Node.js application from scratch and deploy to AZ AKS Cluster

Part 1 - create application

On ubuntu:
Create working directory:

- **mkdir nodejs-webap && cd nodejs-webap**

npm stands for "Node Package Manager." It is a widely used package manager for the JavaScript programming language and is primarily used for managing packages and dependencies in Node.js applications.
With npm, developers can easily install, share, and manage libraries, tools, and frameworks that are commonly used in JavaScript development. These packages are usually published by other developers or organizations, and npm provides a centralized repository for hosting and distributing them.
Some common commands used with npm include:

- npm install <package-name>: Installs a package and its dependencies.
- npm uninstall <package-name>: Uninstalls a package and removes it from the project's dependencies.
- npm init: Initializes a new project by creating a package.json file that holds information about the project and its dependencies.
- npm update <package-name>: Updates a package to its latest version.
- npm outdated: Lists packages that have newer versions available.
- npm publish: Publishes a package to the npm registry, making it available for others to install.
- npm search <search-term>: Searches for packages in the npm registry based on the provided search term.

Initialize a new Node.js project by creating a package.json file. The package.json file is a configuration file that contains metadata about the project, including its name, version, description, entry point, dependencies, and other important information. This file is essential for managing the project's dependencies, scripts, and various settings.
To make everything automatic:

- **npm init -y**

Better to do:

- **npm init**

```
irina@irina-Inspiron-5379:~/nodejs-webap$ npm init
This utility will walk you through creating a package.json file.
It only covers the most common items, and tries to guess sensible defaults.

See `npm help init` for definitive documentation on these fields
and exactly what they do.

Use `npm install <pkg>` afterwards to install a package and
save it as a dependency in the package.json file.

Press ^C at any time to quit.
package name: (nodejs-webap)
version: (1.0.0)
description: A simple Node JS application
entry point: (index.js) server.js
test command:
git repository:
keywords:
author: Irina Zakharova yrenamm@gmail.com
license: (ISC) MIT
About to write to /home/irina/nodejs-webap/package.json:

{
  "name": "nodejs-webap",
  "version": "1.0.0",
  "description": "A simple Node JS application",
  "main": "server.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "author": "Irina Zakharova yrenamm@gmail.com",
  "license": "MIT"
}


Is this OK? (yes)
```

Install dependencies: (on **ls** will be node_modules)

- **npm install express mocha**

Edit package.json by adding testing by mocha

- **gedit package.json**

```
{
  "name": "nodejs-webap",
  "version": "1.0.0",
  "description": "A simple Node JS application",
  "main": "server.js",
  "scripts": {
    "test": "mocha"
  },
  "author": "Irina Zakharova yrenamm@gmail.com",
  "license": "MIT",
  "dependencies": {
    "express": "^4.18.2",
    "mocha": "^10.2.0"
  }
}
```

Mocha is a popular testing framework for JavaScript applications, both on the server-side (Node.js) and in the browser. It provides a flexible and feature-rich environment for writing and executing tests, making it a common choice for unit testing, integration testing, and more.

Create entry point

- **gedit server.js**

Open

server.js
~/nodejs-webap

```
1 const express = require('express');
2 const app = express();
3 const path = require('path');
4
5
6 app.get('/', (req,res) => {
7   res.sendFile(path.join(__dirname+'/index.html'));
8
9 });
10
11
12 app.get('/about', (req,res) => {
13     res.sendFile(path.join(__dirname+'/about.html'));
14
15   });
16
17 app.listen(3000, () => {
18     console.log('Listening on port 3000');
19 });
20
```

Create index.html

- **gedit index.html**

Open

***index.html**
~/nodejs-webap

Save

```
1 <!DOCTYPE html>
2 <head>
3    <title> IBT Learning </title>
4    <script src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.2/jquery.min.js"></script>
5    <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.1/css/bootstrap.min.css">
6    <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.1/css/bootstrap-theme.min.css">
7    <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.1/js/bootstrap.min.js"></script>
8  </head>
9  <body>
10   <div style="margin:6px;">
11     <nav class="navbar navbar-inverse navbar-static-top">
12   <div class="container">
13     <a class="navbar-brand" href="/"> Welcome to IBT Learning </a>
14     <ul class="nav navbar-nav">
15       <li class="active">
16         <a href="/"> Home </a>
17       </li>
18       <li>
19         <a href="/about"> About </a>
20       </li>
21     </ul>
22   </div>
23 </nav>
24     <div class="jumbotron"  style="padding:40px;">
25       <h1> This is a Hello Message from  <span class="text-primary"> IBT </span><span class="text-danger"> Learning </span></h1>
26       <p> Kudos to Everyone of you <span class="bg-info"> For </span> Building this Nodejs Project <span class="bg-success"> From Scratch </span></p>
27       <p><a class="btn btn-primary btn-lg" href="#" role="button">Keep Winning</a></p>
28     </div>
29   </div>
30 </body>
31 </html>
32
```

Create about.html

- **gedit about.html**


```
<!DOCTYPE html>
<head>
    <title> About Us Page </title>
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.2/jquery.min.js"></script>
    <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.1/css/bootstrap.min.css">
    <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.1/css/bootstrap-theme.min.css">
    <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.1/js/bootstrap.min.js"></script>
  </head>
  <body>
    <div style="margin:6px;">
      <nav class="navbar navbar-inverse navbar-static-top">
    <div class="container">
      <a class="navbar-brand" href="/"> About IBT Learning </a>
      <ul class="nav navbar-nav">
        <li>
          <a href="/">Home</a>
        </li>
        <li class="active">
          <a href="/about">About</a>
        </li>
      </ul>
    </div>
  </nav>
      <div class="jumbotron"  style="padding:40px;">
        <h1>About Us</h1>
        <strong><p>IBT Learning is a digital economy Instructor-led and career-oriented Bootcamp offering training to working professionals, career changers, and corporate institutions across industries and sectors. IBT Learning is an official training partner of IBM, AWS Partner Network, Linux Professional Institute, Comptia, EC-Council, and more. </p></strong>

        <strong><p>Learn the skills that companies are looking for and land a job that you will love. At IBT, we are outcome driven and are not merely training you to bag certifications, we are training you for successful careers in tech.</p></strong>
        <p><a class="btn btn-primary btn-lg" href="#" role="button">Start Learning</a></p>
      </div>
    </div>
  </body>
</html>
```

Run the application

- **node server.js**

Check if works:
http://localhost:3000/

Part 2 - containerize application

Create Dockerfile:

- **gedit Dockerfile**

Open

Dockerfile
~/nodejs-webap

```
# Base Image

FROM node:lts-alpine3.17

# Set the working Directory
WORKDIR /app

# Copy Package.json

COPY package*.json ./

# Install Dependencies

RUN npm install

# Copy source code to the container work directory

COPY . .

# Expose Port

EXPOSE 3000

# Entry for CMD

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

```

Create .dockerignore

- **gedit .dockerignore**

Open

.dockerignore
~/nodejs-webap

```
.env
node_modules
```

Containerize the application:

- **docker build -t yrenamm/nodejs-webapp .**

To check if the docker image was created:

- **docker images**

To test the docker image:

- **docker run -itd -p 3000:3000 yrenamm/nodejs-webapp:latest**

Check docker processes

- **docker ps**
- **docker ps -a**

Check the dockerized application is running good:
http://localhost:3000/

Stop the containerized application:

- **docker kill fe68a71c2993c059e72b47ed749cc64ff087284db4ef924193f8d053c6d095e0**
- **docker rm fe68a71c2993c059e72b47ed749cc64ff087284db4ef924193f8d053c6d095e0**
- **docker stop fe68a71c2993**

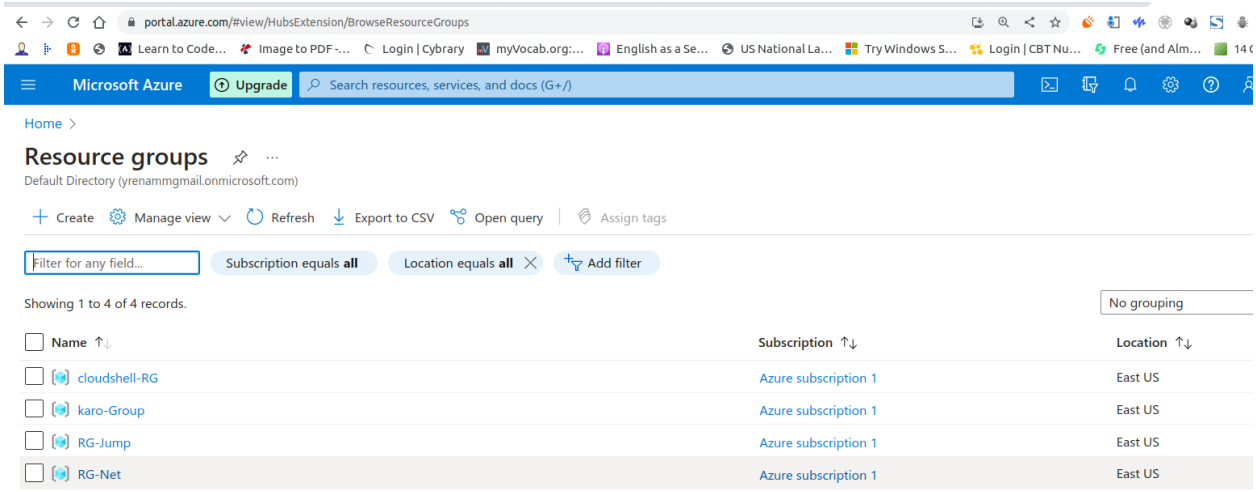
Part 3 - cloud part azure

Login to azure:

- **az login**

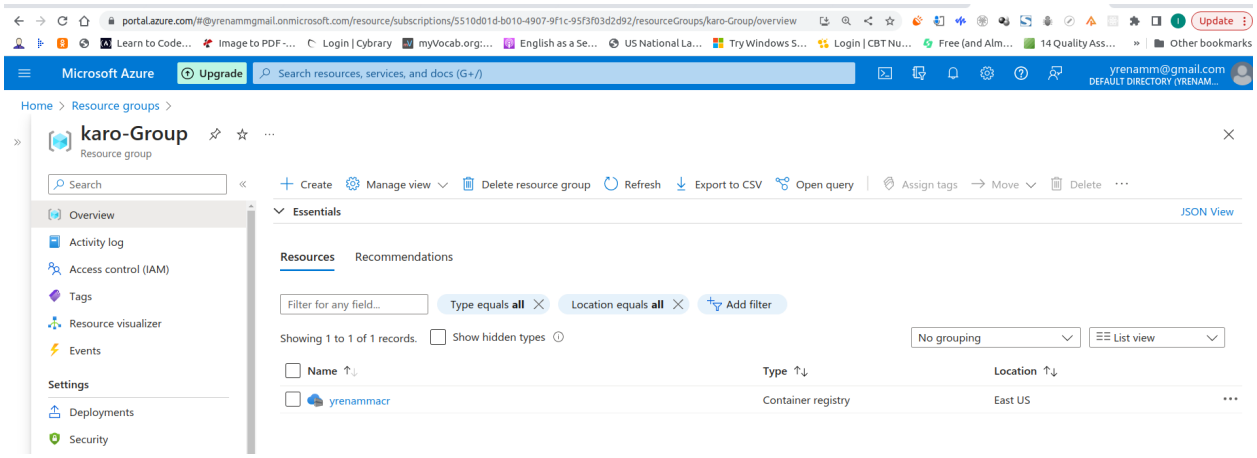
Create resource group:

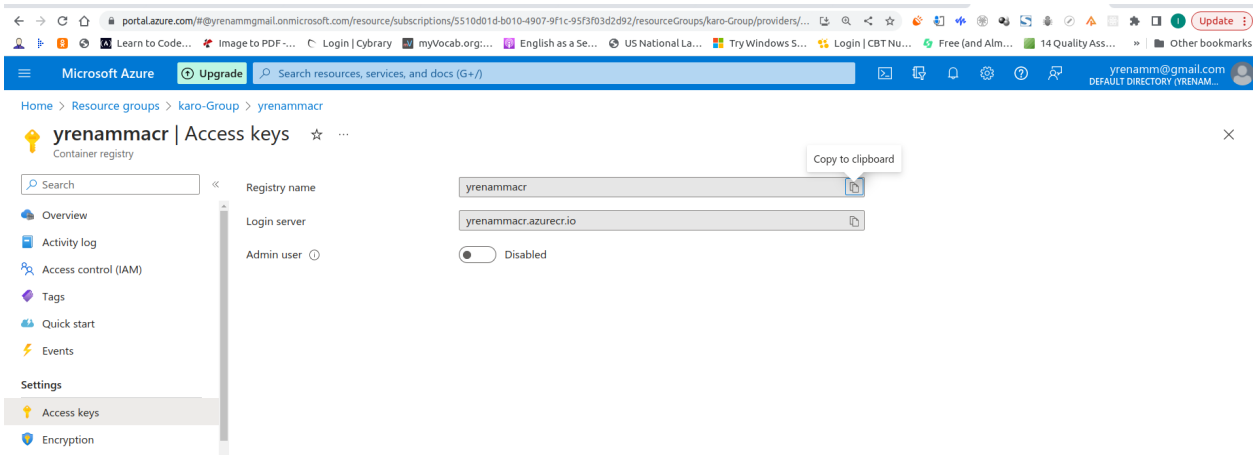
- **az group create --name karo-Group --location eastus**



Create an Azure Container Registry

- **az acr create --resource-group karo-Group --name yrenammacr --sku Standard --location eastus**





Log into an Azure Container Registry (ACR) allowing you to push and pull container images to and from that registry

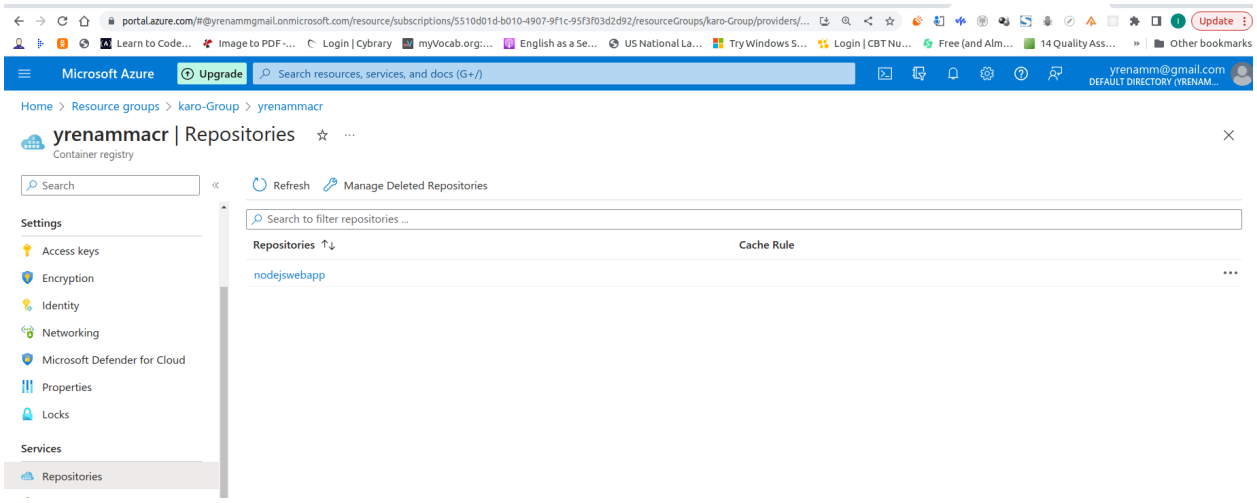
- **az acr login --name yrenammacr**

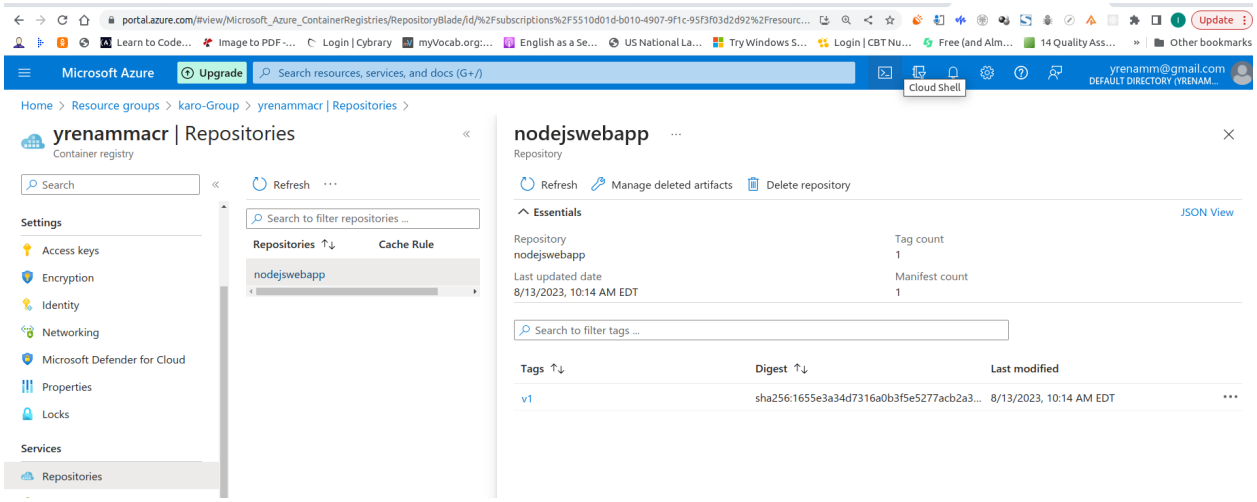
Retag image:

- **docker tag yrenamm/nodejs-webapp:latest yrenammacr.azurecr.io/nodejswebapp:v1**

Push image

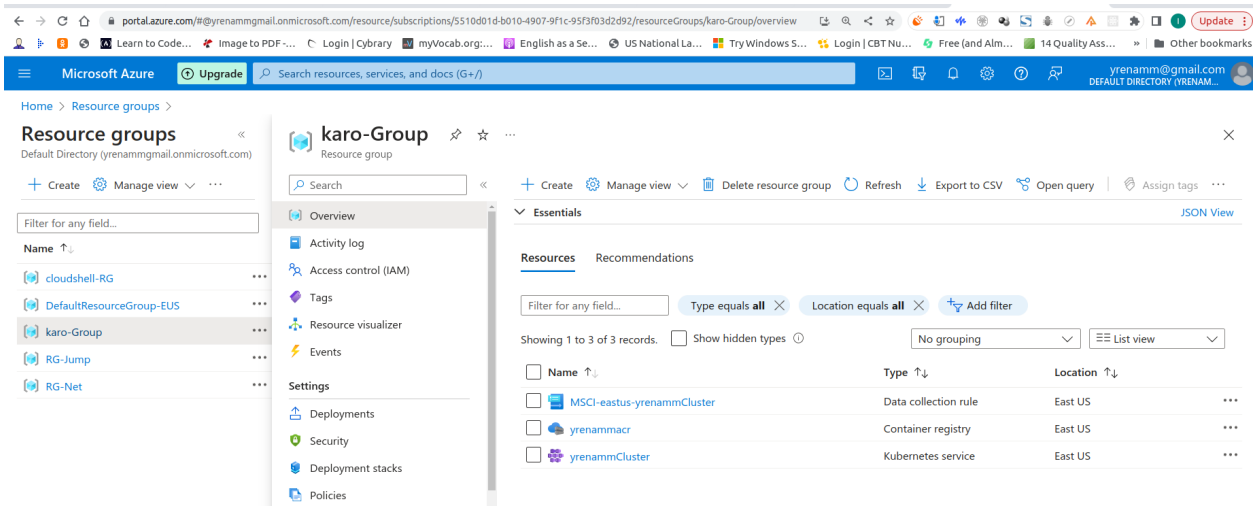
- **docker push yrenammacr.azurecr.io/nodejswebapp:v1**





Create an Azure Kubernetes Service (AKS) cluster

- **az provider register --namespace Microsoft.OperationalInsights**
- **az provider register --namespace microsoft.insights**
- **az aks create --resource-group karo-Group --name yrenammCluster --node-count 2 --generate-ssh-keys --enable-addons monitoring**



Set up Kubernetes configuration on your local machine for accessing an Azure Kubernetes Service (AKS) cluster.

- **az aks get-credentials --resource-group karo-Group --name yrenammCluster --overwrite-existing**

Retrieve information about the nodes in your Kubernetes cluster.

- **kubectl get nodes**

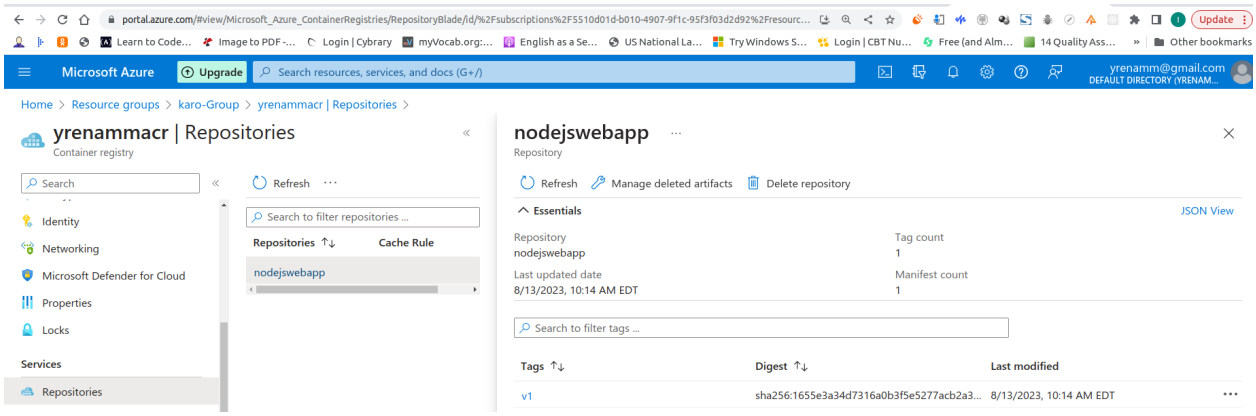
Connect our cluster to ACR - update an existing Azure Kubernetes Service (AKS) cluster to attach it to an Azure Container Registry (ACR) for easy integration with container images stored in the ACR.

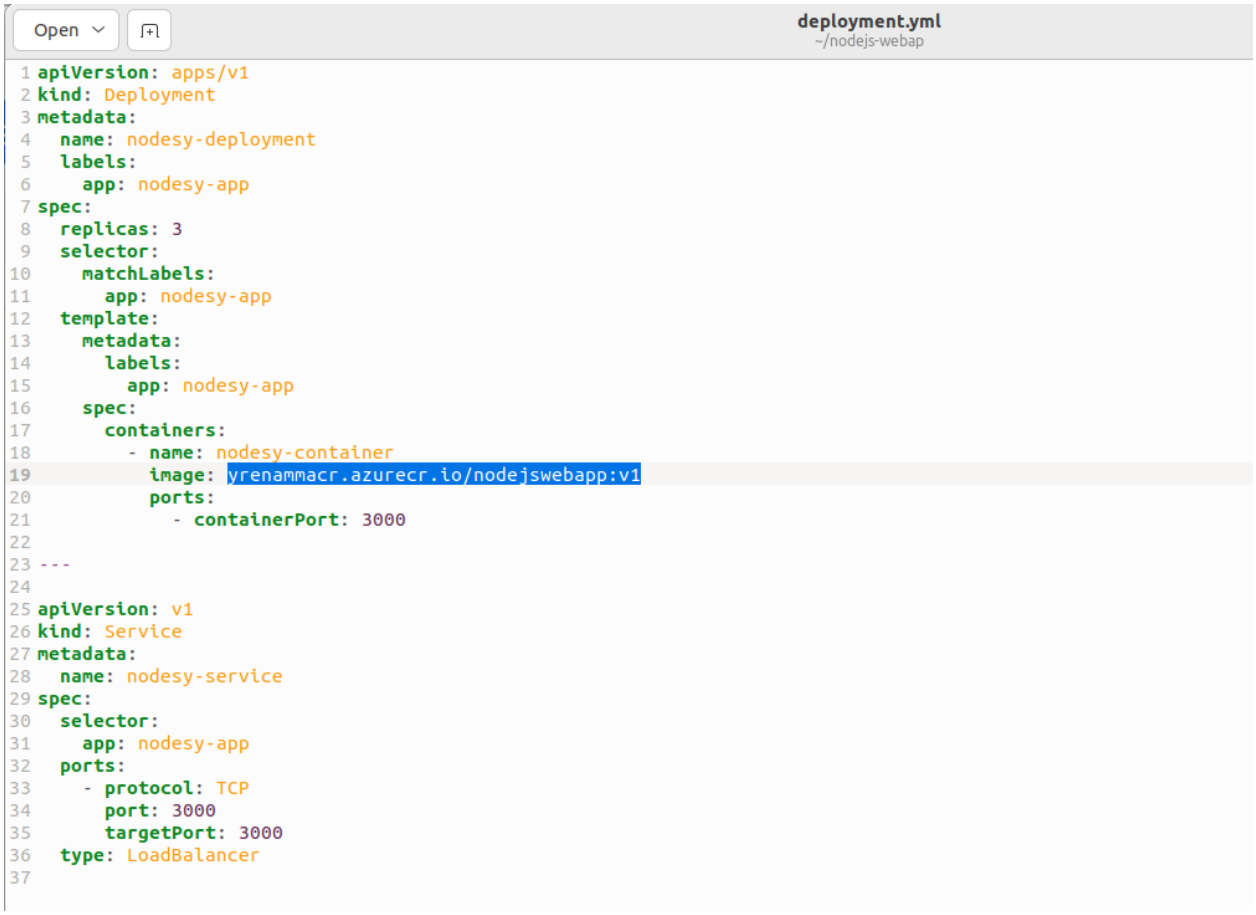
- **az aks update -n yrenammCluster -g karo-Group --attach-acr yrenammacr**

Create deployment.yml in the working directory:

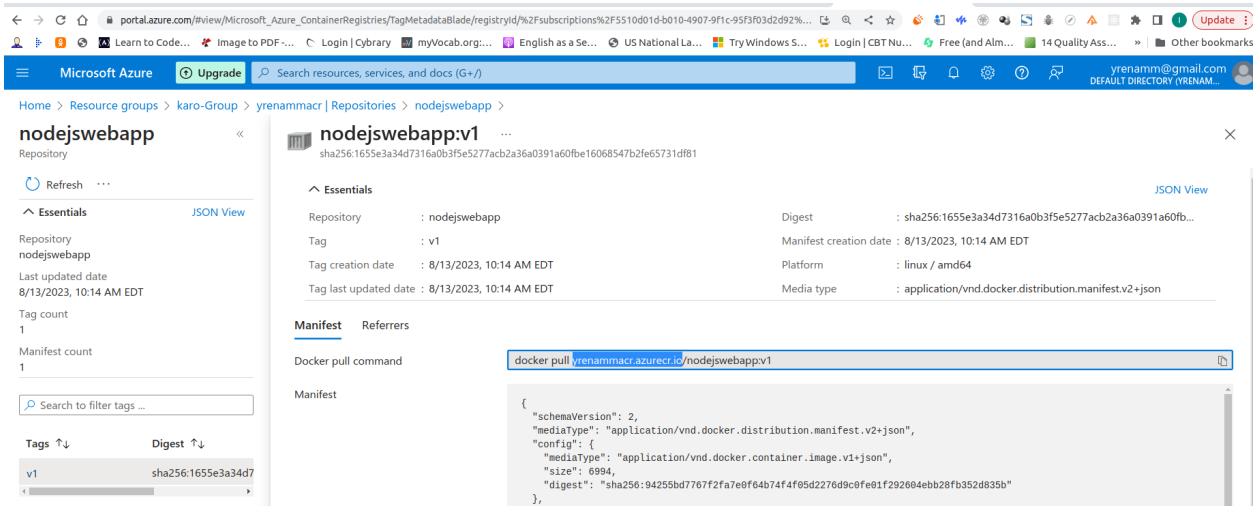
- **gedit deployment.yml**

```
irina@irina-Inspiron-5379:~/nodejs-webapp$ ls
about.html  deployment.yml  Dockerfile  index.html  node_modules  package.json  package-lock.json  server.js
irina@irina-Inspiron-5379:~/nodejs-webapp$
```


Microsoft Azure
Upgrade
Search resources, services, and docs (G+/)
yrenamm@gmail.com
DEFAULT DIRECTORY (YRENAM...
Home > Resource groups > karo-Group > yrenammacr | Repositories >
yrenammacr | Repositories
Container registry
Search
Identity
Networking
Microsoft Defender for Cloud
Properties
Locks
Services
Repositories
Refresh
Search to filter repositories ...
Repositories
Cache Rule
nodejswebapp
nodejswebapp
Repository
Refresh
Manage deleted artifacts
Delete repository
Essentials
JSON View
Repository
nodejswebapp
Tag count
1
Last updated date
8/13/2023, 10:14 AM EDT
Manifest count
1
Search to filter tags ...
Tags
Digest
Last modified
v1
sha256:1655e3a34d7316a0b3f5e5277acb2a3...
8/13/2023, 10:14 AM EDT


Open
deployment.yml
~/nodejs-webap
1 apiVersion: apps/v1
2 kind: Deployment
3 metadata:
4 name: nodesy-deployment
5 labels:
6 app: nodesy-app
7 spec:
8 replicas: 3
9 selector:
10 matchLabels:
11 app: nodesy-app
12 template:
13 metadata:
14 labels:
15 app: nodesy-app
16 spec:
17 containers:
18 - name: nodesy-container
19 image: yrenammacr.azurecr.io/nodejswebapp:v1
20 ports:
21 - containerPort: 3000
22
23 ---
24
25 apiVersion: v1
26 kind: Service
27 metadata:
28 name: nodesy-service
29 spec:
30 selector:
31 app: nodesy-app
32 ports:
33 - protocol: TCP
34 port: 3000
35 targetPort: 3000
36 type: LoadBalancer
37

Like:



Apply or update Kubernetes resources defined in a YAML file

- **kubectl apply -f deployment.yml**

Retrieve information about the pods running in your Kubernetes cluster.

- **kubectl get pods**

Retrieve information about the services running in your Kubernetes cluster

- **kubectl get svc** (or)
- **kubectl get services**

Copy EXTERNAL-IP to browser:

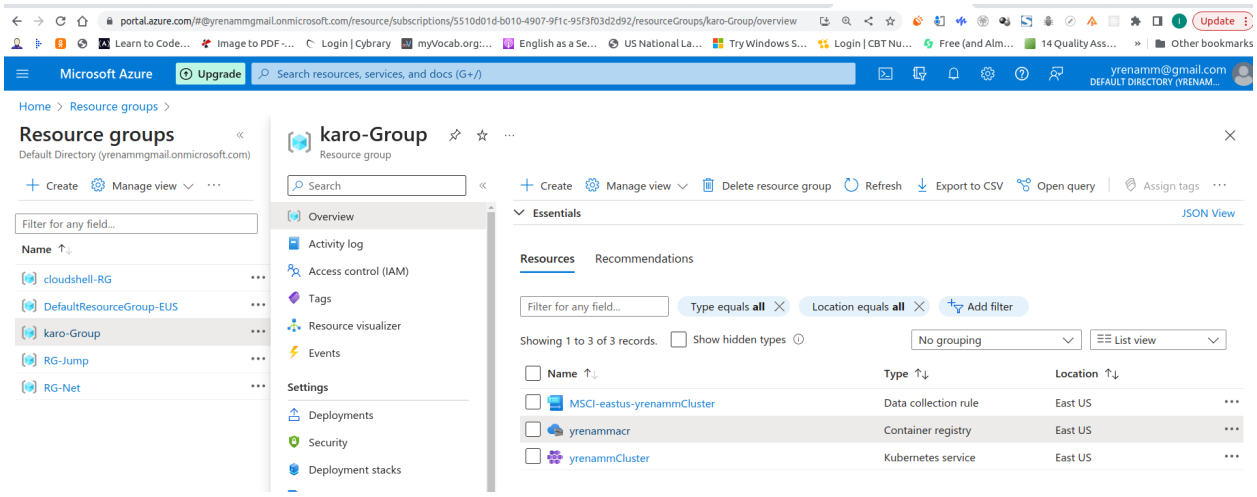
```
irina@irina-Inspiron-5379:~/nodejs-webap$ kubectl get svc
NAME             TYPE           CLUSTER-IP     EXTERNAL-IP      PORT(S)          AGE
kubernetes       ClusterIP      10.0.0.1       <none>           443/TCP          72m
nodesy-service   LoadBalancer   10.0.169.60    20.246.177.214   3000:32327/TCP   18m
irina@irina-Inspiron-5379:~/nodejs-webap$ kubectl get services
NAME             TYPE           CLUSTER-IP     EXTERNAL-IP      PORT(S)          AGE
kubernetes       ClusterIP      10.0.0.1       <none>           443/TCP          73m
nodesy-service   LoadBalancer   10.0.169.60    20.246.177.214   3000:32327/TCP   19m
```

http://20.246.177.214:3000/ (the app is running correctly)

Part 3 - clean up

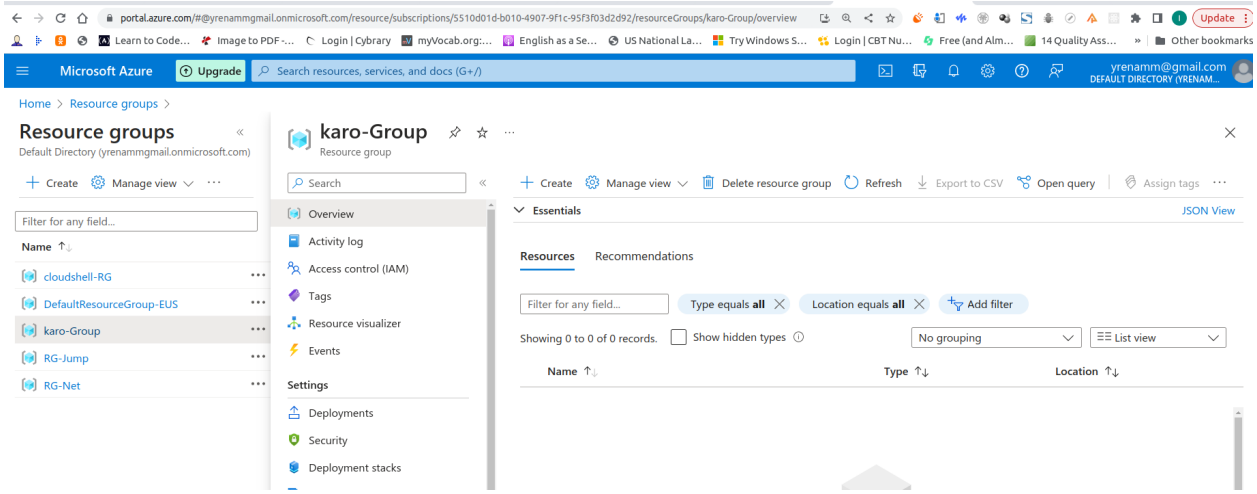
Clean up:

- **kubectl delete -f deployment.yml**
- **kubectl get pods**



Delete everything (resources)

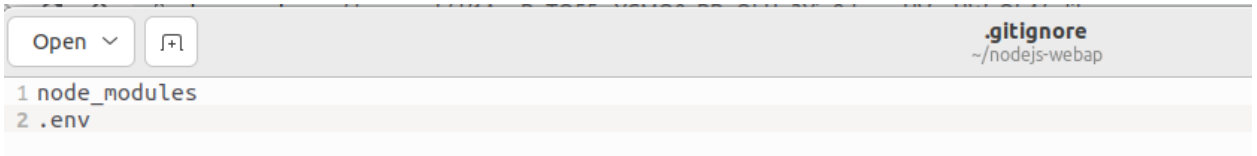
- **az group delete --name karo-Group**



Part 4 - Push project to GitHub

Create .gitignore

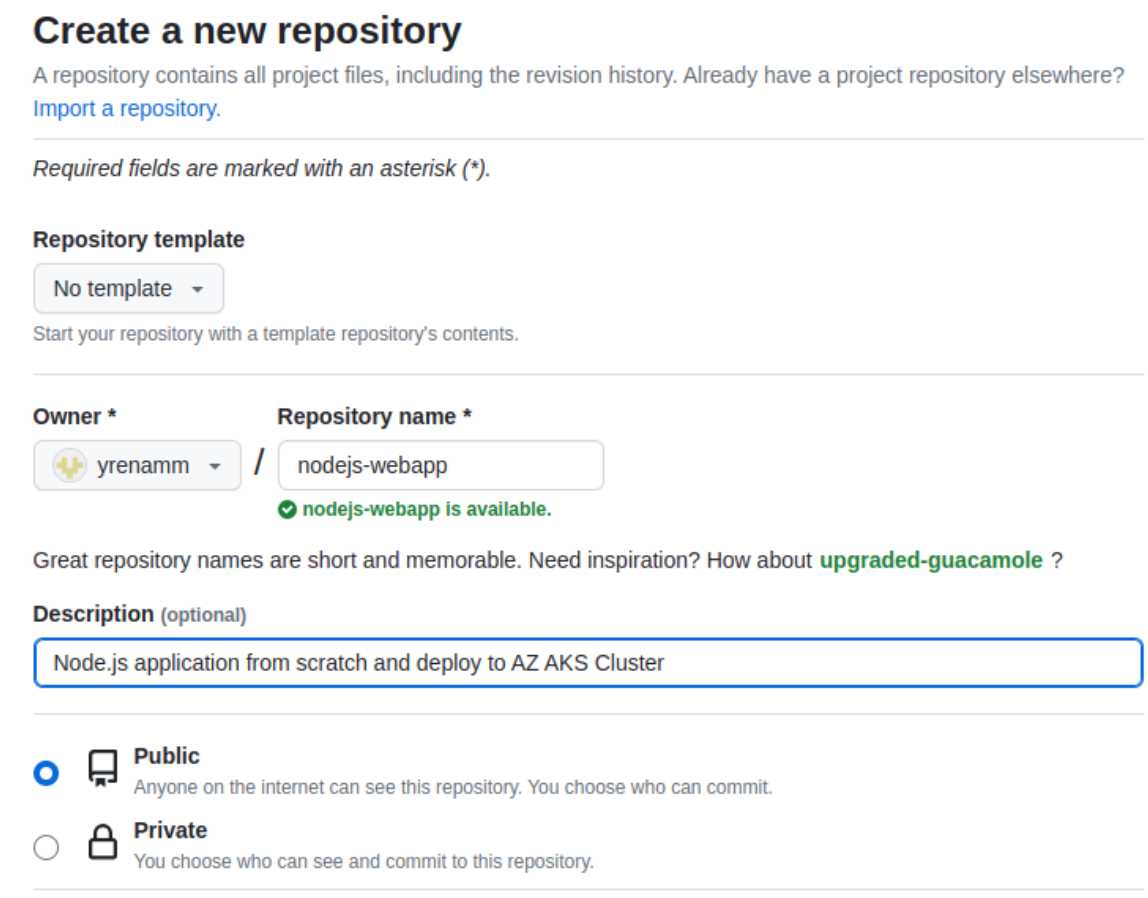
- **gedit .gitignore**



ls -al:

```
irina@irina-Inspiron-5379:~/nodejs-webap$ ls -al
total 136
drwxrwxr-x   3 irina irina  4096 Aug 13 12:39 .
drwxr-x---  38 irina irina  4096 Aug 12 19:32 ..
-rw-rw-r--   1 irina irina  1699 Aug 12 20:39 about.html
-rw-rw-r--   1 irina irina   602 Aug 13 11:51 deployment.yml
-rw-rw-r--   1 irina irina   304 Aug 12 21:18 Dockerfile
-rw-rw-r--   1 irina irina    18 Aug 12 21:45 .dockerignore
-rw-rw-r--   1 irina irina    18 Aug 13 12:39 .gitignore
-rw-rw-r--   1 irina irina  1344 Aug 12 20:36 index.html
drwxrwxr-x 129 irina irina  4096 Aug 12 20:02 node_modules
-rw-rw-r--   1 irina irina   306 Aug 12 20:26 package.json
-rw-rw-r--   1 irina irina 93390 Aug 12 20:02 package-lock.json
-rw-rw-r--   1 irina irina   351 Aug 12 20:32 server.js
```

Create new repository on GitHub



Init git on working directory

- **git init**

- **git branch -M main**
- **git remote add origin https://github.com/yrenamm/nodejs-webapp.git**
- **git add .**
- **git commit -m "Init"**
- **git push -u origin main**
- **git config --global user.email "yrenamm@gmail.com"**
- **git config --global user.name "Irina Zakharova"**

Part 5 - Push docker image to DockerHub

For the image name:

- **docker images**

Push the image to Docker Hub

- **docker login**
- **docker push yrenamm/nodejs-webapp:latest**

The content of files:

—----------SERVER.JS—----------

```
const express = require('express');
const app = express();
const path = require('path');


app.get('/', (req,res) => {
  res.sendFile(path.join(__dirname+'/index.html'));

});


app.get('/about', (req,res) => {
    res.sendFile(path.join(__dirname+'/about.html'));

  });

app.listen(3000, () => {
    console.log('Listening on port 3000');
});
```

—----------

—----------index.html—----------

```
<!DOCTYPE html>
<head>
   <title> IBT Learning </title>
   <script src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.2/jquery.min.js"></script>
   <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.1/css/bootstrap.min.css">
   <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.1/css/bootstrap-theme.min.css">
   <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.1/js/bootstrap.min.js"></script>
 </head>
 <body>
   <div style="margin:6px;">
     <nav class="navbar navbar-inverse navbar-static-top">
   <div class="container">
     <a class="navbar-brand" href="/"> Welcome to IBT Learning </a>
     <ul class="nav navbar-nav">
       <li class="active">
         <a href="/"> Home </a>
       </li>
       <li>
         <a href="/about"> About </a>
       </li>
     </ul>
   </div>
 </nav>
     <div class="jumbotron"  style="padding:40px;">
       <h1> This is a Hello Message from  <span class="text-primary"> IBT </span><span
class="text-danger"> Learning </span></h1>
       <p> Kudos to Everyone of you <span class="bg-info"> For </span> Building this Nodejs
Project <span class="bg-success"> From Scratch </span></p>
       <p><a class="btn btn-primary btn-lg" href="#" role="button">Keep Winning</a></p>
     </div>
   </div>
 </body>
</html>
```

—----------

—----------about.html—----------

```
<!DOCTYPE html>
<head>
   <title> About Us Page </title>
   <script src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.2/jquery.min.js"></script>
   <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.1/css/bootstrap.min.css">
   <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.1/css/bootstrap-theme.min.css">
   <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.1/js/bootstrap.min.js"></script>
 </head>
 <body>
   <div style="margin:6px;">
    <nav class="navbar navbar-inverse navbar-static-top">
   <div class="container">
    <a class="navbar-brand" href="/"> About IBT Learning </a>
    <ul class="nav navbar-nav">
      <li>
       <a href="/">Home</a>
      </li>
      <li class="active">
       <a href="/about">About</a>
      </li>
    </ul>
   </div>
 </nav>
    <div class="jumbotron"  style="padding:40px;">
      <h1>About Us</h1>
      <strong><p>IBT Learning is a digital economy Instructor-led and career-oriented
Bootcamp offering training to working professionals, career changers, and corporate institutions
across industries and sectors. IBT Learning is an official training partner of IBM, AWS Partner
Network, Linux Professional Institute, Comptia, EC-Council, and more. </p></strong>

      <strong><p>Learn the skills that companies are looking for and land a job that you will
love. At IBT, we are outcome driven and are not merely training you to bag certifications, we are
training you for successful careers in tech.</p></strong>
      <p><a class="btn btn-primary btn-lg" href="#" role="button">Start Learning</a></p>
    </div>
   </div>
 </body>
</html>
```

—----------Dockerfile—----------

```
# Base Image

FROM node:lts-alpine3.17

# Set the working Directory
WORKDIR /app

# Copy Package.json

COPY package*.json ./

# Install Dependencies

RUN npm install

# Copy source code to the container work directory

COPY . .

# Expose Port

EXPOSE 3000

# Entry for CMD

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

—----------

—----------deployment.yml—----------

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nodesy-deployment
  labels:
    app: nodesy-app
spec:
  replicas: 3
  selector:
    matchLabels:
      app: nodesy-app
```

```
  template:
    metadata:
      labels:
        app: nodesy-app
    spec:
      containers:
        - name: nodesy-container
          image: boboacr.azurecr.io/nodejswebapp:v1.0.0
          ports:
            - containerPort: 3000

---

apiVersion: v1
kind: Service
metadata:
  name: nodesy-service
spec:
  selector:
    app: nodesy-app
  ports:
    - protocol: TCP
      port: 3000
      targetPort: 3000
  type: LoadBalancer
```

AWS:

Search Elastic Container registry

Create repository

```
eksctl cluster create --name yrenamm-eks --region us-east-1
```

Some quick commands::

1. npm init
2. npm install express mocha

3a. vi/nano server.js or app.js

3b. Create Index.html and About.hmtl

4. run app locally

5. dockerise application (Create Dockerfile --production)
6. build docker image
7. push docker image to registry, Create Registry first before Push (Dockerhub, ECR, ACR, DOCR). See note for ACR below.
8. Create kubernetes clusters (Use Terraform, or CLI or UI)
9. use kubectl and create resources via cli or manifest files (deployments, services, ingress also?)
10. Create a remote git repo in Github
11. Push Code to Github (Create .gitinore file)
12. Create a Jenkins Job

----- FOR ACR -----
// do az login first
a. az group create --name karo-Group --location westeurope
b. az acr create --resource-group karo-Group --name boboacr --sku Standard --location westeurope
c. az acr login --name <registry-name>
d. docker tag ooghenekaro/nodejswebapp:v1.0.0 boboacr.azurecr.io/nodejswebapp:v1
d. docker push boboacr.azurecr.io/nodejswebapp:v1
e. az acr repository list --name boboacr --output table
f. az acr repository show-tags --name boboacr --repository nodejswebapp --output table
g. az group delete --name <myResourceGroup>

Create AKS Cluster from CLI
a. az login
b. az group create --name myResourceGroup --location westeurope
c. az aks create --resource-group karo-Group --name boboCluster --node-count 2 --generate-ssh-keys --enable-addons monitoring
d. az aks show --name boboCluster --resource-group karo-Group

e. az acr create --resource-group karo-Group --name boboacr --sku Standard --location westeurope //only use this command if you didn't have an existing acr or you will like a create a different one instead.

f. az aks get-credentials --resource-group karo-Group --name boboCluster --overwrite-existing
g. az aks update -n boboCluster -g karo-Group --attach-acr boboacr