Deployment project : Final

Monday, September 28, 2020 6:01 PM

Priya : Note to the reviewer:

Prepared the below document wherein have listed the entire flow with the output and screenshots while working on the project.

Changes are covered in the commit message.
Took help from the mentors when encountered roadblocks.
Links referred :
https://github.com/jungleBadger/FSND-Deploy-Flask-App-to-Kubernetes-Using-EKS/blob/master/troubleshooting/deploy.md#step-7---fill-the-ci-cd-codepipelinecfnyml-file
Since was getting SSL failure referred to the link : https://github.com/aws/aws-cli/issues/1545 and used --no-verify-ssl option along with the aws eks kubectl upgrade command.

Have deleted everything and tried creating from scratch multiple number of times, changes can be verified with the git commit message.

IP to be used : **10.100.163.178**

```
pv252n:FSND-Deploy-Flask-App-to-Kubernetes-Using-EKS :~$ kubectl get services simple-jwt-api -o wide
NAME           TYPE           CLUSTER-IP      EXTERNAL-IP                                                              PORT(S)        AGE
SELECTOR
simple-jwt-api   LoadBalancer   10.100.163.178   adf56b6e69ade45208e2f8e0906236d4-54416909.us-west-2.elb.amazonaws.com   80:31002/TCP   17m   app=simple-jwt-api
pv252n:FSND-Deploy-Flask-App-to-Kubernetes-Using-EKS :~$

pv252n:FSND-Deploy-Flask-App-to-Kubernetes-Using-EKS :~$ export TOKEN=`curl -d '{"email":"pv252n@att.com","password":"pass"}' -H "Content-Type: application/json" -X POST $URL/auth  | jq -r '.token'`
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100   218  100   174  100    44    194     49 --:--:-- --:--:-- --:--:--   194
pv252n:FSND-Deploy-Flask-App-to-Kubernetes-Using-EKS :~$ curl --request GET $URL:80/contents -H "Authorization: Bearer ${TOKEN}" | jq
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100    61  100    61    0     0     69      0 --:--:-- --:--:-- --:--:--    70
{
  "email": "pv252n@att.com",
  "exp": 1614956196,
  "nbf": 1613746596
}
pv252n:FSND-Deploy-Flask-App-to-Kubernetes-Using-EKS :~$

pv252n:FSND-Deploy-Flask-App-to-Kubernetes-Using-EKS :~$ echo $TOKEN
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJleHAiOjE2MTQ5NTYxOTYsIm5iZiI6MTYxMzc0NjU5NiwiZW1haWwiOiJwdjI1Mm5AYXR0LmNvbSJ9.rLWKV1oyFIQE41qQ8U497XP0aLYkM413kvp4l6VDHFA
pv252n:FSND-Deploy-Flask-App-to-Kubernetes-Using-EKS :~$
```

Introduction to Flask App

1. Initial setup - Done
   a. Fork FSND-Deploy-Flask-App-to-Kubernetes-Using-EKS github repo to your Github account. - Done : https://github.com/priyavprashanth/FSND-Deploy-Flask-App-to-Kubernetes-Using-EKS.git
   b. Locally clone your forked version to begin working on the project - Done

```
pv252n@priya01:~/fswd/FSND/projects/04_aws_kubernetes_docker$ ls
FSND-Deploy-Flask-App-to-Kubernetes-Using-EKS  iam-role-policy.json  trust.json
pv252n@priya01:~/fswd/FSND/projects/04_aws_kubernetes_docker$ pwd
/home/pv252n/fswd/FSND/projects/04_aws_kubernetes_docker
pv252n@priya01:~/fswd/FSND/projects/04_aws_kubernetes_docker$
```

2. Dependencies - DONE
   a. Docker Engine - Installation instructions for all OSes can be found here - Done

```
pv252n@priya01:~/fswd/FSND/projects/04_aws_kubernetes_docker$ sudo docker version
[sudo] password for pv252n:
Client:
 Version:           18.09.7
 API version:       1.39
 Go version:        go1.10.4
 Git commit:        2d0083d
 Built:             Fri Aug 16 14:19:38 2019
 OS/Arch:           linux/amd64
 Experimental:      false

Server:
 Engine:
  Version:          18.09.7
  API version:      1.39 (minimum version 1.12)
  Go version:       go1.10.4
  Git commit:       2d0083d
  Built:            Thu Aug 15 15:12:41 2019
  OS/Arch:          linux/amd64
  Experimental:     false
```

pv252n@priya01:~/fswd/FSND/projects/04_aws_kubernetes_docker$

b. AWS Account - You can create an AWS account by signing up here - Done

c. Python 3.7 or higher - DONE
Note - Priya - Had python 3.5 running
Ran into many issues, since am using 3.5 for remaining projects, didn't want to just upgrade to 3.7
Created a new virtual environment with python3.7
Referenced links :
   i. https://www.liquidweb.com/kb/creating-virtual-environment-ubuntu-16-04/ - for creating a new virutal environment
   ii. https://stackoverflow.com/questions/1534210/use-different-python-version-with-virtualenv#:~:text=By%20default%2C%20that%20will%20be,%2Flocal%2Fbin%2Fpython3.
(fenv) pv252n:FSND-Deploy-Flask-App-to-Kubernetes-Using-EKS :~$ python --version
Python 3.7.9
(fenv) pv252n:FSND-Deploy-Flask-App-to-Kubernetes-Using-EKS :~$ deactivate
pv252n@priya01:~/fswd/FSND/projects/04_aws_kubernetes_docker/FSND-Deploy-Flask-App-to-Kubernetes-Using-EKS$ python --version
Python 3.5.2
pv252n@priya01:~/fswd/FSND/projects/04_aws_kubernetes_docker/FSND-Deploy-Flask-App-to-Kubernetes-Using-EKS$

3. Steps to run the API Locally using the Flask Server (No Containerization) - DONE
The following steps describe how to run the Flask API locally with the standard Flask server, so that you can test endpoints before you containerize the app:

a. DONE : Install python dependencies - These dependencies are kept in a requirements.txt file. To install them, go to the project directory that you've just downloaded, and use pip as follows:
`pip install -r requirements.txt`
b. DONE : Set up the environment - You do not need to create an .env_file to run locally but you do need the following two variables available in your terminal environment:
   - **JWT_SECRET** - The secret used to make the JWT, for the purpose of this course the secret can be any string.
   - **LOG_LEVEL** - It represents the level of logging. It is optional to be set. It has a default value as 'INFO', but when debugging an app locally, you may want to set it to 'DEBUG'. To add these to your terminal environment, run the following:

```
export JWT_SECRET='myjwtsecret'
export LOG_LEVEL=DEBUG
```

(penv) pv252n:FSND-Deploy-Flask-App-to-Kubernetes-Using-EKS :~$ export JWT_SECRET='myjwtsecret'
(penv) pv252n:FSND-Deploy-Flask-App-to-Kubernetes-Using-EKS :~$ export LOG_LEVEL=DEBUG
(penv) pv252n:FSND-Deploy-Flask-App-to-Kubernetes-Using-EKS :~$

c. Run the app using the Flask server, from the top directory, run:
`python main.py`

Open a new browser to run http://127.0.0.1:8080/ and it will give you a response as "Healthy".

4. Try the API endpoints on Command-Line - DONE
a. Open a new shell and install jq, which is a package that helps to read or manipulate JSON processors. For Linux,
`sudo apt-get install jq`

b. To try the /auth endpoint, use the following command, replacing <EMAIL> and <PASSWORD> with any values::
`export TOKEN=`curl -d '{"email":"pv252n@att.com","password":"pass1234"}' -H "Content-Type: application/json" -X POST localhost:8080/auth | jq -r '.token'``

This calls the endpoint 'localhost:8080/auth' with the {"email":"<EMAIL>","password":"<PASSWORD>"} as the message body. The return value is a JWT token based on the secret string you supplied. We are assigning that secret to the environment variable 'TOKEN'. To see the JWT token, run:
`echo $TOKEN`

pv252n:FSND-Deploy-Flask-App-to-Kubernetes-Using-EKS :~$ export TOKEN=`curl -d '{"email":"pv252n@att.com","password":"pass1234"}' -H "Content-Type: application/json" -X POST localhost:8080/auth | jq -r '.token'`

```
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100   227  100   179  100    48  11966   3208 --:--:-- --:--:-- --:--:-- 12785
```

pv252n:FSND-Deploy-Flask-App-to-Kubernetes-Using-EKS :~$ echo $TOKEN
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJleHAiOjE2MDIzNjA2MjAsIm5iZiI6MTYwMTE1MTAyMCwiZW1haWwiOiJwdjI1Mm5AYXR0LmNvbSJ9.cdhn63zXRCVbuI3iPHXUjGxqJJ0cHiR4j9ar265vD-0
pv252n:FSND-Deploy-Flask-App-to-Kubernetes-Using-EKS :~$

Priya Note :
export TOKEN=`curl -d '{"email":"pv252n@att.com","password":"pass1234"}' -H "Content-Type: application/json" -X POST localhost:8080/auth | jq -r '.token'`
This was giving error saying Failed writing body.
Logs gave the error :
File "/home/pv252n/fswd/FSND/projects/04_aws_kubernetes_docker/FSND-Deploy-Flask-App-to-Kubernetes-Using-EKS/main.py", line 82, in auth
  return jsonify(token=_get_jwt(user_data).decode('utf-8'))

AttributeError: 'str' object has no attribute 'decode'

Based on the learnings from the below link :removed .decode('utf-8')
https://stackoverflow.com/questions/28583565/str-object-has-no-attribute-decode-python-3-error

```
pv252n:FSND-Deploy-Flask-App-to-Kubernetes-Using-EKS :~$ export TOKEN=`curl -d
'{"email":"pv252n@att.com","password":"pass1234"}' -H "Content-Type: application/json" -
X POST 0.0.0.0:8080/auth  | jq -r '.token'`
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100   227  100   179  100    48   9486   2543 --:--:-- --:--:-- --:--:-- 10529
pv252n:FSND-Deploy-Flask-App-to-Kubernetes-Using-EKS :~$ echo $TOKEN
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJleHAiOjE2MTQ4OTAzNDQsIm5iZiI6MTYxMzY4M
Dc0NCwiZW1haWwiOiJwdjI1Mm5AYXR0LmNvbSJ9.qt7otJsNYKd0agZr3n50lOGyH1q905M9
W_p2b5pUhEI
pv252n:FSND-Deploy-Flask-App-to-Kubernetes-Using-EKS :~$
```

c. To try the `/contents` endpoint which decrypts the token and returns its content, run:
`curl --request GET 'http://127.0.0.1:8080/contents' -H "Authorization: Bearer ${TOKEN}" | jq .`

You should see the email id that you passed in as one of the values.

```
pv252n:FSND-Deploy-Flask-App-to-Kubernetes-Using-EKS :~$ curl --request GET
'http://127.0.0.1:8080/contents' -H "Authorization: Bearer ${TOKEN}" | jq .
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100    76  100    76    0     0   3127      0 --:--:-- --:--:-- --:--:--  3454
{
  "email": "pv252n@att.com",
  "exp": 1614890344,
  "nbf": 1613680744
}
pv252n:FSND-Deploy-Flask-App-to-Kubernetes-Using-EKS :~$
```

Containerize the Flask App and Run Locally

Steps to follow :

1. Install docker - DONE
2. Create a Dockerfile named `Dockerfile` in the app repo. - DONE
   Note : Priya - Referred the link https://www.geeksforgeeks.org/dockerize-your-flask-app/ and https://docs.docker.com/develop/develop-images/dockerfile_best-practices/ when creating the docker file

   Your Dockerfile should:
   - Use the `python:stretch` image as a source image
   - Set up an app directory for your code
   - Install `pip` and needed Python packages from `requirements.txt`
   - Define an entrypoint which will run the main app using the Gunicorn WSGI server. The Gunicorn should run with the arguments as follows: `["gunicorn", "-b", ":8080", "main:APP"]`.

   ```
   pv252n:FSND-Deploy-Flask-App-to-Kubernetes-Using-EKS :~$ cat Dockerfile
   FROM python:stretch
   COPY . /app
   WORKDIR /app
   RUN pip install -r requirements.txt
   EXPOSE 8080
   ENTRYPOINT  ["gunicorn", "-b", ":8080", "main:APP"]

   pv252n:FSND-Deploy-Flask-App-to-Kubernetes-Using-EKS :~$
   ```

3. Create a file named `.env_file` and save both `JWT_SECRET` and `LOG_LEVEL` into `.env_file`. These environment variables will run locally in your container. Here, we do **not** need the `export` command, just an equals sign:
   ```
   JWT_SECRET=myjwtsecret
   LOG_LEVEL=DEBUG
   ```

   This `.env_file` is only for the purposes of running the container locally, you do not want to check it into github or other public repositories. You can prevent this by adding it to your `.gitignore` file, which will cause git to ignore it. To safely store and use secrets in the cloud, use a secure solution such as AWS's parameter store.

   ```
   pv252n:FSND-Deploy-Flask-App-to-Kubernetes-Using-EKS :~$ cat .env_file
   SECRET=myjwtsecret
   LOG_LEVEL=DEBUG
   pv252n:FSND-Deploy-Flask-App-to-Kubernetes-Using-EKS :~$


   pv252n:FSND-Deploy-Flask-App-to-Kubernetes-Using-EKS :~$ grep env .gitignore
   .env
   .flaskenv
   env/
   env*
   .env_file
   # Direnv
   .envrc
   .direnv
   ```

#Priya : Added the .env_file as per the instructions in the project
.env_file
pv252n:FSND-Deploy-Flask-App-to-Kubernetes-Using-EKS :~$

4. Build a local Docker image with the tag `jwt-api-test`

docker build -t "jwt-api-test" .

```
pv252n:FSND-Deploy-Flask-App-to-Kubernetes-Using-EKS :~$ sudo docker build -t "jwt-api-test" .
Sending build context to Docker daemon  223.7kB
Step 1/6 : FROM python:stretch
 ---> b9d77e48a75c
Step 2/6 : COPY . /app
 ---> 5892ae94f212
Step 3/6 : WORKDIR /app
 ---> Running in 0be8d3545355
Removing intermediate container 0be8d3545355
 ---> 06387e7eb8ec
Step 4/6 : RUN pip install -r requirements.txt
 ---> Running in 2cc0766e7c96
.
.
.
Successfully installed Jinja2-2.11.2 MarkupSafe-1.1.1 Werkzeug-1.0.1 attrs-20.2.0 click-7.1.2
flask-1.1.2 gunicorn-20.0.4 importlib-metadata-2.0.0 iniconfig-1.0.1 itsdangerous-1.1.0 more-
itertools-8.5.0 packaging-20.4 pluggy-0.13.1 py-1.9.0 pyjwt-1.7.1 pyparsing-2.4.7 pytest-6.0.2
six-1.15.0 toml-0.10.1 zipp-3.2.0
WARNING: You are using pip version 19.2.3, however version 20.2.3 is available.
You should consider upgrading via the 'pip install --upgrade pip' command.
Removing intermediate container 2cc0766e7c96
 ---> 3f26a06eadd6
Step 5/6 : EXPOSE 8080
 ---> Running in ca2707e5131c
Removing intermediate container ca2707e5131c
 ---> 99ffae90756d
Step 6/6 : ENTRYPOINT  ["gunicorn", "-b", ":8080", "main:APP"]
 ---> Running in 9bc7aa38e838
Removing intermediate container 9bc7aa38e838
 ---> 8d28dc790a06
Successfully built 8d28dc790a06
Successfully tagged jwt-api-test:latest
pv252n:FSND-Deploy-Flask-App-to-Kubernetes-Using-EKS :~$
```

docker image ls

```
(fenv) pv252n:FSND-Deploy-Flask-App-to-Kubernetes-Using-EKS :~$ sudo docker image ls
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
jwt-api-test        latest              6054844dd2d6        About a minute ago   955MB
hello-world         latest              bf756fb1ae65        13 months ago       13.3kB
python              stretch             b9d77e48a75c        17 months ago       940MB
(fenv) pv252n:FSND-Deploy-Flask-App-to-Kubernetes-Using-EKS :~$
```

If required, you can delete an image using `docker image rm -f <image_name>`

5. Create and run a Container using the image locally:

Before running the container:

```
(fenv) pv252n:FSND-Deploy-Flask-App-to-Kubernetes-Using-EKS :~$ sudo docker ps -a
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS                      PORTS
NAMES
e538ec986456        hello-world         "/hello"            4 months ago        Exited (0) 4 months ago
jolly_edison
(fenv) pv252n:FSND-Deploy-Flask-App-to-Kubernetes-Using-EKS :~$
```

- You can pass the name of the env file using the flag `--env-file=<YOUR_ENV_FILENAME>`.
- You should expose the port 8080 of the container to the port 80 on your host machine.

docker run *--env-file=.env_file -p 80:8080 jwt-api-test*

```
(fenv) pv252n:FSND-Deploy-Flask-App-to-Kubernetes-Using-EKS :~$ sudo docker run --env-
file=.env_file -p 8080:8080 jwt-api-test
[2021-02-19 04:53:24 +0000] [1] [INFO] Starting gunicorn 20.0.4
[2021-02-19 04:53:24 +0000] [1] [INFO] Listening at: http://0.0.0.0:8080 (1)
[2021-02-19 04:53:24 +0000] [1] [INFO] Using worker: sync
[2021-02-19 04:53:24 +0000] [8] [INFO] Booting worker with pid: 8
2021-02-19 04:53:24,699 - main - DEBUG - Starting with log level: DEBUG
```

```
pv252n:FSND-Deploy-Flask-App-to-Kubernetes-Using-EKS :~$ sudo docker ps
[sudo] password for pv252n:
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS
PORTS               NAMES
9c152a509aa3        jwt-api-test        "gunicorn -b :8080 m..."   35 seconds ago      Up 34
seconds             0.0.0.0:8080->8080/tcp   upbeat_lederberg
pv252n:FSND-Deploy-Flask-App-to-Kubernetes-Using-EKS :~$
```

docker container ls

```
pv252n:FSND-Deploy-Flask-App-to-Kubernetes-Using-EKS :~$ sudo docker container ls
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS
PORTS               NAMES
9c152a509aa3        jwt-api-test        "gunicorn -b :8080 m..."   About a minute ago   Up About
a minute            0.0.0.0:8080->8080/tcp   upbeat_lederberg
pv252n:FSND-Deploy-Flask-App-to-Kubernetes-Using-EKS :~$
```

If required, you can stop a container using `docker stop [OPTIONS] CONTAINER [CONTAINER...]` or delete a container using `docker rm [OPTIONS] CONTAINER [CONTAINER...]`

6. To use the endpoints, you can use the same curl commands as before, except using port 80 this time:

- To try the /auth endpoint, use the following command:
  export TOKEN=`curl -d '{"email":"<EMAIL>","password":"<PASSWORD>"}' -H "Content-Type: application/json" -X POST localhost:80/auth | jq -r '.token'`

  export TOKEN=`curl -d '{"email":"pv252n@att.com","password":"pass1234"}' -H "Content-Type: application/json" -X POST localhost:8080/auth | jq -r '.token'`

  ```
  (penv) pv252n:FSND-Deploy-Flask-App-to-Kubernetes-Using-EKS :~$ export TOKEN=`curl -d '{"email":"pv252n@att.com","password":"pass1234"}' -H "Content-Type: application/json" -X POST localhost:8080/auth | jq -r '.token'`
    % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                   Dload  Upload   Total   Spent    Left  Speed
  100   222  100   174  100    48   7055   1946 --:--:-- --:--:-- --:--:--  7250
  (penv) pv252n:FSND-Deploy-Flask-App-to-Kubernetes-Using-EKS :~$
  ```

- To try the /contents endpoint which decrypts the token and returns its content, run:
  curl --request GET 'http://127.0.0.1:8080/contents' -H "Authorization: Bearer ${TOKEN}" | jq .

  ```
  (penv) pv252n:FSND-Deploy-Flask-App-to-Kubernetes-Using-EKS :~$ curl --request GET 'http://127.0.0.1:8080/contents' -H "Authorization: Bearer ${TOKEN}" | jq .
    % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                   Dload  Upload   Total   Spent    Left  Speed
  100    61  100    61    0     0   2272      0 --:--:-- --:--:-- --:--:--  2346
  {
    "email": "pv252n@att.com",
    "exp": 1602307741,
    "nbf": 1601098141
  }
  (penv) pv252n:FSND-Deploy-Flask-App-to-Kubernetes-Using-EKS :~$
  ```

Create an EKS Cluster and IAM Role

Priya Note to the Reviewer :
Was getting stuck at create stack step so based on the suggestion from mentor, followed link :
https://github.com/jungleBadger/FSND-Deploy-Flask-App-to-Kubernetes-Using-EKS/blob/master/troubleshooting/deploy.md#step-7---fill-the-ci-cd-codepipelinecfnyml-file

Above was provided by one of the mentors from Udacity, reference question is :
https://knowledge.udacity.com/questions/341318

1. Meeting Dependencies : DONE
   - AWS Account

     ```
     (penv) pv252n:FSND-Deploy-Flask-App-to-Kubernetes-Using-EKS :~$ aws --version
     aws-cli/1.18.146 Python/3.5.2 Linux/4.4.0-62-generic botocore/1.18.5
     (penv) pv252n:FSND-Deploy-Flask-App-to-Kubernetes-Using-EKS :~$
     ```

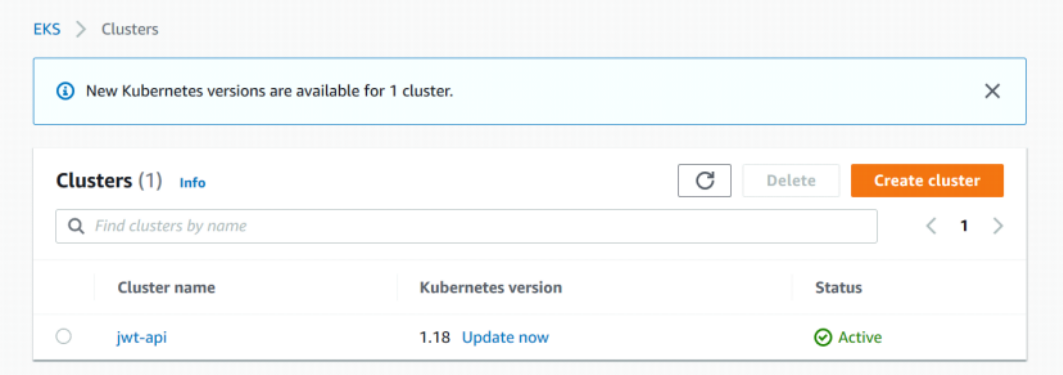
   - kubectl installed :

     ```
     pv252n:FSND-Deploy-Flask-App-to-Kubernetes-Using-EKS :~$ kubectl version
     Client Version: version.Info{Major:"1", Minor:"19", GitVersion:"v1.19.2",
     GitCommit:"f5743093fd1c663cb0cbc89748f730662345d44d", GitTreeState:"clean",
     BuildDate:"2020-09-16T13:41:02Z", GoVersion:"go1.15", Compiler:"gc",
     Platform:"linux/amd64"}
     Server Version: version.Info{Major:"1", Minor:"17+", GitVersion:"v1.17.12-eks-7684af",
     GitCommit:"7684af4ac41370dd109ac13817023cb8063e3d45", GitTreeState:"clean",
     BuildDate:"2020-10-20T22:57:40Z", GoVersion:"go1.13.15", Compiler:"gc",
     Platform:"linux/amd64"}
     pv252n:FSND-Deploy-Flask-App-to-Kubernetes-Using-EKS :~$
     ```

2. Create EKS Cluster: Done
   Before creating cluster - created a user with Administration Access :
   User - pv252n
   Prisha@20271324
   AKIAY5VJBRIHQUWSKG54
   Mjg2m3NVC5TYqwPeSUcW84BuBDoIDuI4iuX05Tzo

   ```
   pv252n:FSND-Deploy-Flask-App-to-Kubernetes-Using-EKS :~$ aws configure
   AWS Access Key ID [****************KG54]: AKIAY5VJBRIHQUWSKG54
   AWS Secret Access Key [****************5Tzo]:
   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
   Default region name [None]: us-west-2
   Default output format [None]: text
   pv252n:FSND-Deploy-Flask-App-to-Kubernetes-Using-EKS :~$ aws configure list
       Name                    Value             Type    Location
       ----                    -----             ----    --------
    profile                <not set>             None    None
   access_key     ****************KG54 shared-credentials-file
   secret_key     ****************5Tzo shared-credentials-file
     region                   us-west-2      config-file    ~/.aws/config
   pv252n:FSND-Deploy-Flask-App-to-Kubernetes-Using-EKS :~$
   ```

   https://us-west-2.console.aws.amazon.com/eks/home?region=us-west-2#/clusters



3. aws ssm put-parameter --name JWT_SECRET --value "myjwtsecret" --type SecureString - DONE

   pv252n:FSND-Deploy-Flask-App-to-Kubernetes-Using-EKS :~$ aws ssm put-parameter --name JWT_SECRET --value "myjwtsecret" --type SecureString --overwrite
   Standard 2
   pv252n:FSND-Deploy-Flask-App-to-Kubernetes-Using-EKS :~$
   Priya - Had to add the --overwrite option for this to work.

4. Create additional role and fetch aws file:
   a. pv252n:FSND-Deploy-Flask-App-to-Kubernetes-Using-EKS :~$ ACCOUNT_ID=$(aws sts get-caller-identity --query Account --output text)
   b. pv252n:FSND-Deploy-Flask-App-to-Kubernetes-Using-EKS :~$ TRUST="{ \"Version\": \"2012-10-17\", \"Statement\": [{ \"Effect\": \"Allow\", \"Principal\": { \"AWS\": \"arn:aws:iam::${ACCOUNT_ID}:root\" }, \"Action\": \"sts:AssumeRole\" }] }"

      pv252n:FSND-Deploy-Flask-App-to-Kubernetes-Using-EKS :~$ echo $TRUST
      { "Version": "2012-10-17", "Statement": [{ "Effect": "Allow", "Principal": { "AWS": "arn:aws:iam::613461101071:root" }, "Action": "sts:AssumeRole" }] }
      pv252n:FSND-Deploy-Flask-App-to-Kubernetes-Using-EKS :~$

   c. pv252n:FSND-Deploy-Flask-App-to-Kubernetes-Using-EKS :~$ aws iam create-role --role-name UdacityFlaskDeployCBKubectlRole --assume-role-policy-document "$TRUST" --output text --query 'Role.Arn'
      arn:aws:iam::613461101071:role/UdacityFlaskDeployCBKubectlRole
      pv252n:FSND-Deploy-Flask-App-to-Kubernetes-Using-EKS :~$

   d. pv252n:FSND-Deploy-Flask-App-to-Kubernetes-Using-EKS :~$ echo '{ "Version": "2012-10-17", "Statement": [{ "Effect": "Allow", "Action": ["eks:Describe*", "ssm:GetParameters"], "Resource": "*" }] }' > ./iam-role-policy
      pv252n:FSND-Deploy-Flask-App-to-Kubernetes-Using-EKS :~$

   e. pv252n:FSND-Deploy-Flask-App-to-Kubernetes-Using-EKS :~$ aws iam put-role-policy --role-name UdacityFlaskDeployCBKubectlRole --policy-name eks-describe --policy-document file://./iam-role-policy
      pv252n:FSND-Deploy-Flask-App-to-Kubernetes-Using-EKS :~$

   f. pv252n:FSND-Deploy-Flask-App-to-Kubernetes-Using-EKS :~$ kubectl get -n kube-system configmap/aws-auth -o yaml > ./aws-auth-patch.yml
      pv252n:FSND-Deploy-Flask-App-to-Kubernetes-Using-EKS :~$

   g. Here we will actually declare the new Role and patch the configuration back to our account. This step will in fact enable the UdacityFlaskDeployCBKubectlRole to perform the operations as expected.

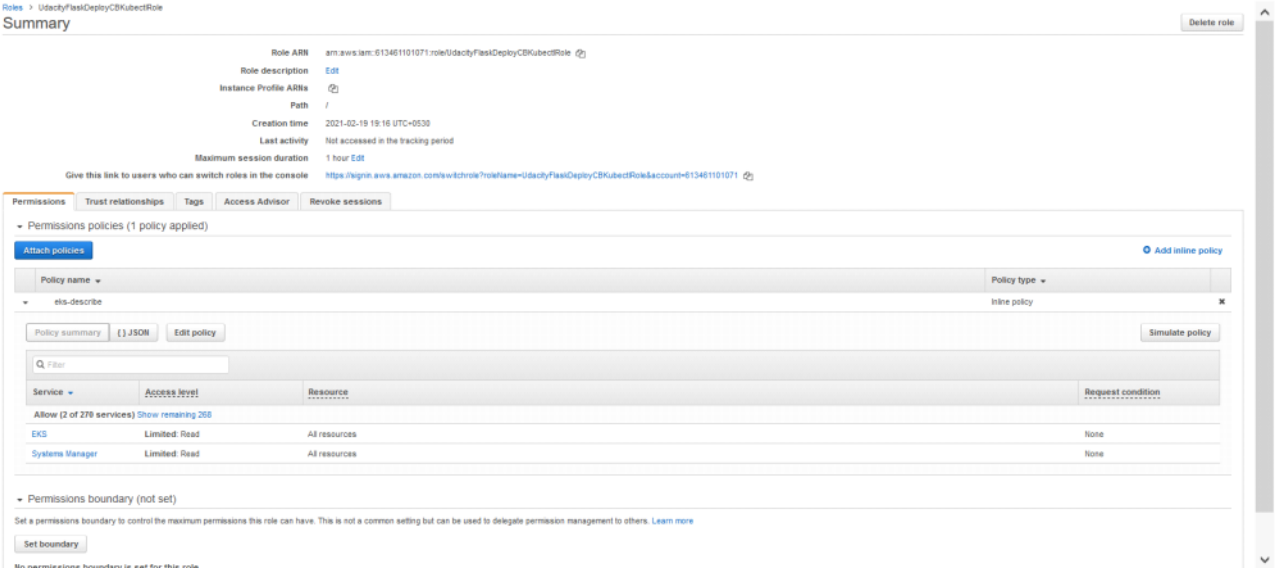
      pv252n:FSND-Deploy-Flask-App-to-Kubernetes-Using-EKS :~$ cat aws-auth-patch.yml
      apiVersion: v1
      data:
        mapRoles: |
          - groups:
            - system:bootstrappers
            - system:nodes
            rolearn: arn:aws:iam::613461101071:role/eksctl-jwt-api-nodegroup-ng-3eae5-NodeInstanceRole-Y75XDJVCRC0X
            username: system:node:{{EC2PrivateDNSName}}
          - groups:
            - system:masters
            rolearn: arn:aws:iam::613461101071:role/UdacityFlaskDeployCBKubectlRole
            username: build
        mapUsers: |
          []
      kind: ConfigMap
      metadata:
        creationTimestamp: "2021-02-19T13:27:30Z"
        managedFields:
        - apiVersion: v1
          fieldsType: FieldsV1
          fieldsV1:
            f:data:
              .: {}
              f:mapRoles: {}
              f:mapUsers: {}
            manager: eksctl
            operation: Update

```
  time: "2021-02-19T13:27:30Z"
 name: aws-auth
 namespace: kube-system
 resourceVersion: "1708"
 selfLink: /api/v1/namespaces/kube-system/configmaps/aws-auth
 uid: 2ed7938c-f16f-459d-bda8-f670eb4f1657
pv252n:FSND-Deploy-Flask-App-to-Kubernetes-Using-EKS :~$
```

h. Patch the modified aws-auth-patch.yml

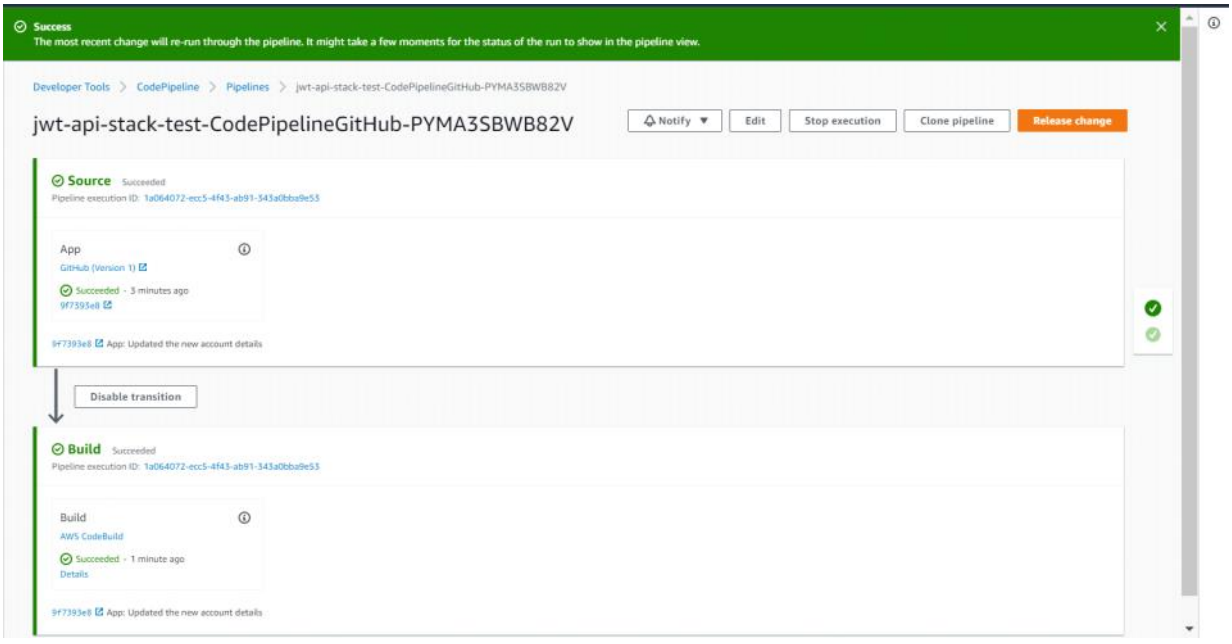
```
pv252n:FSND-Deploy-Flask-App-to-Kubernetes-Using-EKS :~$ kubectl patch configmap/aws-
auth -n kube-system --patch "$(cat ./aws-auth-patch.yml)"
configmap/aws-auth patched
pv252n:FSND-Deploy-Flask-App-to-Kubernetes-Using-EKS :~$
```

i.



j. Step 7 - Fill the ci-cd-codepipeline.cfn.yml file - DONE

k.



l. Grab the EKS Cluster endpoint URL -

Final successful result :

```
pv252n:FSND-Deploy-Flask-App-to-Kubernetes-Using-EKS :~$ kubectl get services simple-
jwt-api -o wide
NAME          TYPE          CLUSTER-IP      EXTERNAL-IP
PORT(S)       AGE   SELECTOR
simple-jwt-api   LoadBalancer   10.100.163.178
adf56b6e69ade45208e2f8e0906236d4-54416909.us-west-2.elb.amazonaws.com
80:31002/TCP   17m   app=simple-jwt-api
pv252n:FSND-Deploy-Flask-App-to-Kubernetes-Using-EKS :~$

pv252n:FSND-Deploy-Flask-App-to-Kubernetes-Using-EKS :~$ export TOKEN=`curl -d
'{"email":"pv252n@att.com","password":"pass"}' -H "Content-Type: application/json" -X
```

```
POST $URL/auth  | jq -r '.token`
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100   218  100   174  100    44    194     49 --:--:-- --:--:-- --:--:--   194
pv252n:FSND-Deploy-Flask-App-to-Kubernetes-Using-EKS :~$ curl --request GET
$URL:80/contents -H "Authorization: Bearer ${TOKEN}" | jq
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100    61  100    61    0     0     69      0 --:--:-- --:--:-- --:--:--    70
{
  "email": "pv252n@att.com",
  "exp": 1614956196,
  "nbf": 1613746596
}
pv252n:FSND-Deploy-Flask-App-to-Kubernetes-Using-EKS :~$

pv252n:FSND-Deploy-Flask-App-to-Kubernetes-Using-EKS :~$ echo $TOKEN
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJleHAiOjE2MTQ5NTYxOTYsIm5iZiI6MTYxMzc0NjU
5NiwiZW1haWwiOiJwdjI1Mm5AYXR0LmNvbSJ9.rLWKV1oyFIQE41qQ8U497XP0aLYkM413kv
p4l6VDHFA
pv252n:FSND-Deploy-Flask-App-to-Kubernetes-Using-EKS :~$
```

Adding Tests to the Build

a. **Changes to buildspec.yml file**:

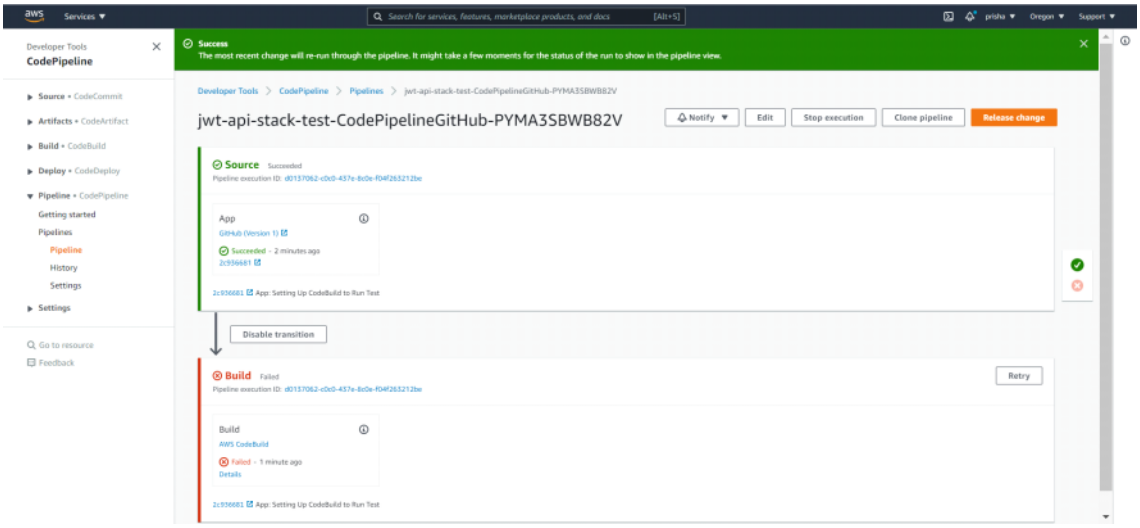
```
pv252n:FSND-Deploy-Flask-App-to-Kubernetes-Using-EKS :~$ git diff buildspec.yml
diff --git a/buildspec.yml b/buildspec.yml
index 457cc42..db2d875 100644
--- a/buildspec.yml
+++ b/buildspec.yml
@@ -22,10 +22,14 @@ phases:
      - apt-get update && apt-get -y install jq  && pip install --upgrade awscli
  pre_build:
    commands:
+     - docker --version
+     - docker help
      - TAG="$REPOSITORY_NAME.$REPOSITORY_BRANCH.$ENVIRONMENT_NAME.$(date
+%Y-%m-%d.%H.%M.%S).$(echo $CODEBUILD_RESOLVED_SOURCE_VERSION | head -c 8)"
      - sed -i 's@CONTAINER_IMAGE@'"$REPOSITORY_URI:$TAG"'@' simple_jwt_api.yml
      - $(aws ecr get-login --no-include-email)
      - export KUBECONFIG=$HOME/.kube/config
+     - pip3 install -r requirements.txt
+     - python -m pytest test_main.py
  build:
    commands:
     - docker build --tag $REPOSITORY_URI:$TAG .
pv252n:FSND-Deploy-Flask-App-to-Kubernetes-Using-EKS :~$
```

b. **Changes to test_main.py**

```
pv252n:FSND-Deploy-Flask-App-to-Kubernetes-Using-EKS :~$ git diff test_main.py
diff --git a/test_main.py b/test_main.py
index 3c6c898..10eb552 100644
--- a/test_main.py
+++ b/test_main.py
@@ -26,6 +26,7 @@ def test_health(client):
    response = client.get('/')
    assert response.status_code == 200
    assert response.json == 'Healthy'
+   assert False
 def test_auth(client):
pv252n:FSND-Deploy-Flask-App-to-Kubernetes-Using-EKS :~$

pv252n:FSND-Deploy-Flask-App-to-Kubernetes-Using-EKS :~$ git add buildspec.yml
test_main.py
pv252n:FSND-Deploy-Flask-App-to-Kubernetes-Using-EKS :~$ git commit -m "Setting Up
CodeBuild to Run Test"
[master 2c93668] Setting Up CodeBuild to Run Test
 2 files changed, 5 insertions(+)
pv252n:FSND-Deploy-Flask-App-to-Kubernetes-Using-EKS :~$ git push origin master
Username for 'https://github.com': priyavprashanth
Password for 'https://priyavprashanth@github.com':
Counting objects: 8, done.
Compressing objects: 100% (8/8), done.
Writing objects: 100% (8/8), 1.33 KiB | 0 bytes/s, done.
Total 8 (delta 5), reused 0 (delta 0)
remote: Resolving deltas: 100% (5/5), completed with 4 local objects.
To https://github.com/priyavprashanth/FSND-Deploy-Flask-App-to-Kubernetes-Using-
EKS.git
   9f7393e..2c93668  master -> master
pv252n:FSND-Deploy-Flask-App-to-Kubernetes-Using-EKS :~$
```

Build failed after checking in:

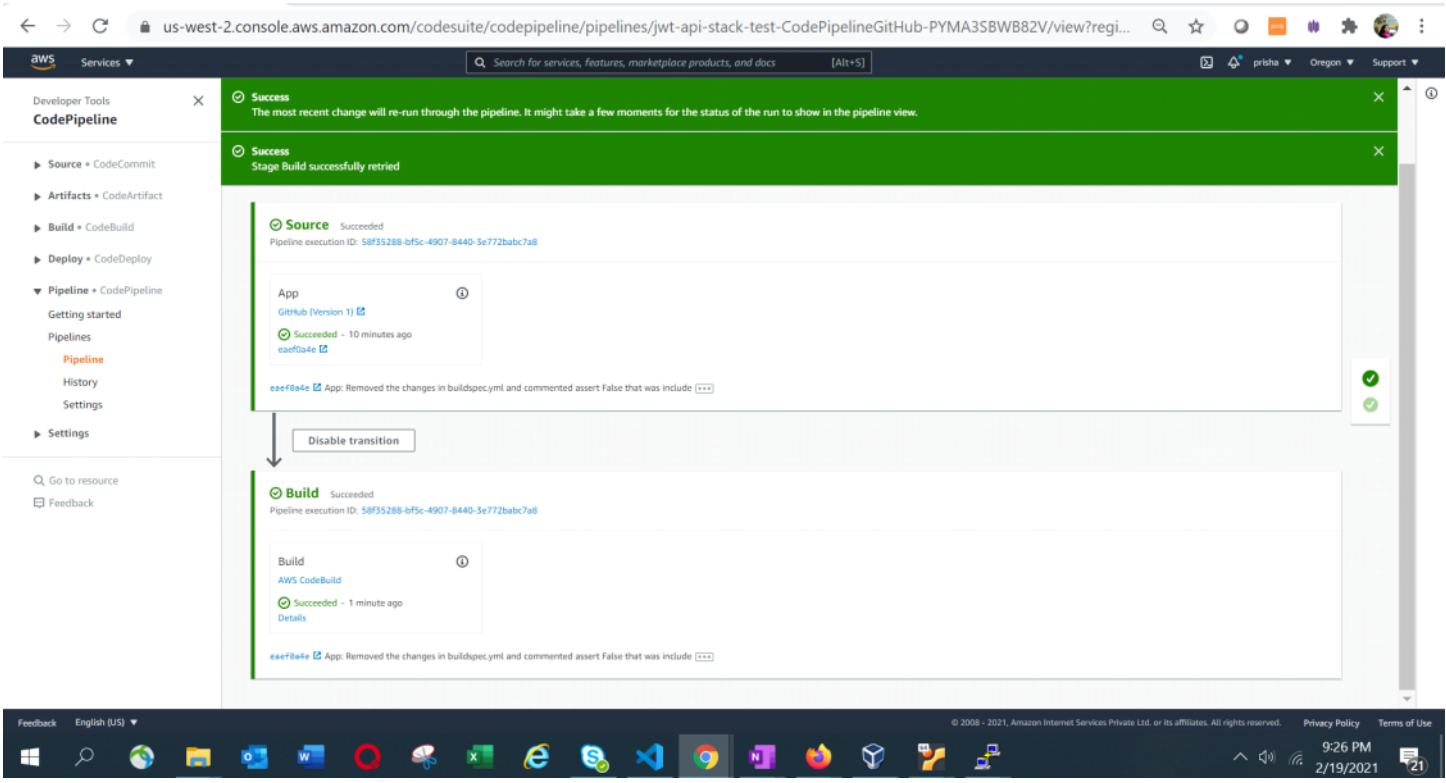


Priya : Commenting out the change in buildspec.yml and test_main.py and checking in the changes to show that now it is working.

pv252n:FSND-Deploy-Flask-App-to-Kubernetes-Using-EKS :~$ **git diff buildspec.yml**
diff --git a/buildspec.yml b/buildspec.yml
index db2d875..457cc42 100644
--- a/buildspec.yml
+++ b/buildspec.yml
@@ -22,14 +22,10 @@ phases:
      - apt-get update && apt-get -y install jq && pip install --upgrade awscli
  pre_build:
    commands:
-     - docker --version
-     - docker help
      - TAG="$REPOSITORY_NAME.$REPOSITORY_BRANCH.$ENVIRONMENT_NAME.$(date
+%Y-%m-%d.%H.%M.%S).$(echo $CODEBUILD_RESOLVED_SOURCE_VERSION | head -c 8)"
      - sed -i 's@CONTAINER_IMAGE@'"$REPOSITORY_URI:$TAG"'@' simple_jwt_api.yml
      - $(aws ecr get-login --no-include-email)
      - export KUBECONFIG=$HOME/.kube/config
-     - pip3 install -r requirements.txt
-     - python -m pytest test_main.py
  build:
    commands:
      - docker build --tag $REPOSITORY_URI:$TAG .
pv252n:FSND-Deploy-Flask-App-to-Kubernetes-Using-EKS :~$

pv252n:FSND-Deploy-Flask-App-to-Kubernetes-Using-EKS :~$ **git diff test_main.py**
diff --git a/test_main.py b/test_main.py
index 10eb552..6e07098 100644
--- a/test_main.py
+++ b/test_main.py
@@ -26,7 +26,7 @@ def test_health(client):
    response = client.get('/')
    assert response.status_code == 200
    assert response.json == 'Healthy'
-   assert False
+#   assert False

 def test_auth(client):
pv252n:FSND-Deploy-Flask-App-to-Kubernetes-Using-EKS :~$



Build is successful