

1장 - 배포v1 - 엘라스틱빈스톡 샘플코드 사용해보기


▶ 실행중인 인스턴스 종료

- ① 좌측 탄력적 아이피 연결 → 연결 해제
- ② 우측 상단 작업 콤보상자 → 탄력적ip주소 릴리스 → 릴리스 함.

● IAM 생성하기

- OS(EC2) 설치가 필요 없음(이미 설치가 되어 있음)
- JDK 설치가 필요 없음(이미 설치가 되어 있음)
- 오토 스케일링, 각종 소프트웨어 구성, 로드밸런서, 모니터링, 업데이트 버전관리

- ① 검색 창에 iam → 역할 만들기(역할을 자동으로 생성 안하고 직접 만들어 줘야함.)
- ② IAM 대시보드 - 좌측 역할 - 우측 역할 만들기
- ③ 엔티티 유형 - AWS 서비스
- ④ 사용사례 - EC2 → 다음
- ⑤ 권한추가 - 검색창에 awselastic



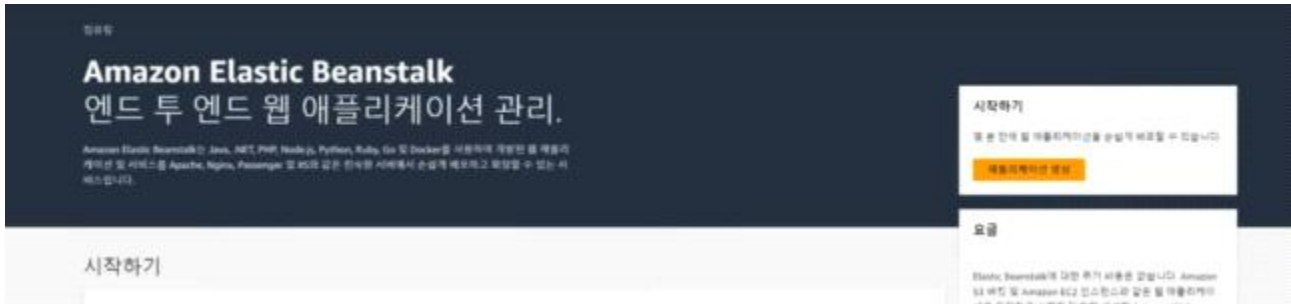
<input checked="" type="checkbox"/>	<input type="checkbox"/>	AWSElasticBeanstalkMulticontainerDocker	AWS 관리형	Prov
<input type="checkbox"/>	<input type="checkbox"/>	AWSElasticBeanstalkReadOnly	AWS 관리형	Gran
<input type="checkbox"/>	<input type="checkbox"/>	AWSElasticBeanstalkRoleCore	AWS 관리형	AWS
<input type="checkbox"/>	<input type="checkbox"/>	AWSElasticBeanstalkRoleCW	AWS 관리형	{Elas
<input type="checkbox"/>	<input type="checkbox"/>	AWSElasticBeanstalkRoleEC2	AWS 관리형	{Elas
<input type="checkbox"/>	<input type="checkbox"/>	AWSElasticBeanstalkRoleRDS	AWS 관리형	{Elas
<input type="checkbox"/>	<input type="checkbox"/>	AWSElasticBeanstalkRoleSNS	AWS 관리형	{Elas
<input type="checkbox"/>	<input type="checkbox"/>	AWSElasticBeanstalkRoleWorkerTier	AWS 관리형	{Elas
<input checked="" type="checkbox"/>	<input type="checkbox"/>	AWSElasticBeanstalkWebTier	AWS 관리형	Prov
<input checked="" type="checkbox"/>	<input type="checkbox"/>	AWSElasticBeanstalkWorkerTier	AWS 관리형	Prov

▷ 세개 체크 (WebTier, WorkerTier, MulticontainerDocker) → 엘라스틱 빈스톡에서 실행되는 아마존 EC2 인스턴스들이 파일저장소 등 접근할 수 있는 권한.

▷ 다음 누르고 역할 이름 : aws-elasticbeanstalk-ec2-role
→ 역할 생성 됨.

● 엘라스틱 빈스톡 생성하기

- 검색창에 beanstalk 검색 후 이동



① 애플리케이션 생성



- 환경티어 : 웹서버 환경
- 애플리케이션 정보 : aws-v2
- 환경 정보는 자동으로 만들어짐.
- 플랫폼 : java
- 다음 클릭

② 서비스 액세스 구성

▷ 서비스 액세스

- 서비스 역할 : 새 서비스 역할 생성 및 사용
- 서비스 역할 이름

※ 엘라스틱 빈스톡 서비스 롤 : 엘라스틱 빈스톡에서 AWS 리소스를 생성하고 관리하려면 이 역할이 필요... 나중에 jar 파일을 엘라스틱 빈스톡에 배포하고 엘라스틱 빈스톡은 EC2 서버에 배포를 해야하는데 이때 권한이 필요하다... 그래서 서비스 롤이 필요한 것임.

- 키패어는 내꺼, EC2 인스턴스 프로파일은 자동으로 잡힐거임...

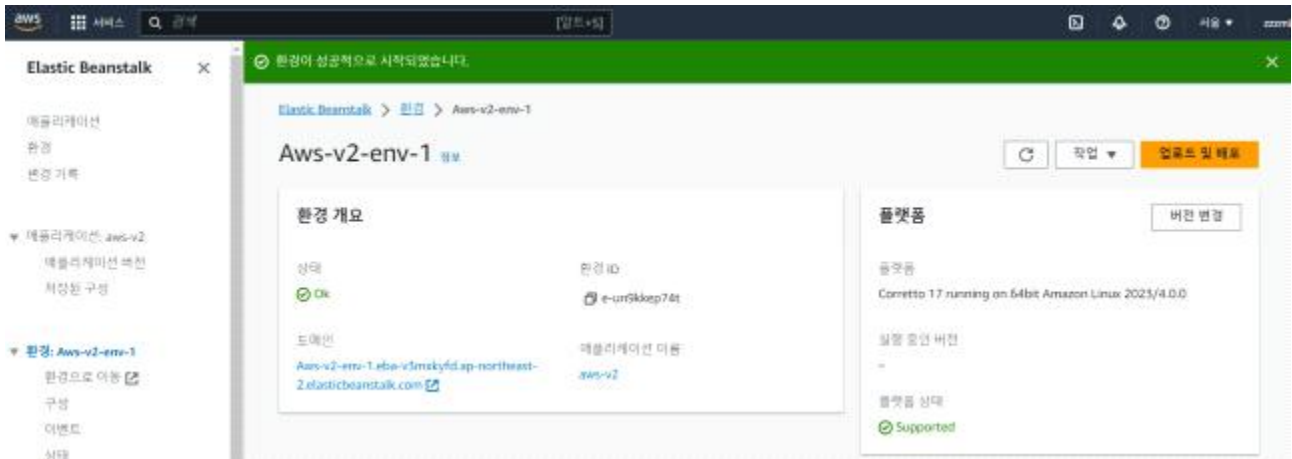
▷ 다음

③ 네트워킹, 데이터베이스 및 태그설정

- Virtual Private Cloud - 아래 콤보 누르고 보이는 것 선택
- 인스턴스 설정 → 퍼블릭 IP 주소 ---- 반드시 체크
- 인스턴스 서브넷 : ap-northeast-2a 선택 - 다른 것도 상관 없음.

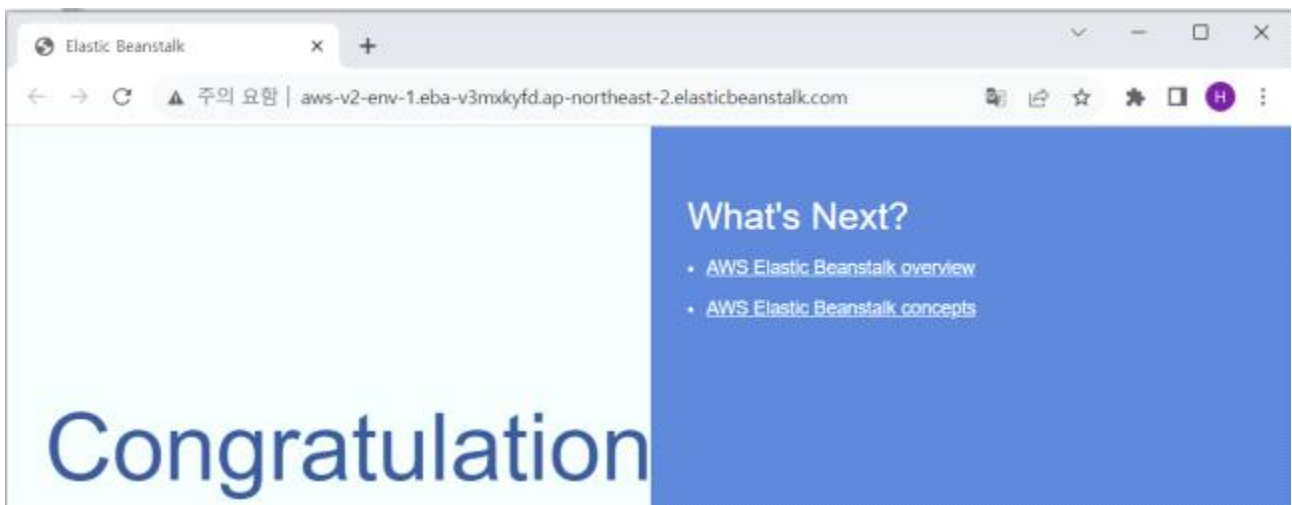
▶ 제출하기

그러면 환경이 만들어 진다.

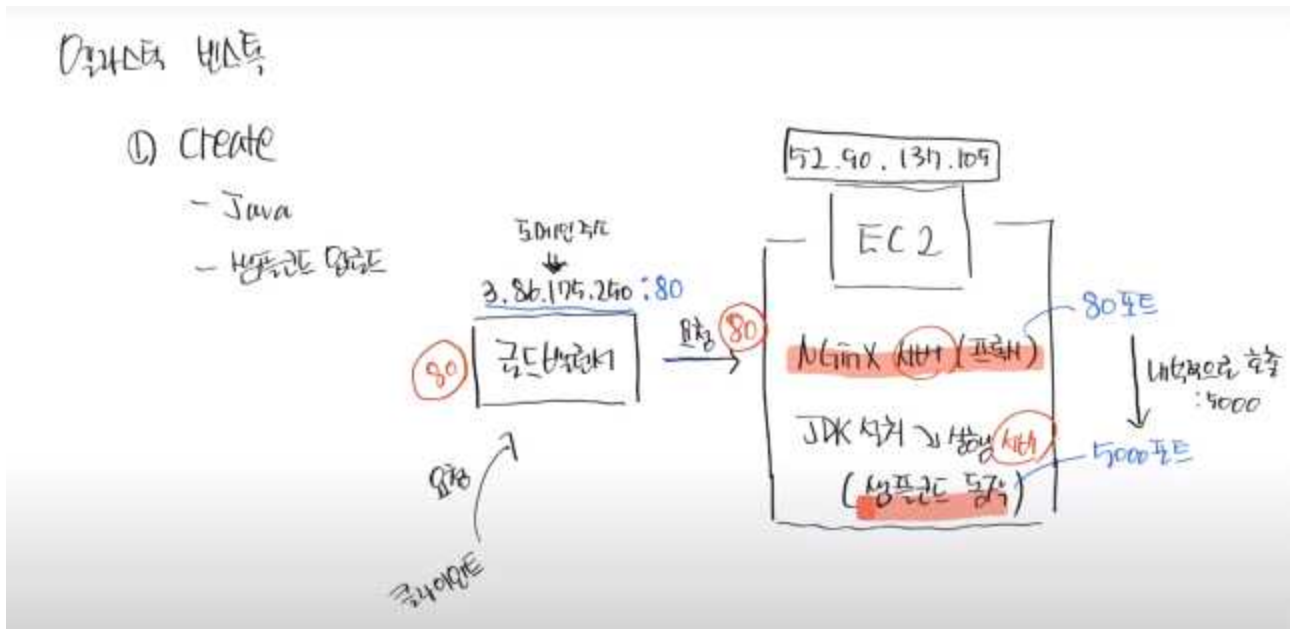


③ 좌측 - 환경으로 이동

버그나면 F5 한번 누르고 다시 시도하면 Congratulations 화면 나옴.. 샘플코드가 실행되는 거임.



2장. 배포 v2 - 엘라스틱빈스톡 내부 구성이해하기

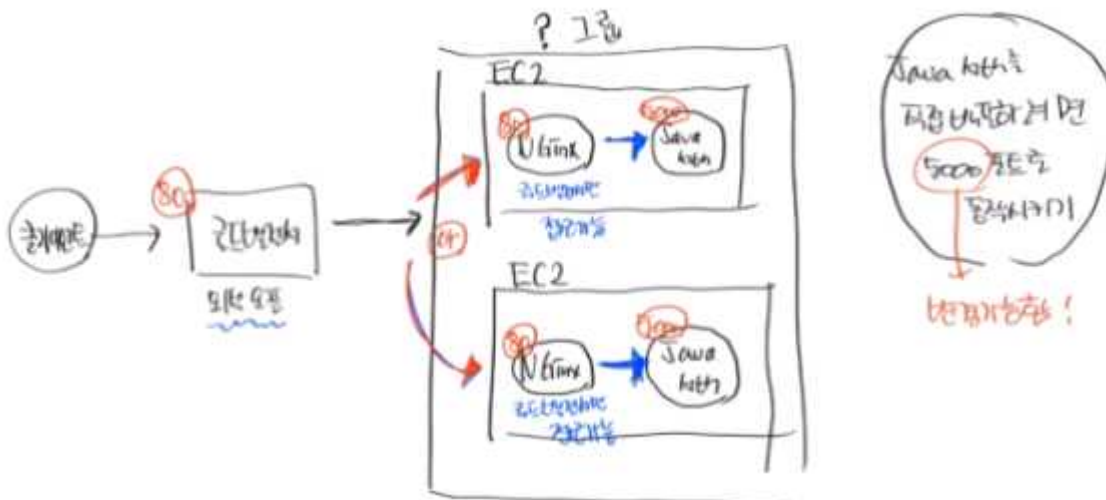


로드밸런서와 EC2(NginX-프록시)는 같은 보안그룹에 묶여 있다. Client 가 개별적으로 NginX로 요청해도 거부함.

- ▷ EC2 - 보안규칙 - 인바운드 규칙 - 80포트
- ▷ 인바운드 규칙 편집 화면
- 80포트는 거기 써 있는 보안그룹만 들어올 수 있다. 누구나로 바꾸면 거부된다.

◎ Client 요청 → 로드밸런서 통과 → NginX 통과 그 다음에 → 내부 자바 스프링 서버가 동작 함.

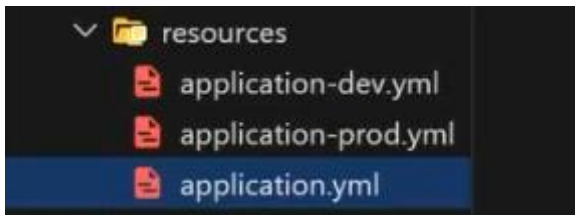
▶ NginX서버는 로드밸런서에서만 접근이 가능하다.



3장. 배포 v2 - 엘라스틱빈스톡 프로젝트 배포해보기

▷ 깃헙에 있는 aws-2

내려서 IntelliJ에서 열어보면 프로젝트가 열리고 Controller 가 하나있음.
그 중에 resources 밑에 application 관련 yml 파일이 세 개 있음.



현재는 prod 가 쓰이고 있음.

▶ application-dev.yml

```
server:  
  port: 8080
```

실제로 배포될 yml 파일 포트

▶ application-prod.yml

```
server:  
  port: 5000
```

테스트는 5000 번 포트 사용

▶ 인텔리제이 아래쪽에 Terminal 을 선택하고

\$./gradlew build

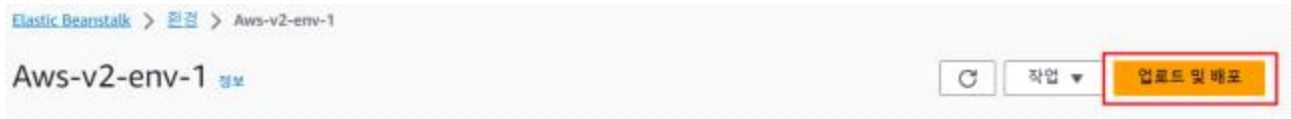
해당 프로젝트를 테스트하고 문제없으면 실행파일 만들어 줌
위치는

```
/build/libs/
```



● jar 파일 Elastic Beanstalk에 업로드 및 배포하기

▶ AWS Elastic Beanstalk 로 와서 좌측 상단 업로드 및 배포 단추 선택 -- 파일 선택 -- 위 jar 파일 선택 후 배포



좌측 메뉴 환경으로 이동 클릭



주소창 맨 뒤에 /aws/v1 하면 잘 불러와 짐

4장. 배포 v2 - 엘라스틱빈스톡 SSH 접속해 보기

- ① 원격으로 엘라스틱빈스톡이 만들어 놓은 EC2로 접속할거야 → 접속하려면 키페어 필요
- ② 엘라스틱빈스톡에 와서 좌측 구성 클릭 → 서비스 액세스 클릭 → 키페어 선택

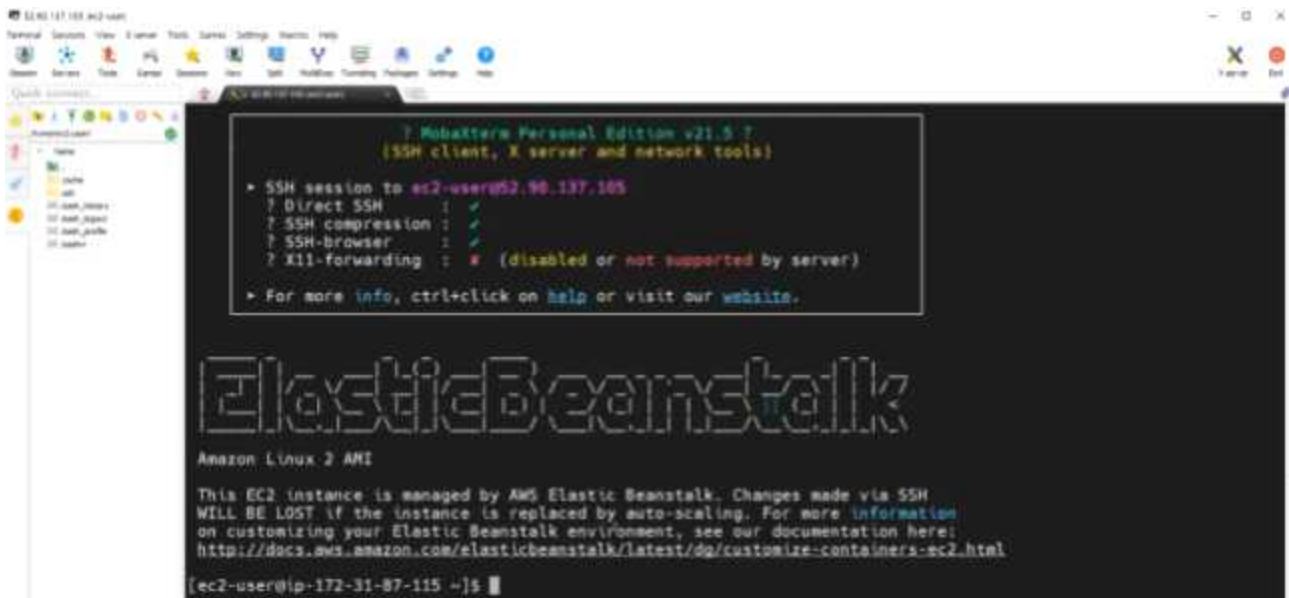
▷ 빈스톡 창 확인해서 완료 되는 것 확인하고...

▶ MobaXTerm 열어 접속

- ① 새로운 세션



- ② ElasticBeanStalk의 username → ec2-user 기억할 것. → 키페어 선택



▷ 어떤 포트 어떤 프로세스 실행되고 있는지 확인

> ps -ef

```
root      11790      1   0 May22 ?           00:00:06 /usr/sbin/rsyslogd -n
webapp    11987      1   0 May22 ?           00:02:02 java -jar application.jar
root      12038      1   0 May22 ?           00:00:00 nginx: master process /usr/sbin/nginx
```

jar를 던지면 파일명을 application.jar로 파일명을 치환시켜서 실행 함....

파일명을 어떤걸로 자동실행 시킬까... 일관성이 없으므로 그래서 그냥 application.jar로 파일명을 바꾸어 버림...

nginx 도 돌고 있음...

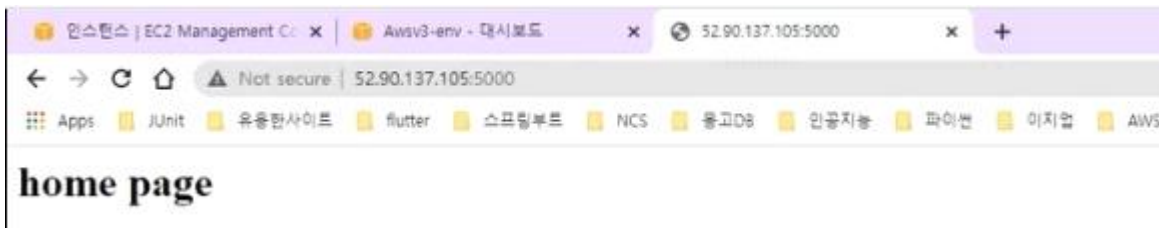
포트 확인해 볼까요?

> netstat -nlpt

```
[ec2-user@ip-172-31-87-115 ~]$ netstat -nlpt
(No info could be read for "-p": geteuid( )=1000 but you should be root.)
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 127.0.0.1:22221         0.0.0.0:*                LISTEN      -
tcp        0      0 0.0.0.0:111            0.0.0.0:*                LISTEN      -
tcp        0      0 0.0.0.0:80             0.0.0.0:*                LISTEN      -
tcp        0      0 0.0.0.0:22             0.0.0.0:*                LISTEN      -
tcp        0      0 127.0.0.1:25           0.0.0.0:*                LISTEN      -
tcp6       0      0 :::5000                :::*                    LISTEN      -
tcp6       0      0 :::111                 :::*                    LISTEN      -
```

80 포트는 EnginX Server

5000 포트는 Spring Server



포트 넣고 테스트 함.... 80 포트는 접근 못함...

> sudo find / -name nginx

nginx 가 들어있는 폴더가 다 보임...

EnginX 관련 설정파일 문서는 /etc/nginx 폴더 안에 있다. --- nginx.conf 파일임.

> cd /etc/enginx

> vi eginx.conf

```
server {
    listen      80 default_server;
    access_log  /var/log/nginx/access.log main;

    client_header_timeout 60;
    client_body_timeout   60;
    keepalive_timeout     60;
    gzip                  off;
    gzip_comp_level       4;
    gzip_types            text/plain text/css application/json application/javascript application/xml application/xml+rss text/javascript;

    # Include the Elastic Beanstalk generated locations
    include conf.d/elasticbeanstalk/*.conf;
}
```

포트 80 사용하고 있구...

더 밑에 include 에 conf.d/elasticbeanstalk/*.conf 를 포함하고 있으니 거기 가서 확인해 봐야 함..

esc :q

해당 폴더로 이동...

```
[ec2-user@ip-172-31-87-115 elasticbeanstalk]$ ls
00_application.conf  healthd.conf
[ec2-user@ip-172-31-87-115 elasticbeanstalk]$ ls -l
total 8
-rw-r--r-- 1 root root 397 May 22 11:42 00_application.conf
-rw-r--r-- 1 root root 215 May 22 11:42 healthd.conf
[ec2-user@ip-172-31-87-115 elasticbeanstalk]$
```

권한을 확인.... ec2-user 는 권한이 없으므로 수정 및 실행하려면 sudo 권한을 이용해야 한다든지.. 변경이 필요하나 수정은 안할거임...

> sudo vi 00_application.conf

여기 중요한 파일이 있음...


```
location / {
    proxy_pass      http://127.0.0.1:5000;
    proxy_http_version 1.1;

    proxy_set_header    Connection      $connection_upgrade;
    proxy_set_header    Upgrade         $http_upgrade;
    proxy_set_header    Host            $host;
    proxy_set_header    X-Real-IP      $remote_addr;
    proxy_set_header    X-Forwarded-For $proxy_add_x_forwarded_for;
}
```

루트를 받았을 때 자기 자신의 5000번 포트를 통해 열어 줌... 그래서 우리가 5000 번 포트를 이용해 배포한 거임.

5장. 배포 v2 - 엘라스틱빈스톡 종료하기

- ▶ RDS 연결 - 데이터베이스 연결
- ▶ 엘라스틱빈스톡 + RDS(데이터베이스)

- ① AWS Elasticbeanstalk에서 이전 환경 삭제합니다.....
모든 환경 → 우측 상단 → 환경 종료

위에 꺼 복사해서 아래에 붙여넣고... → 종료하기
시간이 좀 걸림.....

끝나면 좌측에 애플리케이션 선택 → 애플리케이션 선택 → 우 상단 애플리케이션 삭제

위와 같은 방법으로 위에 꺼 복사해서 아래에 붙여넣고 삭제 처리하면 됨.



깔끔하게 요 화면 보여야 되구요..

- ② 좌측 EC2 검색해 보면 EC2도 종료 되고 있음....
- ③ 보안그룹 이름에 default 빼고 모두 삭제 되어야 함..(default 는 무조건 하나 있어야 함.)

▷ EC2로 가서...

: 좌측 보안그룹 → default 만 빼고 정리(인스턴스도 종료가 되어 있어야 함.)

6장. 배포 v2 - 엘라스틱빈스톡 + RDS 생성하기

① 검색 → Elastic BeanStalk 이동 → 우측 Create Application

▷ 이름 : aws-v2-beanstalk

▷ 플랫폼 : java

▷ 버전 : 추천버전

▷ 샘플 애플리케이션

▷ Elastic Beanstalk → 구성 → 업데이트, 모니터링 및 로깅 → 편집 단추 클릭

환경속성 : 애플리케이션에 전달된다.(여기서 애플리케이션이란 우리가 만든 프로젝트를 뜻함) - 즉 프로젝트(웹서버)에서 인식하는 환경

RDS_DB_NAME	book	제거
RDS_PORT	3306	제거
RDS_USERNAME	zzzmini	제거
RDS_PASSWORD	1111	제거
RDS_HOSTNAME	IP	제거

- RDS_HOSTNAME 옆에는 원래 IP가 들어와야하는데 지금은 모르니까 IP로 주자
데이터베이스에 연결할 주소를 아직 모르니까... 일단 IP로 (나중에 Mysql DB ip 주소로 변경)

- RDS_DB_NAME aws-v2-rds

- RDS_PORT 3306

- RDS_USERNAME zzzmini

- RDS_PASSWORD 적당히 8자 이상

→ 적용

▷ 단일 인스턴스로 생성하면 로드밸런서가 자동으로 만들어 지지 않는다.

-- 로드밸런서 - 그냥 놔두고

-- 데이터베이스 - 편집 - 데이터베이스 추가 - 이걸 안쓰고 RDS라는 걸 씀

다하고 앱 생성 누름.....

굉장히 오래걸림... 10분 이상... 엘라스틱 빈스톡은 다 만들거임.

7장 - 배포 v2 - VPC(Virtual Personal Cloud) 구성 이해

좌측 - 환경으로 이동 누르면 잘 설치가 됐는지 확인 가능 함.

Congratulations 나오면 잘 된 거임..

현재 Elastic BeanStalk 주소 표시줄을 그대로 복사하고, 크롬 새 창 생성하고 그대로 주소 붙여넣기...

그리고 EC2 검색

Elastic BeanStalk를 만들면 실제로 EC2 가 만들어지는 거임.

위쪽 실행 중 인스턴스 클릭하고....

각종 정보들을 저장하자...

1. EC2

- 퍼블릭 IPv4 : 52.204.233.129 - 외부 접속 IP
- 프라이빗 IP : 172.31.16.120 - 내부 사설 IP 가짜

▷ 하단 VPC ID 누르면 새 창 열림..

- VPC ID 복사 - vpc-Oe3b어쩌구....

지금 창은 끄고 인스턴스 써 있는 하단에서 보안 탭 선택 - 보안그룹 링크 클릭

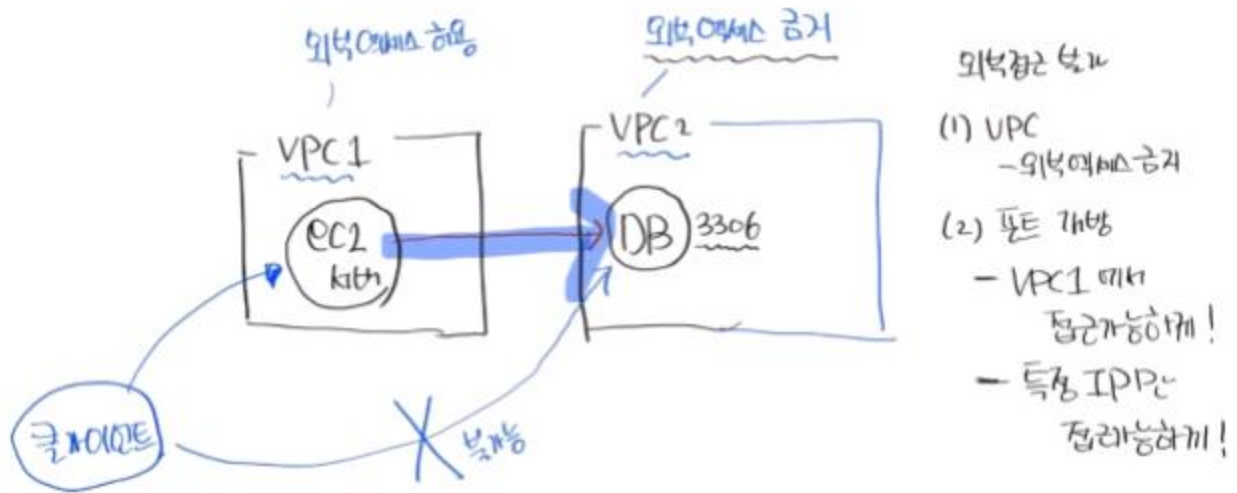
---- 22번 80번 포트 두개만 열려있으면 됨.

우 상단 세부정보

- 보안그룹 이름 : 복사 정리 -- awseb-e-taiw... 어쩌구
- 보안그룹 ID : sg-05766caa어쩌구...

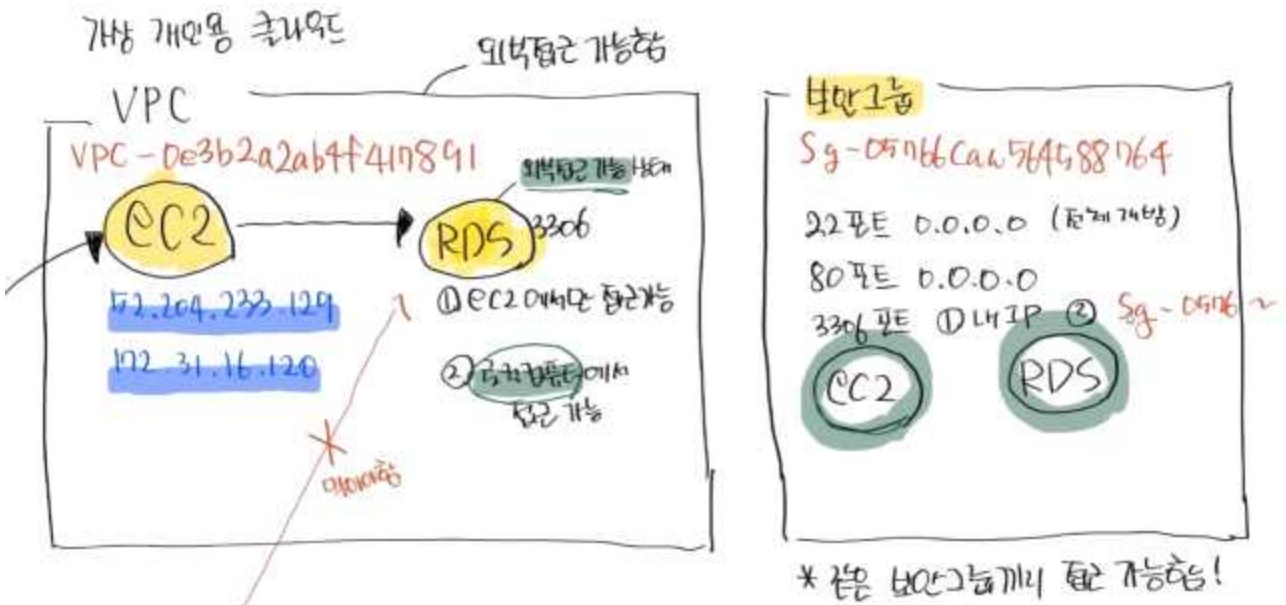
얘네들 메모장 "엘라스틱빈스톡 환경설정 파일.txt" 파일로 저장

일반적으로 VPC를 두개 생성해서 한개는 외부접근 가능(얘는 웹 접근용), 한개는 외부접근 불가능(보통 DB를 담아서 내부 통신으로만 접근 가능하도록...) 이렇게 개인용가상화클라우드를 만들어 사용하나,



여기선 VPC 1개만 생성해서 처리함... 근데 RDS는 어찌 되는가... EC2 에서만 접근 가능하도록 설정... 또 한가지 내 컴에서는 DB에 접속이 가능하도록 3306포트를 열어 놓을거야.. 테이블 생성하고.. 해야니까

음... 그럼 위쪽 두개 IP를 할당해서 접근이 가능하도록 할 수 있지만... 보안 그룹을 통해 설정하는 방법이 있다.



즉, RDS를 같은 보안그룹 내에 포함시킨다. 같은 보안그룹 내부에서는 서로 통신이 가능하니까 괜찮다. 그래서 결국은 EC2, RDS가 같은 보안그룹으로 묶이면 노출되어있는 IP로 외부에서 클라이언트가 접근이 가능해 지고, RDS는 접근이 불가능한 상태가 된다. 이때 RDS를 연결하는 3306포트는 EC2에서만 접근이 가능하게된다. 같은 보안그룹이니까... 근데 로컬에서도 접근이 가능해야 한다. DB 셋팅해야니까... 그러려면 내 컴 ip를 알려주는 법을 쓸거임.

8장 - 배포 v2 - RDS 생성 및 Windows 접속

● RDS 만들어 보자

① Elastic BeanStalk 창 주소를 복사해서 새 창 만들어서 생성 --- 창이 여러개 있는게 좋다.

→ 검색 창에 RDS 검색 - 진입

② 상단 데이터베이스 생성 단추 클릭 → 엔진 옵션 → MySQLDB 선택 → 버전 선택

③ 템플릿 : 프리티어

④ 설정

- DB 인스턴스식별자 : aws-v2-rds

- 마스터 사용자 이름 : zzzmini

- 비밀번호 : 적절히 주는데 환경속성에서 준거랑 일치...

엘라스틱 빈스톡에서 환경설정에 이미 정해 진거라.. 그것 설정해야 함.... 나중에 점검해서 정리

▷ 퍼블릭 액세스 정보 : 외부접근 가능하게 할거냐.. 예 로 합시다.

왜냐면 원격으로 내 컴에서 접근하게 해야니까.... 누구나 접근 못하게 막을거야

퍼블릭 액세스 정보

☒ 예

RDS는 데이터베이스에 퍼블릭 IP 주소를 할당합니다. VPC 외부의 Amazon EC2 인스턴스 및 다른 리소스가 데이터베이스에 연결할 수 있습니다. VPC 내부의 리소스도 데이터베이스에 연결할 수 있습니다. 데이터베이스에 연결할 수 있는 리소스를 지정하는 VPC 보안 그룹을 하나 이상 선택합니다.

☐ 아니요

RDS는 퍼블릭 IP 주소를 데이터베이스에 할당하지 않습니다. VPC 내부의 Amazon EC2 인스턴스 및 다른 리소스만 데이터베이스에 연결할 수 있습니다. 데이터베이스에 연결할 수 있는 리소스를 지정하는 VPC 보안 그룹을 하나 이상 선택합니다.

▷ 기존 VPC 보안그룹 -- 클릭해서 바꿈....default는 지우고 ...

- 아까 메모장에 넣어놨던 보안그룹 선택.

- 이러면 EC2와 VPC(RDS)가 같은 보안 그룹으로 묶이지요..... 둘다 22번 80번 만 열려있음....

그니까 3306을 열어야함..

VPC 보안 그룹(방화벽) 정보

데이터베이스에 대한 액세스를 허용할 VPC 보안 그룹을 하나 이상 선택합니다. 보안 그룹 규칙이 적절한 수신 트래픽을 허용하는지 확인합니다.

☒ 기존 항목 선택
기존 VPC 보안 그룹 선택

☐ 새로 생성
새 VPC 보안 그룹 생성

기존 VPC 보안 그룹

Choose one or more options ▼

awseb-e-kngdhnp-epa-stack-AWSEBSecurityGroup-VFM62Q5WZF95 X

가용 영역 정보

기본 설정 없음 ▼

요 그룹 밑에

▷ 추가구성 - 데이터베이스 포트 - 값을 3306 으로 아마.. 자동으로 보이는 듯

▼ 추가 구성

데이터베이스 포트 정보

데이터베이스가 애플리케이션 연결에 사용할 TCP/IP 포트입니다.

3306

데이터베이스 인증

월별 추정 요금

Amazon RDS 프리 티어는 12개월 동안 사용할 수 있습니다. 매월 프리 티어를 통해 아래 나열된 Amazon RDS 리소스를 무료로 사용할 수 있습니다.

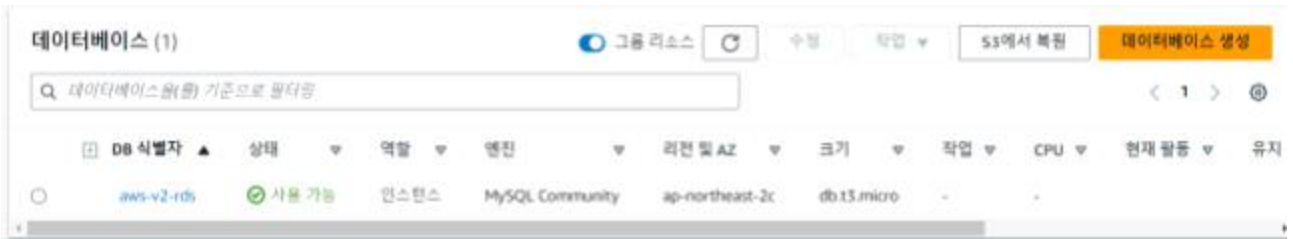
- 단일 AZ db.t2.micro, db.t3.micro 또는 db.t4g.micro 인스턴스에서 Amazon RDS를 750시간 사용.
- 20GB의 범용 스토리지(SSD).
- 20GB의 자동 백업 스토리지 및 사용자가 시작한 모든 DB 스냅샷.

[Learn more about AWS Free Tier.](#)

When your free usage expires or if your application use exceeds the free usage tiers, you simply pay standard, pay-as-you-go service rates as described in the [Amazon RDS Pricing page](#).

이대로 맨 밑 데이터베이스 생성 단추 누름.....

꽤 오래 걸림...



요래 생성 됨..

- ▷ DB 식별자 클릭.... - 아래 엔드포인트 포트 복사.....
- 엔드포인트 : aws-v4-rds.czz... 어쩌구
- 포트 : 3306

- ▷ 우측 보안그룹 클릭...

연결 & 보안

엔드포인트 및 포트

엔드포인트

aws-v2-rds.cjvsp6epglho.ap-northeast-2.rds.amazonaws.com

포트

네트워킹

가용 영역

ap-northeast-2c

VPC

vpc-030eaf76cc74f155d

보안

VPC 보안 그룹

awseb-e-aanzpmtg-stack-AWSEBSecurityGroup-C08CJ5YWDZ5 (sg-0d368604b516c6ce8)

✓ 활성화

현재 보안그룹은 EC와 RDS가 같이 쓰고 있음.

- ▷ 아래쪽 인바운드 규칙 - 우측 인바운드 규칙 편집
- 여기에 3306포트가 추가 되어야 한다. 그리고 내 IP 가 접근 가능하도록 설정 함.



그리고 3306포트에다가 같은 보안그룹을 적용해서 같은 보안 그룹 내에서는 접근 가능하도록... EC2와 RDS 두개가 모두 3306포트 번호로 접근 가능하다.

▶ 규칙 저장

▷ HeidiSQL 애플리케이션은 윈도우용 다운로드 받고 실행한 다음에 새로운 세션 만들자

Installer, 32/64 bit combined

- [illegible]

● HeidiSQL에서 데이터베이스 생성하기

▷ 화면 중간 ▶ 쿼리 클릭 - 내용 다 지우고

```
create database book
```

▷ 애가 커서가 끝에 있을 때는 실행이 안된다. → query 중간에 커서 놓고 마우스 우클릭 실행....

```
use book;
```

```
create table book (  
    id bigint auto_increment primary key,  
    title varchar(255),  
    content varchar(255),  
    author varchar(255)  
);
```

마우스 우클릭 실행

show variables like 'c%'; -- encoding이 UTF-8이 아니면 타입을 UTF-8로 바꿀거야

```
alter database metadb character set= 'utf8mb4' collate='utf8mb4_general_ci';  
-- UTF8 하고 이모지 사용할 수 있게하고.. 실행  
-- UTF8로 바뀜.
```

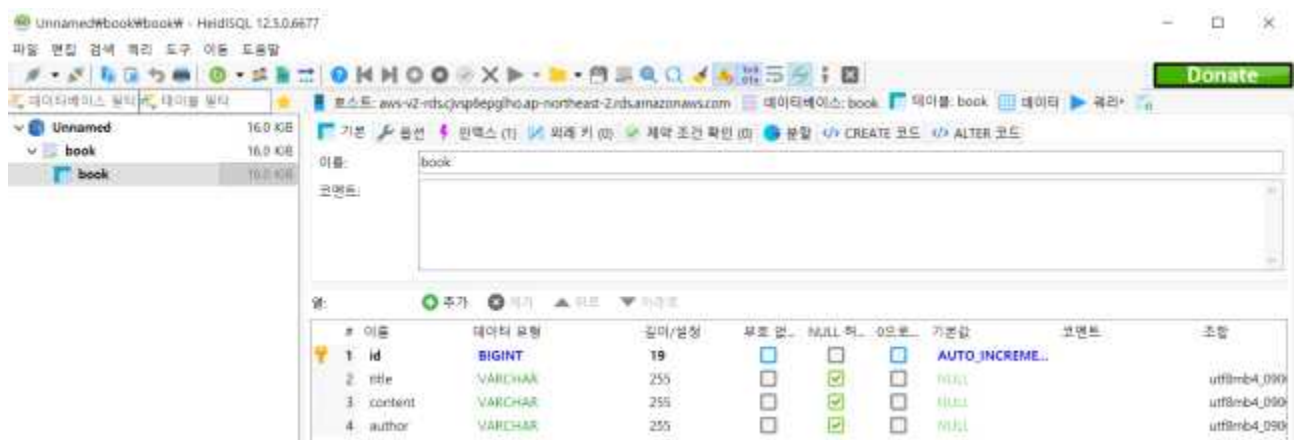
이렇게 하면 DB 셋팅 끝....

잘못 만들면.. 다시 만들어야지... 지우고...

검증....

```
select * from Book;
```

잘 나오면 끝이야.....



9장. 배포 v2 프로젝트 살펴보기

```
application-prod.yml
1  server:
2    port: 5000
3    servlet:
4      encoding:
5        charset: utf-8
6
7    spring:
8      datasource:
9        url: jdbc:mysql://${rds.hostname}:${rds.port}/${rds.db.name}
10       driver-class-name: com.mysql.cj.jdbc.Driver
11       username: ${rds.username}
12       password: ${rds.password}
13
14     jpa:
15       hibernate:
16         ddl-auto: none
```

▷ 환경속성 값을 생성해서 보안 시 필요한 값들을 변수로 저장해 놓는다. 환경변수에 RDS_HOSTNAME 은 RDS.HOSTNAME 와 같이 대문자, 소문자도 지원

ddl-auto: none

▶ Elastic BeanStalk 페이지로 이동... 좌측 구성 → 소프트웨어 → 편집

하단 환경 속성 값을 이용해 적어 준다.... 애가 스프링에 전달되서 최종 값으로 변경 됨.

RDS_HOSTNAME : RDS 엔드포인트 값을 넣어준다.

적용하면 외부로 값이 노출된 우려가 없다.....

요정도 셋팅하면 스프링프로젝트에서 할 일은 끝...

10장. 배포 - v2 배포 완료 및 테스트

jar 파일로 만들자

> ./gradlew clean build

해당 jar 파일 찾아서 적당한 폴더에 위치....

테스트없이 빌드하기

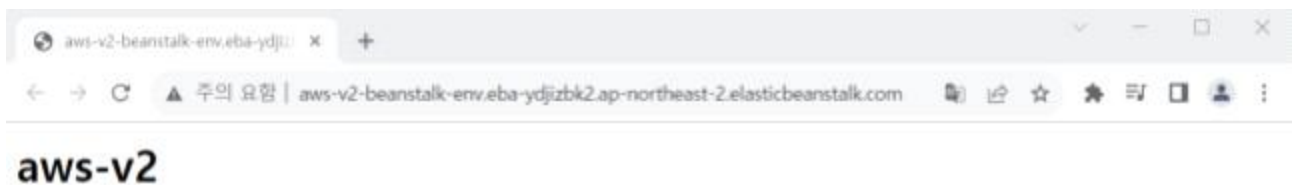
```
> ./gradlew clean build --exclude-task test
```

▶ 다시 Elastic BeanStalk 페이지로 가서

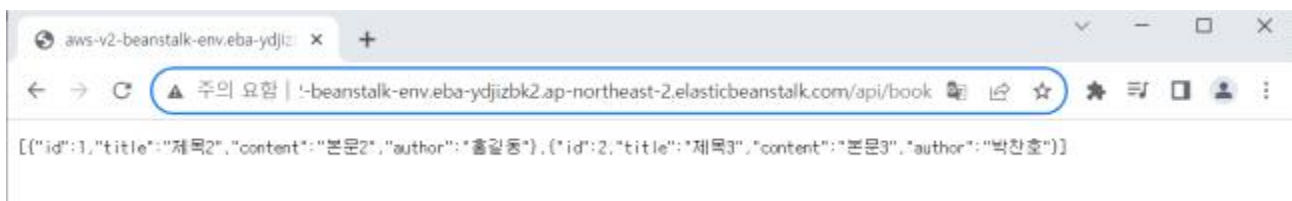
- ① 좌측 환경 클릭 나오는 화면에서 환경이름 클릭 → 업로드 및 배포 → 좀 전 파일 선택... → 배포 끝
- ② 시간이 좀 걸린다..... 땀글땀글 돌아가는 거 밑에 이벤트 보기 누르면... 진행 사항이 보인다.

잘 될 겁니다?

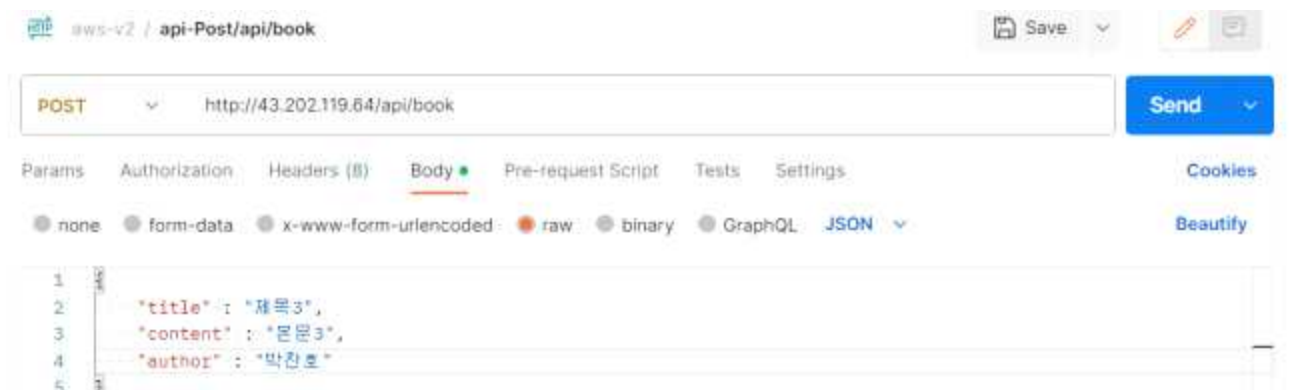
- ③ 좌측 환경으로 이동..... 웹 페이지 보이고



- ④ 주소줄 맨 끝에 /api/book 추가해야 controller 연결된 거 보이겠조?
자료 있음 보이고 없음 안보이니까...



- ⑤ postman으로 테스트 → URi는 EC2 퍼블릭 IP것쥬...



삭제하는 순서는....

1. Elastic BeanStalk 좌측 환경 - 우측 상단 - 환경 종료
2. Elastic BeanStalk 좌측 애플리케이션 - 우측 상단 - 애플리케이션 삭제
3. RDS - 좌측 데이터베이스 - RDS 선택 - 우측 상단 - 작업 - 삭제

다음에 할 것....

Elastic BeanStalk + RDS + GithubAction(자동배포도구) + EC2 서버 2개 + 로드밸런서 + 무중단배포