



AWS Elastic Beanstalk by HPHK

00. 배포 구조

01. 배포 사전 준비

1.1. [IAM 계정 생성](#)

1.2. [IAM 계정 AWS Management Console Login \(Optional\)](#)

02. 배포를 위한 설정

2.1. Github 세팅

2.2. [requirements.txt](#)

2.3. [.ebextensions](#)

2.4. Static 파일 설정

2.5. admin 설정

03. 배포

3.1. AWS CLI 설치 및 계정 등록

3.2. EB 설정

3.3. EB 환경 만들기

3.4. 환경변수 설정

3.5. ALLOWED_HOSTS 설정

3.6. EB로 배포

3.7. Social Login (Optional)

참고 - Client(vue)에서 Server(Django)로 요청

References

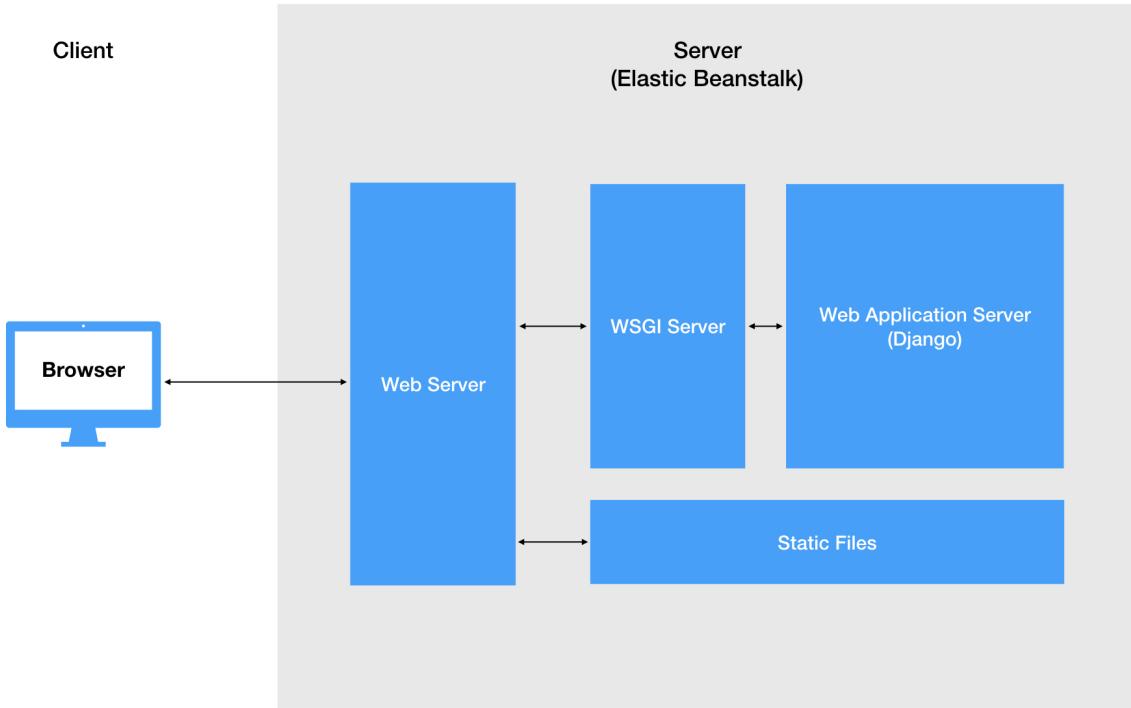


해당 문서는 setting.py 분리를 한 후에 진행한다고 가정한다.

배포 중 발생하는 문제는 반드시 log 를 직접 확인하여 수정한다.

문서 저작권은 주식회사 해피해킹에 있으며 무단 배포 및 수정을 금합니다.

00. 배포 구조



01. 배포 사전 준비

1.1. IAM 계정 생성

- 처음 계정을 생성 했을 때의 ID는 루트 사용자로 해당 계정을 사용해서 작업하는 건 추천되지 않으며 IAM 계정을 만들어서 특정 리소스(ex. EB)에 대한 권한을 부여하여 IAM 계정으로 작업 하는게 좋다.
- AWS에서는 루트 계정이 아닌 IAM 계정을 사용할 것을 강력하게 권장하고 있다.

AWS 계정 생성 후 로그인 뒤 IAM 검색 후 접속한다.

The screenshot shows the AWS Management Console homepage. At the top, there's a search bar with the placeholder "You can enter names, keywords or acronyms." Below it, a search input field contains the text "IAM". To the right of the search input is a close button (X). On the left side, there's a sidebar with sections for "AWS services" and "Build a solution". The "AWS services" section includes a "Find Services" input field with "IAM" selected, a "Build a virtual machine" section with EC2, and a "Build a web app" section with Elastic Beanstalk. On the right side, there are two main sections: "Access resources on the go" (with a mobile phone icon) and "Explore AWS" (with sections for "EC2 Spot Instances" and "CloudEndure Migration").

Users 클릭

The screenshot shows the AWS IAM Management Console. The left sidebar has a tree view under 'Identity and Access Management (IAM)'. Under 'AWS Account (398271762700)', 'Dashboard' is selected. Other options include Groups, Users, Roles, Policies, Identity providers, Account settings, and Credential report. A search bar for 'Search IAM' is also present. Under 'AWS Organizations', there are links for Organization activity and Service control policies (SCPs). The main content area displays a 'Welcome to Identity and Access Management' message with a link to a sign-in page. It shows 'IAM users sign-in link: https://398271762700.signin.aws.amazon.com/console'. Below this, 'IAM Resources' are listed: 'Users: 4', 'Groups: 0', 'Policies: 0', and 'Identity Providers: 0'. A section titled 'Customer Managed Policies: 1' is shown. On the right, a 'Feature Spotlight' video player is displayed with the title 'Introduction to AWS IAM'. Below it, 'Additional Information' links include 'IAM best practices', 'IAM documentation', 'Web Identity Federation Playground', 'Policy Simulator', and 'Videos, IAM release history and additional resources'. The bottom of the page includes a footer with the URL 'https://console.aws.amazon.com/iam/home?#users', copyright information '© 2008 - 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.', and links for 'Privacy Policy' and 'Terms of Use'.

Add user 클릭

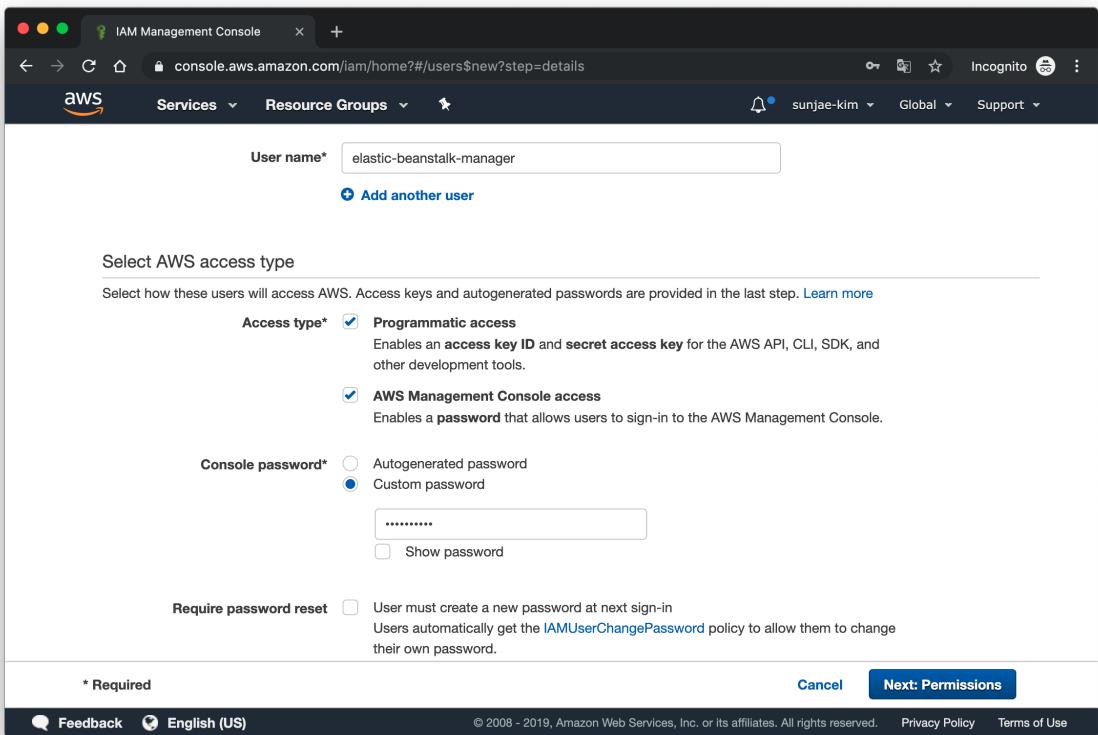
The screenshot shows the AWS IAM Management Console interface. On the left, there's a sidebar with navigation links for Identity and Access Management (IAM) and AWS Organizations. The main area displays a table of users. The columns in the table are: User name, Groups, Access key age, Password age, and Last activity. There are four users listed:

User name	Groups	Access key age	Password age	Last activity
deploy-ec2-1...	N/A	108 days	None	107 days
ec2-1-1-1-1...	N/A	8 days	8 days	8 days
sunjae-kim...	N/A	242 days	242 days	106 days
sunjae-kim...	N/A	243 days	243 days	90 days

- **1. Details**

- **Programmatic access** - CLI 등으로 접속하기 위함
- **AWS Management Console access** - AWS Management Console에서 접속하기 위함
- 둘다 체크하는 것을 추천.
- 비밀번호는 자동 생성 비밀번호보다는 **사용자 지정 비밀번호(Custom Password)**를 추천
- 설정 후에 다음: Permissions 클릭
- **User Name** 으로 설정한 값이 IAM 계정 Login Console에 로그인 할 때 사용자이름(username)이 된다.

Screenshot of the AWS IAM Management Console showing the creation of a new user named "elastic-beanstalk-manager". The "Access type" section is selected, showing both "Programmatic access" and "AWS Management Console access" checked. A custom password has been set, and the "Require password reset" checkbox is unchecked.



- **2. Set Permission**

- EB 모든 권한을 가지고 있는 관리자 IAM 계정을 생성한다.

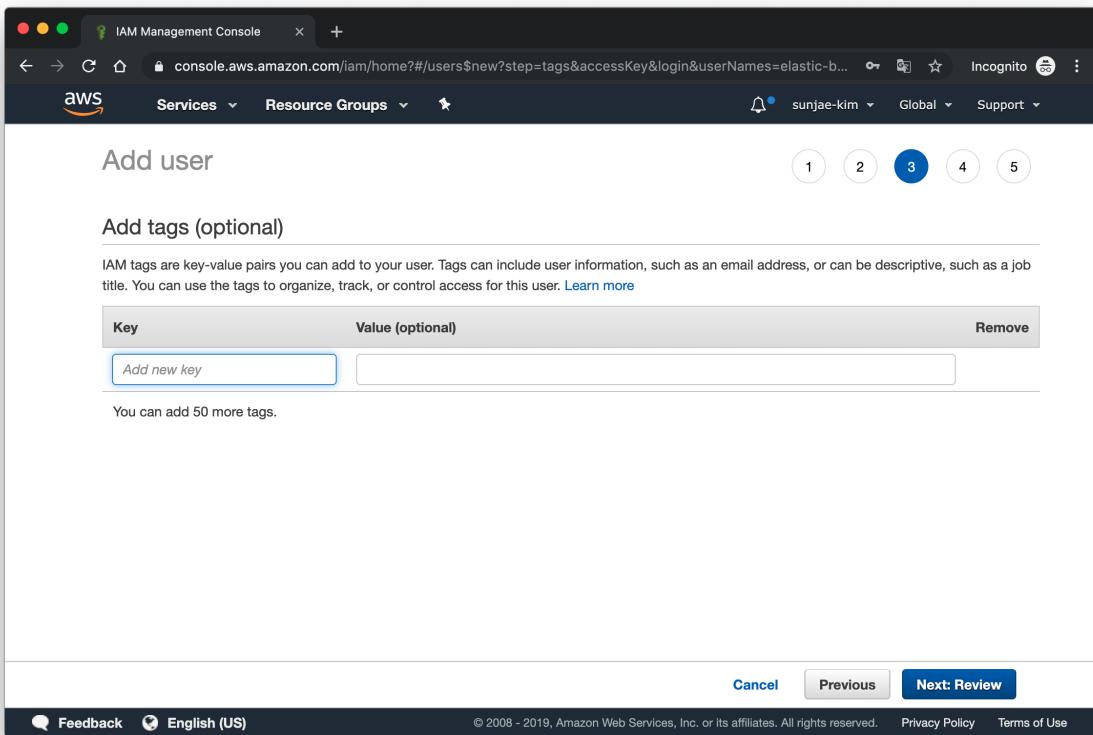


The screenshot shows the AWS IAM Management Console with the URL [console.aws.amazon.com/iam/home?#/users\\$new?step=permissions&accessKey&login&userNames=e...](https://console.aws.amazon.com/iam/home?#/users$new?step=permissions&accessKey&login&userNames=e...). The page is titled "Attach existing policies directly". There are three main buttons at the top: "Add user to group", "Copy permissions from existing user", and "Attach existing policies directly" (which is highlighted with a blue border). Below these are "Create policy" and a refresh icon. A search bar with the text "ElasticBeansTalk" and a "Showing 8 results" message. A table lists policies with columns: Policy name, Type, Used as, and Description. Three policies are listed:

	Policy name	Type	Used as	Description
<input type="checkbox"/>	AWSElasticBeansta...	AWS managed	None	Provide the instance in your custom platfor...
<input type="checkbox"/>	AWSElasticBeansta...	AWS managed	Permissions policy (1)	AWS Elastic Beanstalk Service policy for He...
<input checked="" type="checkbox"/>	AWSElasticBeansta...	AWS managed	Permissions policy (1)	Provides full access to AWS Elastic Beansta...

Below the table, a section titled "AWSElasticBeanstalkFullAccess" describes the policy: "Provides full access to AWS Elastic Beanstalk and underlying services that it requires such as S3 and EC2." It includes "Policy summary" and "JSON" buttons, and a "Filter" search bar. At the bottom, there are "Cancel", "Previous", and "Next: Tags" buttons, along with links for "Feedback", "English (US)", "Privacy Policy", and "Terms of Use".

- 다음: 태그를 클릭해서 넘어간다.
- **3. Tags (Optional)**
 - 태그를 넣으면 나중에 해당 IAM 계정을 Tag 로 찾을수 있다.
 - 다음: 검토를 클릭한다.



- **4. Review**
 - 확인 후 [Create user](#)

Add user

Review

Review your choices. After you create the user, you can view and download the autogenerated password and access key.

User details

User name	elastic-beanstalk-manager
AWS access type	Programmatic access and AWS Management Console access
Console password type	Custom
Require password reset	No
Permissions boundary	Permissions boundary is not set

Permissions summary

The following policies will be attached to the user shown above.

Type	Name
Managed policy	AWSElasticBeanstalkFullAccess

Cancel Previous Create user

- **5. Complete**

Download .csv 버튼 누른 뒤 안전한 곳에 저장한다.

- **Local** : `~/.aws/credentials/` ← 자리 이동시 반드시 삭제
- **Cloud** : Dropbox, Google Drive 등

The screenshot shows the AWS IAM Management Console with the URL [console.aws.amazon.com/iam/home?#users\\$new?step=final&accessKey&login&userNames=elastic-beanstalk-manager](https://console.aws.amazon.com/iam/home?#users$new?step=final&accessKey&login&userNames=elastic-beanstalk-manager). The page title is "Add user". A green "Success" message box states: "You successfully created the users shown below. You can view and download user security credentials. You can also email users instructions for signing in to the AWS Management Console. This is the last time these credentials will be available to download. However, you can create new credentials at any time." Below the message is a link: "Users with AWS Management Console access can sign-in at: <https://398271762700.signin.aws.amazon.com/console>". There is a "Download .csv" button. A table lists the created user "elastic-beanstalk-manager" with columns: User (checkbox), Access key ID, Secret access key, and Email login instructions. The Access key ID is listed as "***** Show" and the Secret access key is listed as "Send email". At the bottom, there are "Feedback", "English (US)", copyright information ("© 2008 - 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved."), and links for "Privacy Policy" and "Terms of Use". A "Close" button is also present.

Access Key ID 는 하얀 박스로 가렸습니다.

1.2. **IAM** 계정 AWS Management Console Login (Optional)

Login 링크 확인 (credential.csv 파일에서 확인하기)

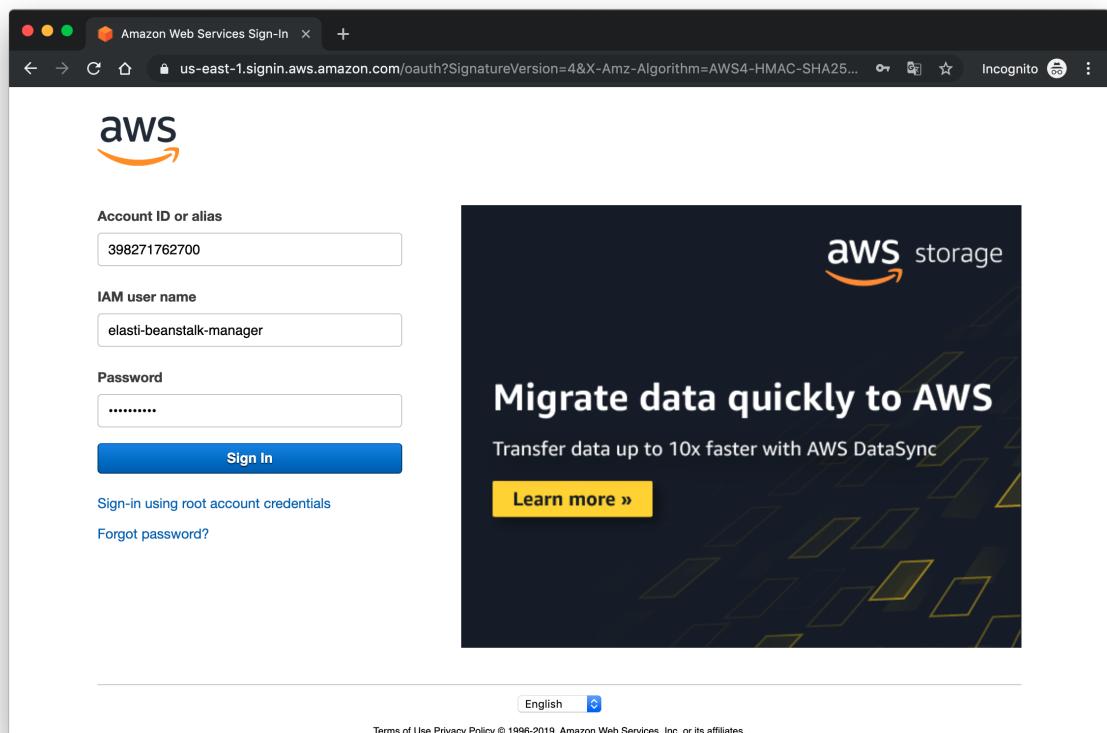
가장 오른쪽 탭의 **Console login link** 통해서 이동

The screenshot shows a Microsoft Excel spreadsheet window titled "credentials — Edited". The ribbon at the top includes "View", "Zoom", "Add Category", "Insert", "Table", "Chart", "Text", "Shape", "Media", "Comment", "Collaborate", "Format", and "Organize". The main area displays a single row of data in a table with the following columns:

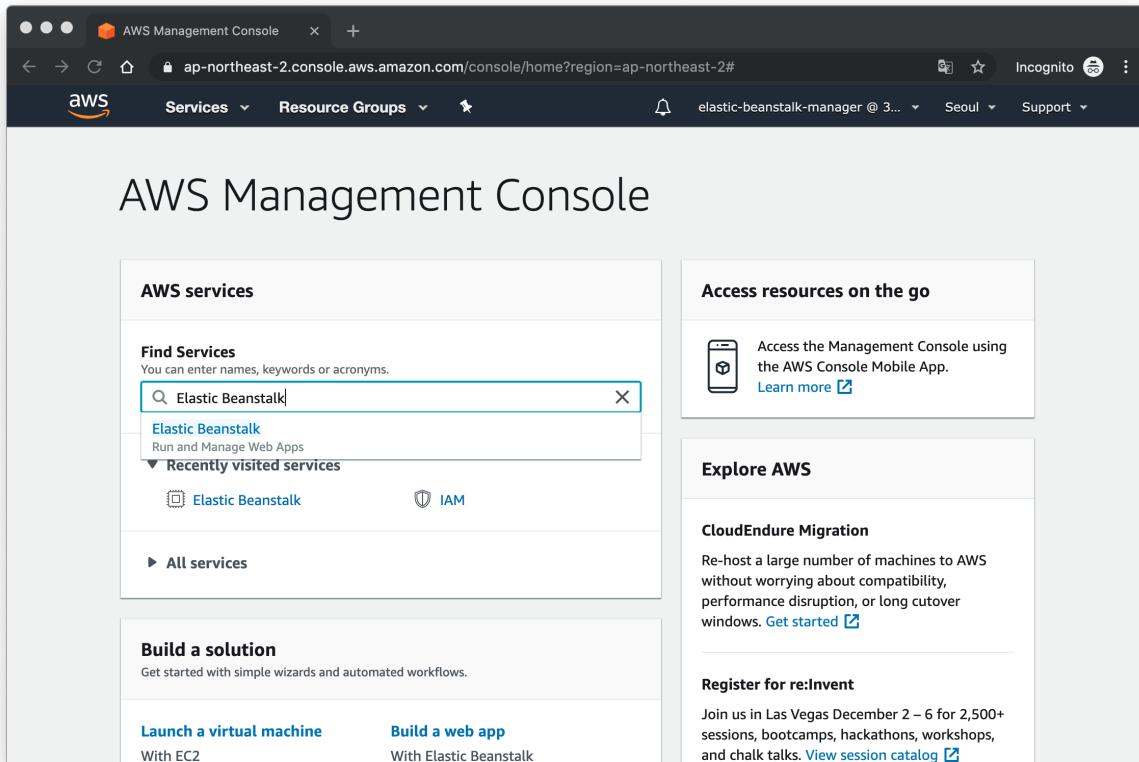
credentials				
User name	Password	Access key ID	Secret access key	Console login link
elastic-beanstalk-manager	-	-	-	https://398271762700.sigin.aws.amazon.com/console

Login Page

- IAM user name에는 방금 전 등록한 **user name**과 **password**를 입력한다. (.csv에 username은 나와 있고 비밀번호는 custom 비번 설정시 했던 번호로 로그인)
- 이후에 바로 비밀번호 변경 창으로 이동하는데 비밀번호를 변경 해주면 된다.



Elastic Beanstalk 검색 후 접속 확인



02. 배포를 위한 설정

- 반드시 에러가 나지 않는 Django Project 를 준비한다.
- .gitignore 는 python, django, venv 으로 설정한다.

2.1. Github 세팅

EB 도 git 으로 버전 관리가 이뤄지기 때문에 모든 세팅 이후 commit 이 있어야 배포 시 적용이 된다.

- 주의! EB는 master 브랜치에서 commit후에 deploy로 넘어가는게 아니라 바로 deploy 브랜치로 이동후에 commit 한다.

```
$ git checkout -b feature/deploy  
$ git add .  
$ git commit -m "..."  
$ git push origin feature/deploy
```

2.2. requirements.txt

- EB에서 해당 파일을 통해 사용된 library들을 확인하고 자동으로 설치한다.
- **requirements.txt 파일 생성**
 - 반드시! 해당 프로젝트를 위한 가상환경이 적용되었는지 확인하자
 - 가상 환경이 뭔가 잘못 되었다면 기존 가상 환경을 날리고 새로 만들어 실행하고 필요한 라이브러리를 재설치하자.

```
# 가상 환경에 문제가 있는 경우  
  
$ rm -rf venv  
$ python -m venv venv  
$ source venv/Scripts/activate
```

```
(venv)  
$ pip freeze > requirements.txt
```

- **requirements.txt**
 - 기존 Django 버전이 2.1.1 보다 높은 경우 삭제하고 다시 설치 후 **requirements.txt** 다시 생성
 - **Django 버전을 2.1.1로 바꾼다. EB는 2.1.1 버전까지만 문제 없이 호환한다. (2019. 09. 23 기준)**
 - [주의] 기존 django 2.2 나 다른 버전에서 만든 migrations들과 호환되지 않기 때문에 makemigrations/migrate 에러가 발생한다. migrations 파일들을 모두 삭제하고 makemigrations 를 새로 해야한다.

2.3. .ebextensions

- Django 프로젝트 세팅을 EB에게 알려주는 곳이다.
https://docs.aws.amazon.com/ko_kr/elasticbeanstalk/latest/dg/ebextensions.html



.ebextensions

- 웹 애플리케이션의 소스 코드에 AWS Elastic Beanstalk 구성 파일(.ebextensions)을 추가하여 환경을 구성하고 환경에 있는 AWS 리소스를 사용자 지정할 수 있습니다.
- 구성 파일은 .config 파일 확장명을 사용하는 YAML이나 JSON 형식 문서로, .ebextensions 폴더에 놓고 애플리케이션 소스 번들로 배포합니다.

Project 최상단에서 `.ebextensions` 폴더를 생성한다.

```
$ mkdir .ebextensions
```

`.ebextensions/django.config`

- wsgi 미들웨어 파일 path 설정
- 모든 `.config` 파일에서는 tab 을 사용해선 안된다.

```
$ touch .ebextensions/django.config
```

```
# django.config
```

```
option_settings:  
aws:elasticbeanstalk:container:python:  
WSGIPath: 프로젝트이름/wsgi.py
```

`.ebextensions/db-migrate.config`

- 프로젝트가 배포된 후 `migrate` 작업 실행 및, DB 권한 설정 및 seed data

```
$ touch .ebextensions/db-migrate.config
```

```
container_commands:  
01_migrate:  
  command: "python manage.py migrate"  
  leader_only: true  
02_chown_sqlitedb:  
  command: "sudo chown wsgi db.sqlite3"  
  leader_only: true  
03_seed: # Seeding 할 데이터가 있을 경우에만 추가  
  command: "python manage.py loaddata seed.json"  
  leader_only: true  
option_settings:  
aws:elasticbeanstalk:application:environment:  
DJANGO_SETTINGS_MODULE: 프로젝트이름.settings # settings 분리했을 경우 - 프로젝트이름.settings.production
```

- 모든 YAML 파일에 있는 # 이후의 주석은 제거해야 한다.
- Seeding 할 데이터가 있다면 `seed.json` 파일을 `<app>/fixtures/` 폴더에 넣는다.
 - 만약 미리 seeding 할 데이터가 없다면 `03_seed` 는 제외해도 된다.

`.ebextensions/pip-upgrade.config`

```
$ touch .ebextensions/pip-upgrade.config
```

- eb 환경의 pip 의 버전이 낮아 에러가 발생할 수 있으니 배포시 배포환경의 pip 버전을 업그레이드.

```
commands:  
  pip_upgrade:  
    command: "sudo /opt/python/run/venv/bin/pip install --upgrade pip"
```

2.4. Static 파일 설정

static root 설정

```
# settings.py or base.py  
  
STATIC_URL = '/static/'  
STATIC_ROOT = 'static' # 추가
```

▼ `STATIC_ROOT` 에 다른 경로를 두고 싶다면?

배포작업이 끝나고 aws console에서 세팅할 예정으로 03.Elastic Beanstalk 까지 완료하고 온다.

- aws console로 로그인 한 뒤 Elastic Beanstalk 접속
- 구성 ⇒ 소프트웨어 ⇒ 수정
- 정적파일 ⇒ "디렉토리 경로" 를 `STATIC_ROOT`에 설정한 경로로 변경

The screenshot shows the AWS Elastic Beanstalk configuration interface for an application named 'crud-test'. The 'Logs' section is active, displaying settings for logging to S3 and CloudWatch Logs. It includes fields for log rotation (7 days) and log retention (keep logs after deployment). The 'Environment Variables' section shows the variable DJANGO_SETTINGS_MODULE set to crud.settings. At the bottom right are 'Cancel', 'Next Step', and 'Apply' buttons.

.ebextensions/db-migrate.config

- 배포 되면서 `collectstatic` 명령어 실행
- seed data가 필요하다면 필수

```

container_commands:
  01_migrate:
    command: "python manage.py migrate"
    leader_only: true
  02_chown_sqlitedb:
    command: "sudo chown wsgi db.sqlite3"
    leader_only: true
  03_seed:
    command: "python manage.py loaddata seed.json"
    leader_only: true
  04_collectstatic: # 추가
    command: "python manage.py collectstatic"
    leader_only: true
option_settings:
  aws:elasticbeanstalk:application:environment:
    DJANGO_SETTINGS_MODULE: 프로젝트이름.settings # settings 분리했을 경우 - 프로젝트이름.settings.production

```

2.5. admin 설정

관리자 계정 생성

```
# 관리자 계정 생성

$ python manage.py createsuperuser
사용자 이름: admin
이메일 주소:
Password:
Password (again):
Superuser created successfully.
```

관리자 계정을 json 파일로 export

```
# Custom User Model 을 만들었다면
$ python manage.py dumpdata 어플리케이션(app).모델 --indent 4 > users.json # ex) accounts.User
# 만들어진 users.json 어플리케이션/fixtures 폴더로 이동
```

```
# Custom User Model 을 만들지 않았다면
$ python manage.py dumpdata auth.user --indent 4 > users.json
# 만들어진 users.json 아무 application 의 fixtures 폴더로 이동
```

- 생성 후에 만들어진 `users.json` 파일을 `어플리케이션/fixtures` 디렉토리로 이동 시킨다.

.ebextensions/db-migrate.config

```
container_commands:
  01_migrate:
    command: "python manage.py migrate"
    leader_only: true
  02_chown_sqlitedb:
    command: "sudo chown wsgi db.sqlite3"
    leader_only: true
  03_seed:
    command: "python manage.py loaddata seed.json"
    leader_only: true
  04_collectstatic:
    command: "python manage.py collectstatic"
    leader_only: true
  05_superuser:
    command: "python manage.py loaddata users.json"
    leader_only: true
option_settings:
  aws:elasticbeanstalk:application:environment:
    DJANGO_SETTINGS_MODULE: 프로젝트이름.settings.production
```



주의! `awsebcli` 설치 이후에 `pip freeze > requirements.txt` 는 절대 진행하면 안됨!

어쩔수 없이 추가 라이브러리 설치를 해야 한다면

가상환경 삭제 → 모두 재설치 → requirements.txt 만들기 → awsebcli 설치 → add commit deploy

03. 배포

3.1. AWS CLI 설치 및 계정 등록

EB 를 cli 에서 사용하기 위해 `awsebcli` 라이브러리를 설치한다.

```
# 배포 브랜치에서(feature/deploy)

$ git add .
$ git commit -m "..."
```

```
$ pip install awsebcli
$ eb --version
EB CLI 3.16.0 (Python 3.7.3)
```

3.2. EB 설정

```
$ eb init
```

1. Region

```
Select a default region
1) us-east-1 : US East (N. Virginia)
2) us-west-1 : US West (N. California)
3) us-west-2 : US West (Oregon)
4) eu-west-1 : EU (Ireland)
5) eu-central-1 : EU (Frankfurt)
6) ap-south-1 : Asia Pacific (Mumbai)
7) ap-southeast-1 : Asia Pacific (Singapore)
8) ap-southeast-2 : Asia Pacific (Sydney)
9) ap-northeast-1 : Asia Pacific (Tokyo)
10) ap-northeast-2 : Asia Pacific (Seoul)
11) sa-east-1 : South America (Sao Paulo)
12) cn-north-1 : China (Beijing)
13) cn-northwest-1 : China (Ningxia)
14) us-east-2 : US East (Ohio)
15) ca-central-1 : Canada (Central)
16) eu-west-2 : EU (London)
```

```
17) eu-west-3 : EU (Paris)
18) eu-north-1 : EU (Stockholm)
(default is 3): 10
```

2. AWS Credential Setting

다운로드 받았던 `credential.csv` 파일에 있는 정보를 넣는다.

```
You have not yet set up your credentials or your credentials are incorrect
You must provide your credentials.
(aws-access-id): [Access key ID]
(aws-secret-key): [Secret access key]
```

3. Application Name (enter)

```
(defualt is "..."): [애플리케이션 이름 | default값은 바로 enter]
```

4. 개발 환경

```
It appears you are using Python. Is this correct? y
```

5. python 버전

```
Select a platform version.
1) Python 3.6
2) Python 3.4
3) Python 3.4 (Preconfigured - Docker)
4) Python 2.7
5) Python
(default is 1): 1
```

6. codecommit 사용 여부

```
Do you wish to continue with CodeCommit? (y/N) (default is n): n
```

▼ 여기서 codec 오류 발생 시

- `eb init --debug` 해보면 되는데, 대부분 `.gitignore` 에 한글 있어서임. `.gitignore` 파일 다시 갈아치우는 것 추천
- 혹시나 그래도 안된다면, 방법은 `~/.pyenv/versions/[가상환경이름]/lib/python3.6/site-packages/ebcli/objects/sourcecontrol.py` 여기에서 실제로 소스코드 수정.
- [열캠기준] `C:\Users\student\Desktop\juno\django\myfrom\[가상환경이름]\Lib\site-packages\ebcli\objects\sourcecontrol.py`

```
# sourcecontrol.py

# 수정 전
with open('.gitignore', 'r') as f:
```

```
# 수정 후  
with open('.gitignore', 'r', encoding="utf-8") as f:
```

7. ssh 사용

```
Do you want to set up SSH for your instances?(Y/n): n
```

8. 완료

- 자동적으로 프로젝트 디렉토리내에 `.elasticbeanstalk/` 생성된다.
- `.gitignore` 에 아래의 정보 추가되어 있음. (직접 추가 NO)

```
# .gitignore  
  
# Elastic Beanstalk Files  
.elasticbeanstalk/*  
!.elasticbeanstalk/*.cfg.yml  
!.elasticbeanstalk/*.global.yml
```

- credentials 정보 잘 입력되었는지 확인

```
$ code ~/.aws/config
```

3.3. EB 환경 만들기

- 배포 환경의 이름은 자유롭게 생성하면 된다.

```
$ eb create [배포하는 환경의 이름] # ex) my-movie-pjt
```

- `eb create [env-name]` 으로 `create` 명령어 뒤에 **이름 옵션**과 함께 환경을 만드는게 이슈가 덜 발생한다.

EB 환경 생성

- 아래 화면처럼 나오며 따로 입력할 내용은 없다.
- 3~5 분 소요

```
Creating application version archive "app-43f7-190512_190916".  
Uploading instagram/app-43f7-190512_190916.zip to S3. This may take a while.  
Upload Complete.  
Environment details for: instagram-dev  
  Application name: instagram  
  Region: ap-northeast-2  
  Deployed Version: app-43f7-190512_190916
```

```

Environment ID: e-b2hcwx3a5q
Platform: arn:aws:elasticbeanstalk:ap-northeast-2::platform/Python 3.6 running on 64bit Amazon Linux/2.8.3
Tier: WebServer-Standard-1.0
CNAME: NONE
Updated: 2019-05-12 10:09:21.046000+00:00
...
2019-05-12 10:13:16    INFO    Application available at instagram-dev.ap-northeast-2.elasticbeanstalk.com.
2019-05-12 10:13:16    INFO    Successfully launched environment: instagram-dev

```

발생할 수 있는 여러 에러 이슈

▼ [에러] Your WSGIPath refers to a file that does not exist

- IAM 으로 aws console 로그인 후 방금 만든 환경설정에서 `WSGIPath` 수동 설정 해준다.
 - 구성 ⇒ 소프트웨어 ⇒ 수정 ⇒ "WSGIPath" 를 `프로젝트이름/wsgi.py` 로 바꾼다.
 - 경로가 올바르다면 환경 재구축을 하면 해결된다.

Modify software

Container Options

The following settings control container behavior and let you pass key-value pairs in as OS environment variables. [Learn more](#)

WSGIPath	<input type="text" value="crud/wsgi.py"/>
The file that contains the WSGI application.	
NumProcesses	<input type="text" value="1"/>
The number of daemon processes that should be started for the process group.	
NumThreads	<input type="text" value="15"/>
The number of threads to create to handle requests in each daemon process.	

▼ [에러] decouple SECRET_KEY 에러

- 아래 3.4 챕터로 해결
- 그 외 다른 에러가 발생한다면 상황에 따라 해결하고 `EB 앱 - 작업 - 환경 재구축` 또는 **add commit 후 deploy**

3.4. 환경변수 설정

- `AWS_ACCESS_KEY_ID`, `AWS_SECRET_ACCESS_KEY`, `SECRET_KEY`, `DEBUG` 값을 추가한다.
- CLI 에서 추가

```
$ eb setenv SECRET_KEY='...' DEBUG='False'
2019-05-12 10:13:58     INFO      Environment update is starting.
2019-05-12 10:14:04     INFO      Updating environment instagram-dev's configuration settings.
```

- 적용 3~5분 소요
- 혹은 aws console에서 [구성 -> 소프트웨어 -> 환경 속성](#)에 직접 입력 가능
 - 본인의 credentials 정보도 기입(대문자로)

환경 속성

다음 속성은 환경 변수로서 애플리케이션에서 전달됩니다. 자세히 알아보십시오

이름	값
AWS_ACCESS_KEY_ID	AKIATKEWMS4JV4RK7QAY
AWS_SECRET_ACCESS_KEY	Y9qlK7LBzzyvT9pnpX0z+4wYI
DJANGO_SETTINGS_MODULE	Last_Project.settings
SECRET_KEY	r+wrzc(=vhcmhr0jdwha7+e+

3.5. ALLOWED_HOSTS 설정

- CNAME 확인 후 [ALLOWED_HOSTS](#)에 넣기

```
$ eb status
```

```
Environment details for: instagram-dev
  Application name: instagram
  Region: ap-northeast-2
  Deployed Version: app-43f7-190512_190916
  Environment ID: e-b2hcwx3a5q
  Platform: arn:aws:elasticbeanstalk:ap-northeast-2::platform/Python 3.6 running on 64bit Amazon Linux/2.8.3
  Tier: WebServer-Standard-1.0
  CNAME: myform-dev.ap-northeast-2.elasticbeanstalk.com
  Updated: 2019-05-12 10:15:11.725000+00:00
  Status: Ready
  Health: Red
```

```
# settings.py or production.py

ALLOWED_HOSTS = [
    'my-movie-pjt.ap-northeast-2.elasticbeanstalk.com'
]
```

- 루트로 보여주고 싶은 페이지가 있다면 아래와 같이 설정하자.

```
# [PROJECT_NAME]/urls.py  
  
from app_name import views as app_name_views  
  
path('', app_name_views.index),
```

- **commit**

```
$ git add .  
$ git commit -m '...'
```

3.6. EB로 배포

- 반드시 배포 환경으로 가서 가상환경이 켜져있는지 확인하고 할 것!

```
$ eb deploy
```

```
$ eb open
```

- 수정 사항이 있는 경우 수정 사항 작성 후 아래 과정 수행

```
$ git add .  
$ git commit -m "..."  
$ eb deploy
```

-
- 배포 이후에 `feature/deploy` push

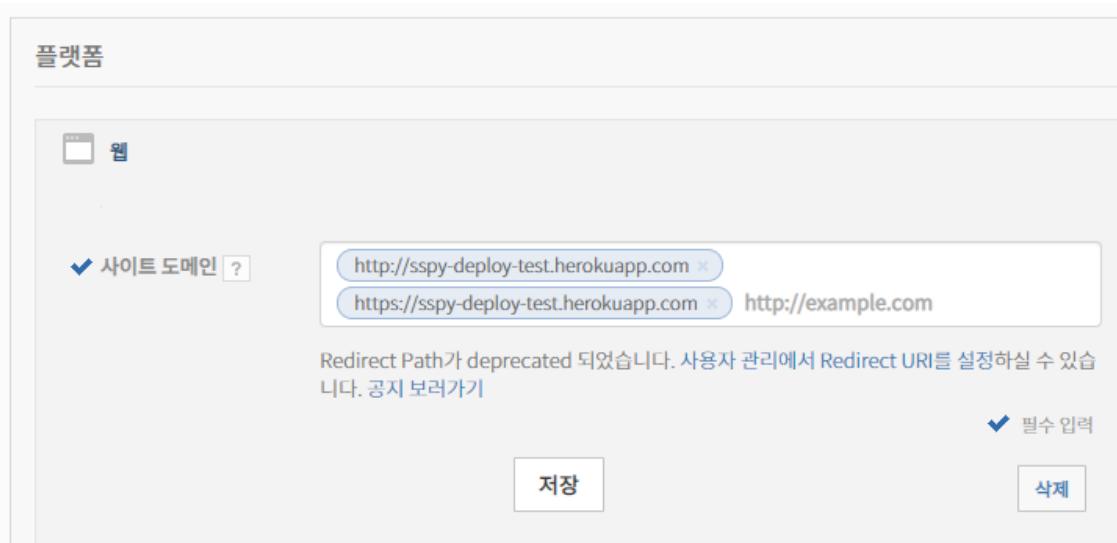
```
$ git push origin feature/deploy  
  
# upstream master에 pull request 보내서 merge  
  
$ git checkout master  
  
$ git pull upstream master  
  
# 개발 끝난 branch는 삭제
```

3.7. Social Login (Optional)

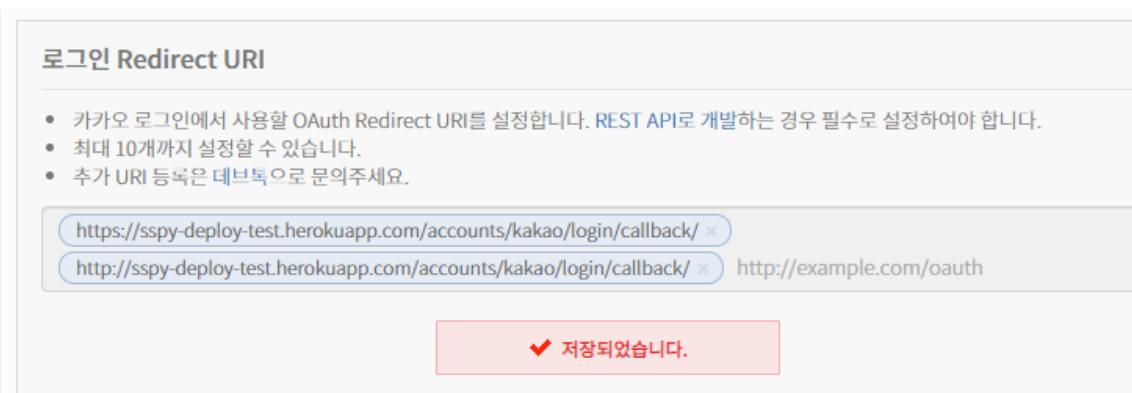
- 배포 이후에 소셜 로그인을 하게 되면 DoesNotExist 에러가 발생한다.
- admin 페이지에서 소셜 어플리케이션 등록을 새로 해주고 카카오 개발자 센터에도 배포 주소를 등록해주어야 한다.
- 카카오 뿐만 아니라 다른 social login api 모두 마찬가지이다.

kakao 개발자 센터

- [카카오 개발자 센터](#) - 설정 - 일반 - 플랫폼 - 웹
 - 사이트 도메인에 자신의 heroku 주소 넣고 저장하기([http](#) , [https](#) 모두 넣어야 함)



- [사용자 관리](#) → [로그인 Redirect URI](#)
 - [/accounts/kakao/login/callback/](#) 추가



admin 페이지

- 소셜 계정 → 소셜 어플리케이션
 - 클라이언트 아이디: 일반 → REST API 키
 - 비밀 키: 일반 → 고급 → 코드

소셜 어플리케이션 변경

히스토리

제공자: Kakao ▾

이름: justin

클라이언트 아이디:
[REDACTED]
앱 아이디 또는 컨슈머 아이디

비밀 키:
[REDACTED]
API 비밀 키, 클라이언트 비밀 키, 또는 컨슈머 비밀 키

키:
[REDACTED]

Sites:

이용 가능한 sites	선택된 sites
<input type="text"/> 필터	example.com
<input type="button"/> 모두 선택	<input type="button"/> 모두 제거
하나 이상을 선택하려면 "Control" 키, Mac은 "Command" 키를 누르세요.	

삭제 저장 및 다른 이름으로 추가 저장 및 편집 계속 저장

참고 - Client(vue)에서 Server(Django)로 요청

- vue에서 axios를 활용해서 AWS EB로 배포한 Django서버로 요청을 날릴 때 주의 할 사항
- 서버 주소도 환경 변수로 환경하기 위해 아래와 같이 `.env.local`에 주소를 적어 줄 때 반드시 `http://`처럼 프로토콜을 명시해줘야 한다.
- 그렇지 않을 경우 `로컬 주소/AWS 주소/요청 uri` 형태로 표현이 된다.

VUE_APP_SERVER_IP=<http://todos-test.twiwzmpmik.ap-northeast-2.elasticbeanstalk.com>

References

Django 애플리케이션을 Elastic Beanstalk에 배포

https://docs.aws.amazon.com/ko_kr/elasticbeanstalk/latest/dg/create-deploy-python-django.html