SUMMARY:

This Lab covers Containers & ECS

Contents

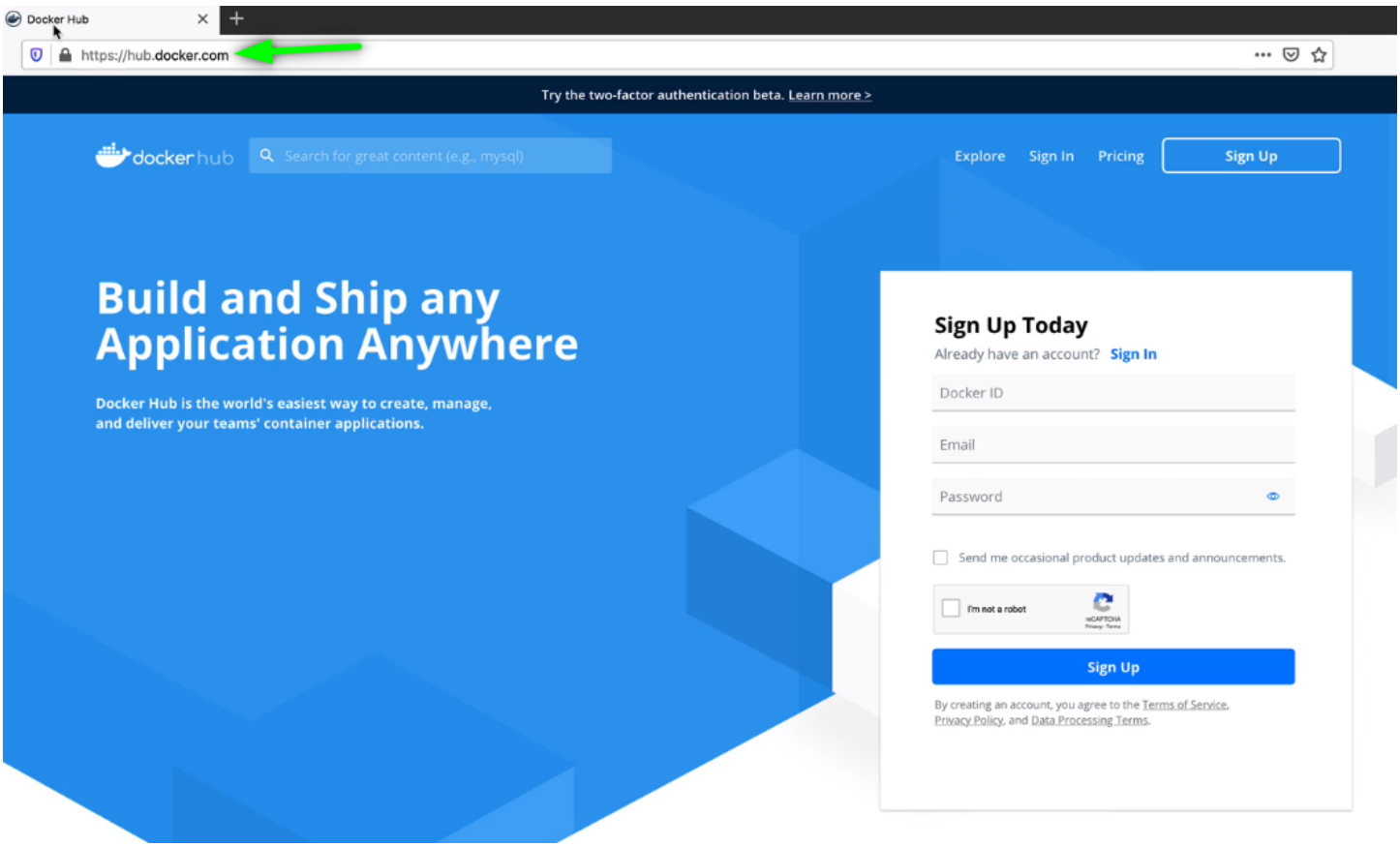
1. Creating 'container of cats' Docker Image

- In this [DEMO] lesson we create a docker image containing the 'container of cats' application.
- We will install the docker engine on an EC2 Instance and use this to create the image.
- To test the image we will 'RUN' the image, creating a docker container and once tested, upload the image to dockerhub.

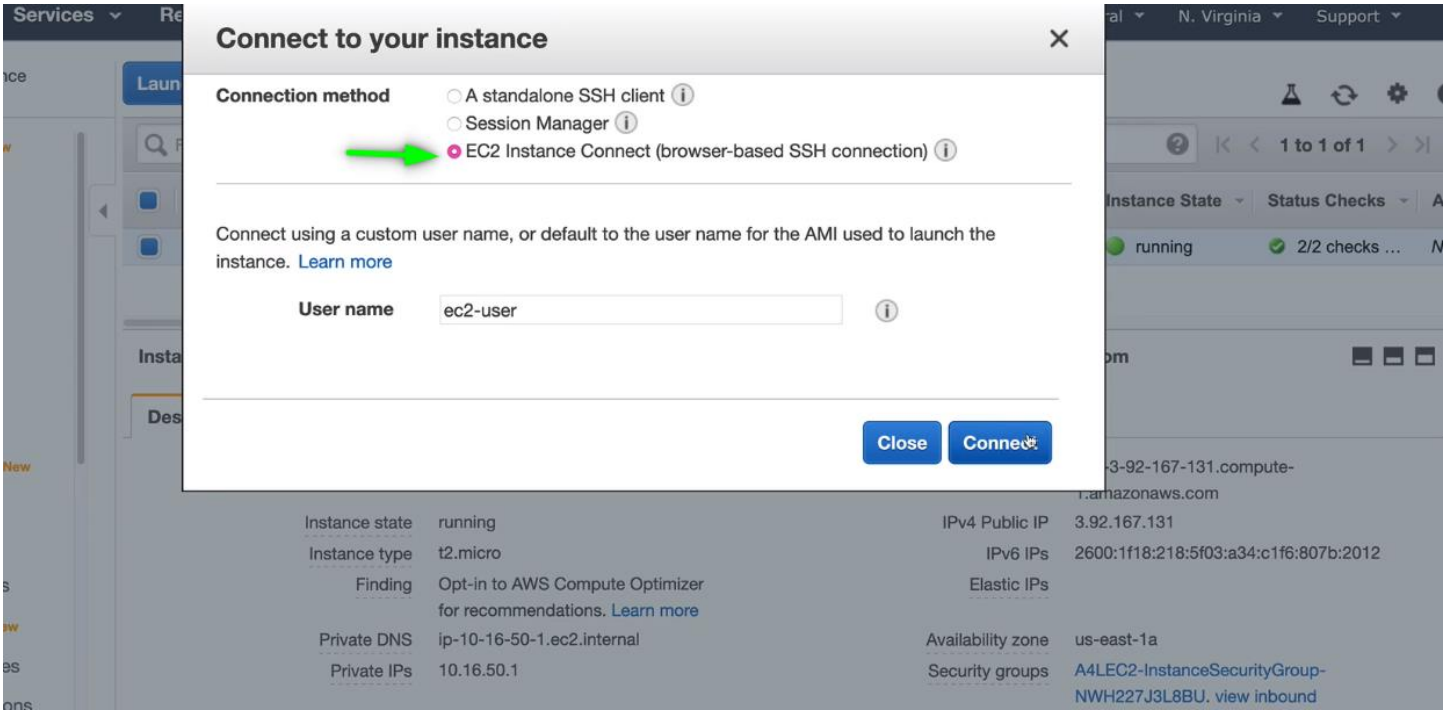
COMMANDS:

```
# Install Docker Engine on EC2 Instance
sudo amazon-linux-extras install docker
sudo service docker start
sudo usermod -a -G docker ec2-user

LOGOUT and login

# Install GIT and Download Course Repo
sudo yum install git
git clone https://github.com/acantril/aws-sa-associate-saac02.git

# Build Docker Image
cd aws-sa-associate-saac02/09-Containers-ECS/container_of_cats/container
docker build -t containerofcats .
docker images --filter reference=containerofcats

# Run Container from Image
docker run -t -i -p 80:80 containerofcats

# Upload Container to Dockerhub (optional)
docker login --username=YOUR_USER
docker images
docker tag IMAGEID YOUR_USER/containerofcats
docker push YOUR_USER/containerofcats:latest
```

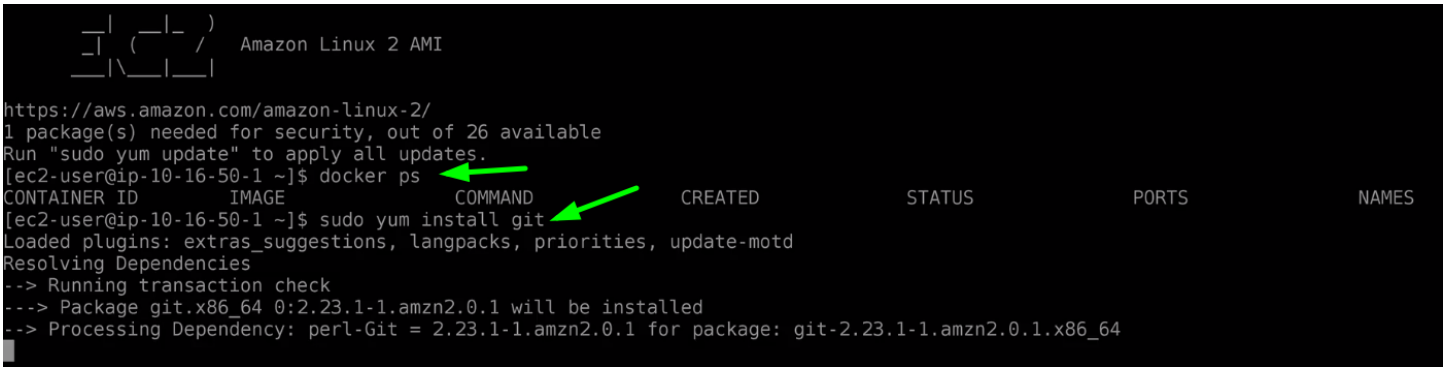
Create Docker hub account – to upload into afterwards:

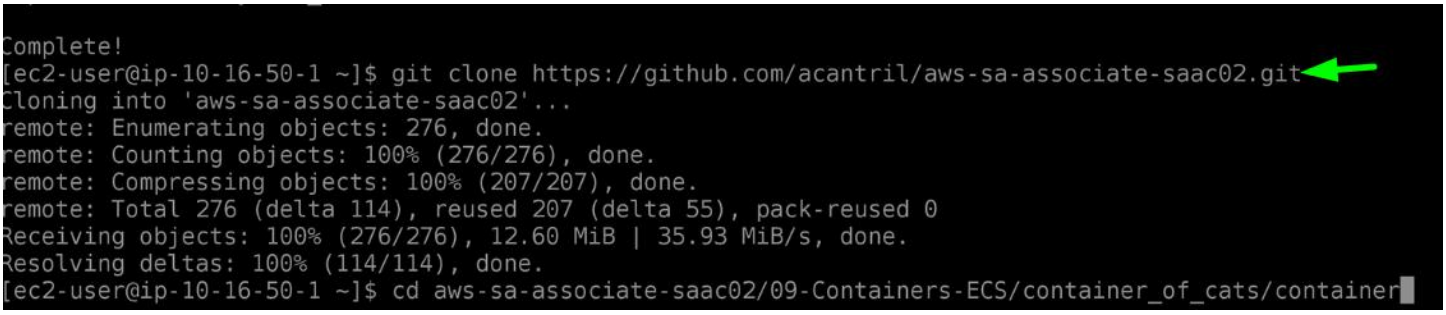


Create EC2 with CFN yaml file, then connect via SSH:

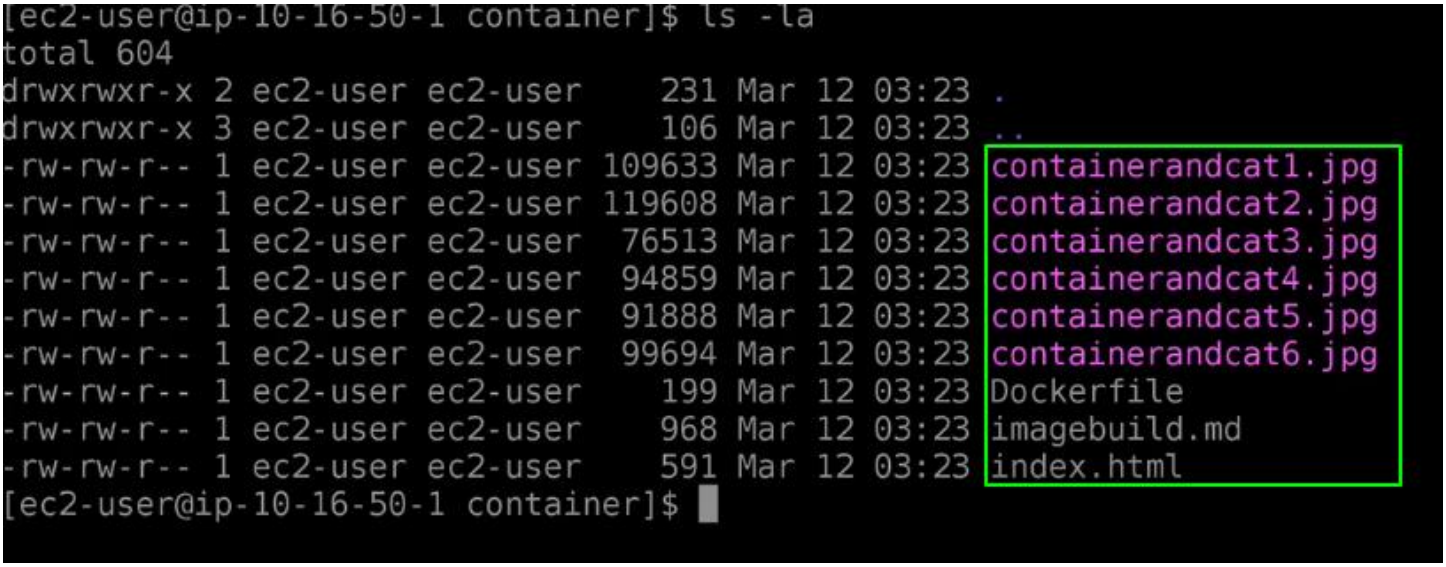


Install docker and git, then clone Adrian repo for lesson:





Locate files for docker image from adrain's cloned repo:



Build container of cats:

Docker file:

```
FROM centos:latest
LABEL maintainer="Animals4life"
RUN yum -y install httpd
COPY index.html /var/www/html/
COPY containerandcat*.jpg /var/www/html/
ENTRYPOINT ["/usr/sbin/httpd", "-D", "FOREGROUND"]
EXPOSE 80
```

```
[ec2-user@ip-10-16-50-1 container]$ docker build -t containerofcats .
Sending build context to Docker daemon  602.1kB
Step 1/7 : FROM centos:latest
latest: Pulling from library/centos
8a29a15cefae: Extracting [==================================================>]  67.4MB/73.23MB
```

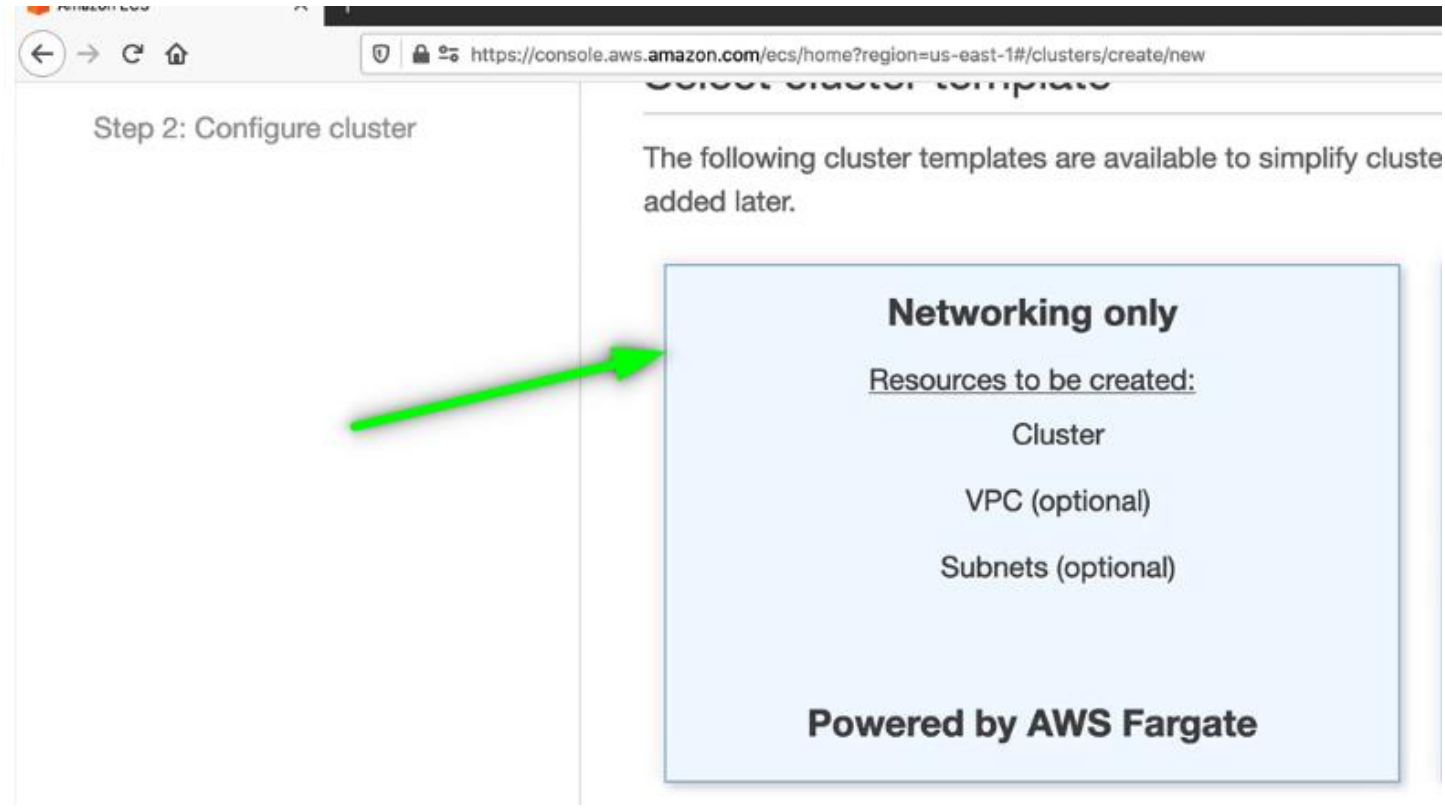
Login with username, create container of cats image:

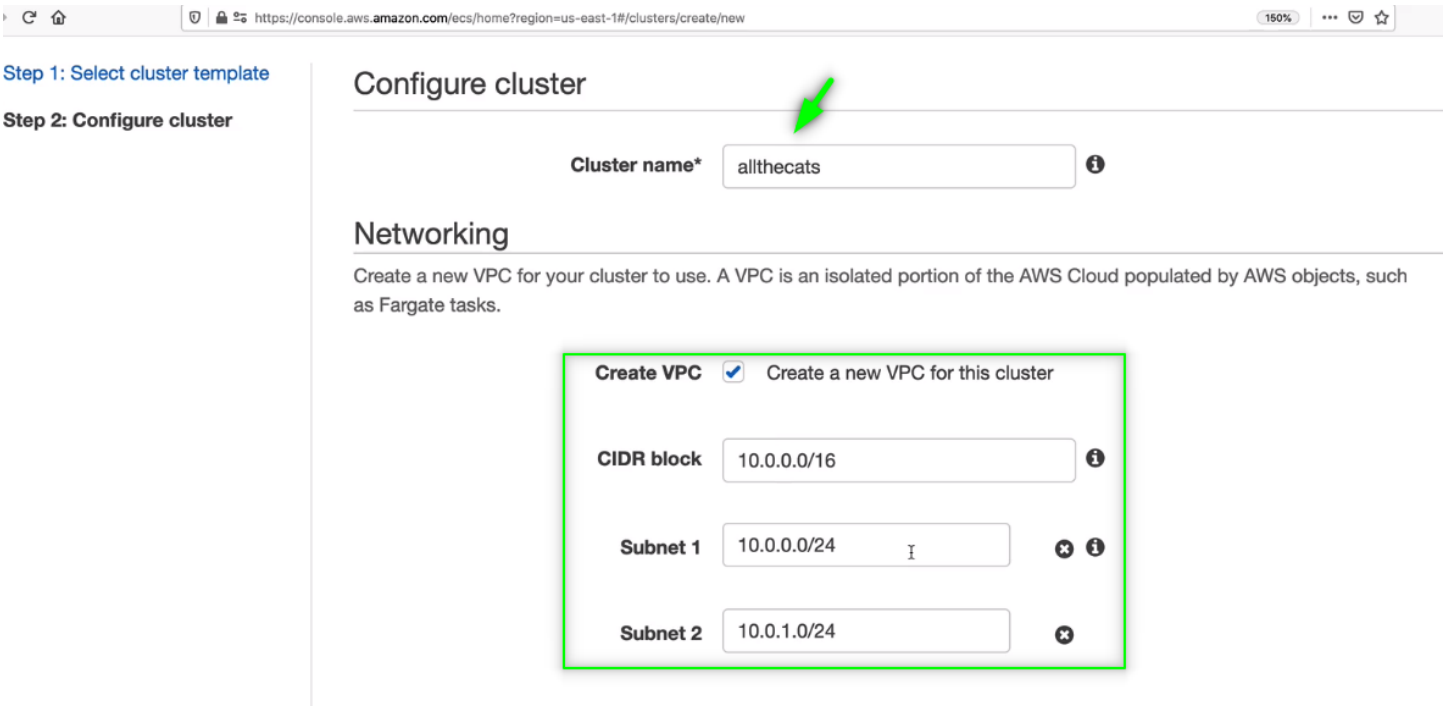
```
https://console.aws.amazon.com/ec2/v2/connect/ec2-user/i-04deb2235dbc0a598
[ec2-user@ip-10-16-50-1 container]$ docker login --username [redacted]
Password:
WARNING! Your password will be stored unencrypted in /home/ec2-user/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store

Login Succeeded
[ec2-user@ip-10-16-50-1 container]$ docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
containerofcats     latest              d3113fce67a5        11 minutes ago      284MB
centos              latest              470671670cac        7 weeks ago         237MB
[ec2-user@ip-10-16-50-1 container]$ docker tag d3113fce67a5 [redacted]containerofcats
[ec2-user@ip-10-16-50-1 container]$ docker push [redacted]containerofcats:latest
The push refers to repository [docker.io/[redacted]containerofcats]
e39773e27609: Pushed
8507357b95bf: Pushed
0bead52d58b2: Pushed
0683de282177: Mounted from library/centos
latest: digest: sha256:43587662332385cd909a9957bd37f2ad5851d291aef2ce88f3e6e74bcdda72a6 size: 1158
[ec2-user@ip-10-16-50-1 container]$
```

2. Deploying 'container of cats' using Fargate

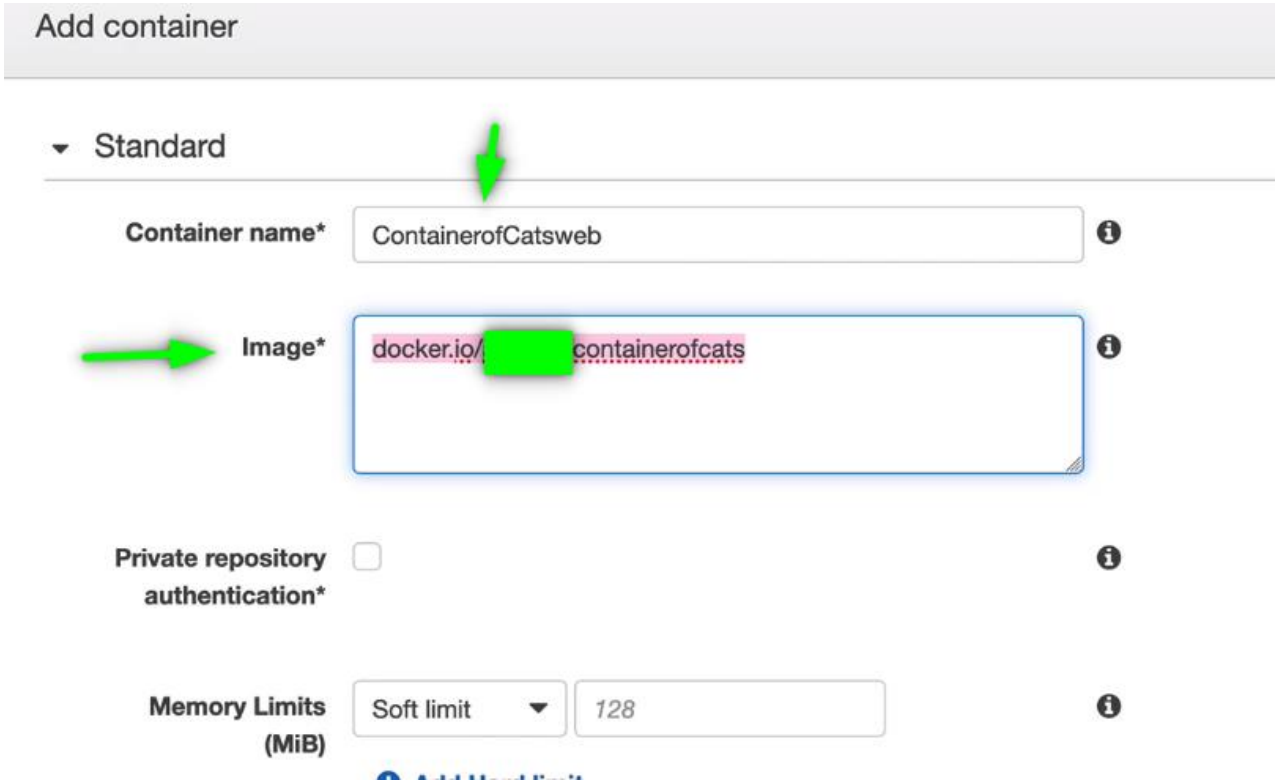
In this [DEMO] lesson you will create a Fargate Cluster, create a task and container definition and deploy the world renowned 'container of cats' Application from Dockerhub into Fargate.

Create fargate cluster:

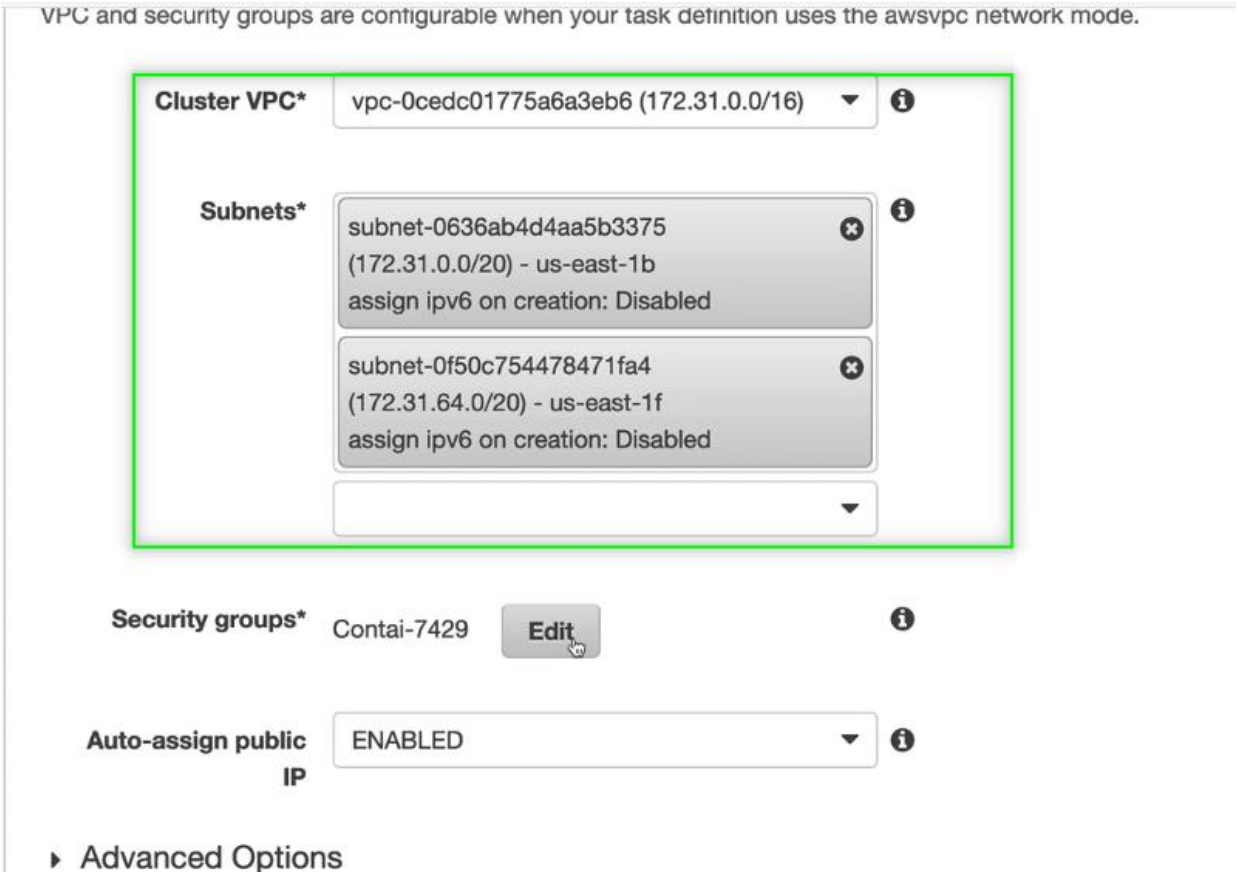


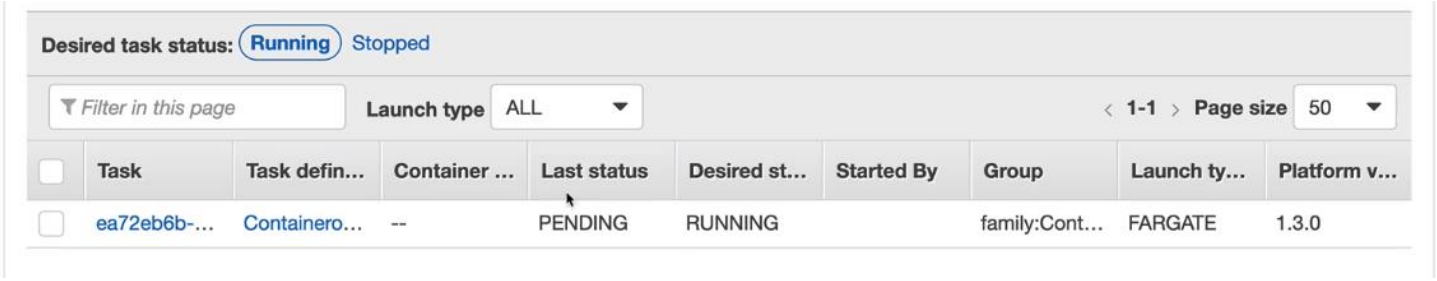


Point to dockerhub account image:



Configure VPC:





Once up, obtain IP and check via browser: