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

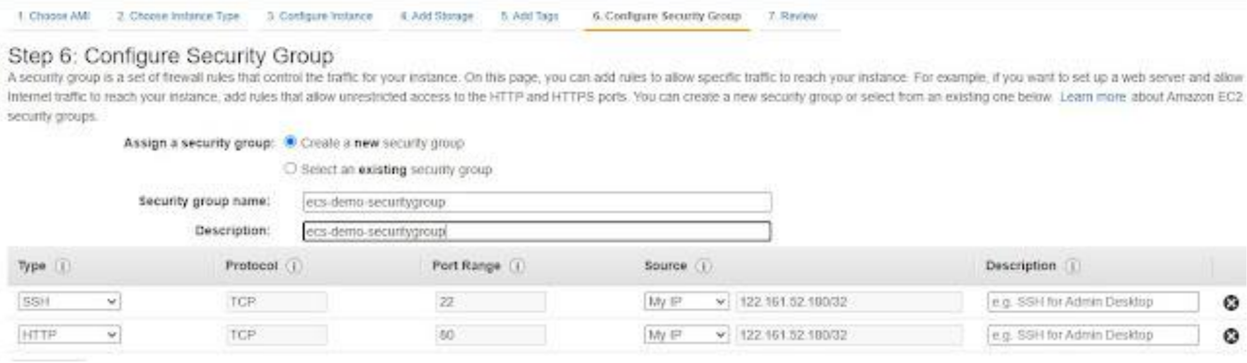
aws ecs demo | amazon elastic container service demo | aws container demo | aws fargate

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



Step 6: Configure Security Group

A security group is a set of firewall rules that control the traffic for your instance. On this page, you can add rules to allow specific traffic to reach your instance. For example, if you want to set up a web server and allow Internet traffic to reach your instance, add rules that allow unrestricted access to the HTTP and HTTPS ports. You can create a new security group or select from an existing one below. Learn more about Amazon EC2 security groups.

Assign a security group: ● Create a **new** security group
○ Select an **existing** security group

Security group name: ecs-demo-securitygroup

Description: ecs-demo-securitygroup

Type	Protocol	Port Range	Source		Description
SSH	TCP	22	My IP	122.161.52.180/32	e.g. SSH for Admin Desktop
HTTP	TCP	80	My IP	122.161.52.180/32	e.g. SSH for Admin Desktop

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

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
```

```
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
```

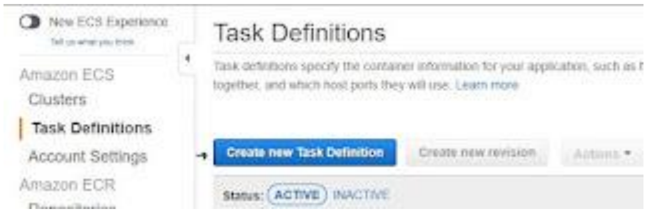
To delete

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

create task definition

go to aws console - service - search for ecs

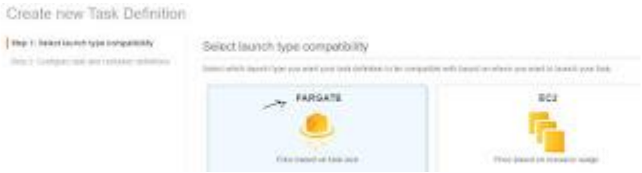
click task definition - and create a new task definition



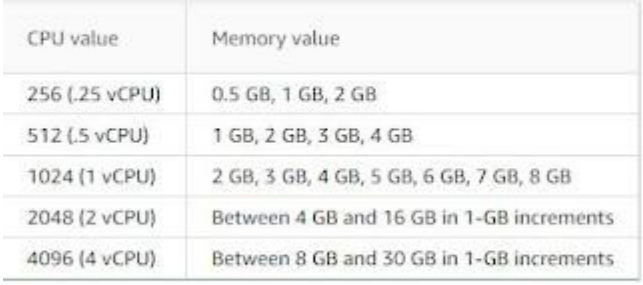
Task Definitions

Task definitions specify the container information for your application, such as t together, and which host ports they will use. Learn more

Click next and select fargate


Create new Task Definition
Step 1: Select launch type compatibility
Step 2: Configure task and container definitions
Select launch type compatibility
FARGATE
EC2

enter task definition name, IAM role (leave none)
task size (need to select from below combination only)

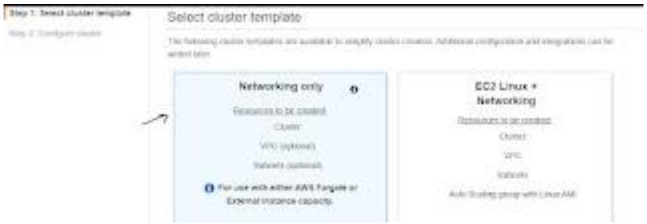


CPU value	Memory value
256 (.25 vCPU)	0.5 GB, 1 GB, 2 GB
512 (.5 vCPU)	1 GB, 2 GB, 3 GB, 4 GB
1024 (1 vCPU)	2 GB, 3 GB, 4 GB, 5 GB, 6 GB, 7 GB, 8 GB
2048 (2 vCPU)	Between 4 GB and 16 GB in 1-GB increments
4096 (4 vCPU)	Between 8 GB and 30 GB in 1-GB increments

add container image - format
leave other options as default and create

create cluster

Click on create a cluster, select networking only



Step 1: Select cluster template

Select cluster template

Networking only

Resources to be created

Cluster

VPC (optional)

Subnets (optional)

For use with either AWS Fargate or External instance capacity.

EC2 Linux + Networking

Resources to be created

Cluster

VPC

Subnets

Auto Scaling group with Linux AMI

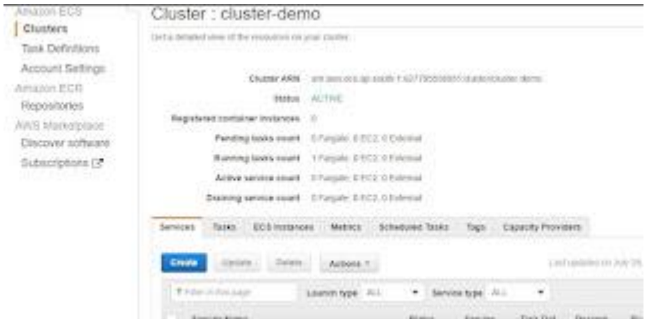
leave the other options as default

create cluster

create service

select the cluster created in the above step

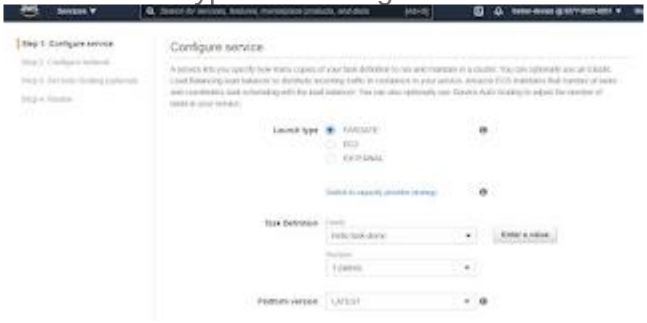
select the service tab



Cluster : cluster-demo

Status	ACTIVE
Registered container instances	0
Pending tasks count	0 Fargate, 0 EC2, 0 External
Running tasks count	1 Fargate, 0 EC2, 0 External
Active service count	0 Fargate, 0 EC2, 0 External
Draining service count	0 Fargate, 0 EC2, 0 External

in the launch type select fargate



Step 1: Configure service

Configure service

Launch type ● FARGATE ○ EC2 ○ EXTERNAL

Task Definition: hello-task-demo

Revision: 1 (latest)

Platform version: LATEST

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

give service name as per choice

number of task : 1

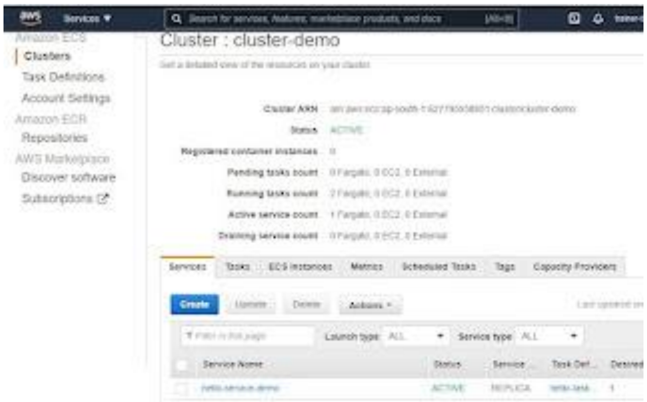
next step select the vpc , and subnet

create

verify

Click on the cluster, select service

wait for the service to get into Active state



Cluster : cluster-demo

Get a detailed view of the resources on your cluster

Cluster ARN arn:aws:ecs:ap-south-1:[illegible]:cluster/cluster-demo

Status ACTIVE

Registered container instances 0

Pending tasks count 0 Fargate, 0 EC2, 0 External

Running tasks count 2 Fargate, 0 EC2, 0 External

Active service count 1 Fargate, 0 EC2, 0 External

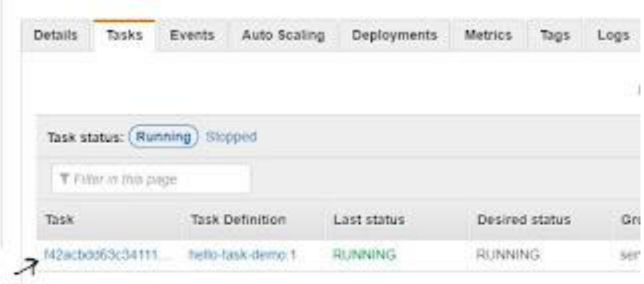
Draining service count 0 Fargate, 0 EC2, 0 External

Launch type ALL

Service type ALL

Service Name	Status	Service ...	Task Def...	Desired
hello-service-demo	ACTIVE	REPLICA	hello-task	1

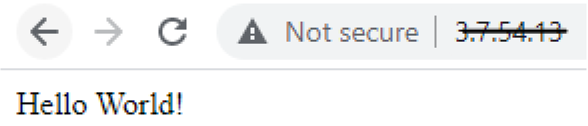
Click on the service, then task tab, and click on the task



Task status: Running Stopped

Task	Task Definition	Last status	Desired status	Gr
f42acbdd63c34111...	hello-task-demo:1	RUNNING	RUNNING	ser

in the network, section copy the public Ip to verify



Not secure | ~~3.7.54.13~~

Hello World!