

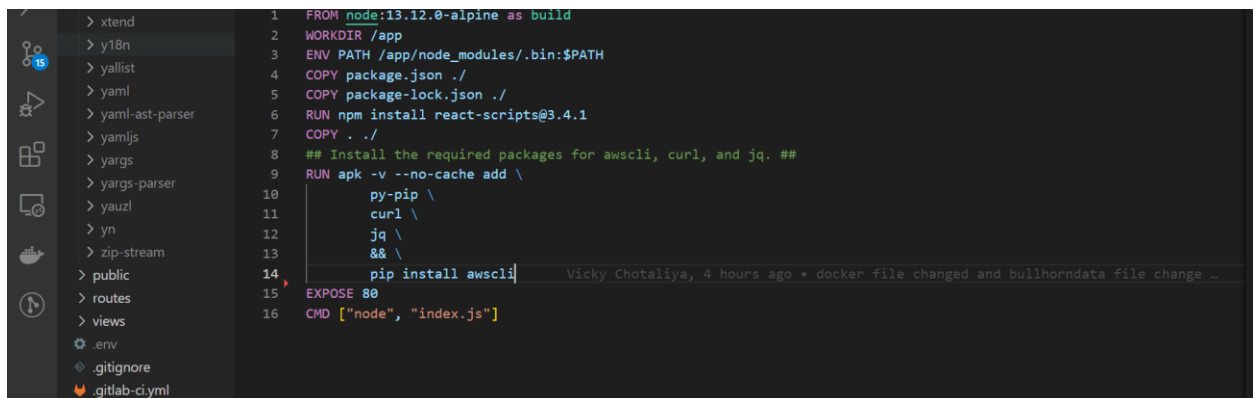
## Building docker image and deploying on ecr

### 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.



```
1 FROM node:13.12.0-alpine as build
2 WORKDIR /app
3 ENV PATH /app/node_modules/.bin:$PATH
4 COPY package.json ./
5 COPY package-lock.json ./
6 RUN npm install react-scripts@3.4.1
7 COPY . ./
8 ## Install the required packages for awscli, curl, and jq. ##
9 RUN apk -v --no-cache add \
10     py-pip \
11     curl \
12     jq \
13     && \
14     pip install awscli
15 EXPOSE 80
16 CMD ["node", "index.js"]
```

The screenshot shows a code editor with a file explorer on the left and a code editor on the right. The file explorer shows a directory structure with files like .env, .gitignore, .gitlab-ci.yml, and folders like public, routes, views, xtend, y18n, yallist, yaml, yml-ast-parser, ymljs, yargs, yargs-parser, yauzl, yn, zip-stream. The code editor shows a Dockerfile with 16 lines of instructions. Line 14 has a comment from Vicky Chotaliya. Line 15 has a red arrow pointing to it.

### 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

## General settings

### Visibility settings [Info](#)

Choose the visibility setting for the repository.

☒ Private

Access is managed by IAM and repository policy permissions.

☐ Public

Publicly visible and accessible for image pulls.

### Repository name

Provide a concise name. A developer should be able to identify the repository contents by the name.

0 out of 256 characters maximum (2 minimum). The name must start with a letter and can only contain lowercase letters, numbers, hyphens, underscores, periods and forward slashes.

### Tag immutability [Info](#)

Enable tag immutability to prevent image tags from being overwritten by subsequent image pushes using the same tag. Disable tag immutability to allow image tags to be overwritten.

☐ Disabled

 Once a repository is created, the visibility setting of the repository can't be changed.

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

For example, we created **dummy\_dealcalc** repositories and pushed our latest image on that.

Images (7)

Find images

< 1 >

Refresh

Delete

Scan

Image tag	Artifact type	Pushed at ▼	Size (MB) ▼	Image URI	Digest	Scan status
latest	<a href="#">Image</a>	July 19, 2022, 17:10:16 (UTC+05.5)	454.81	<div></div> <div>Copy URI</div>	<div></div> <div>sha256:e02a6a0...</div>	-
<untagged>	<a href="#">Image</a>	July 19, 2022, 15:48:47 (UTC+05.5)	454.55	<div></div> <div>Copy URI</div>	<div></div> <div>sha256:ce4391b...</div>	-
<untagged>	<a href="#">Image</a>	July 18, 2022, 17:58:13 (UTC+05.5)	513.08	<div></div> <div>Copy URI</div>	<div></div> <div>sha256:849ed22...</div>	-

## 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].dkr.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 <account\_id>.dkr.ecr.us-east-1.amazonaws.com/dummy\_dealcalc:latest**


## 5. Creating service on elastic container service

### Configure service

A service lets you specify how many copies of your task definition to run and maintain in a cluster. You can optionally use an Elastic Load Balancing load balancer to distribute incoming traffic to containers in your service. Amazon ECS maintains that number of tasks and coordinates task scheduling with the load balancer. You can also optionally use Service Auto Scaling to adjust the number of tasks in your service.

Launch type ☒ FARGATE   
☐ EC2  
☐ EXTERNAL

[Switch to capacity provider strategy](#) 

Operating system family Linux ▼ 

**Task Definition**

Family  
[redacted]-dev-ec2-dummy ▼

Revision  
3 (latest) ▼

**Cluster** [redacted] FargateCluster-... ▼ ⓘ

**Service name**  ⓘ

**Service type\*** ☒ REPLICA ☐ DAEMON ⓘ

**Number of tasks**  ⓘ

**Minimum healthy percent**  ⓘ

**Maximum percent**  ⓘ

[Enter a value](#)

And after this click on the continue button and save service

## 6. Creating task

### Run Task

Select the cluster to run your task definition on and the number of copies of that task to run. To apply container overrides or target particular container instances, click Advanced Options.

**Launch type** ☒ FARGATE ☐ EC2 ☐ EXTERNAL ⓘ

[Switch to capacity provider strategy](#) ⓘ

**Operating system family**  ▼

**Task Definition**

Family  
[redacted]-dev-ec2-dummy ▼

Revision  
2 ▼

Platform version LATEST



Cluster CO-...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\* vp-...c (10.19.230.0/2...



Subnets\*



subnet-0a7871017ed31209a  
private1a - us-east-1a  
assign ipv6 on creation: Disabled  
subnet-0c972dae6f3f502ca  
private1b - us-east-1b  
assign ipv6 on creation: Disabled

## 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 ☐ Create new security group  
security ☒ Select existing security group  
groups

### 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
<input type="checkbox"/>	sg-025b021920070a71d	con-web-ecsservice...	This SG allows traffic ...	<a href="#">Copy to new</a>
<input type="checkbox"/>	sg-08e7b1b091fab9702	WebOneUAT-Migratio...	2020-08-04T11:15:17...	<a href="#">Copy to new</a>
<input checked="" type="checkbox"/>	sg-05426d6d155d79799	...	...	<a href="#">Copy to new</a>
<input type="checkbox"/>	sg-04e27ffb81098b90f	ALB-WEBAPIDEV-SG	Security group for Ap...	<a href="#">Copy to new</a>
<input type="checkbox"/>	sg-068e725ea468c0589	...	Allow load balancer to...	<a href="#">Copy to new</a>

Auto-assign public IP

ENABLED

Advanced Options

Task tagging configuration

☒ Enable ECS managed tags

Propagate tags from

Do not propagate

Tags

Key	Value
<input type="text" value="Add key"/>	<input type="text" value="Add value"/>

Cancel
Run Task

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.fetchDeItemCategories (/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`