**Dockerizing a Node.js web app**

**Prerequisites:**

1. Python2
2. Awscli
3. Docker

Create EC2 Instance 🡪 Here I used RHEL Amazon Machine Image

**Update yum command:**

yum update -y

**Install java and wget:**

yum install java wget -y

**Install Python2:**

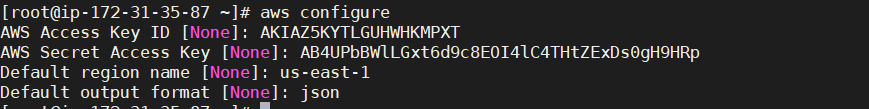
yum install python2-pip -y

alternatives --config python

**Install Awscli:**

pip2 install awscli

aws configure



**Install Docker on RHEL:**

Dnf config-manager –add repo=https://download.docker.com/linux/centos/docker-ce.repo

dnf install docker-ce-3:18.09.1-3.el7

systemctl enable docker

systemctl start docker

**Create Directory:**

mkdir DockerNodejs

cd DockerNodejs/

Create a file with the name of “**package.json”** and insert below data**:**

vi package.json

{

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

"version": "1.0.0",

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

"author": "First Last <first.last@example.com>",

"main": "server.js",

"scripts": {

"start": "node server.js"

},

"dependencies": {

"express": "^4.16.1"

}

}

Create file name with “**server.js”** and insert below data**:**

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\n');

});

app.listen(PORT, HOST);

console.log(`Running on http://${HOST}:${PORT}`);

Create Dockerfile and insert below data:

vi Dockerfile

FROM node:10

# Create app directory

WORKDIR /usr/src/app

# Install app dependencies

# A wildcard is used to ensure both package.json AND package-lock.json are copied

# where available (npm@5+)

COPY package\*.json ./

RUN npm install

# If you are building your code for production

# RUN npm ci --only=production

# Bundle app source

COPY . .

EXPOSE 8080

CMD [ "node", "server.js" ]

Create a file with the name “**.dockerignore”** and insert below data:

node\_modules

npm-debug.log

**With Docker commands:**

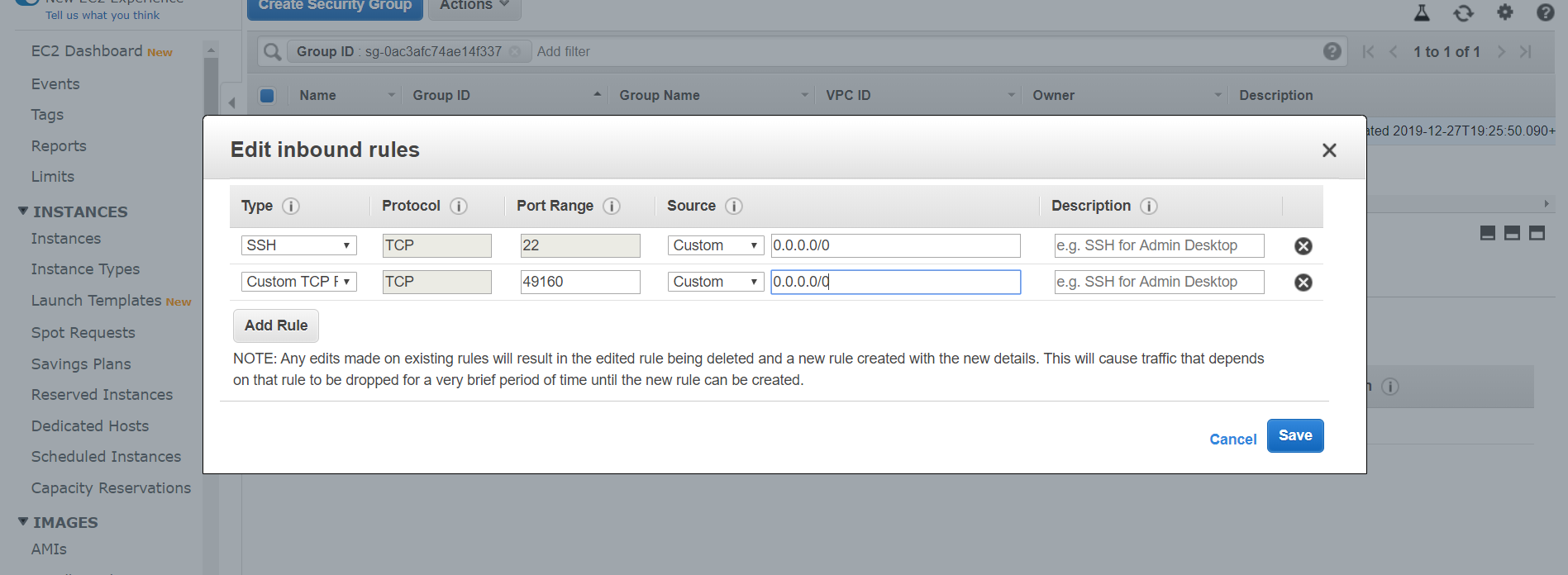
* Build Docker image

docker build -t naresh/node-web-app .

* Run Docker image

docker run --name=nodejswebapp -p 49160:8080 -d naresh/node-web-app:latest

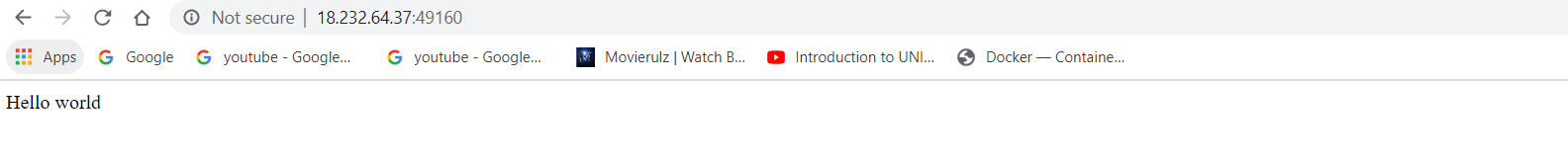
Allow PortNumber at Security-Group For EC2 Instance:



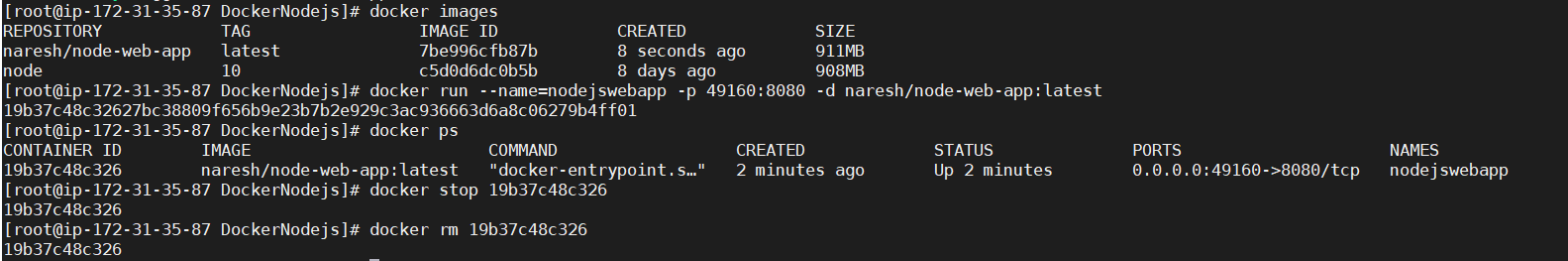
Give Public Ip address with port number

<http://18.232.64.37:49160/>

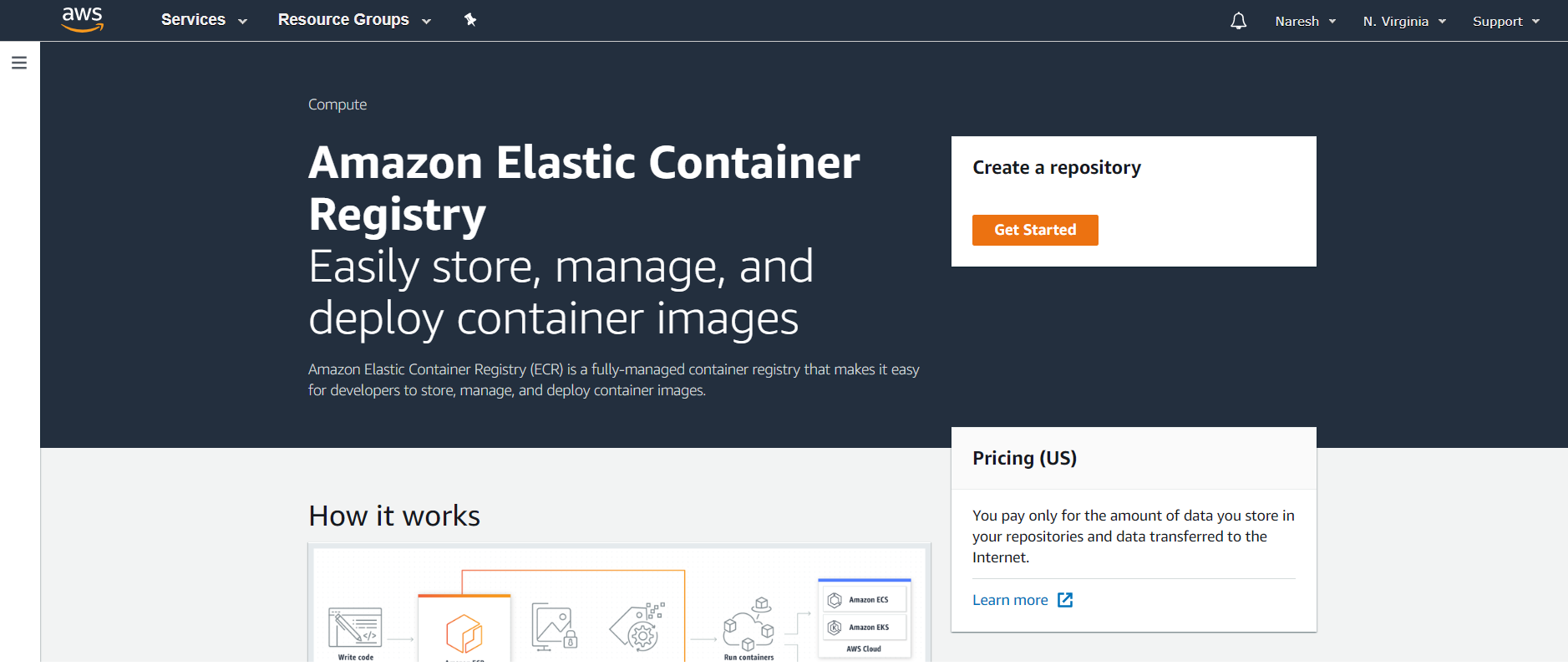
**Output with Docker:**



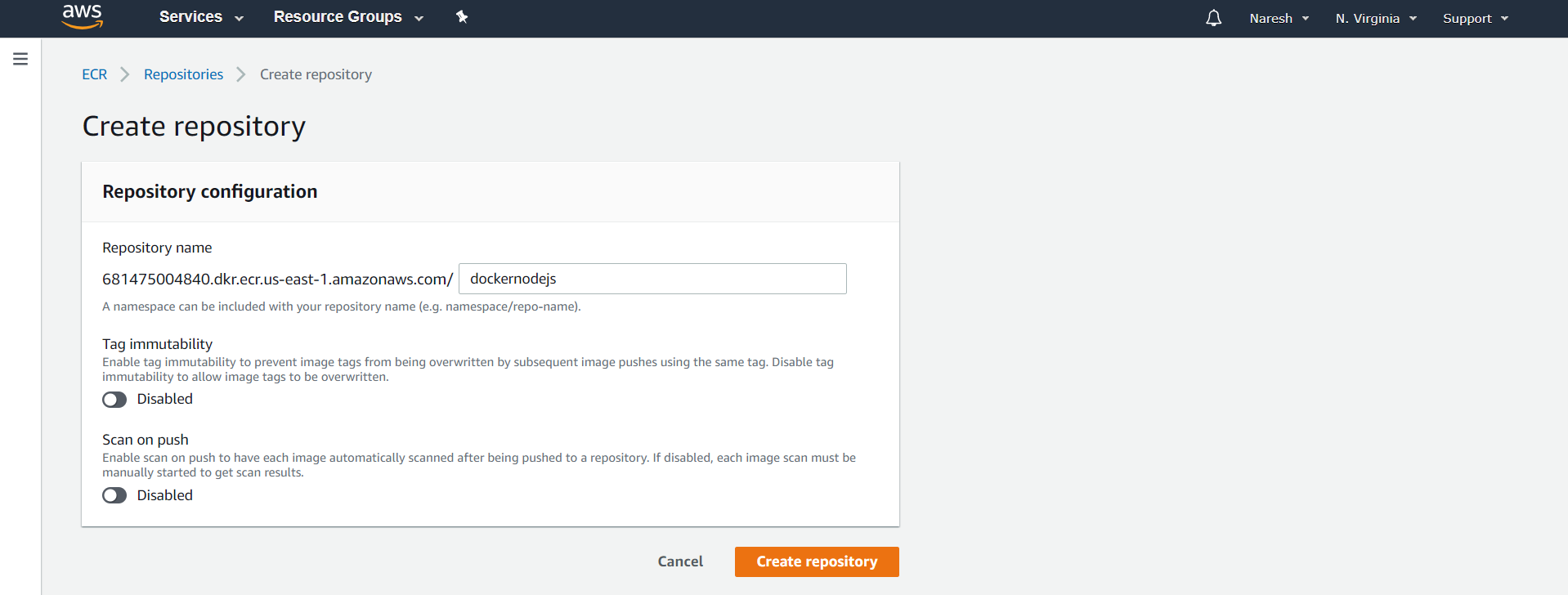
Stop and Delete Docker Container:



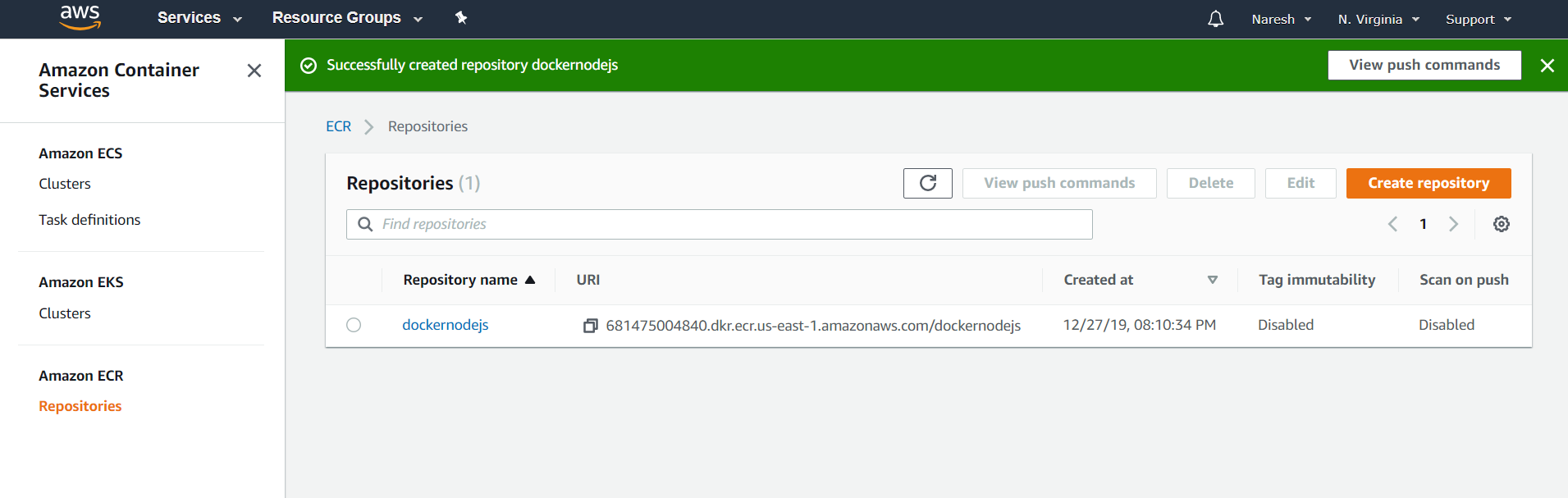
**Creating Repository on ECR using below steps:**



* Click on **Get Started**



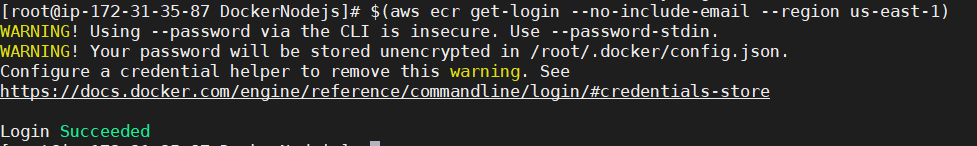
* Give Repository name as “**dockernodejs”** and Click on **Create repository**
* Below figure shows created Repository



**Steps to push docker image to ECR:**

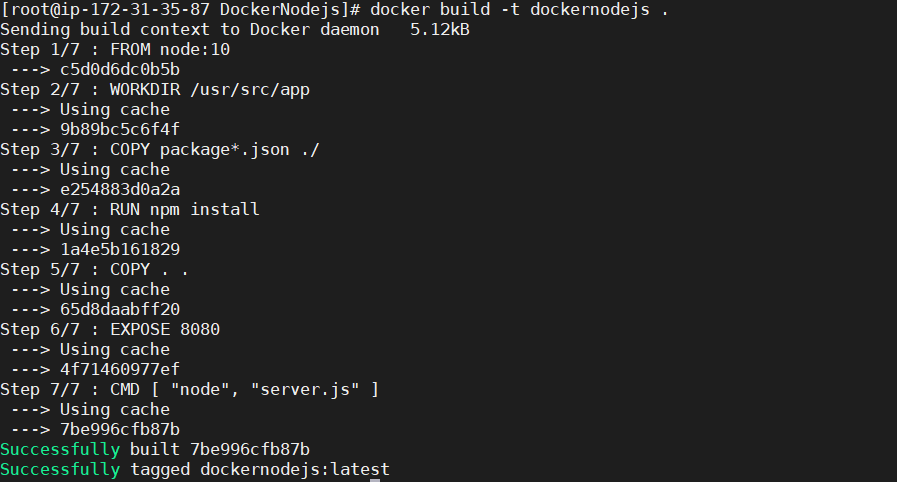
1. Login to aws ECR:

$(aws ecr get-login --no-include-email --region us-east-1)



1. Build Docker images:

docker build -t dockernodejs .



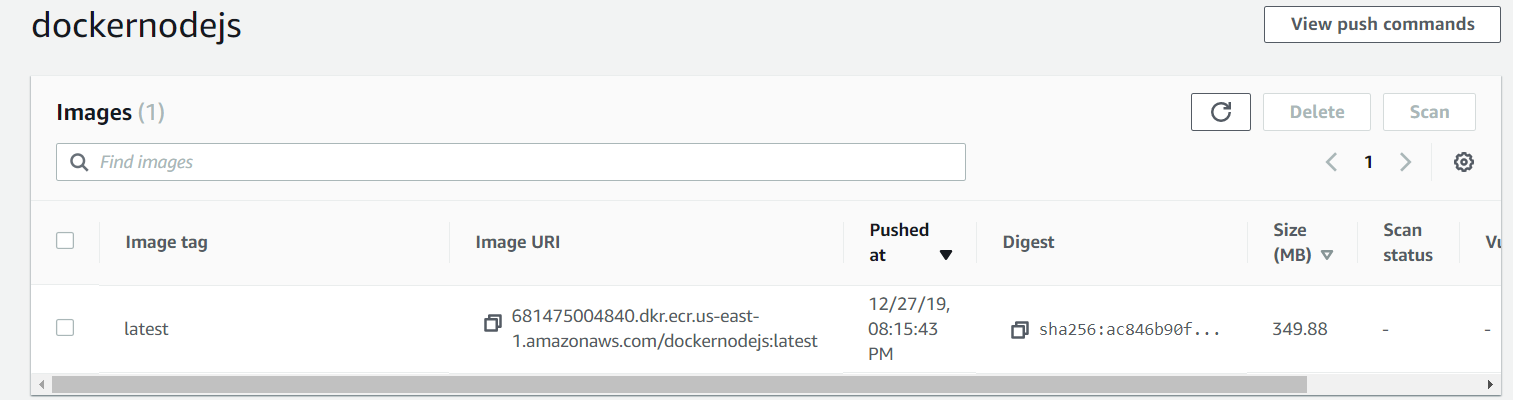
1. Tag Docker image:

Docker tag dockernodejs:latest 681475004840.dkr.ecr.us-east-1.amazonaws.com/dockernodejs:latest

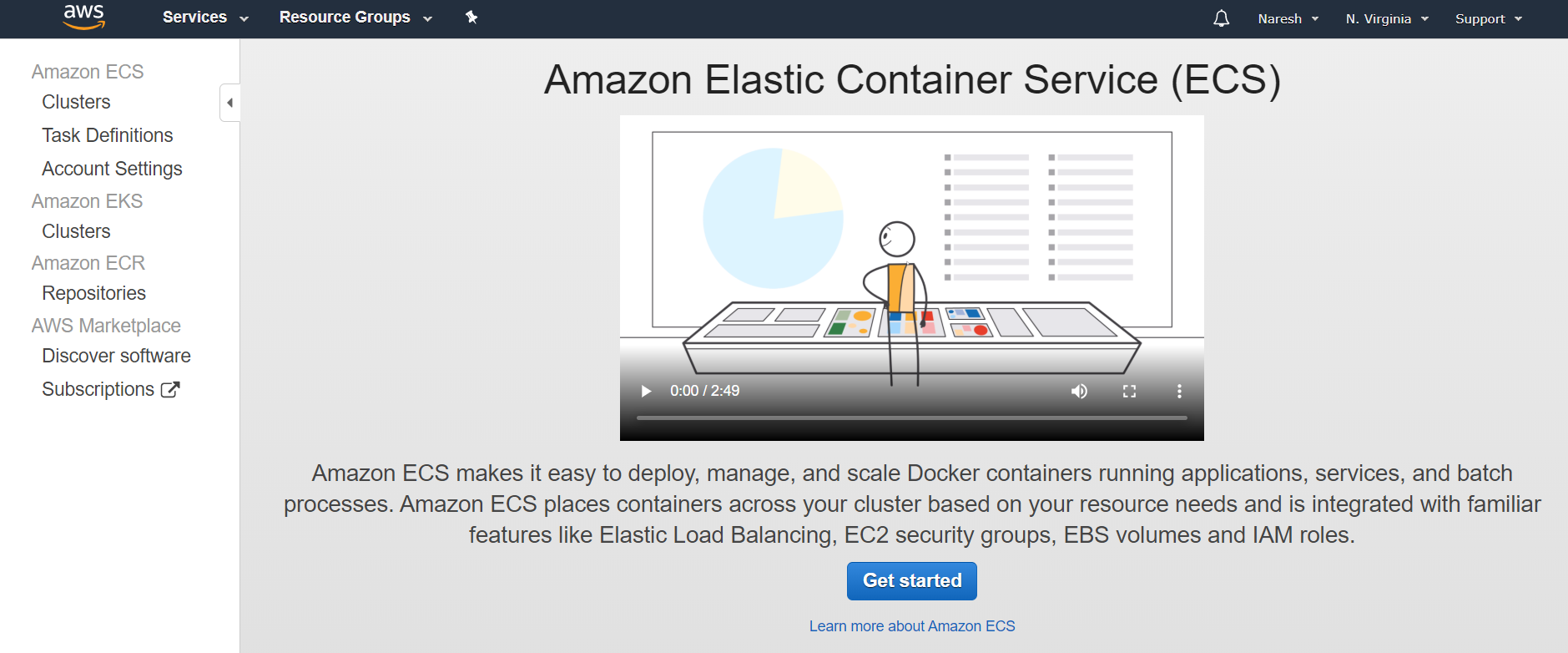
1. Push Docker image to ECR:

docker push 681475004840.dkr.ecr.us-east-1.amazonaws.com/dockernodejs:latest

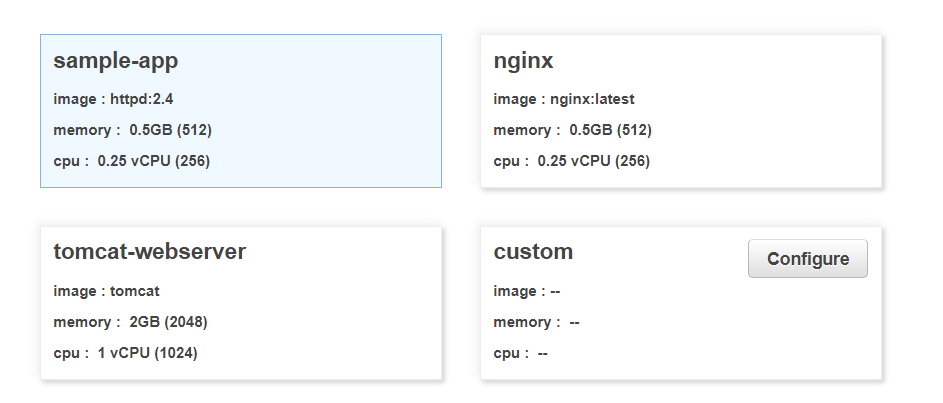
**Finally Repository stored in ECR:**



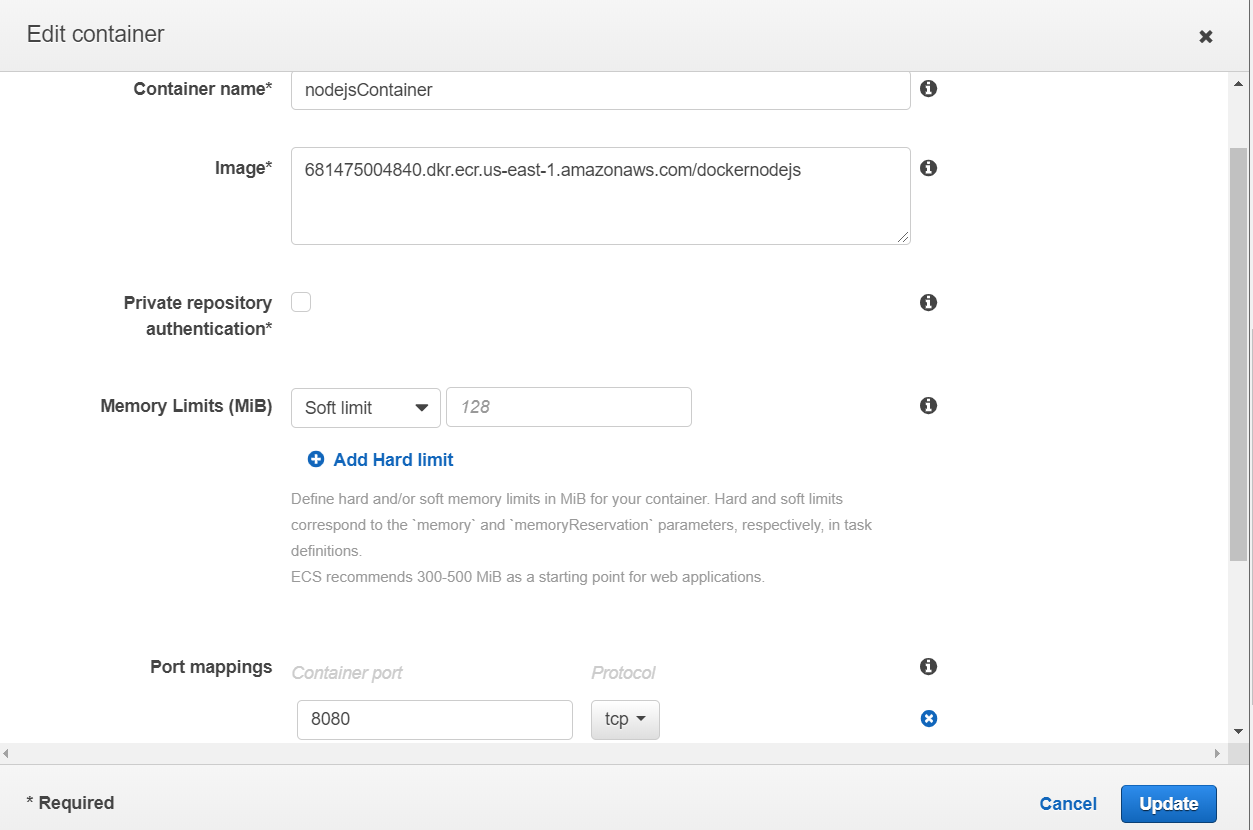
**Open ECS on AWS and Create Cluster using below steps and fill details:**



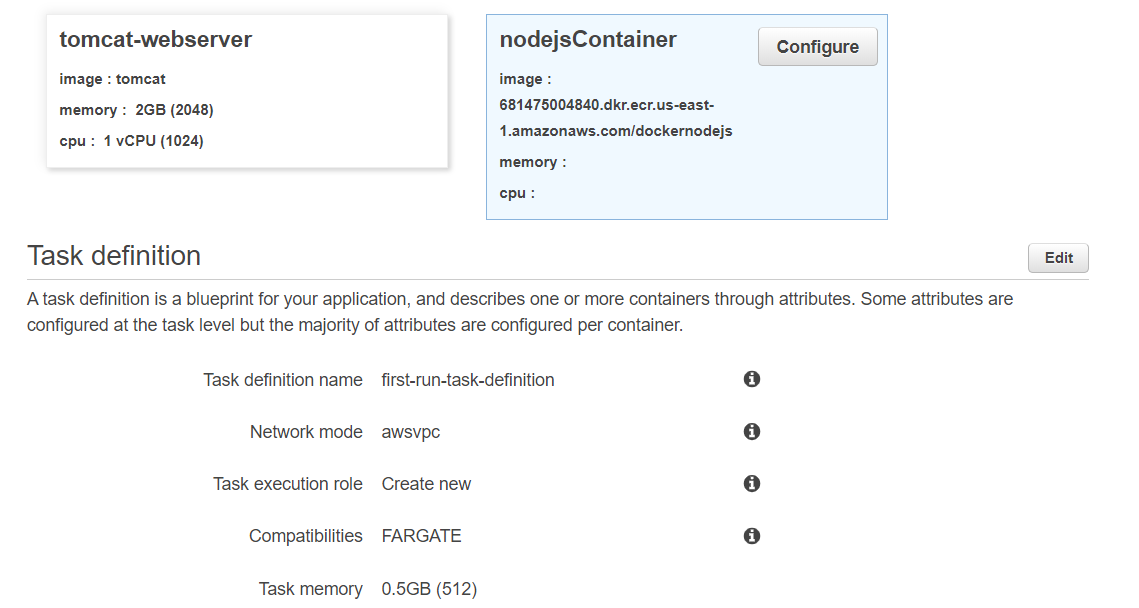
* Click on **Get started**



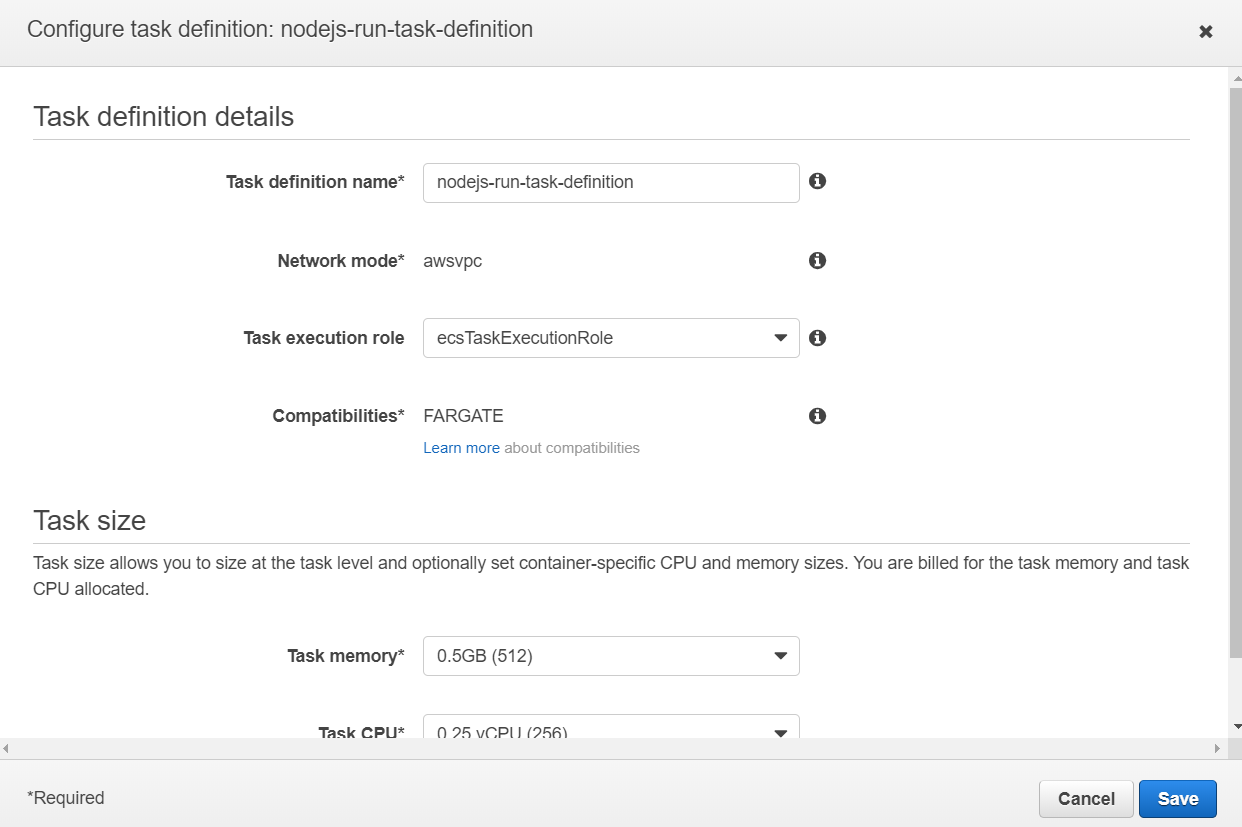
* Click on **Configure**



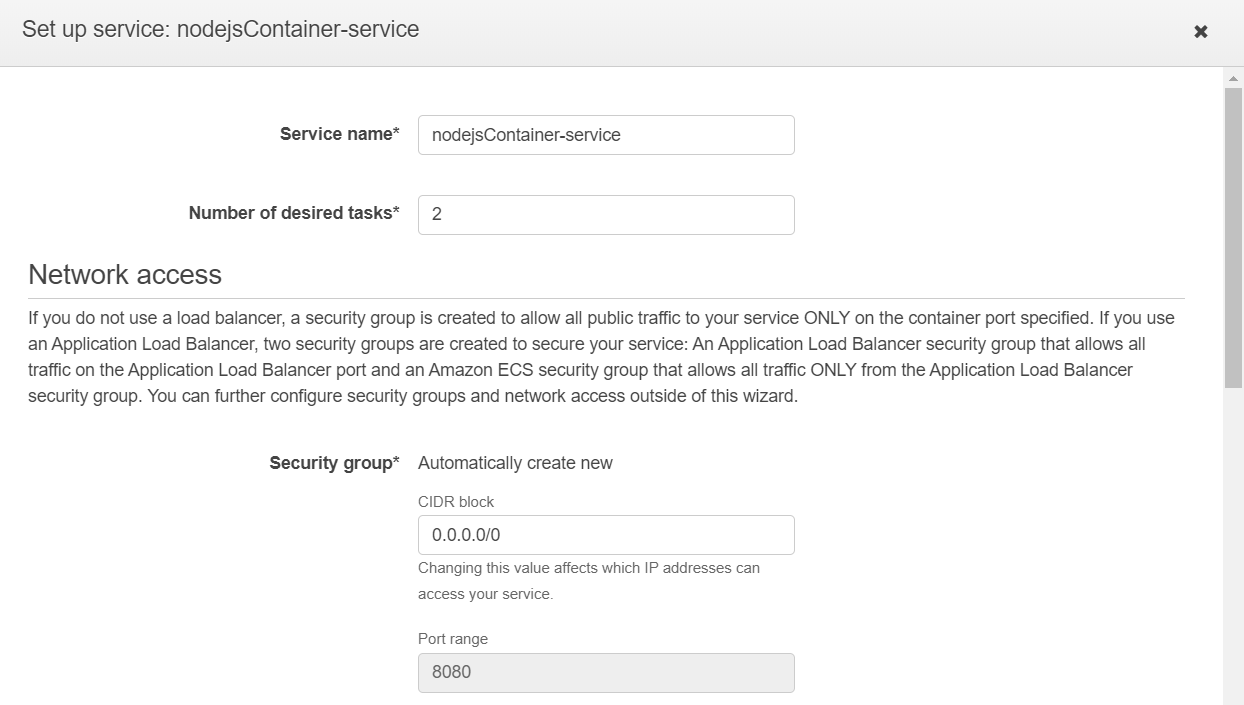
* Click on **update**

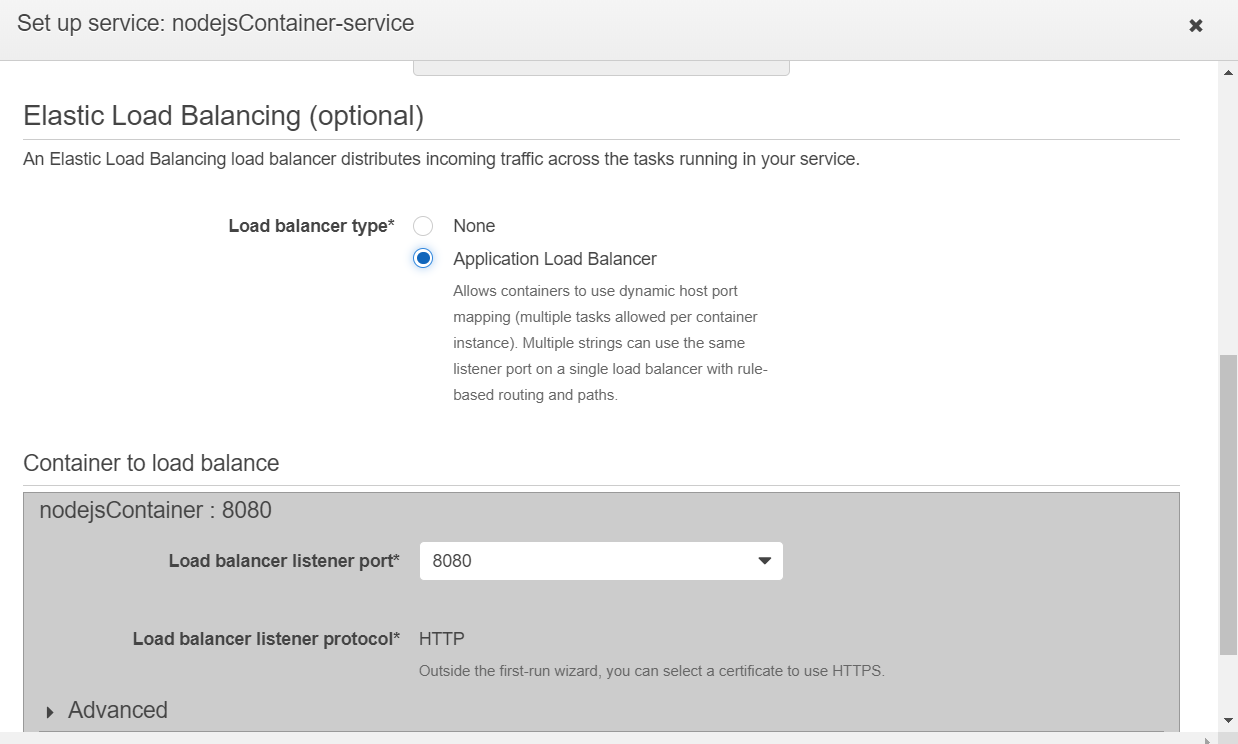


* Click on **Edit** to give Run Task defination details

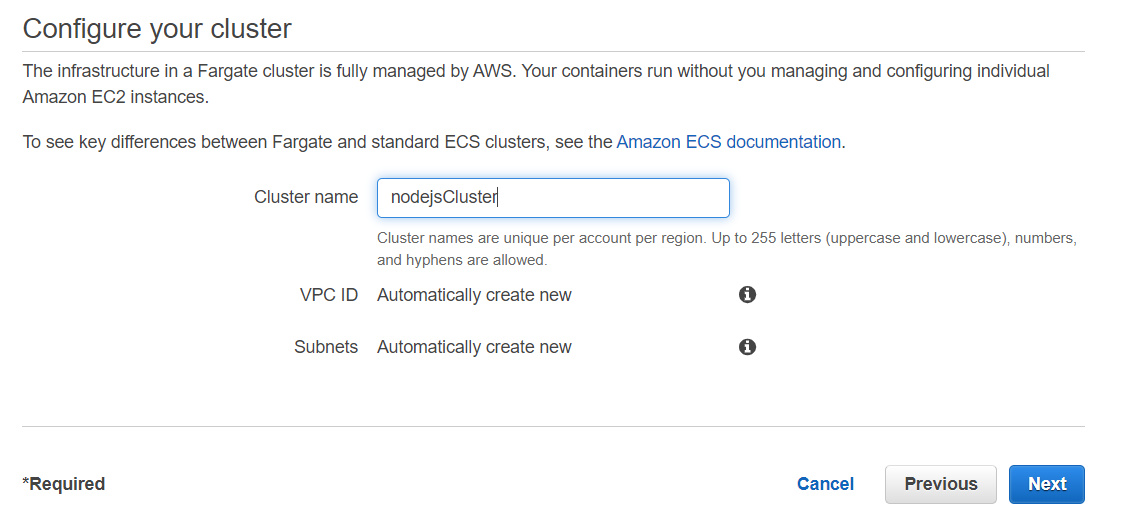


* Click on **save** and click **Next** and Edit details for Container Service



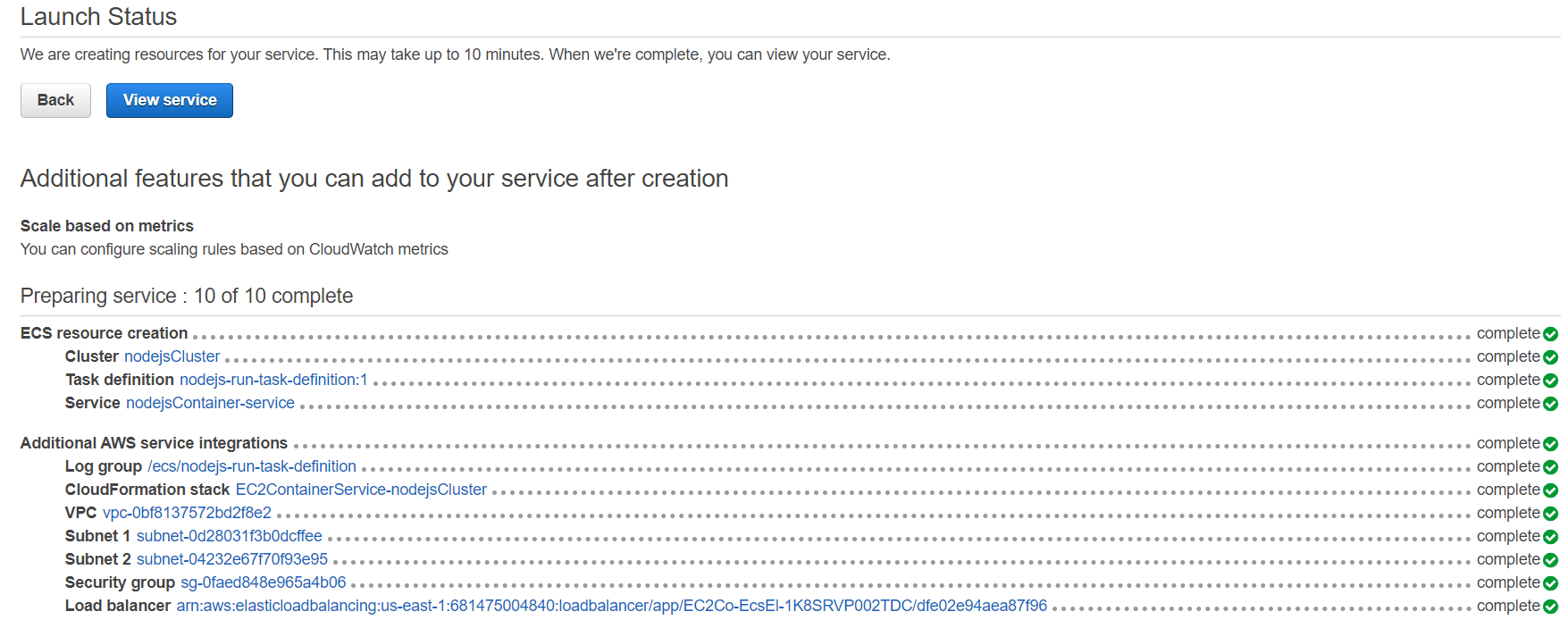


* Click on **save** and click on **Next**

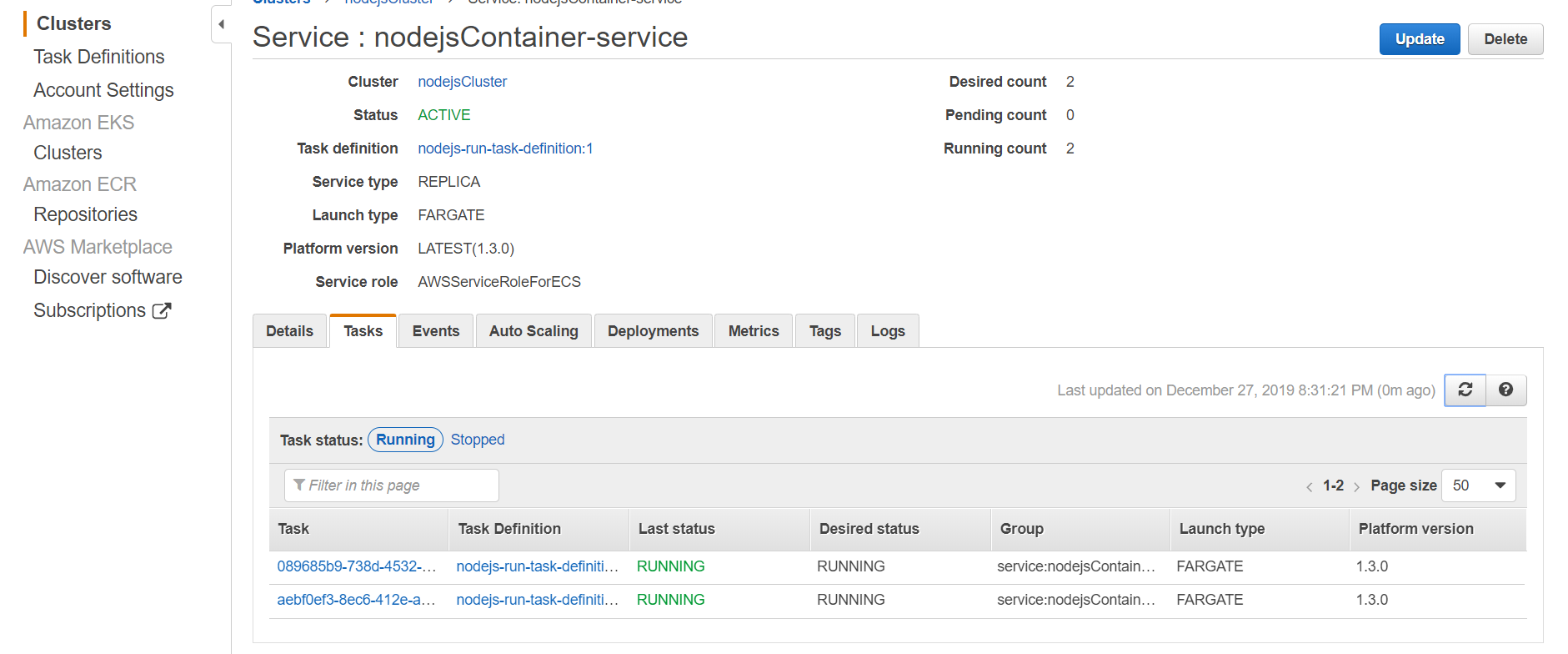


* Click on **Next**
* Click on **Create**

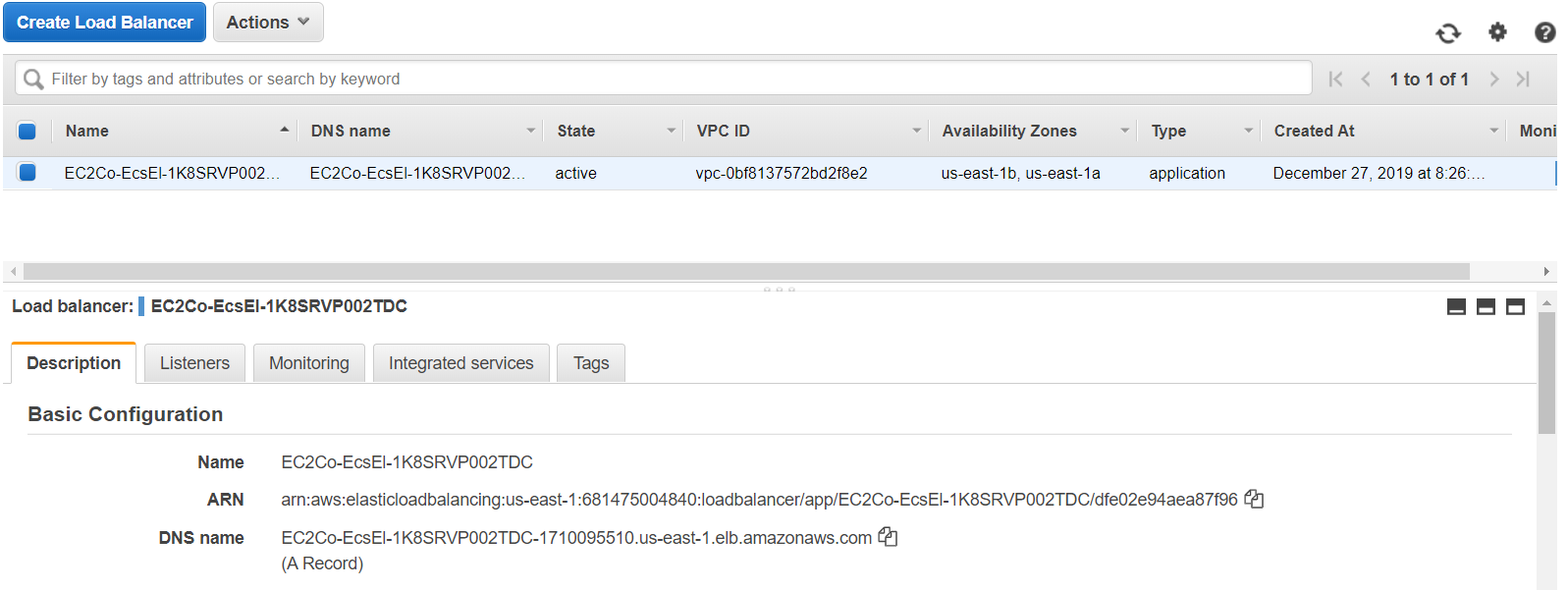
Check below figure shows Cluster succefully created



After Creation of Cluster need to wait untill task status shows in **RUNNING** mode

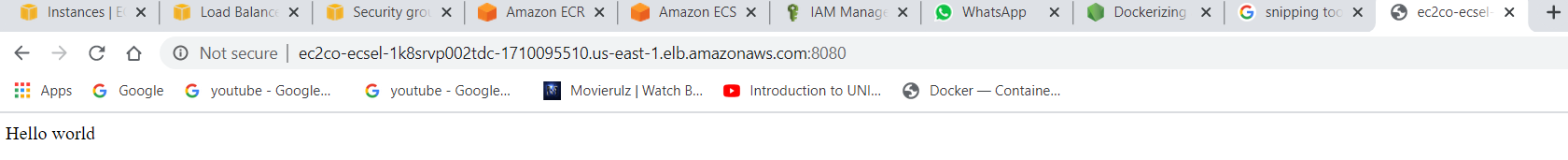


Check Load Balancer, you will get new ELB



Copy DNS Name and give at GUI with port number

<http://ec2co-ecsel-1k8srvp002tdc-1710095510.us-east-1.elb.amazonaws.com:8080/>



**References:**

For docker Installation:

<https://linuxconfig.org/how-to-install-docker-in-rhel-8>

# Dockerizing a Node.js web app:

# <https://nodejs.org/de/docs/guides/nodejs-docker-webapp/#dockerizing-a-node-js-web-app>