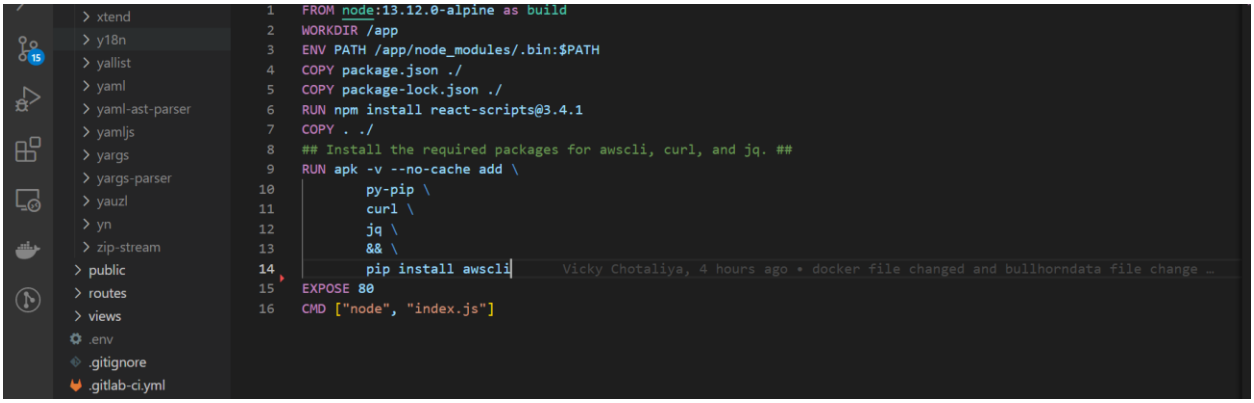
Building docker image and deploying on ecr

1. Prerequisite
   - To build docker image locally we require nodejs,npm and Docker Desktop in our machine.

1. Creating docker file in node js code

To create a docker image we created a .dockerfile in the root of directory.



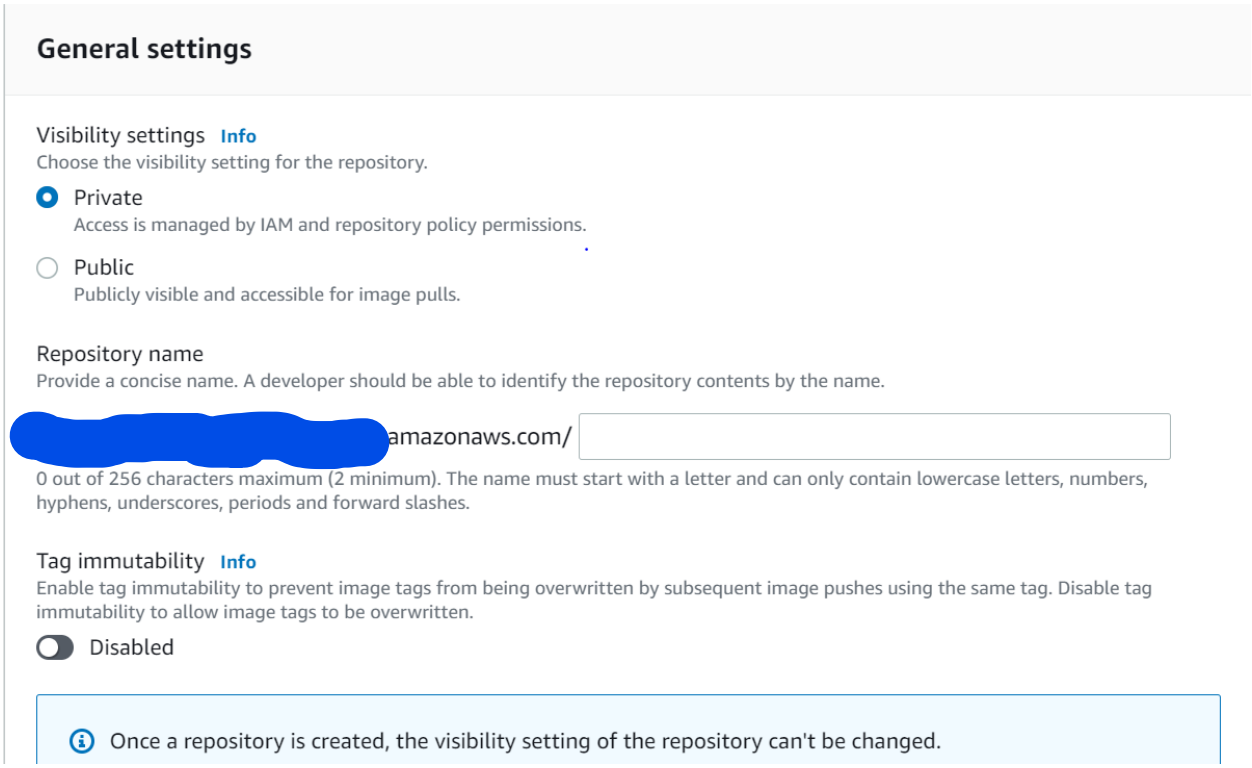
2. Building a docker image

To build docker images we run **docker build -t <image-name> .**

For Example: docker build -t dummy_dealcalc .

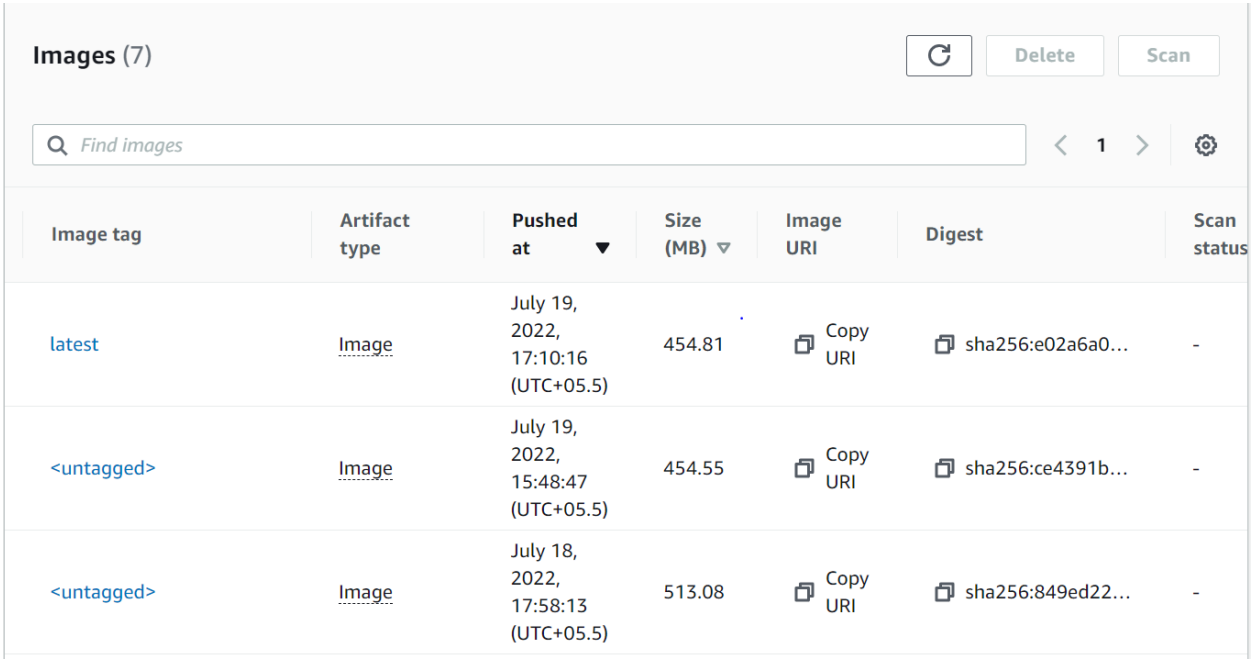
This command builds an image for us.

3. Creating registry in ecr



Write **<repository-name>** inside input.

For example, we created **dummy_dealcalc** repositories and pushed our latest image on that.



4. How to push image in repositories

Below are steps for the windows PowerShell. You can paste the command.

Make sure that you have the latest version of AWS Tools for PowerShell and Docker installed.

Use the following steps to authenticate and push an image to your repository.

1. Retrieve an authentication token and authenticate your Docker client to your registry.

   Use AWS Tools for PowerShell:

   aws ecr get-login-password --region us-east-1 | docker login --username AWS --password-stdin <account._id>.dkr.ecr.us-east-1.amazonaws.com

   For example: aws ecr get-login-password --region us-east-1 | docker login --username AWS --password-stdin [redacted]dkr.ecr.us-east-1.amazonaws.com

2. Build your Docker image using the following command.

   docker build -t <image-name>.

   For example: docker build -t dummy_dealcalc .

3. After the build completes, tag your image so you can push the image to this repository:

   docker tag <image-name>: latest [redacted]r.ecr.us-east-1.amazonaws.com/ <image-name>: latest

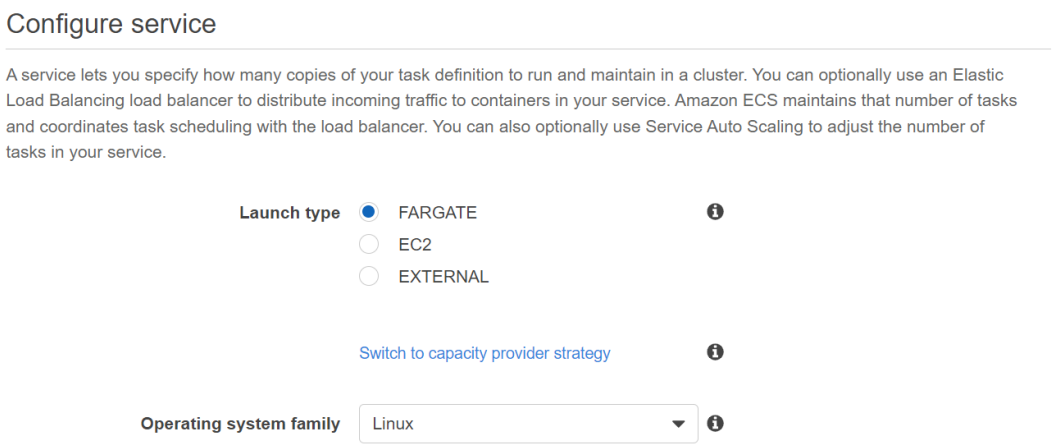
   For example: docker tag dummy_dealcalc:latest [redacted].dkr.ecr.us-east-1.amazonaws.com/dummy_dealcalc:latest

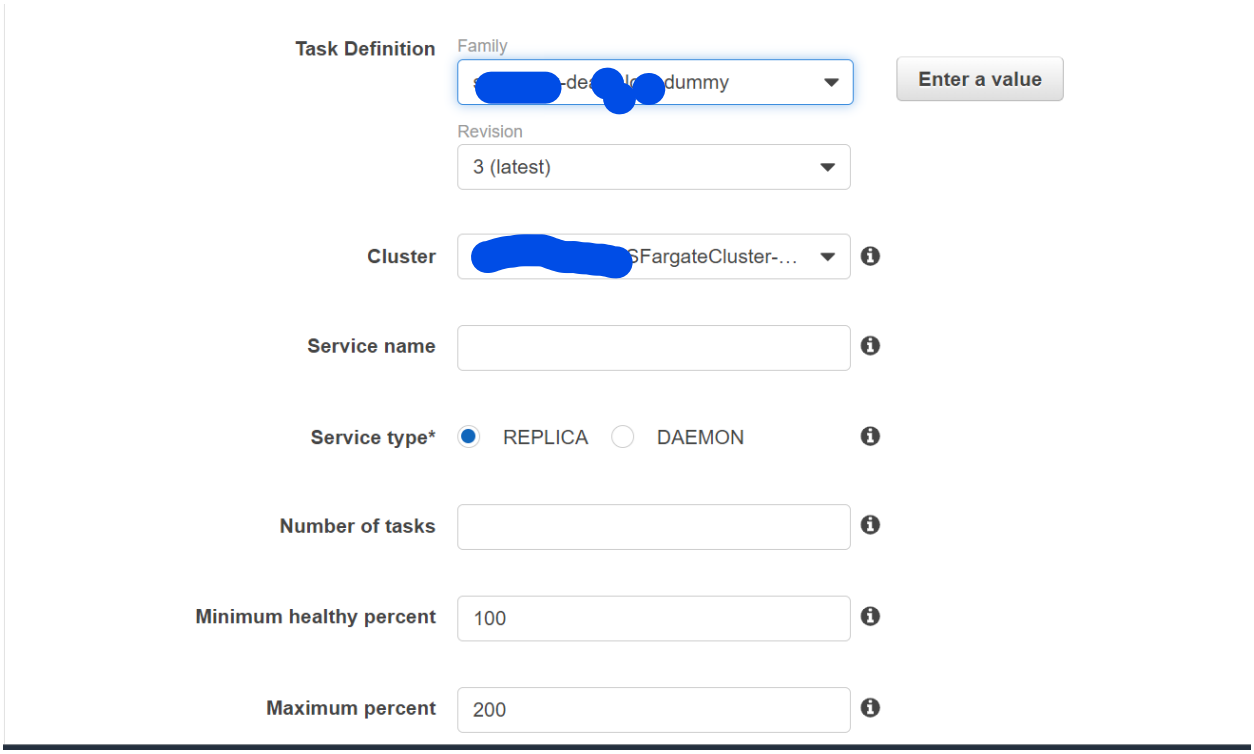
4. Run the following command to push this image to your newly created AWS repository:

docker push <account_id>.dkr.ecr.us-east-1.amazonaws.com/< image-name>:latest

For example: docker push [redacted]dkr.ecr.us-east-1.amazonaws.com/dummy_dealcalc:latest

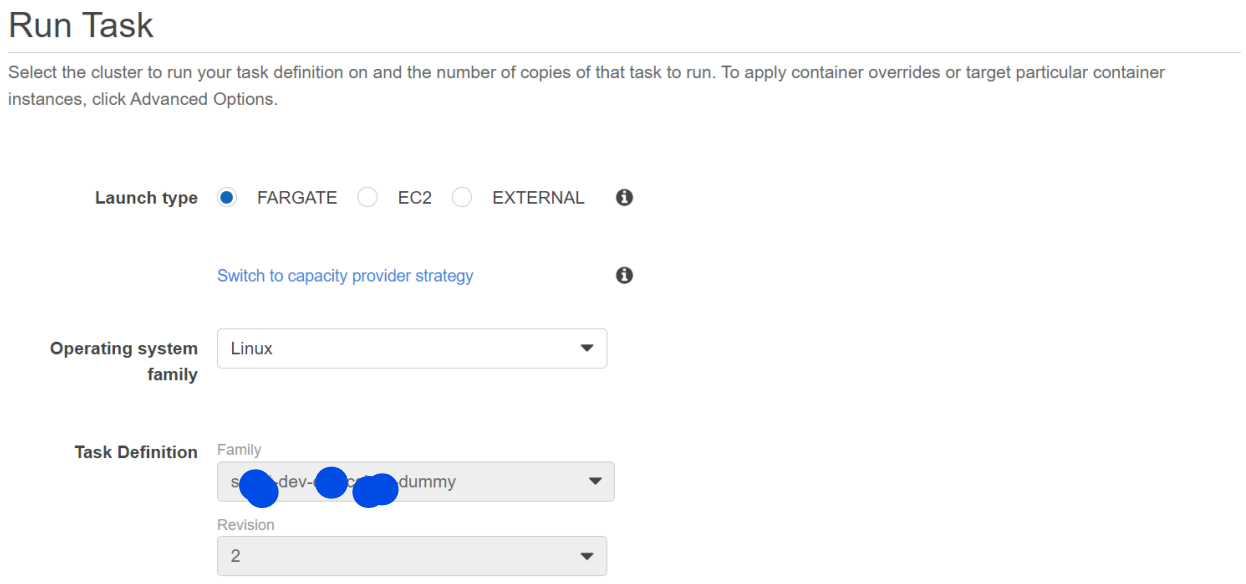
5. Creating service on elastic container service

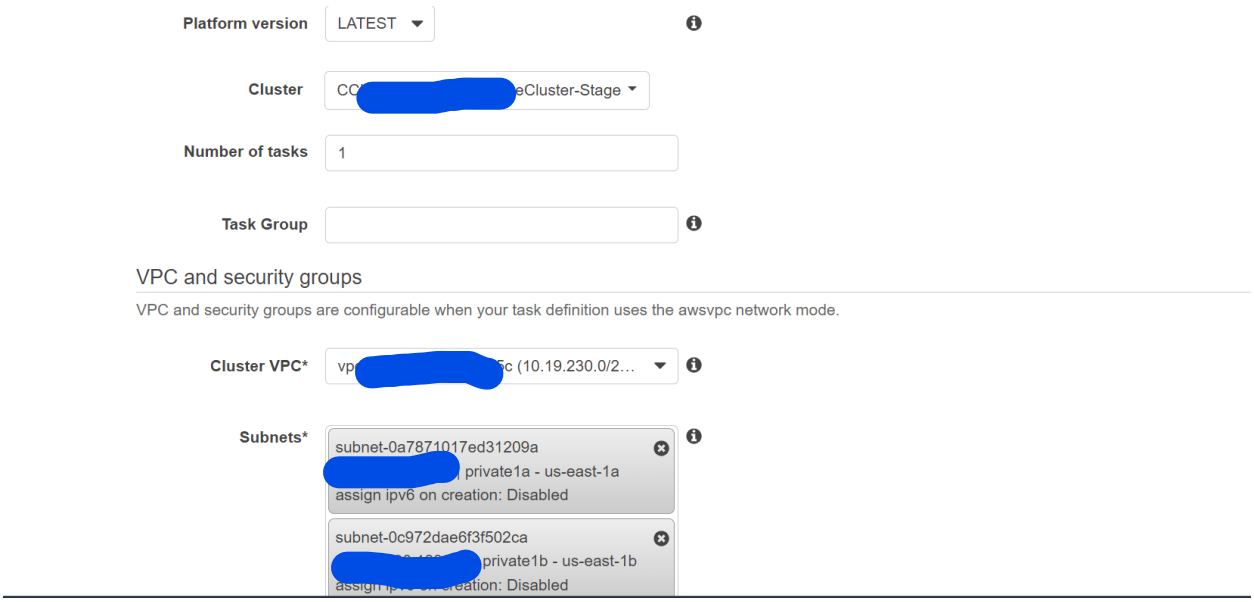


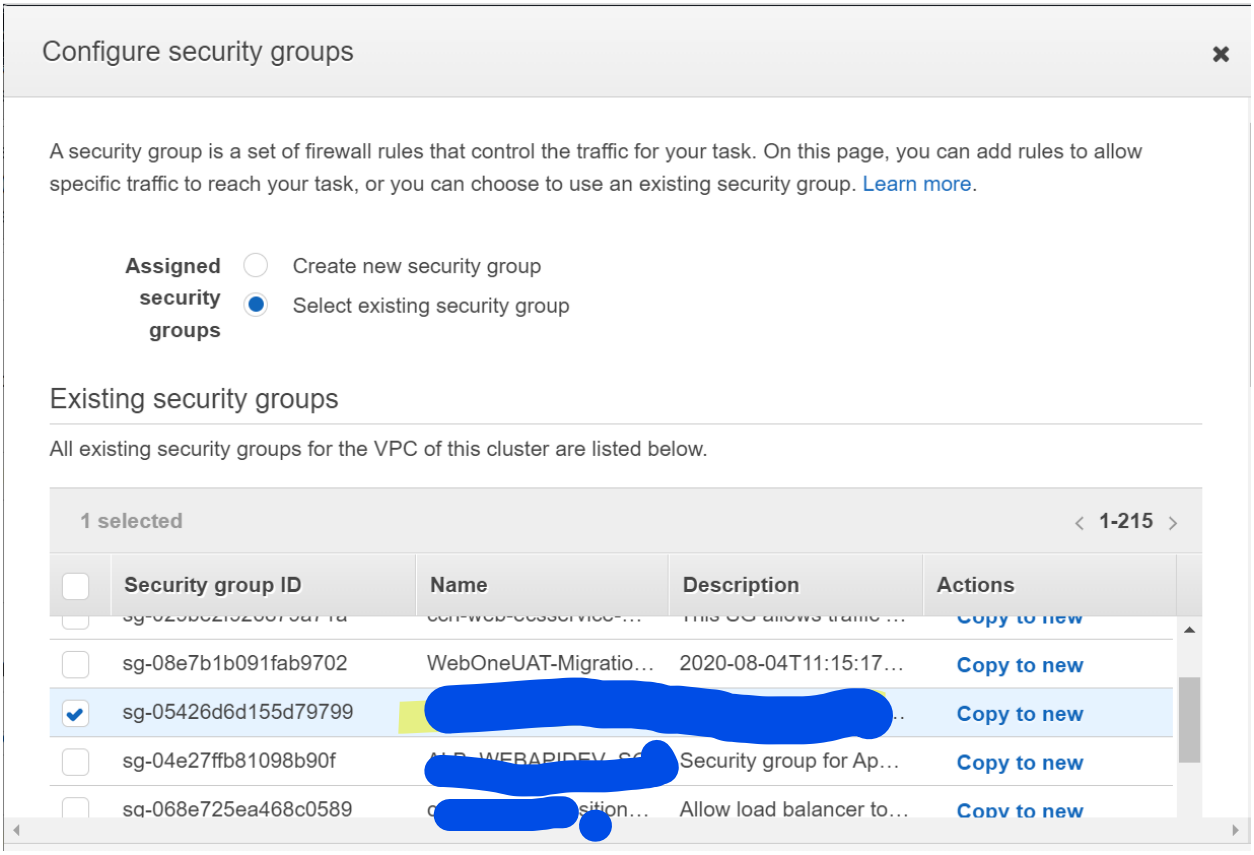


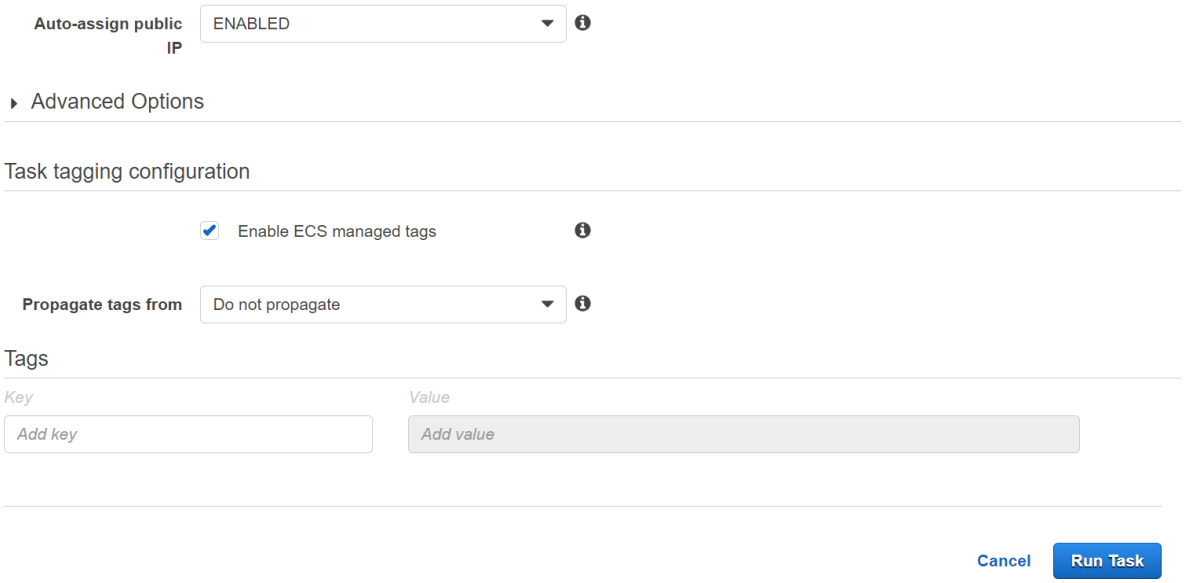
And after this click on the continue button and save service

6. Creating task




Platform version
LATEST
Cluster
eCluster-Stage
Number of tasks
1
Task Group
VPC and security groups
VPC and security groups are configurable when your task definition uses the awsvpc network mode.
Cluster VPC*
(10.19.230.0/2…
Subnets*
subnet-0a7871017ed31209a
private1a - us-east-1a
assign ipv6 on creation: Disabled
subnet-0c972dae6f3f502ca
private1b - us-east-1b


Configure security groups
A security group is a set of firewall rules that control the traffic for your task. On this page, you can add rules to allow specific traffic to reach your task, or you can choose to use an existing security group. Learn more.
Assigned security groups
Create new security group
Select existing security group
Existing security groups
All existing security groups for the VPC of this cluster are listed below.
1 selected
1-215
Security group ID
Name
Description
Actions
sg-08e7b1b091fab9702
WebOneUAT-Migratio…
2020-08-04T11:15:17…
Copy to new
sg-05426d6d155d79799
Copy to new
sg-04e27ffb81098b90f
Security group for Ap…
Copy to new
sg-068e725ea468c0589
Allow load balancer to…
Copy to new



After running dockers locally you might be facing the errors below.

7. Missing AWS_config_file error

```
err :  TypeError: Cannot read properties of undefined (reading 'authenticate')
    at DB.init (/src/app/config/db.js:21:18)
    at processTicksAndRejections (node:internal/process/task_queues:96:5)
    at async Object.connectDB (/src/app/api/modules/cache/repository/ItemCategoriesRepository.js:7:17)
    at async Object.fetchDcItemCategories (/src/app/api/modules/cache/service/ItemCategoriesService.js:10:11)
    at async Object.runAllFunctions (/src/CallingFunctions.js:9:9)
DB connection closed
Init called in db
Error: connect ECONNREFUSED 169.254.169.254:80
    at TCPConnectWrap.afterConnect [as oncomplete] (node:net:1187:16)
    at TCPConnectWrap.callbackTrampoline (node:internal/async_hooks:130:17) {
  message: 'Missing credentials in config, if using AWS_CONFIG_FILE, set AWS_SDK_LOAD_CONFIG=1',
  errno: -111,
  code: 'CredentialsError',
  syscall: 'connect',
  address: '169.254.169.254',
  port: 80,
  time: 2022-06-30T10:19:27.859Z,
  originalError: {
    message: 'Could not load credentials from any providers',
```

```
AWS_SDK_LOAD_CONFIG = 1;
```

To resolve this error, paste the above line on .env file.

8. Missing .aws file error

```
/app/node_modules/aws-sdk/lib/node_loader.js:129
        if (fileInfo.isConfig) throw err;
                               ^

Error: ENOENT: no such file or directory, open '/root/.aws/config'
    at Object.openSync (fs.js:462:3)
    at Object.readFileSync (fs.js:364:35)
    at Object.readFileSync (/app/node_modules/aws-sdk/lib/util.js:95:26)
    at IniLoader.parseFile (/app/node_modules/aws-sdk/lib/shared-ini/ini-loader.js:6:47)
    at IniLoader.loadFrom (/app/node_modules/aws-sdk/lib/shared-ini/ini-loader.js:56:30)
    at getRegion (/app/node_modules/aws-sdk/lib/node_loader.js:125:32)
    at Config.region (/app/node_modules/aws-sdk/lib/node_loader.js:182:18)
    at Config.set (/app/node_modules/aws-sdk/lib/config.js:524:39)
    at Config.<anonymous> (/app/node_modules/aws-sdk/lib/config.js:359:12)
    at Config.each (/app/node_modules/aws-sdk/lib/util.js:512:32) {
  errno: -2,
  syscall: 'open',
  code: 'ENOENT',
```

To resolve this error, push your image on ecr and run on aws fargate.

9. Run Docker image locally

Command: **docker run -v <your .aws folder located path>:/root/.aws <image-name>: latest**

For Example: **docker run -v C:/Users/vchotali/.aws:/root/.aws dummy_dealcalc:latest**