

Create a Nginx image for company XYZ  
This lab seeks to help company XYZ to provision their website as a container.  
In order to achieve this , you have been contracted to create an image from the existing code base.

Follow

the instructions below:

1. Create a directory named **WebServer**
2. Use the file web.html as the web page.
3. Copy the web.html file into the nginx image. (**COPY** command)
4. Use the nginx daemon off -g group command in your dockerfile. (hint use the **CMD**

command)

1. Expose the image to port 8080
2. Build image from the dockerfile and call it **my\_pod\_image** (use **docker build** command and tag it web\_site)
3. Create 2 pods from the image **my\_pod\_image** ensure you bind the ports 80 to 8080**.**
4. Check the logs of each pod and provide screenshots.
5. Provide screenshots of the pods created.

***10. Ensure the website is running. (NB Services)*** 11. **Write a pod definition file using yaml.**

**Solution:**

1. **Create a directory named WebServer:**

Command:

***mkdir WebServer***

***cd WebServer***

2. **Use the file index.html as the web page:**

Create a file named `index.html` and add your HTML content.

A screen shot of a computer

Description automatically generated

3. **Create a Dockerfile:**

Create a file named `Dockerfile` in the `WebServer` directory and add the following content:

***```Dockerfile***

***FROM nginx (you can add the version number eg nginx:1 or nginx:2 or nginx:latest, if not docker will download the latest version)***

***COPY index.html /usr/share/nginx/html/***

***CMD ["nginx", "-g", "daemon off;"]***

***EXPOSE 8080***A screen shot of a computer

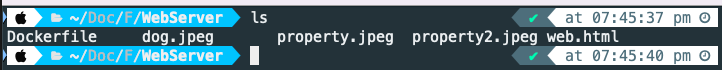
Description automatically generated

4. **Build the Docker image: (For point 6. 1 build the dockerfile and call it my\_pod\_image 2 and then when you push image to dockerhub call it web\_site)**

**# Navigate to the WebServer directory**

cd WebServer





**# Build the Docker image and tag it as my\_pod\_image**

*docker build -t my\_pod\_image .*

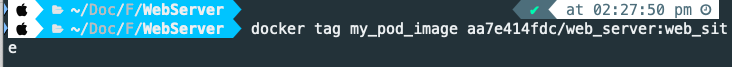
A screen shot of a computer

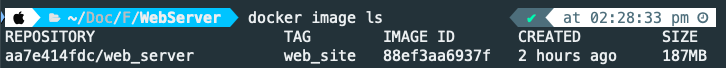
Description automatically generated

# Pushing a Docker image to Docker registry

* Go on your docker hub account and create a repository name. Our repository name is web\_server.
* Then return to terminal to create a tag name of your image : ‘**docker tag SOURCE\_IMAGE[:TAG] TARGET\_IMAGE[:TAG]’**
* In our example**, the *SOURCE\_IMAGE[:TAG]***is the name and the tag of the existing **(my\_pod\_image)** image. **The *TARGET\_IMAGE[:TAG]***is the alias consisting of our user name and the name of the repository that we want to push the image to.We want to give the image a new tag called **web\_site** when it pushes to our repository.
* **See the image below:**

docker tag my\_pod\_image:latest aa7e414fdc/web\_server:web\_site





Then push to the repository on our docker hub called aa7e414fdc/web\_server

A screen shot of a computer

Description automatically generated

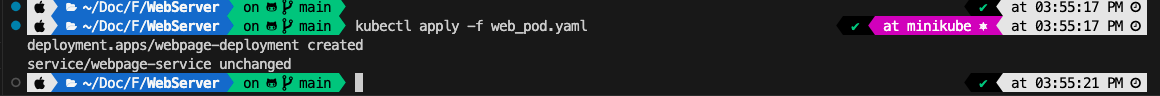
5. **Create 2 pods from the image using kubernetes:**

Create a yaml file for deployment.

A screen shot of a computer

Description automatically generated

Cd to the folder where the file is located in you terminal and run the following command



Check if pods are running   


Check the logs  
A screenshot of a computer

Description automatically generated

**View in web browser**

* To view webpage in the browser you need the minikube IP address and Nodeport number.
* From our web\_pod.yaml file we know the node port number is 30100.
* To find minikube ip use the command **“minikube ip”** below



* Now go to your web brower and type MinikubeIP:NodePort

In our case ***“192.168.49.2:30100”***

* If this doesn’t load up the webpage in your web browser then use

minikube service webpage-service