

SUMMARY:

This Lab covers Containers & ECS

Contents

1. Creating 'container of cats' Docker Image 1
 2. Deploying 'container of cats' using Fargate 4
-

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

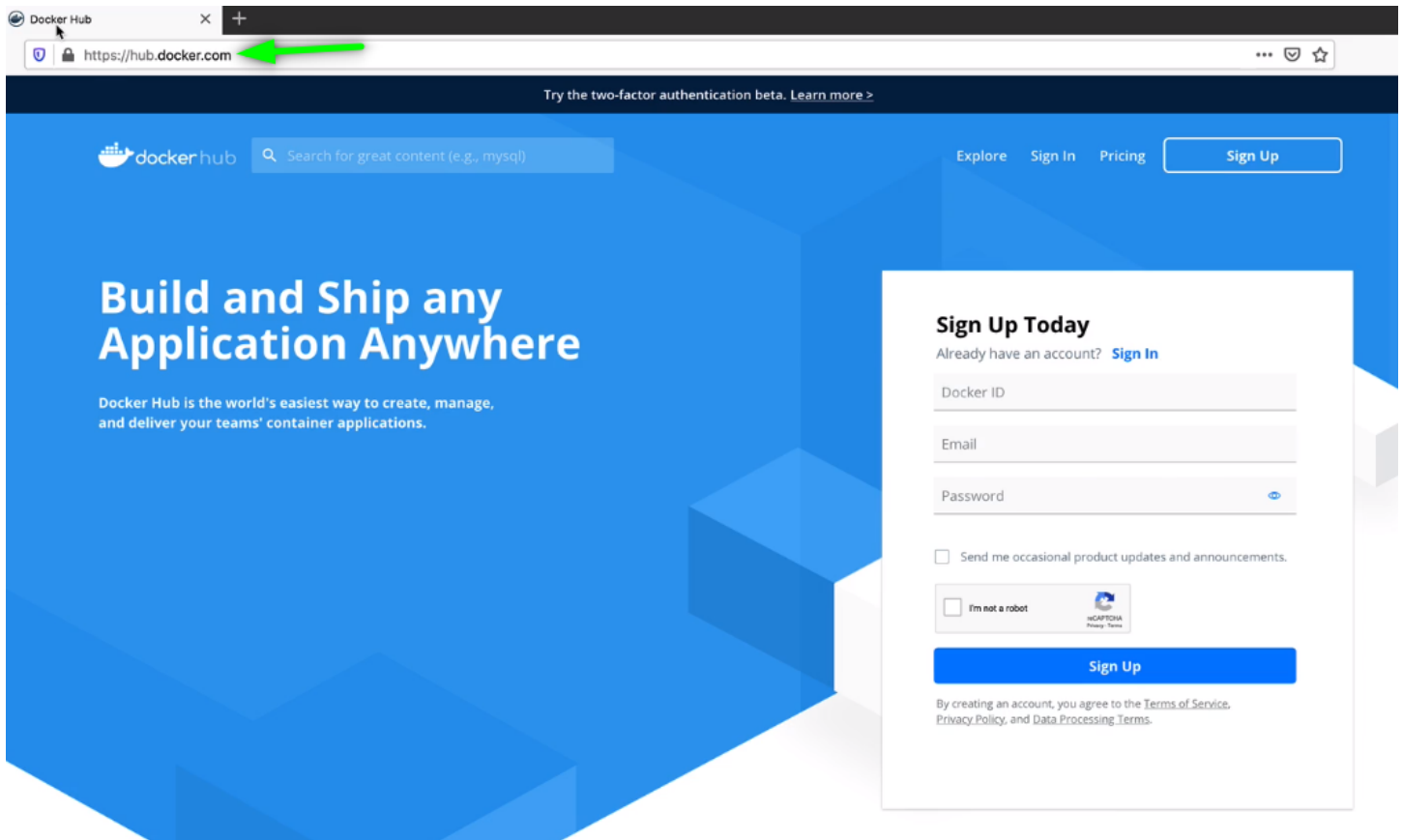
# Install GIT and Download Course Repo
sudo yum install git
git clone https://github.com/acantril/aws-sa-associate-saac02.git

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

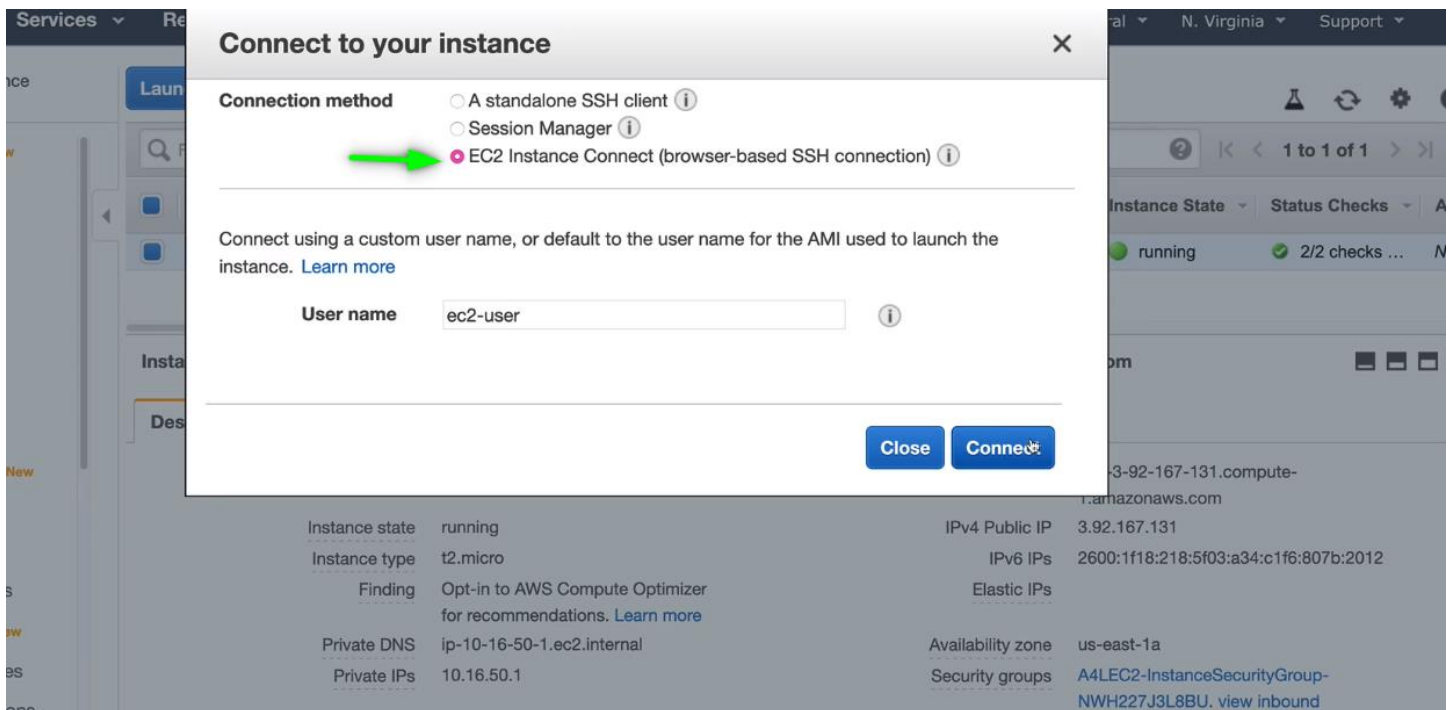
# Run Container from Image
docker run -t -i -p 80:80 containerofcats

# 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:

```

 _| ( _|_ )
 _| ( _|_ /   Amazon Linux 2 AMI
 _|\_|\_|\_|\_

https://aws.amazon.com/amazon-linux-2/
1 package(s) needed for security, out of 26 available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-10-16-50-1 ~]$ sudo amazon-linux-extras install docker

```

```

 _| ( _|_ )
 _| ( _|_ /   Amazon Linux 2 AMI
 _|\_|\_|\_|\_

https://aws.amazon.com/amazon-linux-2/
1 package(s) needed for security, out of 26 available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-10-16-50-1 ~]$ docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS              PORTS              NAMES
[ec2-user@ip-10-16-50-1 ~]$ sudo yum install git
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
Resolving Dependencies
--> Running transaction check
---> Package git.x86_64 0:2.23.1-1.amzn2.0.1 will be installed
--> Processing Dependency: perl-Git = 2.23.1-1.amzn2.0.1 for package: git-2.23.1-1.amzn2.0.1.x86_64

```

```

Complete!
[ec2-user@ip-10-16-50-1 ~]$ git clone https://github.com/acantril/aws-sa-associate-saac02.git
Cloning into 'aws-sa-associate-saac02'...
remote: Enumerating objects: 276, done.
remote: Counting objects: 100% (276/276), done.
remote: Compressing objects: 100% (207/207), done.
remote: Total 276 (delta 114), reused 207 (delta 55), pack-reused 0
Receiving objects: 100% (276/276), 12.60 MiB | 35.93 MiB/s, done.
Resolving deltas: 100% (114/114), done.
[ec2-user@ip-10-16-50-1 ~]$ cd aws-sa-associate-saac02/09-Containers-ECS/container_of_cats/container

```

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

```

[ec2-user@ip-10-16-50-1 container]$ ls -la
total 604
drwxrwxr-x 2 ec2-user ec2-user    231 Mar 12 03:23 .
drwxrwxr-x 3 ec2-user ec2-user    106 Mar 12 03:23 ..
-rw-rw-r-- 1 ec2-user ec2-user 109633 Mar 12 03:23 containerandcat1.jpg
-rw-rw-r-- 1 ec2-user ec2-user 119608 Mar 12 03:23 containerandcat2.jpg
-rw-rw-r-- 1 ec2-user ec2-user  76513 Mar 12 03:23 containerandcat3.jpg
-rw-rw-r-- 1 ec2-user ec2-user  94859 Mar 12 03:23 containerandcat4.jpg
-rw-rw-r-- 1 ec2-user ec2-user  91888 Mar 12 03:23 containerandcat5.jpg
-rw-rw-r-- 1 ec2-user ec2-user  99694 Mar 12 03:23 containerandcat6.jpg
-rw-rw-r-- 1 ec2-user ec2-user    199 Mar 12 03:23 Dockerfile
-rw-rw-r-- 1 ec2-user ec2-user    968 Mar 12 03:23 imagebuild.md
-rw-rw-r-- 1 ec2-user ec2-user    591 Mar 12 03:23 index.html
[ec2-user@ip-10-16-50-1 container]$

```

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
3a29a15cefae: Extracting [=====> ] 67.4MB/73.23MB
```

Login with username, create container of cats image:

```
[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:

The screenshot displays the AWS Management Console interface for creating a new ECS cluster. The left sidebar shows the navigation pane with 'Step 1: Select cluster template' and 'Step 2: Configure cluster'. The main content area is titled 'Configure cluster' and shows the 'Networking only' template selected. A green arrow points to the 'allthecats' cluster name input field. Below the cluster name, the 'Networking' section is visible, showing the 'Create VPC' checkbox checked. A green box highlights the VPC configuration details, including the CIDR block (10.0.0.0/16) and two subnets (Subnet 1: 10.0.0.0/24, Subnet 2: 10.0.1.0/24).

Step 2: Configure cluster

Select cluster template

The following cluster templates are available to simplify cluster creation.

Networking only

Resources to be created:

- Cluster
- VPC (optional)
- Subnets (optional)

Powered by AWS Fargate

Cluster name* allthecats

Networking

Create a new VPC for your cluster to use. A VPC is an isolated portion of the AWS Cloud populated by AWS objects, such as Fargate tasks.

Create VPC ☒ Create a new VPC for this cluster

CIDR block 10.0.0.0/16

Subnet 1 10.0.0.0/24

Subnet 2 10.0.1.0/24

Point to dockerhub account image:


Add container


▼ Standard

Container name*

ContainerofCatsweb

i

 **Image***

docker.io/containerofcats

i

Private repository authentication*

☐

i

Memory Limits (MiB)

Soft limit ▼

128

i

Configure VPC:

VPC and security groups are configurable when your task definition uses the awsvpc network mode.

Cluster VPC*

vpc-0cedc01775a6a3eb6 (172.31.0.0/16)

i

Subnets*

subnet-0636ab4d4aa5b3375
(172.31.0.0/20) - us-east-1b
assign ipv6 on creation: Disabled

subnet-0f50c754478471fa4
(172.31.64.0/20) - us-east-1f
assign ipv6 on creation: Disabled

▼

i

Security groups*

Contai-7429

Edit

i

Auto-assign public IP

ENABLED

i

► Advanced Options

Desired task status: Running Stopped									
Filter in this page		Launch type ALL		< 1-1 >		Page size 50			
<input type="checkbox"/>	Task	Task defin...	Container ...	Last status	Desired st...	Started By	Group	Launch ty...	Platform v...
<input type="checkbox"/>	ea72eb6b-...	Containero...	--	PENDING	RUNNING		family:Cont...	FARGATE	1.3.0

Once up, obtain IP and check via browser:

Network

Network mode awsvpc

ENI Id [eni-06de002e907a5a073](#)

Subnet Id subnet-0636ab4d4aa5b3375

Private IP 172.31.2.144

Public IP **3.91.19.104**

Mac address 02:12:93:af:bd:6f

Containers

Container or cats

3.91.19.104

IF IT FITS, I SITS (.... in a container.... in a container)

