

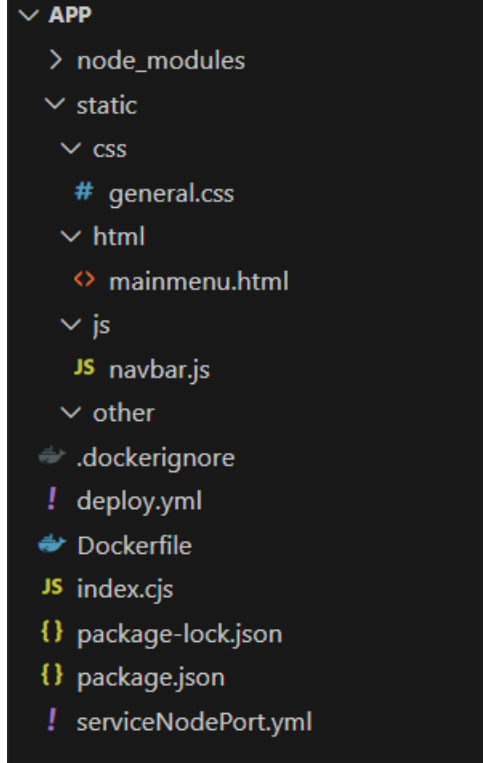
Adım Adım Amazon EKS Rehberi

Şükrü ÇİRİŞ

Arçelik A.Ş.

- 1- Canlıya almak istediğin node.js uygulamasını hazırla.

Basit node.js uygulamanın dosya yapısı:



index.cjs dosyası:

```
const express = require("express");
const path = require("path");

const app = express();

app.use(express.static(path.join(__dirname, "static")));

const PORT = 8080;
const HOST = "0.0.0.0";

app.get("/", (req, res) => {
  res.sendFile(path.join(__dirname, "static/html/mainmenu.html"));
});

app.get("/main", (req, res) => {
  res.sendFile(path.join(__dirname, "static/html/mainmenu.html"));
});

app.get("/healthcheck", (req, res) => {
```

```
res.send({ status: "ALL GOOD" });
});

app.get("/authors", (req, res) => {
  res.send({ author: "SUKRU CIRIS" });
});

app.listen(PORT, HOST, () => {
  console.log("Server started at http://localhost:" + PORT);
});
```

- 2- AWS hesabı oluştur daha sonra IAM konsolundan AWS CLI'yi kurmak için access key'lerini oluştur.
- 3- AWS CLI kur. 'aws configure' komutu ile key'lerini kullanarak hesabını lokalde aç.
- 4- Bilgisayarında yoksa Docker indir ve çalıştır. Docker dosyalarını hazırla.

Dockerfile dosyası:

```
FROM node:20

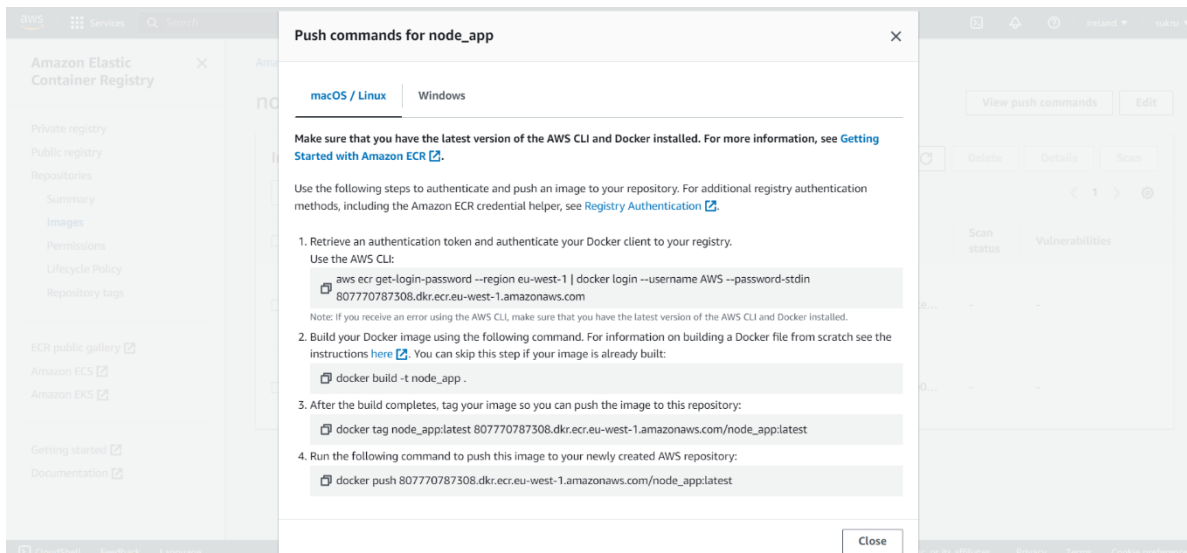
WORKDIR /usr/src/app
COPY package*.json ./
RUN npm install
COPY . .
EXPOSE 8080
CMD [ "node", "index.cjs" ]
```

'EXPOSE'dan sonra kodunda belirttiğin port numarası gelmeli ve son komutta index.cjs yerine kendi ana node.js dosyanı yaz. Base image olarak node'un resmi image'ının versiyon 20'sini kullandım. Başka bir versiyon da kullanılabilir. Resmi node image'ları debian tabanlı.

.dockerignore dosyası:

```
node_modules
npm-debug.log
```

- 5- Amazon ECR servisinde bir repository oluştur. Repository içine girip view push commands'e tıkla. Oradaki komutları app klasörünün içinde çalıştır. Hepsi çalıştıktan sonra uygulamanın image'ı latest tag'i ile repository'e yüklenmiş olması lazım.




6- EKS servisinde bir cluster oluřtur. Bütün cluster ayarlarını default olarak bıraktım. Cluster için kullandığım rol tek bir policy'e sahip:

Permissions Trust relationships Tags Access Advisor Revoke sessions

Permissions policies (1) [Info](#)

You can attach up to 10 managed policies.

Filter policies by property or policy name and press enter.

<input type="checkbox"/>	Policy name ↗	Type	Description
<input type="checkbox"/>	 AmazonEKSClusterPolicy	AWS managed	This policy provides Kubernetes the




7- Cluster içinde bir node group oluřtur. Node group için kullandığım rol üç policy'e sahip:

Permissions Trust relationships Tags Access Advisor Revoke sessions

Permissions policies (3) [Info](#)

You can attach up to 10 managed policies.

Filter policies by property or policy name and press enter.

<input type="checkbox"/>	Policy name ↗	Type	Description
<input type="checkbox"/>	 AmazonEC2ContainerRegistryReadOnly	AWS managed	Provides read-only access to A
<input type="checkbox"/>	 AmazonEKS_CNI_Policy	AWS managed	This policy provides the Amazc
<input type="checkbox"/>	 AmazonEKSWorkerNodePolicy	AWS managed	This policy allows Amazon EKS

Default bırakmadığım ve önemli node group ayarları:

Node group compute configuration

These properties cannot be changed after the node group is created.

AMI type [Info](#)
Select the EKS-optimized Amazon Machine Image for nodes.

Capacity type
Select the capacity purchase option for this node group.

Instance types [Info](#)
Select instance types you prefer for this node group.

t3.micro
vCPU: 2 vCPUs Memory: 1 GiB Network: Up to 5 Gigabit Max ENI: 2 Max IPs: 4

Disk size
Select the size of the attached EBS volume for each node.
 GiB

Node group scaling configuration

Desired size
Set the desired number of nodes that the group should launch with initially.
 nodes

Minimum size
Set the minimum number of nodes that the group can scale in to.
 nodes

Maximum size
Set the maximum number of nodes that the group can scale out to.
 nodes

- 8- 'aws eks update-kubeconfig --name <cluster_ismi>' komutunu çalıştırarak otomatik olarak kubectl için config oluştur.
- 9- deploy.yml dosyasını yaz ve 'kubectl create -f deploy.yml' komutunu çalıştır.
deploy.yml dosyası:

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: testing-kube-deploy
  labels:
    name: testing-kube-deploy
    app: testing-kube-app
spec:
  replicas: 1
  selector:
    matchLabels:
      name: testing-kube-pod
      app: testing-kube-app
  template:
    metadata:
      name: testing-kube-pod
      labels:
        name: testing-kube-pod
        app: testing-kube-app
    spec:
      containers:
        - name: testing-kube-container
          image: "807770787308.dkr.ecr.eu-west-1.amazonaws.com/node_app:latest"
          imagePullPolicy: Always
          resources:
            limits:
              memory: 512Mi
              cpu: "1"
            requests:
              memory: 128Mi
              cpu: "0.2"
          livenessProbe:
            httpGet:
              path: /healthcheck
              port: 8080
            initialDelaySeconds: 5
            periodSeconds: 5
```

Port parametresini kodunda belirttiğin port yapmak ve image parametresini kendi ECR repondaki image'inin url'si yapmak kritik önem arz ediyor.

- 10- serviceNodePort.yml dosyanı oluştur. 'kubectl create -f serviceNodePort.yml' komutunu çalıştır.

serviceNodePort.yml dosyası:

```
apiVersion: v1
kind: Service
metadata:
  name: testing-kube-service-np
  labels:
    name: testing-kube-deploy
    app: testing-kube-app
spec:
  type: NodePort
  selector:
    app: testing-kube-app
  ports:
    - protocol: TCP
      port: 8080
      nodePort: 30072
```

Port parametresini kodunda belirttiğin port yapmak önemli. Buradaki nodePort parametresini daha sonra kullanacağız.

11- EC2 servisine gidip cluster'ına ait instance'ların security group'una serviceNodePort.yml dosyasındaki nodePort portuna sahip custom TCP ipv4 ve ipv6 için iki tane inbound rule ekle.

Edit inbound rules [Info](#)

Inbound rules control the incoming traffic that's allowed to reach the instance.

Inbound rules [Info](#)

Security group rule ID	Type Info	Protocol Info	Port range Info	Source Info	Description - optional Info	
sgr-0d6e73b96c72dee2f	All traffic	All	All	Custom		Delete
-	Custom TCP	TCP	30072	Anywh...	sg-011ad6dc0fcd38da8	Delete
-	Custom TCP	TCP	30072	Anywh...	0.0.0.0/0	Delete
-	Custom TCP	TCP	30072	Anywh...	:::0	Delete

Add rule

Cancel Preview changes Save rules

12- Node grubuna ait EC2 servisindeki instance'lardan biri için `http://<Public-IPv4-DNS>:<nodePort>` url'sine gidersen uygulamanın çalıştığını görebilirsin.

