Creating Sample application - Build that application in docker - convert that docker container into docker image - deploy that image into kubernetes cluster

My Task is :-

I am having a task
1. Take a sample application
2. Build that application in the docker
3. Push that docker image to the container registry
4. Deploy that image to Kubernetes using Yaml

all this should be in Azure so help be each task how to do in detail

==========

1. Go to this youtube link and create a sample application

   https://www.youtube.com/watch?v=-ERWlp828kY&list=WL&index=18

2. Login to u r azure portal search for Container registries

3. Create a one container — it is a docker container

4. Write a docker file like this

```
# Use an official Node.js runtime as the base image
FROM node:14-alpine

# Set the working directory inside the container
WORKDIR /app

# Copy package.json and package-lock.json to the working directory
COPY package*.json ./

# Install dependencies
```

```
RUN npm install

# Copy the rest of the application code
COPY . .

# Build the React application
RUN npm run build

# Expose the port the application will run on
EXPOSE 3000

# Command to start the application
CMD ["npm", "start"]
```

5. Open terminal in visual studio code

   az login

6. Place this Dockerfile in the root directory of your application

7. Use this command this will contact to your application

   az acr login --name < your application name >

8. Open a terminal, navigate to your application's directory, and run the following Docker commands to build the Docker image

   docker build -t <your-image-name > : latest

9. This command will create your docker image in the place of your-image-name give whatever u want for your image name.

10. Now go to your portal and search for Kubernetes services and create one Create Kubernetes cluster

11. Now go to u r visual studio code and open terminal and use this command this will connect u r AKC cluster from local to portal

    az acr login --name your-acr-name

12. Now open your created cluster and you may need to generate an authentication token for secure access to your ACR. This can be done through the Azure portal as well ,

    In the ACR's overview page in the Azure portal, navigate to "Access keys" under the "Settings" section.

    You can configure a service principal or use a username and password for authentication, depending on your requirements. Generate the necessary credentials and tokens as needed.

    Remember to keep your ACR credentials secure, especially in a production environment, and consider using Azure's managed identities or other security best practices for your container registry.

    This is a command u have to use:-

    docker push your-acr-name.azurecr.io/your-image-name:latest

Task 4: Deploy the Image to Kubernetes Using YAML

13. Now in Create a Kubernetes Deployment YAML file go to the root folder of your application and save the file with this extension (e.g., deployment.yaml) to define how your application should be deployed. Here's a basic example:

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: sample-app-deployment
spec:
  replicas: 0
  selector:
    matchLabels:
      app: sample-app
  template:
    metadata:
      labels:
        app: sample-app
    spec:
    containers:
        - name: myapp-container
```

```
          image: sampleapplication.azurecr.io/myapp:latest
          ports:
            - containerPort: 8080
---
apiVersion: v1
kind: Service
metadata:
  name: sample-app-service-nodeport
spec:
  selector:
    app: sample-app
  ports:
    - protocol: TCP
      port: 80
      targetPort: 8080
  type: LoadBalancer
```

14. Now use this command

```
kubectl apply -f deployment.yaml
```

15. Kubectl get pods

That's it……..