aws ecs demo | amazon elastic container service demo | aws container demo | aws fargate

Below topics will cover

1. aws ec2 machine creation
2. install docker on aws linux machine 2
3. create docker image
4. create iam user with programmatic access and give permission to ecs, ecr, ec2
5. configure user created in above step in linux machine
6. create ecr repository and push docker image
7. create task definition
8. create cluster
9. create service
10. demo

## **step to create ec2 machine**

go the aws console, in the search bar search for ec2

click launch instance from the top right side

next select the machine - Amazon Linux 2 AMI

in the choose instance type select - t2.micro

in the configure instance and add storage step - go with default values

give the name of your machine in the add tags section

in the configure security open port 22 and port 80 in the inbound ref screenshot



Click on the next review and launch

create new keypair - give name as per choice, download it on a local machine (key pair require to login to machine )

Download git-bash from https://git-scm.com/downloads

download for windows

go the location where key pair is saved

right-click anywhere on the free space and select git bash from here

## **install docker on aws linux machine 2**

sudo yum update -y

sudo amazon-linux-extras install docker

sudo yum install docker

sudo service docker start

sudo usermod -a -G docker ec2-user

## **create docker image**

create one file - touch Dockerfile

update below content in the Dockerfile create in above step

FROM ubuntu:18.04

# Install dependencies

RUN apt-get update && \

apt-get -y install apache2

# Install apache and write hello world message

RUN echo 'Hello World!' > /var/www/html/index.html

# Configure apache

RUN echo '. /etc/apache2/envvars' > /root/run\_apache.sh && \

echo 'mkdir -p /var/run/apache2' >> /root/run\_apache.sh && \

echo 'mkdir -p /var/lock/apache2' >> /root/run\_apache.sh && \

echo '/usr/sbin/apache2 -D FOREGROUND' >> /root/run\_apache.sh && \

chmod 755 /root/run\_apache.sh

EXPOSE 80

CMD /root/run\_apache.sh

create image

docker build -t hello-world .

search image created in the above step

docker images --filter reference=hello-world

now verify image working properly or not

docker run -t -id -p 80:80 hello-world

## **create iam user with programmatic access and give permission to ecs, ecr, ec2**

<https://www.blogger.com/blog/post/edit/4755733674328893075/6044184154345895909?hl=en>

go to aws console, search for IAM

click on create user, give name as per choice

select programmatic access, next in the policy select administrator policy

next, create ( make sure to download the excel file )

## 

## **configure user created in above step in linux machine**

aws configure

<https://www.blogger.com/blog/post/edit/4755733674328893075/6044184154345895909?hl=en>

AWS\_ACCESS\_KEY\_ID=

AWS\_SECRET\_ACCESS\_KEY=

AWS\_DEFAULT\_REGION=

## **create ecr repository and push docker image**

aws ecr create-repository --repository-name hello-repository --region ap-south-1

next tagging a repository

docker tag hello-world awsaccountnumberhere.dkr.ecr.ap-south-1.amazonaws.com/hello-repository

login

aws ecr get-login-password | docker login --username AWS --password-stdin awsaccountnumberhere.dkr.ecr.ap-south-1.amazonaws.com/hello-repository

to push docker image to ecr

docker push awsaccountnumberhere.dkr.ecr.ap-south-1.amazonaws.com/hello-repository

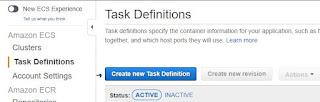
To delete

aws ecr delete-repository --repository-name hello-repository --region region --force

## **create task definition**

go to aws console - service - search for ecs

click task definition - and create a new task definition

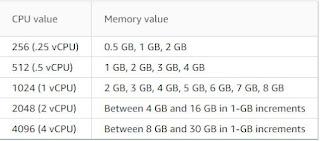


Click next and select fargate



enter task definition name, IAM role ( leave none )

task size ( need to select from below combination only)

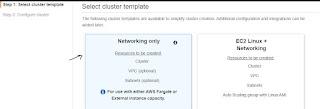


add container image - format

leave other options as default and create

## **create cluster**

Click on create a cluster, select networking only



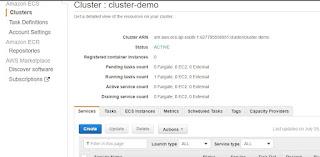
leave the other options as default

create cluster

## **create service**

select the cluster created in the above step

select the service tab



in the launch type select fargate



fill all the other details like task definition, revision, cluster name

give service name as per choice

number of task : 1

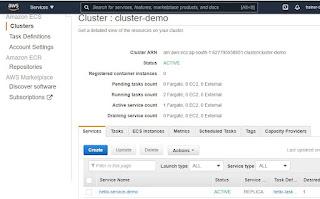
next step select the vpc , and subnet

create

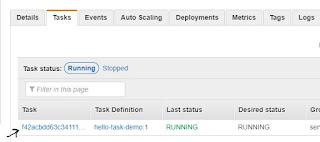
## **verify**

Click on the cluster, select service

wait for the service to get into Active state



Click on the service, then task tab, and click on the task



in the network, section copy the public Ip to verify

