

- AWS EC2 인스턴스 생성 및 환경 설정
- 1. EC2 인스턴스 생성 & 탄력적 IP 연결
- 2. EC2 서버 접속하기
- 3. AWS EC2(Linux) 서버 생성시 꼭 해야 할 설정들
 - 3-1. Java 8 또는 Java 11 설치
 - 3-2. 타임존 변경
 - 3-3. hostname 설정

AWS EC2 인스턴스 생성 및 환경 설정

EC2(Elastic Compute Cloud)는 AWS에서 제공하는 성능, 용량 등을 유동적으로 사용할 수 있는 서버이다.

AWS에서 무료로 제공하는 프리티어 플랜에서는 EC2 사용에 다음과 같은 제한이 있다.

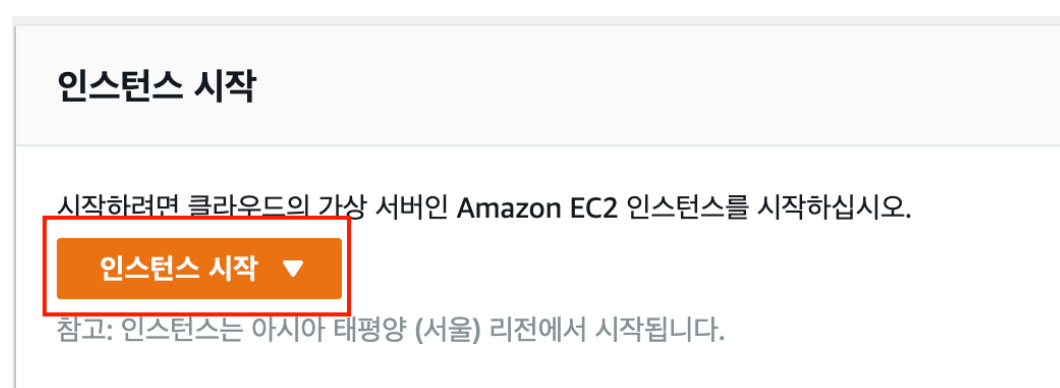
- 사양이 t2.micro만 가능하다.
- 월 750시간의 제한이 있다. 초과하면 비용이 부과된다.
- $24 \times 31 = 744$ 시간이다. 즉 1대의 t2.micro만 사용한다면 24시간 사용할 수 있다.
- 저번에 2~3개 프리티어 인스턴스를 1달 24시간 돌리니깐 20불정도 부과된 기억이 있다.
 - 탄력적 IP 비용 포함. 탄력적 IP는 1개가 초과되면 비용이 부과되기 시작한다. ([요금 정책](#))

1. EC2 인스턴스 생성 & 탄력적 IP 연결

1) 본인의 리전을 '[서울](#)'로 변경해준다.




2) ec2에 접속 > [인스턴스 시작](#) 버튼 클릭



인스턴스 시작

3) AMI : Amazon Linux 2 AMI 선택

책에서는 Amazon Linux AMI를 사용하였지만 해당 버전은 2020년 12월 31일 종료되었다. (참고)


Amazon Linux
프리 티어 사용 가능

Amazon Linux 2 AMI (HVM), SSD Volume Type -
ami-0a0de518b1fc4524c (64비트 x86) / ami-03daae882e94f15e0 (64비트 Arm)

Amazon Linux 2는 5년간 지원을 제공합니다. Amazon EC2에 성능 최적화된 Linux kernel 4.14와 systemd 219, GCC 7.3, Glibc 2.26, Binutils 2.29.1, 최신 소프트웨어 패키지를 추가적으로 제공합니다.

루트 디바이스 유형: ebs 가상화 유형: hvm ENA 활성화: 예

선택

☒ 64비트(x86)
☐ 64비트(Arm)

Linux 2 AMI

Amazon Linux2는 Centos 7버전 자료들을 그대로 사용할 수 있다. 그럼 굳이 Centos AMI를 사용하지 않고 아마존 리눅스 AMI를 사용한 이유는 다음과 같다.

- 아마존이 개발하고 있기 때문에 지원받기 쉽다.
- 레드햇 베이스이므로 레드햇 계열의 배포판을 많이 다뤄본 사람일수록 문제없이 사용할 수 있다.
- AWS의 각종 서비스와의 상성이 좋다.
- Amazon 독자적인 개발 레포지토리를 사용하고 있어 yum이 매우 빠르다.

4) 인스턴스 유형 t2.micro 선택 > 인스턴스 세부 정보 구성은 패스

현재 선택된 항목: t2.micro (- ECU, 1 vCPUs, 2.5 GHz, -, 1 GiB 메모리, EBS 전용)

	그룹	유형	vCPUs	메모리 (GiB)	인스턴스 스토리지 (GB)	EBS 최적화 사용 가능	네트워크 성능	IPv6 지원
<input type="checkbox"/>	t2	t2.nano	1	0.5	EBS 전용	-	낮음에서 중간	예
<input checked="" type="checkbox"/>	t2	t2.micro 프리 티어 사용 가능	1	1	EBS 전용	-	낮음에서 중간	예
<input type="checkbox"/>	t2	t2.small	1	2	EBS 전용	-	낮음에서 중간	예
<input type="checkbox"/>	t2	t2.medium	2	4	EBS 전용	-	낮음에서 중간	예
<input type="checkbox"/>	t2	t2.large	2	8	EBS 전용	-	낮음에서 중간	예
<input type="checkbox"/>	t2	t2.xlarge	4	16	EBS 전용	-	보통	예
<input type="checkbox"/>	t2	t2.2xlarge	8	32	EBS 전용	-	보통	예
<input type="checkbox"/>	t3	t3.nano	2	0.5	EBS 전용	예	최대 5기가비트	예
<input type="checkbox"/>	t3	t3.micro	2	1	EBS 전용	예	최대 5기가비트	예
<input type="checkbox"/>	t3	t3.small	2	2	EBS 전용	예	최대 5기가비트	예
<input type="checkbox"/>	t3	t3.medium	2	4	EBS 전용	예	최대 5기가비트	예
<input type="checkbox"/>	t3	t3.large	2	8	EBS 전용	예	최대 5기가비트	예
<input type="checkbox"/>	t3	t3.xlarge	4	16	EBS 전용	예	최대 5기가비트	예
<input type="checkbox"/>	t3	t3.2xlarge	8	32	EBS 전용	예	최대 5기가비트	예
<input type="checkbox"/>	t3a	t3a.nano	2	0.5	EBS 전용	예	최대 5기가비트	예

취소

이전

검토 및 시작

다음: 인스턴스 세부 정보 구성

5) 스토리지 구성 > 크기(GiB) 30으로 설정

서버의 용량의 기본값은 8GB이다. 모르고 넘어가는 분들이 있는데 30GB까지 프리티어로 가능하다. 물론 그 이상은 비용이 청구된다.

단계 4: 스토리지 추가
인스턴스가 다음 스토리지 디바이스 설정으로 시작됩니다. 추가 EBS 볼륨 및 인스턴스 스토어 볼륨을 인스턴스에 연결하거나 루트 볼륨의 설정을 편집할 수 있습니다. 인스턴스를 시작한 후 추가 EBS 볼륨을 연결할 수도 있지만, 인스턴스 스토어 볼륨은 연결할 수 없습니다. Amazon EC2의 스토리지 옵션에 대해 자세히 알아보십시오.

볼륨 유형	디바이스	스냅샷	크기(GiB)	볼륨 유형	IOPS	처리량(MB/초)	종료 시 삭제	암호화
루트	/dev/xvda	snap-0e9d6c6149b136780	30	범용 SSD(gp2)	100/3000	해당 사항 없음	<input checked="" type="checkbox"/>	암호화도

스토리지 구성

6) 태그 추가 > 키 (Name) , 값 (인스턴스 표현할 이름) 입력

웹 콘솔에서 표기될 태그인 Name 태그를 등록한다. 태그는 해당 인스턴스를 표현하는 여러 이름으로 사용될 수 있다.

단계 5: 태그 추가
태그는 대소문자를 구별하는 키-값 페어로 이루어져 있습니다. 예를 들어 키가 Name이고 값이 Webserver인 태그를 정의할 수 있습니다. 태그 복사본은 볼륨, 인스턴스 또는 둘 다에 적용될 수 있습니다. 태그는 모든 인스턴스 및 볼륨에 적용됩니다. Amazon EC2 리소스 태그 지정에 대해 자세히 알아보기.

키 (최대 128자)	값 (최대 256자)	인스턴스	볼륨	네트워크 인터페이스
Name	spring-deploy-test	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<div>다른 태그 추가 (최대 50개 태그)</div>				

태그 추가

7) 보안 그룹 추가 > [내 IP 설정, 기본 8080 포트 설정] > 검토 및 시작 클릭

보안 그룹은 방화벽을 이야기한다. '서버로 80포트 외에는 허용하지 않는다'는 역할을 하는 방화벽이 AWS에서는 보안 그룹으로 사용된다.

- SSH하면서 포트 항목에서 22인 경우
 - AWS EC2에서 터미널로 접속할 때를 이야기한다. pem키가 없으면 접속이 안 되니 전체 오픈(0.0.0.0/0, ::/0)하는 경우가 종종 있는데 이렇게 되면 이후 파일 공유디렉토리나 깃허브 등에 실수로 pem키가 노출되는 순간 서버에서 가상화폐가 채굴되는 것을 볼 수 있을 것이다.
 - 지정된 IP만 접속이 가능하도록 설정하고 다른 장소에서 접속할 경우 해당 장소 IP를 다시 능동적으로 SSH 규칙에 추가하는 것이 안전하다.

보안 그룹 할당: ☒ 새 보안 그룹 생성
☐ 기존 보안 그룹 선택

보안 그룹 이름:
 설명:

유형	프로토콜	포트 범위	소스	설명
SSH	TCP	22	내 IP	예: SSH for Adm
사용자 지정 TCP	TCP	8080	사용자 지정 0.0.0.0/0, ::/0	예: SSH for Adm
HTTPS	TCP	443	사용자 지정 0.0.0.0/0, ::/0	예: SSH for Adm

규칙 추가

보안 그룹 설정

8) 인스턴스 검토 > 시작하기 > pem키 생성 혹은 선택 > 인스턴스 시작

인스턴스로 접근하기 위해서는 pem키가 필요하다. 그래서 마지막 단계는 할당할 pem키를 선택해준다. **일종의 마스터 키이기 때문에 절대 유출하면 안된다.** 잘 관리할 수 있는 디렉토리로 저장한다. 기존에 생성된 pem키가 있다면 선택하고 없으면 생성한다.

참고: 선택한 키 페어가 이 인스턴스에 대해 승인된 키 세트에 추가됩니다. 퍼블릭 AMI에서 기존 키 페어 제거에 대해 자세히 알아보십시오.

새 키 페어 생성

Key pair type
☒ RSA ☐ ED25519

키 페어 이름

키 페어 다운로드

계속하려면 먼저 프라이빗 키 파일(*.pem 파일)을 다운로드해야 합니다. 액세스할 수 있는 안전한 위치에 저장합니다. 파일은 생성되고 나면 다시 다운로드할 수 없습니다.

취소 **인스턴스 시작**

pem키 생성 혹은 선택

9) 인스턴스 생성 완료

인스턴스 시작 연결 작업

search : i-02010aaaa30a909db 필터 추가

Name	인스턴스 ID	인스턴스 유형	가용 영역	인스턴스 상태	상태 검사
spring-deplo...	i-02010aaaa30a909...	t2.micro	ap-northeast-2a	running	초기...

인스턴스 생성 완료

10) EIP 할당하기

인스턴스도 결국 하나의 서버이기 때문에 IP가 존재한다. 인스턴스 생성 시에 항상 새 IP를 할당하는데, 한 가지 조건이 더 있다. 같은 인스턴스를 중지하고 다시 시작할 때도 새 IP가 할당된다. 즉, 요금을 아끼기 위해 잠깐 인스턴스를 중지하고 다시 시작하면 IP가 변경되는 것이다.

매번 접속해야 하는 IP가 변경돼서 매우 번거로우므로 인스턴스의 IP가 매번 변경되지 않고 고정 IP를 가지게 해야 한다. 그래서 **고정 IP**를 할당해준다.

10-1) 메뉴 : 탄력적 IP 클릭 > 탄력적 IP 주소 할당 버튼 클릭 > 할당 버튼 클릭

탄력적 IP 주소 할당

탄력적 IP 주소 설정

네트워크 경계 그룹

ap-northeast-2

퍼블릭 IPv4 주소 풀

Amazon의 IPv4 주소 풀

AWS 계정으로 가져오는 퍼블릭 IPv4 주소 (풀을 찾을 수 없으므로 옵션이 비활성화됨) 자세히 알아보기

IPv4 주소의 고객 소유 풀 (고객 소유 풀을 찾을 수 없기 때문에 옵션이 비활성화됨) 자세히 알아보기

글로벌 정적 IP 주소

AWS Global Accelerator는 AWS 엣지 로케이션의 애니캐스트를 사용하여 전 세계에 발표된 글로벌 정적 IP 주소를 제공할 수 있습니다. 이를 통해 Amazon 글로벌 네트워크를 사용하여 사용자 트래픽의 가용성과 지연 시간을 개선할 수 있습니다. 자세히 알아보기

액셀러레이터 생성

태그 - 선택 사항

태그는 사용자가 AWS 리소스에 할당하는 레이블입니다. 각 태그는 키와 값(선택 사항)으로 구성됩니다. 태그를 사용하여 리소스를 검색 및 필터링하거나 AWS 비용을 추적할 수 있습니다.

리소스에 연결된 태그가 없습니다.

새로운 태그 추가

최대 50개의 태그를 더 추가할 수 있습니다.

취소

할당

탄력적 IP 할당

10-2) 생성된 탄력적 IP 클릭 > 작업 > 탄력적 IP 주소 연결 클릭

탄력적 IP 주소

작업

세부 정보 보기

탄력적 IP 주소 릴리스

탄력적 IP 주소 연결

탄력적 IP 주소 연결 해제

할당된 IP 주소

탄력적 IP 주소 연결

10-3) 생성한 EC2 인스턴스(spring-deploy-test) 선택 > 연결

리소스 유형
탄력적 IP 주소를 연결할 리소스의 유형을 선택합니다.

☒ 인스턴스
☐ 네트워크 인터페이스

⚠

탄력적 IP 주소가 이미 연결되어 있는 인스턴스에 탄력적 IP 주소를 연결하면 이전에 연결된 탄력적 IP 주소가 연결 해제되지만 계정에는 계속 할당된 상태로 남아 있습니다. [자세히 알아보기](#)

인스턴스

Q i-02010aaaa30a909db

×

↻

프라이빗 IP 주소

탄력적 IP 주소를 연결할 프라이빗 IP 주소입니다.

Q 프라이빗 IP 주소 선택

재연결

이미 리소스에 연결되어 있는 탄력적 IP 주소를 다른 리소스에 재연결할 수 있는지 여부를 지정합니다.

☐ 이 탄력적 IP 주소를 재연결하도록 허용

취소

연결

인스턴스 연결

EC2 인스턴스로 돌아가 확인해보면 탄력적 IP가 할당되었음을 확인할 수 있다.

필터 추가

	Name	인스턴스 ID	인스턴스 유형	가용 영역	인스턴스 상태	상태 검사	경보 상태	퍼블릭 DNS(IPv4)	IPv4 퍼블릭 IP	IP
	spring-deplo	i-02010aaaa30a909...	t2.micro	ap-northeast-2a	running	2/2개 검사 ...	없음	ec2-3-35-90-28.ap-nort...	3.35.90.28	-

인스턴스 확인

2. EC2 서버 접속하기

Mac에서 EC2로 접속해보자. 만약 오랜 시간 접속이 안 되거나, 권한이 없어서 안 된다고 메시지가 나온다면 다음을 확인해보자.

- HostName 값이 정확히 탄력적 IP로 되어있는지 확인
- EC2 인스턴스가 running 상태인지 확인
- EC2 인스턴스의 보안그룹 → 인바운드 규칙에서 현재 본인의 IP가 등록되어 있는지 확인

AWS와 같은 외부 서버로 SSH 접속을 하려면 다음과 같이 긴 명령어를 입력해야 한다.

- **ssh -i pem 키 위치 EC2의 탄력적 IP 주소**

1) pem키 ~/.ssh 로 복사하기

매번 입력하기 상당히 귀찮으므로 쉽게 접속할 수 있도록 설정하자. 좀 전에 받은 키페어 pem 파일을 ~/.ssh/로 복사한다. ~/.ssh/ 디렉토리 로 pem 파일을 옮겨 놓으면 ssh 실행 시 pem 키 파일을 자동으로 읽어 접속을 진행한다.

이후부터는 별도로 pem 키 위치를 명령어로 지정할 필요가 없게 된다.

- **cp pem 키를 내려받은 위치 ~/.ssh/**

```
→ ~ cp ~/Documents/spring-deploy-test.pem ~/.ssh/  
→ ~
```

pem키 복사

2) ~/.ssh로 이동 및 파일 목록 확인하기

pem 키가 잘 복사되었는지 ~/.ssh 디렉토리로 이동해서 파일 목록을 확인해 보자.

- `cd ~/.ssh/`
- `ll`

```
→ ~ cd ~/.ssh/  
→ .ssh ll | grep deploy  
-rw-r--r--@ 1 ijong-won  staff   1.7K  8 23 22:35 spring-deploy-test.pem
```

~/.ssh/

3) pem키 권한 변경

복사되었다면 pem 키의 권한을 변경해주자. chmod는 [file | directory][user (rwx)][group (rwx)][others (rwx)]로 구성 된다. 각 권한은 r= 4, w= 2, x=1으로 나타내진다. 만약 user와 group에게 read 권한을 부여하려면 chmod 440으로 적용하면 된다.

- `chmod 600 ~/.ssh/{pem 키 이름}`

```
→ .ssh chmod 600 ~/.ssh/spring-deploy-test.pem
```

pem 키 권한 변경

4) config 파일 생성 및 권한 설정

권한을 변경하였다면 config 파일 생성 후 본인이 원하는 Host로 등록한다. **Host를 앞으로 접속할 키 값으로 보면 된다.** Host 외에 **HostName은 탄력적 IP 주소**를 사용하면 된다. 해당 파일 설정 완료 후 실행 권한이 필요하므로 권한 설정을 해준다.

- `vim ~/.ssh/config`
 - i: 입력, :wq: 저장 후 종료
- `chmod 700 ~/.ssh/config`

```
# spring-deploy-test  
Host test  
    HostName 3.35.90.28  
    User ec2-user  
    IdentityFile ~/.ssh/spring-deploy-test.pem
```

config 설정

```
→ .ssh chmod 700 ~/.ssh/config
```

권한 설정

5) EC2 접속

이제 ssh {Host 이름}으로 EC2에 접속해보자. yes를 입력하면 다음과 같이 EC2 접속에 성공한다.

- ssh test

```
→ .ssh ssh test
The authenticity of host '3.35.90.28 (3.35.90.28)' can't be established.
ECDSA key fingerprint is SHA256:1Fi0HDC+/hyK+vQeK0l/miBLaCfLKA0B5EDdg2VQMpU.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '3.35.90.28' (ECDSA) to the list of known hosts.

      __|  __|_  )
      _| (      /   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-172-31-7-15 ~]$
```

EC2 접속 성공

SSH 접속시 RSA 공유키 충돌 문제 해결 방법

```
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@      WARNING: REMOTE HOST IDENTIFICATION HAS CHANGED!      @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
IT IS POSSIBLE THAT SOMEONE IS DOING SOMETHING NASTY!
Someone could be eavesdropping on you right now (man-in-the-middle attack)!
It is also possible that a host key has just been changed.
The fingerprint for the ECDSA key sent by the remote host is
SHA256:1Fi0HDC+/hyK+vQeK0l/miBLaCfLKA0B5EDdg2VQMpU.
Please contact your system administrator.
```

해당 충돌 발생 이유는 기존에 접속한 적이 있는 서버와 RSA 공유키를 교환한 상태에서 또 다른 서버로 접속을 시도해서 그렇다. 경고 메시지는 Man in the Middle Attack 이라는 일명 '중간자 공격'에 대해 경고한다. 즉, 기존에 서버가 알고있던 정보를 찾아서 따라갔더니- 기존과는 전혀 다른 서버로 접속되었다는 것이다.

하지만 이 경우는 운영자인 내가 고의적으로 변경한 것이기 때문에 해킹 당한 것은 아니다. (스푸핑)

이를 해결하기 위해 간단하게 명령어를 통해 초기화 시켜주면 된다.

- ssh-keygen -R [IP or DomainName]

```
→ .ssh ssh-keygen -R 3.35.90.28
# Host 3.35.90.28 found: line 17
/Users/ijong-won/.ssh/known_hosts updated.
Original contents retained as /Users/ijong-won/.ssh/known_hosts.old
```

ssh-keygen 초기화

3. AWS EC2(Linux) 서버 생성시 꼭 해야 할 설정들

3-1. Java 8 또는 Java 11 설치

yum으로 설치 가능한 Java는 버전 8까지 가능하다. 설치 가능한 jdk를 확인해보면 7, 8버전이 있다.

- `yum list java*jdk-devel`

```
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
```

설치 가능한 jdk 확인

Java 8 설치하기

Java 8을 사용하려면 다음 명령어를 실행해주면 된다. 설치 후 버전을 확인해주면 다음과 같이 나온다.

- `sudo yum install -y java-1.8.0-openjdk-devel.x86_64`
- `java -version`

```
[ec2-user@ip-172-31-7-15 ~]$ 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)
```

java 8 설치 및 확인

Java 11 설치하기

Java 11은 직접 설치를 해줘야 한다. 설치 방법은 두 가지가 있다. 나는 아마존에서 제공하는 **AWS Corretto**를 이용해서 다운을 받을 것이다. 왜냐하면 오라클에서는 ec2에 파일을 직접 받아 .sh파일 수정 후 직접 실행해줘야 되어서 좀 더 귀찮기 때문이다.

- 아마존에서 제공하는 OpenJDK인 AWS Corretto를 다운받아 설치 ([check](#))
- 오라클에서 직접 OpenJDK 파일을 받아 설치 후 실행 ([check](#))

AWS Corretto 다운 방법은 다음과 같다.

- `wget https://corretto.aws/downloads/latest/amazon-corretto-11-x64-al2-jdk.rpm -O jdk11.rpm`
- 또는 `curl -L https://corretto.aws/downloads/latest/amazon-corretto-11-x64-al2-jdk.rpm -o jdk11.rpm`

```
[ec2-user@ip-172-31-7-15 ~]$ curl -L https://corretto.aws/downloads/latest/amazon-corretto-11-x64-al2-jdk.rpm -o java11.rpm
```

Corretto 설치

JDK 11를 설치해준다. yes를 입력하면 설치가 완료된다.

- `sudo yum localinstall jdk11.rpm`

```
[ec2-user@ip-172-31-7-15 ~]$ sudo yum localinstall jdk11.rpm
```

jdk11 설치

해당 명령어로 jdk 버전을 11로 변경해준다.

- `sudo /usr/sbin/alternatives --config java`
- `java -version`

```
[ec2-user@ip-172-31-7-15 ~]$ java -version
openjdk version "11.0.12" 2021-07-20 LTS
OpenJDK Runtime Environment Corretto-11.0.12.7.1 (build 11.0.12+7-LTS)
OpenJDK 64-Bit Server VM Corretto-11.0.12.7.1 (build 11.0.12+7-LTS, mixed mode)
```

jdk 11 설치 완료

3-2. 타임존 변경

EC2 서버의 기본 타임존은 UTC이다. 이는 세계 표준 시간으로 한국의 시간대가 아니다. 즉, 한국의 시간과는 9시간 차이가 발생한다. 이렇게 되면 서버에서 수행되는 Java 애플리케이션에서 생성되는 시간 모두 9시간씩 차이 나기 때문에 꼭 수정해줘야 한다. 수정 후 date로 타임존이 **KST**로 변경된 것을 확인할 수 있다.

- `sudo rm /etc/localtime`
- `sudo ln -s /usr/share/zoneinfo/Asia/Seoul /etc/localtime`
- `date` 변경된 타임존을 확인해본다.

```
[ec2-user@ip-172-31-7-15 ~]$ sudo rm /etc/localtime
[ec2-user@ip-172-31-7-15 ~]$ sudo ln -s /usr/share/zoneinfo/Asia/Seoul /etc/localtime
[ec2-user@ip-172-31-7-15 ~]$ date
2021. 08. 23. (월) 23:36:04 KST
```

타임존 변경하기

3-3. hostname 설정

여러 서버를 관리 중일 경우 IP만으로 어떤 서비스의 서버인지 확인이 어렵기 때문에 각 서버가 어느 서비스인지 표현하기 위해 HOSTNAME을 변경해주는 것이 좋다.

- **sudo hostnamectl set-hostname {변경할 hostname}**
- **cat /etc/hostname** 으로 변경된 hostname 확인
- **sudo reboot** 변경한 후 서버를 재부팅시켜준다

```
[ec2-user@ip-172-31-7-15 ~]$ hostname
ip-172-31-7-15.ap-northeast-2.compute.internal
[ec2-user@ip-172-31-7-15 ~]$ sudo hostnamectl set-hostname spring-deploy-test
[ec2-user@ip-172-31-7-15 ~]$ cat /etc/hostname
spring-deploy-test
```

hostname 변경하기

재부팅 후 재접속하면 변경된 HOSTNAME을 확인할 수 있다.

```
Connection to spring-deploy-test closed.
→ .ssh ssh test
Last login: Mon Aug 23 23:00:07 2021 from 59.5.192.12

  __|  __|_  )
 _| (      /   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@spring-deploy-test ~]$ █
```

재부팅 후 Hostname 변경 확인