<https://github.com/Iforgotaboutitin2seconds/Sprint.git>

I already know I need AWS and docker as that is what I am exploring.

I need to find how to use AWS and docker to host my Django app.

[Deploying Django to AWS with Docker and Let's Encrypt | TestDriven.io](https://testdriven.io/blog/django-docker-https-aws/)

Find the first link at the google and trying it out.

Ran into issue when I try to access the server, it said permission denied (publickey).

No idea why it won’t work so I look around and try connecting to instance. It gave me an example, so I used it and it worked.

ssh -i "Django.pem" [ubuntu@ec2-52-14-187-43.us-east-2.compute.amazonaws.com](mailto:ubuntu@ec2-52-14-187-43.us-east-2.compute.amazonaws.com)

but this will change once you added Elastic IP addresses

Turns out I am just stupid and did not read carefully about the instruction.

$ sudo apt update

$ sudo apt install apt-transport-https ca-certificates curl software-properties-common

$ curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -

$ sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu bionic stable"

$ sudo apt update

$ sudo apt install docker-ce

$ sudo usermod -aG docker **${**USER**}**

$ sudo curl -L "https://github.com/docker/compose/releases/download/1.29.2/docker-compose-**$(**uname -s**)**-**$(**uname -m**)**" -o /usr/local/bin/docker-compose

$ sudo chmod +x /usr/local/bin/docker-compose

By following the above I installed docker and compose of it.

Same for AWS CLI.

For some unknown reasons the author did not add how to do the DNS record. So, I did some research to understand what it is trying to say. That it is already done. Since I got the Elastic IP associated to the instance.

Since I am not familiar with the UI, so I just use the search bar on the top to get to what the author wants me to go to.

Seems like they messed up the connectivity part’s screen shots. Need to click additional configuration to see database port. 图形用户界面, 文本, 电子邮件

描述已自动生成

You can use dG in command mode to delete rest of the file and use right click to paste stuff into it.

Ran into

ERROR: invalid tag "<aws-account-id>.dkr.ecr.<aws-region>.amazonaws.com/django-ec2:web": invalid reference format

ERROR: Service 'web' failed to build : Build failed

It happened because I just copy and paste and did not actually read what it wants me to do.

Ran into issue during Running the container’s part.

$ scp -i /path/to/your/djangoletsencrypt.pem **\**

-r **$(**pwd**)**/{app,nginx,.env.staging,.env.staging.proxy-companion,docker-compose.staging.yml} **\**

ubuntu@public-ip-or-domain-of-ec2-instance:/path/to/django-on-docker

This part to be exact. It keeps saying:

+ CategoryInfo : ParserError: (:) [], ParentContainsErrorRecordException

+ FullyQualifiedErrorId : MissingArgument

I just skip this part since it did not really do anything anyway.

I tried to run the container. Did not work because I am stupid and did not change .env.staging file’s virtual\_host and stuff.

Just need to replace all <YOUR\_DOMAIN.COM> to your public IPv4 address.

图形用户界面, 文本, 应用程序, 电子邮件

描述已自动生成

After that I ran into 503 Service Temporarily Unavailable.

I take a look over all steps, I cannot find the different other than the scp command. After some research all it did is to copy files from my local machine to the ubuntu. But I do not have any files about app,nginx, stuff.

I take a guess and think it might mean to get my Django project in it. But it won’t make sense since I already have a Django project from the example.

But I had no other idea so might as well try. Before I tried, I realized that I never run the Django server yet. When I tried, I noticed that since it is different system it used python3 instead of just python. Then I realized that is exactly what docker is doing.

I tried to do load balancer and target group, nothing happened.

Turns out I should had taken a deeper look into the step of Add DNS Record

I use route 53 in AWS to get a hosted zone to have it to get an A record to my DNS.

Turns out it did not do anything.

So, I reach my limit and move on, since I know all this is meant for staging just an example to show how it work. Maybe it will be fixed somehow when it deploys.

Took me 2 hours straight by the way.

After I read again, I realize I am supposed to run everything in my local machine before uploading it to the AWS.

But whatever.

After a while I just gave up and upload my Django project and give it a try.

I uploaded my Django project. Make a Dockerfile and made it with my Django project. Luckily it worked fine.

The thing is when I run it, it did not show up at the IP.

I tried to use windows Dockerfile for a long time to realize it might be the reason why it won’t work. So, I modify it to ubuntu version.

After some mess around I decided to follow documentations of AWS themselves.

<https://docs.aws.amazon.com/AmazonECR/latest/userguide/getting-started-cli.html>

that did not work out, so I try the following:

<https://dev.to/ki3ani/deploying-your-first-dockerized-django-rest-api-on-aws-elastic-beanstalk-a-comprehensive-guide-2m77>

I got so annoyed that I stopped writing down what I did.

After some nice tries, I went back to the first link again.

Then I realized that this is 3rd part of the 3 parts tutorials. I will just reset everything and follow all of them.

So I decided to restart.

I need to find how to use AWS and docker to host my Django app.

[Dockerizing Django with Postgres, Gunicorn, and Nginx | TestDriven.io](https://testdriven.io/blog/dockerizing-django-with-postgres-gunicorn-and-nginx/)

I setup an ec2 instance. Install pip. And put my github stuff in it.

Install my requirements.

Install docker and docker compose.

sudo apt update

sudo apt install -y apt-transport-https ca-certificates curl software-properties-common

curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o /usr/share/keyrings/docker-archive-keyring.gpg

echo "deb [arch=amd64 signed-by=/usr/share/keyrings/docker-archive-keyring.gpg] https://download.docker.com/linux/ubuntu $(lsb\_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list > /dev/null

sudo apt update

sudo apt install -y docker-ce docker-ce-cli containerd.io

sudo systemctl start docker

sudo systemctl enable docker

sudo docker –version

sudo curl -L "https://github.com/docker/compose/releases/latest/download/docker-compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose

learned to use sudo su, instead of sudo -I as this will put me at the /root.

I decide to use chat gpt to get me through the simple stuff.

For some reason when I tried to deploy it, it keeps saying I don’t have Dockerfile.

I could not figure out why, so I redo everything. By delete the elastic bean env, app and docker images.

Whenever I deploy from eb deploy, it creates a new instance with update code. After it successfully pass the health check it will slowly replace everything in the old instance with new one. Therefore, to minimize the down time.

I decided to put the Dockerfile in my commit of git hub. Maybe that will work.

That is the indeed the issue, now the issue is that according to the logs it worked fine, but when I tried to visit it at the website it refused to work.

It seems to be the heath checks. It is red.

Running instances: 1

i-0e58894e40ec42770: unhealthy

Description: Health checks failed with these codes: [302]

Reason: Target.ResponseCodeMismatch

I fixed it by changing the port to match each other.