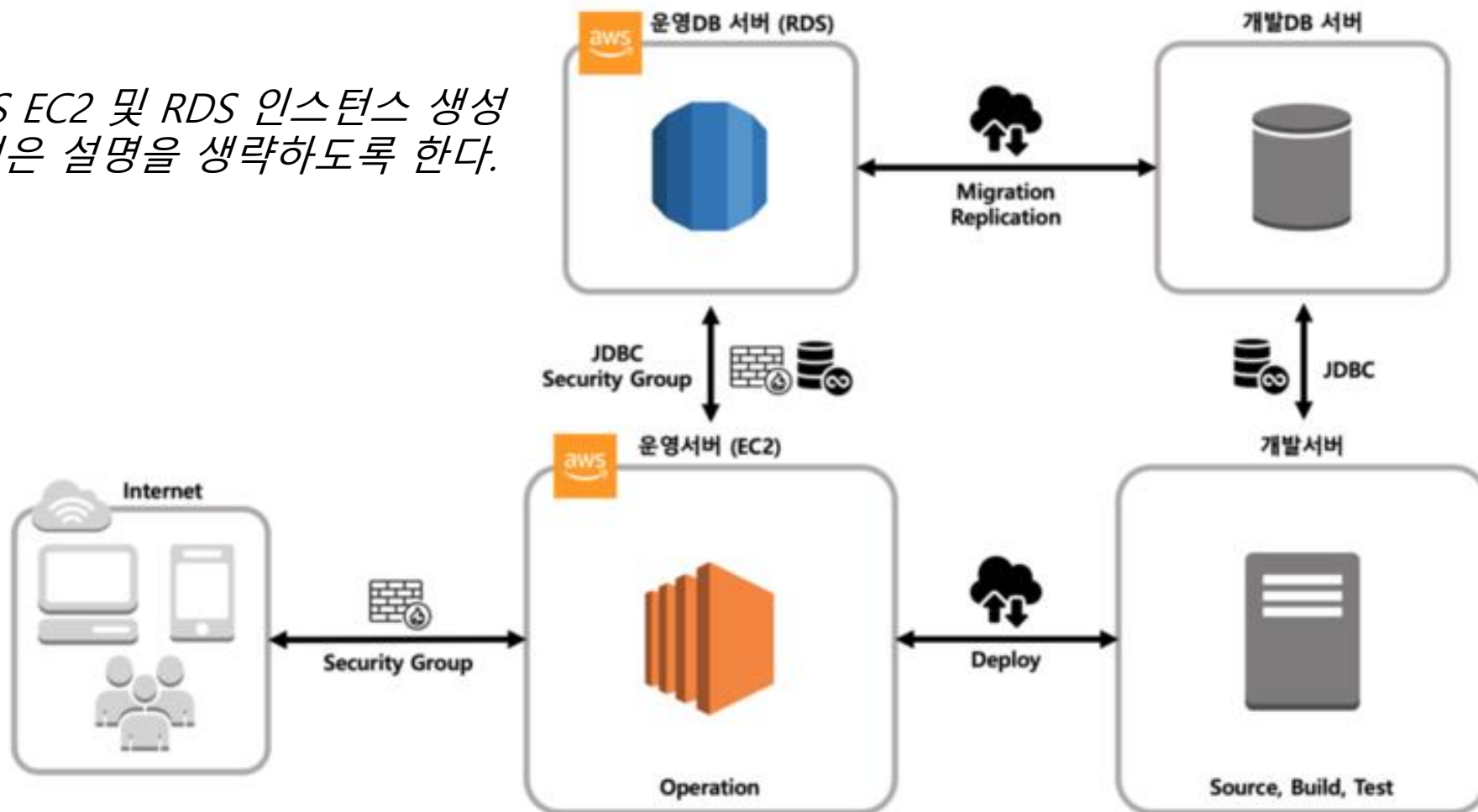


정상 작동을 확인

리눅스 서버를 통해 웹 배포 확인
아마존 EC2 퍼블릭 액세스 IP 주소와 프로젝트 경로를 입력
후 정상적으로 작동되는지 테스트

1. 시스템 설명

AWS EC2 및 RDS 인스턴스 생성 과정은 설명을 생략하도록 한다.



소개



목차



내용



배운 점



반성할 점

1. 시스템 설명

리눅스 웹배포 환경 조성

Putty 터미널에서 리눅스 명령어를 통한 여러 가지 환경을 구성

FileZilla

파일질라를 통해 리눅스 환경에 War 파일을 업로드

Putty

AWS EC2 Linux 인스턴스

JDK & Tomcat 설치

FileZilla

웹 배포

Putty

Putty 를 통해 EC2 인스턴스의 퍼블릭 액세스 IP와 KEY 파일 인증 후 리눅스 터미널 환경 조성

STS에서 war 파일 Export

세미 프로젝트 Givengel 프로젝트의 War파일을 Export 한다.



소개



목차



내용



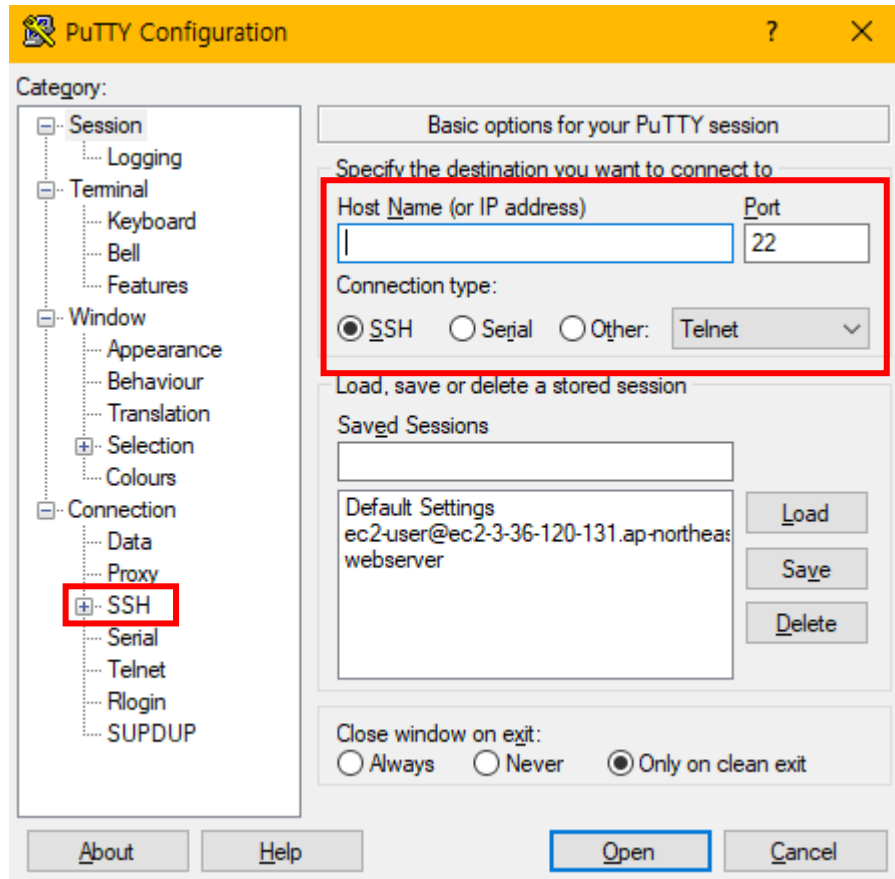
배운 점



반성할 점

2. JDK 설치

1. Putty 사용

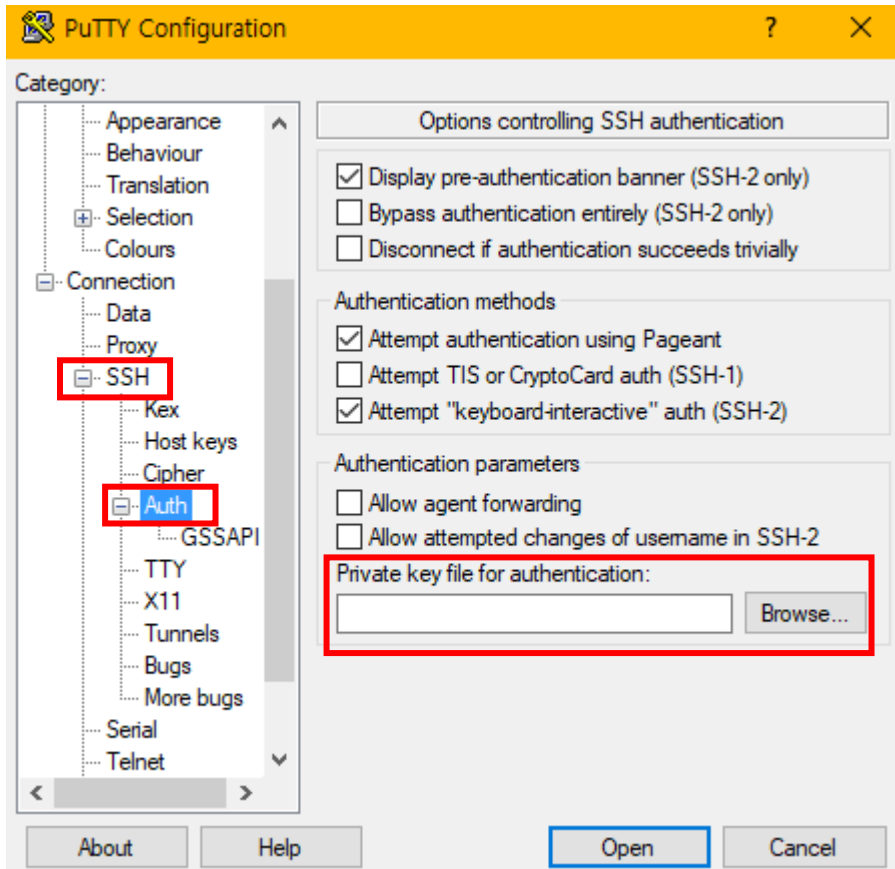


Putty를 이용하여 EC2 리눅스 인스턴스에 접근하기 위해서는 다음과 같은 조건과 설정이 필요하다.

- ① Host Name (or IP address)
: AWS EC2의 경우 해당 칸이 유저아이디와 엔드포인트로 구성되어 있다.
(유저아이디 default : ec2-user)
 - ② Connection type
: SSH 타입으로 설정하여 KEY를 통한 인증 접속 방식으로 접속한다.
(KEY는 AWS EC2 인스턴스 생성과 함께 발급받은 KEY이다.)
- 다음 슬라이드

2. JDK 설치

1. Putty 사용



Putty를 이용하여 EC2 리눅스 인스턴스에 접근하기 위해서는 다음과 같은 조건과 설정이 필요하다.

- ① Host Name (or IP address)
: AWS EC2의 경우 해당 칸이 유저아이디와 엔드포인트로 구성되어 있다.
(유저아이디 default : ec2-user)
- ② Connection type
: SSH 타입으로 설정하여 KEY를 통한 인증 접속 방식으로 접속한다.
(KEY는 AWS EC2 인스턴스 생성과 함께 발급받은 KEY이다.)
- ③ Private key file for authentication
: 발급받은 키를 첨부할 수 있는 Auth 항목에 발급받은 키파일을 첨부

 webserver.ppk



소개



목차



내용



배운 점



반성할 점



소개



목차



내용



배운 점



반성할 점

2. JDK 설치

2. Root 계정 사용

```
2-31-14-215~  
username "ec2-user".  
Communicating with public key "webservice"  
login: Wed Aug 11 07:31:50 2021 from 115.91.81.108  
_ _ _ _ _  
_ _ _ _ _ / Amazon Linux 2 AMI  
_ _ _ _ _  
https://aws.amazon.com/amazon-linux-2/  
4 package(s) needed for security, out of 16 available  
Run "sudo yum update" to apply all updates.  
[ec2-user@ip-172-31-14-215 ~]$
```

```
[ec2-user@ip-172-31-14-215 ~]$ sudo su  
[root@ip-172-31-14-215 ec2-user]#
```

Root 권한 변경

현재 사용자의 권한이 root가 아니기 때문에
이후 환경변수 설정이나 원활한 환경 설정을 위해
root 권한으로 변경해주도록 한다.

Putty 실행 화면

Putty를 통해 정상적으로 EC2 인스턴스에 연결

AWS EC2 LINUX

2. JDK 설치

3. Yum – list, install

```
[root@ip-172-31-14-215 ec2-user]# yum list java*jdk-devel
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
Available Packages
java-1.7.0-openjdk-devel.x86_64 1:1.7.0.261-2.6.22.2.amzn2.0.1 amzn2-core
java-1.8.0-openjdk-devel.x86_64 1:1.8.0.302.b08-0.amzn2.0.1 amzn2-core
java-11-openjdk-devel.x86_64 1:11.0.11.0.9-1.amzn2.0.1 amzn2extra-java-o
penjdk11
[root@ip-172-31-14-215 ec2-user]#
```

Yum list 명령어 사용 – jdk version

Yum list 명령어를 사용하여 설치할 수 있는 jdk 버전
이 무엇이 있는지 확인합니다.

확인 후

Java-1.8.0-openjdk-devel.x86_64

를 설치하였습니다.



소개



목차



내용



배운 점



반성할 점

2. JDK 설치

3. Yum – list, install

```
[root@ip-172-31-41-249 ~]#
[root@ip-172-31-41-249 ~]# yum install -y java-1.8.0-openjdk-devel.x86_64
Loaded plugins: priorities, update-motd, upgrade-helper
amzn-main
amzn-updates
Resolving Dependencies
--> Running transaction check
---> Package java-1.8.0-openjdk-devel.x86_64 1:1.8.0.242.b08-0.50.amzn1 will be installed
--> Processing Dependency: java-1.8.0-openjdk(x86-64) = 1:1.8.0.242.b08-0.50.amzn1 for package: 1:java-1.8.0-openjdk-devel-1.8.0.242.b08-0.50.amzn1.x86_64
--> Processing Dependency: libjvm.so()(64bit) for package: 1:java-1.8.0-openjdk-devel-1.8.0.242.b08-0.50.amzn1.x86_64
--> Processing Dependency: libjava.so()(64bit) for package: 1:java-1.8.0-openjdk-devel-1.8.0.242.b08-0.50.amzn1.x86_64
--> Running transaction check
---> Package java-1.8.0-openjdk.x86_64 1:1.8.0.242.b08-0.50.amzn1 will be installed
---> Package java-1.8.0-openjdk-headless.x86_64 1:1.8.0.242.b08-0.50.amzn1 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

=====
Package                                Arch                                Version
=====
Installing:
java-1.8.0-openjdk-devel                x86_64                            1:1.8.0.242.b08-0.50.amzn1
Installing for dependencies:
java-1.8.0-openjdk                     x86_64                            1:1.8.0.242.b08-0.50.amzn1
java-1.8.0-openjdk-headless             x86_64                            1:1.8.0.242.b08-0.50.amzn1
=====

Transaction Summary
-----
Install 1 Package (+2 Dependent packages)
```

Yum install 명령어 사용 – jdk 설치

Yum install 명령어를 사용하여 다음 버전을 설치하였습니다.



소개



목차



내용



배운 점



반성할 점



소개



목차



내용



배운 점



반성할 점

2. JDK 설치

4. Java -version

```
[root@ip-172-31-14-215 ec2-user]# java -version
openjdk version "1.8.0_302"
OpenJDK Runtime Environment (build 1.8.0_302-b08)
OpenJDK 64-Bit Server VM (build 25.302-b08, mixed mode)
[root@ip-172-31-14-215 ec2-user]#
```

Java -version

Jdk가 제대로 설치되었는지 확인하기 위해
Java -version 명령어를 통해 설치를 확인합니다.



소개



목차



내용



배운 점



반성할 점

3. Tomcat 설치

1. Tomcat .tar

```
[root@ip-172-31-14-215 ec2-user]# wget https://mirror.navercorp.com/apache/tomcat/tomcat-8/v8.5.69/bin/apache-tomcat-8.5.69.tar.gz
```

Tomcat 버전.tar 다운로드 링크 주소 확보

톰캣 버전.tar 링크주소를 가져와 wget 명령어를 이용하여
링크의 다운로드 파일을 다운받는다

톰캣 홈페이지 : <https://tomcat.apache.org/download-80.cgi>

```
[root@ip-172-31-14-215 tomcat8.5]# tar xvfz apache-tomcat-8.5.69.tar.gz
```

받은 tar 파일의 gz(압축) 해제

다운 받은 파일의 압축을 풀기위해
Tar xvfz 명령어를 사용하여 압축을 풀어준다.



소개



목차



내용



배운 점



반성할 점

3. Tomcat 설치

1. Tomcat .tar

```
[root@ip-172-31-14-215 tomcat8.5]# pwd
/usr/local/tomcat8.5
[root@ip-172-31-14-215 tomcat8.5]#
```

Tomcat 서버를 관리할 디렉토리 확보

원하는 디렉토리 공간을 만들어 톰캣 서버를 관리할 공간을 지정해 둔다. 해당 공간을 만든 뒤, mv 명령어를 이용하여 압축 해제된 파일을 만든 디렉토리로 옮기도록 한다.

```
mv apache-tomcat-8.5.69 /usr/local/tomcat8.5
```



소개



목차



내용



배운 점



반성할 점

3. Tomcat 설치

2. 환경변수 설정 - vi /etc/profile

```
export JAVA_HOME=/usr/lib/jvm/java-1.8.0-openjdk-1.8.0.302.b08-0.amzn2.0.1.x86_64/jre/bin/java
export PATH=$PATH:$JAVA_HOME/bin
export CLASSPATH=$JAVA_HOME/jre/lib:$JAVA_HOME/lib/tools.jar
export CATALINA_HOME=/usr/local/tomcat8.5
```

profile 파일에 환경변수 값을 추가하여 앞서 설치한 JDK와 톰캣을 사용할 수 있는 환경 조성

vi editor 를 이용하여 profile 파일을 열고 맨 아래에 다음과 같은 환경변수를 선언하였다.

```
export JAVA_HOME=/usr/lib/jvm/java-1.8.0-openjdk-1.8.0.302.b08-0.amzn2.0.1.x86_64/jre/bin/java
export PATH=$PATH:$JAVA_HOME/bin
export CLASSPATH=$JAVA_HOME/jre/lib:$JAVA_HOME/lib/tools.jar
export CATALINA_HOME=/usr/local/tomcat8.5
```



소개



목차



내용



배운 점



반성할 점

3. Tomcat 설치

3. 한글 인코딩 설정 - vi ~/tomcat8.5/conf/server.xml

```
<Connector port="8080" protocol="HTTP/1.1"
    URIEncoding="UTF-8"
    connectionTimeout="20000"
    redirectPort="8443" />
<!-- A "Connector" using the shared thread pool-->
<!--
```

tomcat 폴더 안에 있는 server.xml 의 인코딩 방식을 설정

vi editor 를 이용하여 server.xml 파일을 열어 다음 코드를 찾아 이와 같은 **문구**를 추가하였다.

```
<Connector port="8080"
    protocol="HTTP/1.1"
    URIEncoding="UTF-8"
    connectionTimeout="20000"
    redirectPort="8443" />
```



소개



목차



내용



배운 점



반성할 점

3. Tomcat 설치

4. 톰캣 테스트 실행

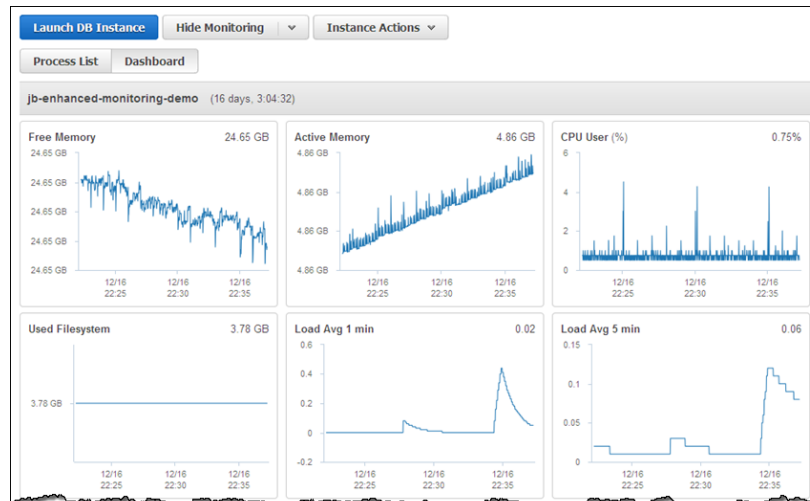
```
[root@ip-172-31-14-215 usr]# ps -ef | grep tomcat
root      7500      1  0 04:16 ?        00:01:18 //bin/java -Djava.util.logging.c
onfig.file=/usr/local/tomcat8.5/conf/logging.properties -Djava.util.logging.mana
ger=org.apache.juli.ClassLoaderLogManager -Djdk.tls.ephemeralDHKeySize=2048 -Dja
va.protocol.handler.pkgs=org.apache.catalina.webresources -Dorg.apache.catalina.
security.SecurityListener.UMASK=0027 -Dignore.endorsed.dirs= -classpath /usr/loc
al/tomcat8.5/bin/bootstrap.jar:/usr/local/tomcat8.5/bin/tomcat-juli.jar -Dcatali
na.base=/usr/local/tomcat8.5 -Dcatalina.home=/usr/local/tomcat8.5 -Djava.io.tmpd
ir=/usr/local/tomcat8.5/temp org.apache.catalina.startup.Bootstrap start
root      8921   8736  0 09:18 pts/0    00:00:00 grep --color=auto tomcat
[root@ip-172-31-14-215 usr]#
```

Ps -ef | grep tomcat 명령어를 통해 서비스 작동 확인

잘 작동하는 것을 확인할 수 있다.

4. MySQL (RDS)

1. AWS RDS 인스턴스



AWS RDS (MySQL)을 사용하는 이유

1. 실시간으로 데이터베이스를 모니터링할 수 있다.
2. AWS EC2와 함께 사용할 경우 더 효과적인 자원관리가 가능하다.
3. 데이터베이스의 스냅샷 관리, 작업 명령 등을 손쉽게 할 수 있다.
4. 데이터베이스 레벨에서 접속이 가능한 ip규칙이나 보안규칙을 만들 수 있다.



소개



목차



내용



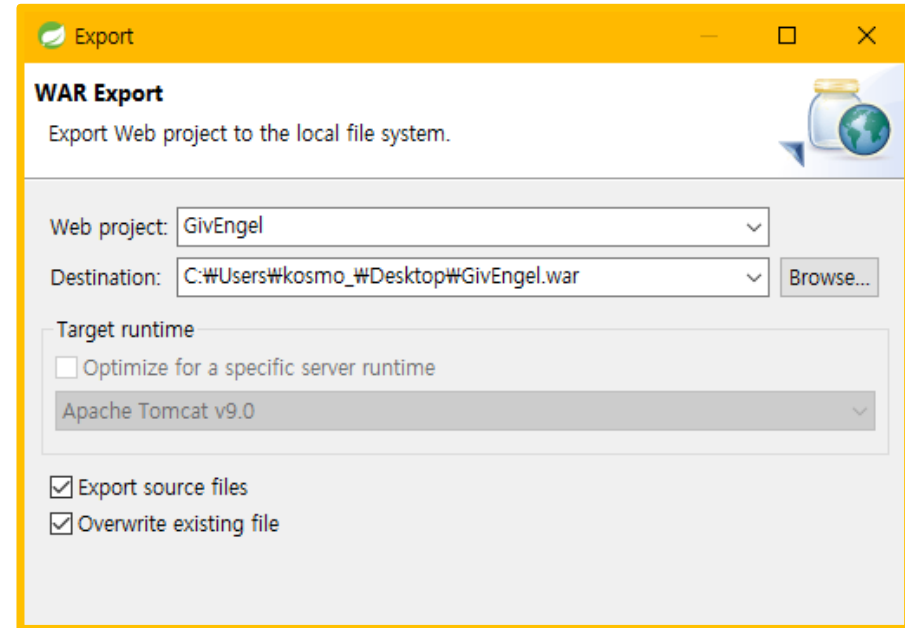
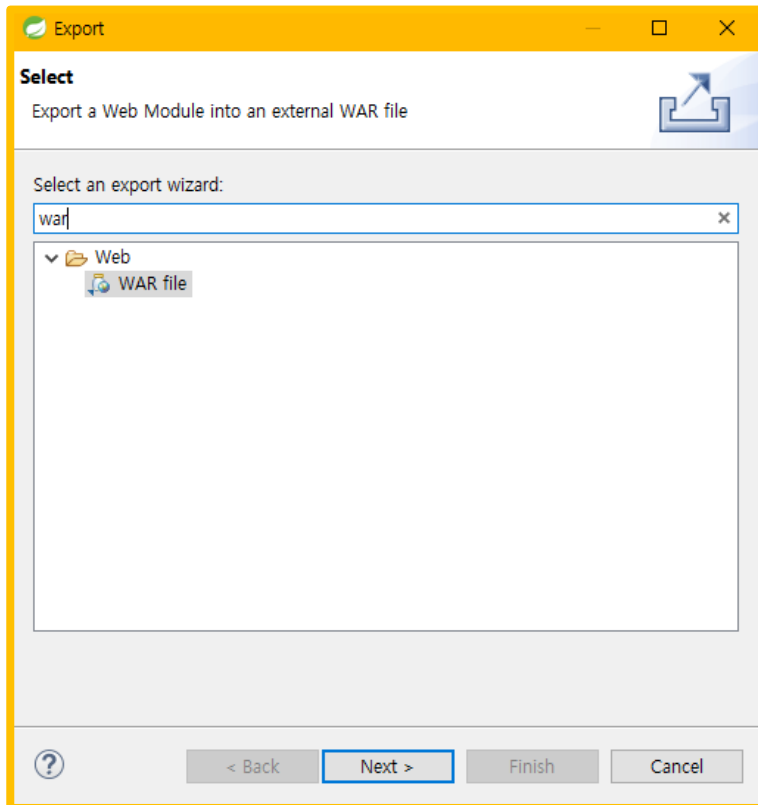
배운 점



반성할 점

5. War 배포

1. STS Export



STS / Export - WAR file

팀원들과 잠을 줄이며 피 땀 눈물로 만든 프로젝트를 war 형식으로 Export한다.



소개



목차



내용



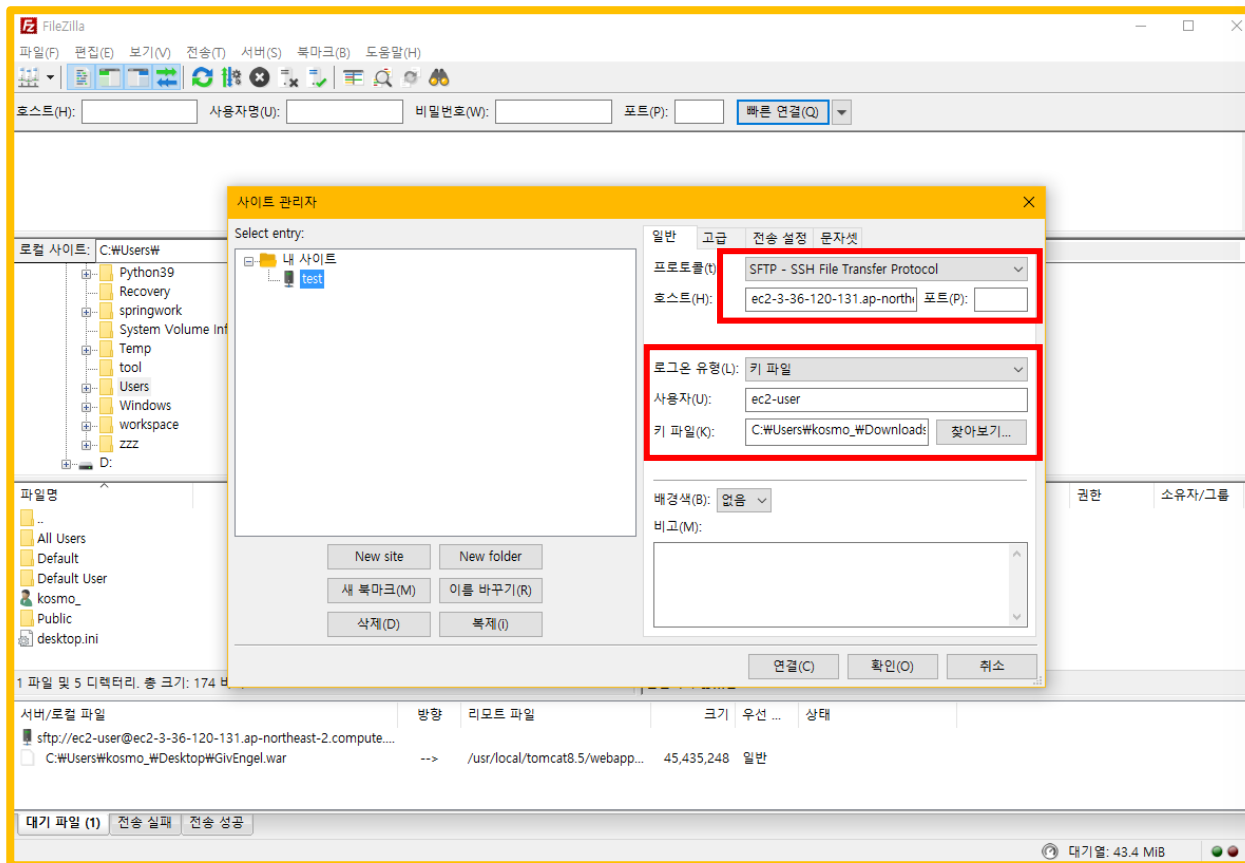
배운 점



반성할 점

5. War 배포

2. FileZilla



Putty와 동일한 조건을 가지고 FileZilla의 사이트 입장 권한을 설정해주도록 한다.

- ① 프로토콜 : SFTP – SSH File Transfer Protocol
- ② 호스트 : EC2 인스턴스의 엔드포인트
- ③ 로그인 유형 : 키 파일
 - EC2 사용자 이름 입력 : (default- ec2-user)
 - EC2 키 파일 첨부 : 인스턴스 생성 시 발급 받은 키 첨부



소개



목차



내용



배운 점



반성할 점

5. War 배포

2. FileZilla

리모트 사이트: /usr/local/tomcat8.5/webapps

파일명	크기	파일 유형	최종 수정	권한	소유자/그룹
..		파일 폴더	2021-08-11 ...	drwxr-x---	ec2-user ec...
docs		파일 폴더	2021-08-11 ...	drwxr-x---	ec2-user ec...
examples		파일 폴더	2021-08-11 ...	drwxr-x---	ec2-user ec...
GivEngel		파일 폴더	2021-08-11 ...	drwxr-x---	root root
host-manager		파일 폴더	2021-08-11 ...	drwxr-x---	ec2-user ec...
manager		파일 폴더	2021-08-11 ...	drwxr-x---	ec2-user ec...
ROOT		파일 폴더	2021-08-11 ...	drwxr-x---	ec2-user ec...
GivEngel.war	45,435,248	WAR 파일	2021-08-11 ...	-rw-rw-r--	ec2-user ec...

1 파일 및 6 디렉터리. 총 크기: 45,435,248 바이트

FileZilla 업로드를 통해 WAR 파일을 앞서 만들어주었던 서버공간 안에 있는 webapps 안에 넣어주도록 한다.



소개



목차



내용



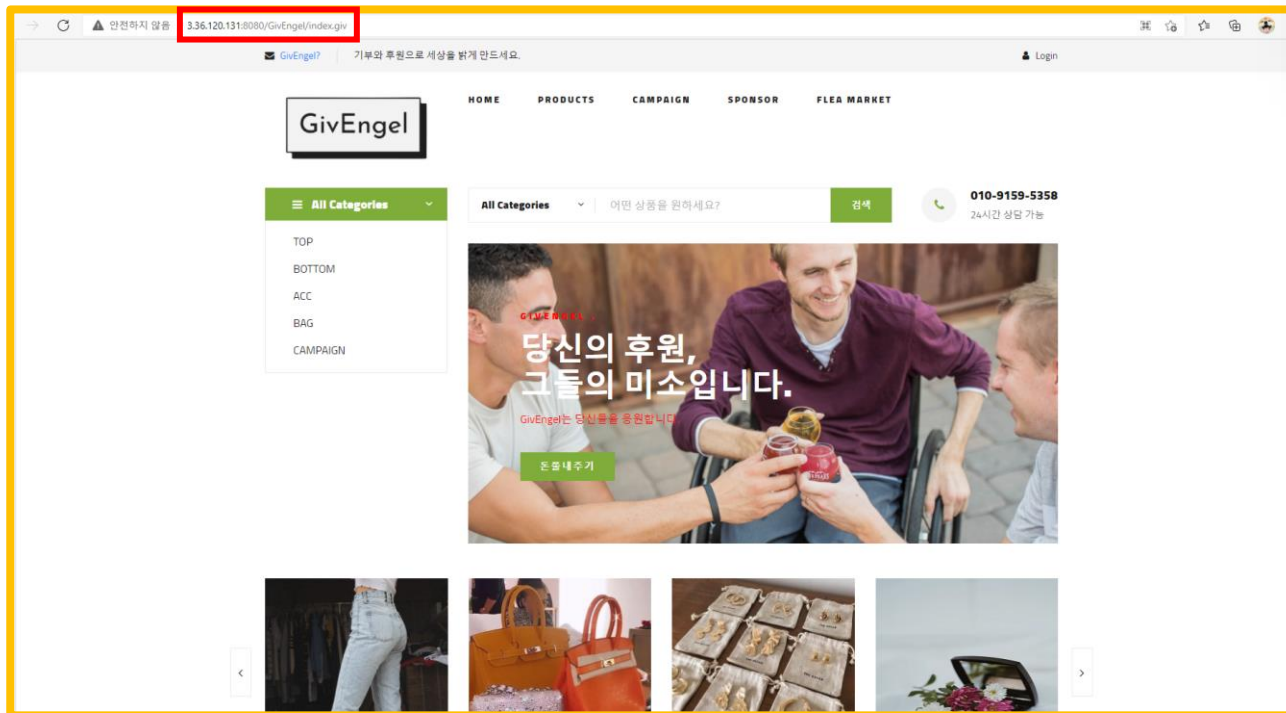
배운 점



반성할 점

6. 정상 실행 확인

1. IP주소 + 프로젝트 + 페이지



정상적으로 작동되는 것을 볼 수 있습니다.

EC2 퍼블릭 액세스 IP:서버포트/프로젝트명/페이지명



소개



목차



내용



배운 점



반성할 점



소개



목차



내용



배운 점



반성할 점

1. 배운 점

1. 만든 프로젝트를 배포하는 방법

2. 다양한 리눅스 명령어

3. 아마존 웹 서비스의 생태계

4. 리눅스를 통한 서버 운용과 웹 배포

5. 보안 관리와 자원 관리의 중요성



소개



목차



내용



배운 점



반성할 점

1. 반성할 점

1. EC2 보안키 분실 – 새로 만듦

2. EC2의 기본 권한이 Root인 줄 알고 파일을 수정하다가 권한 문제로 작업을 날려먹었음

3. 아마존 서비스를 통해 만들어 자원 관리가 미흡할 경우 요금이 부과될 위험에 대해 대처하지 못하고 있음