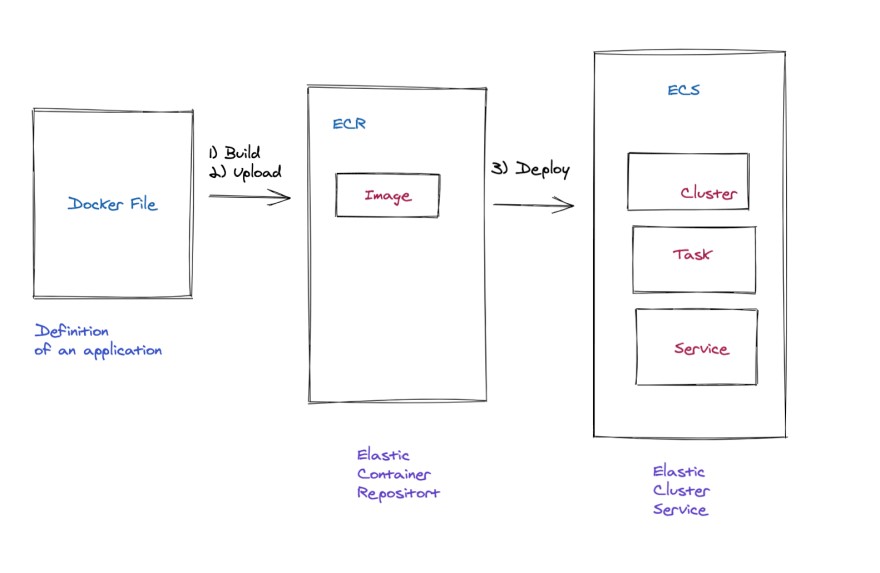
**Project Deploying Docker containers on ECS**

**Name:** Niranjan Chavan

**Email:**Niranjan\_Chavan@epam.com

**Deploying Docker containers on ECS**



**Creating Docker Image on ec2 instance:** sudo yum update

aws configure list

mkdir project

sudo yum install docker

sudo systemctl status docker

sudo systemctl enable docker

sudo systemctl start docker

sudo vi package.json

{

"name": "docker\_web\_app",

"version": "1.0.0",

"description": "Node.js on Docker",

"main": "server.js",

"scripts": {

"start": "node server.js"

},

"dependencies": {

"express": "^4.17.1"

}

}

sudo vi server.js

|  |
| --- |
| "use strict"    const express = require("express")    // Constants const PORT = 8080 const HOST = "0.0.0.0"    // App const app = express() app.get("/", (req, res) => {  res.send(`Hello World - ${new Date().toISOString()}`)  })    app.listen(PORT, HOST)  console.log(`Running on http://${HOST}:${PORT}`) |

sudo vi Dockerfile

|  |
| --- |
| FROM node:14  # Create app directory  WORKDIR /usr/src/app  COPY package\*.json ./    RUN npm install COPY . .    EXPOSE 8080    CMD [ "node", "server.js" ] |

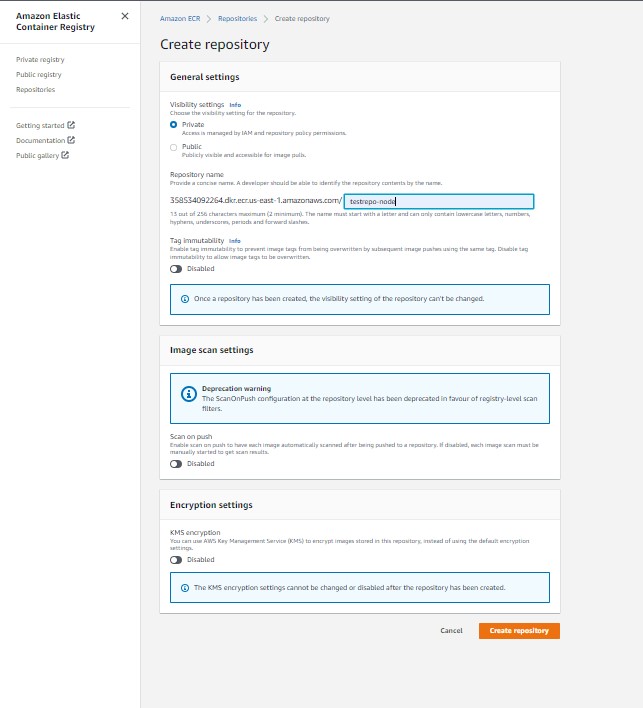
sudo docker build -t node-web-app .

sudo docker run -p 80:8080 -d node-web-app curl [http://localhost:80](http://localhost/)

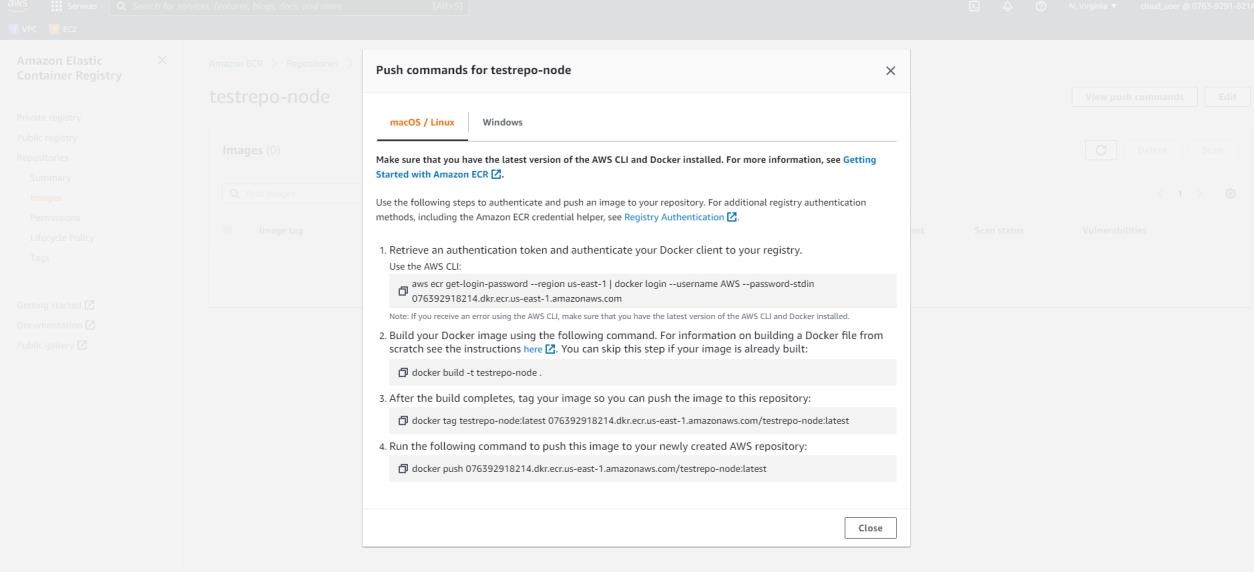
Result:

Hello World - 2021-02-11T05:06:12.739Z

**Create your Amazon ECR in the AWS Console:**



Viewing pushing Command



Uploading image using these command.

Copy the image URI: we need to keep this to create a task definition for the following steps.

# Create an ECS Cluster

Go to the ECS home page and click on the create cluster button:

Choose EC2 Linux + Networking and then click next:

Then enter the following information:

* name of the cluster: ecs01
* EC2 instance type: t3-micro
* Number of instances: 1

sThen choose:

* Default VPC
* Auto assign IP: Enabled
* Security group: default
* Choose one of the subnet

# Create a new Task definition

Click on new Task definition

Choose EC2

Then next

Choose NodeWebAppTask for the name of the task definition.

Enter 128for memory size.

Click Add Container:

Add the name of the container: NodeWebApp

Set the image URI that we have saved to add the end of the add image step

Set the port mappings 80:8080

Click create.

Create Application load Balancer attach it to service and check the DNS:

<http://my-alb-1205436972.us-east-1.elb.amazonaws.com/>

It is working 😊