Scenario 2

**Deploy static Html website as container**

**Step 1:**

\*Docker Images start from a base image.

\*The base image should include the platform dependencies required our application. for example : installed JVM or CLR.

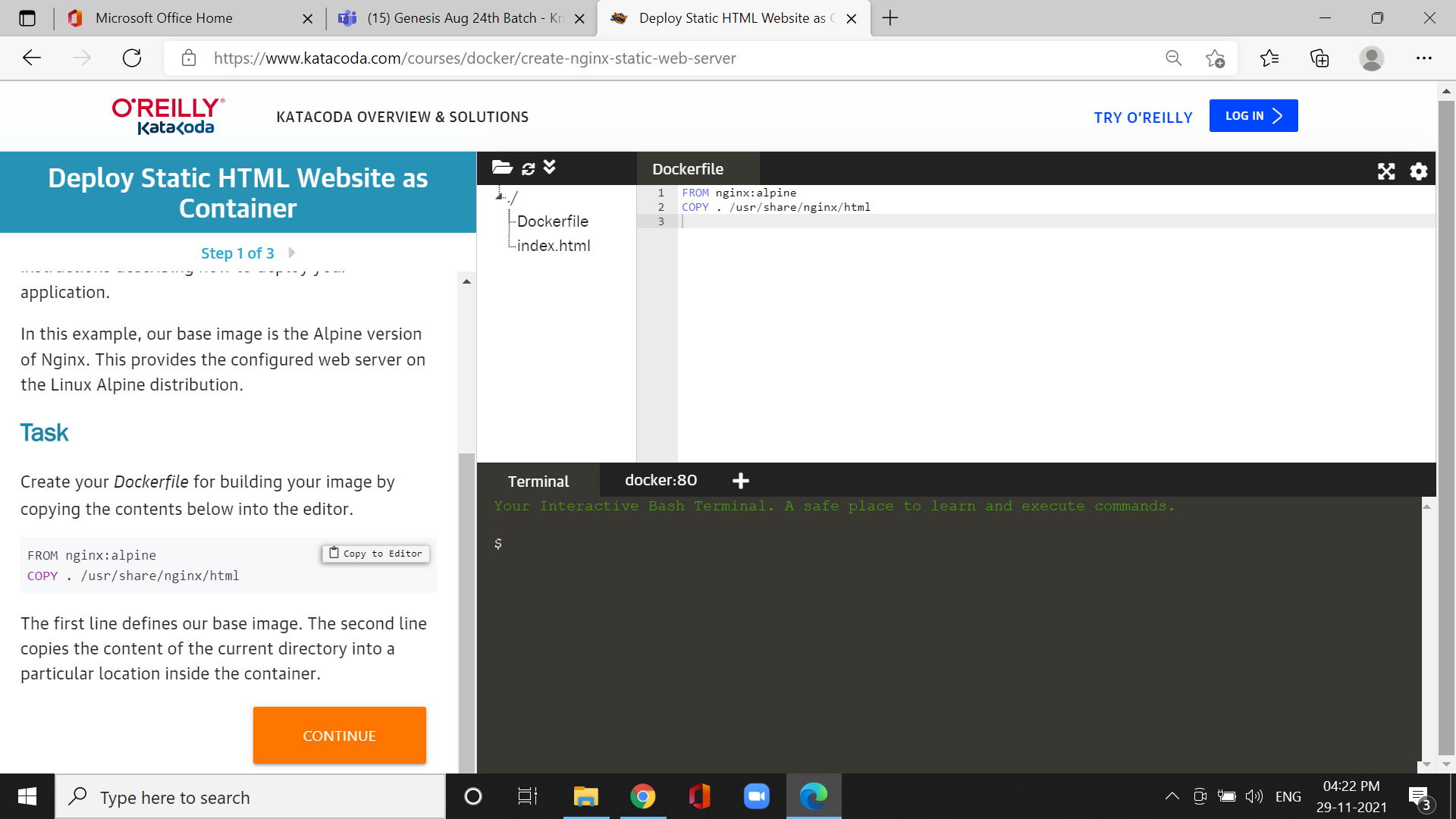
\*Base image is defined as an instruction in the Dockerfile.

\*Docker Images are built based on the contents of a Dockerfile.so dockerfile is a list of instructions describing how to deploy the application.

\*For Example : Base image is the Alpine version of Nginx

Cmd : FROM nginx:alpine

COPY . /usr/share/nginx/html



\*The first line defines our base image.

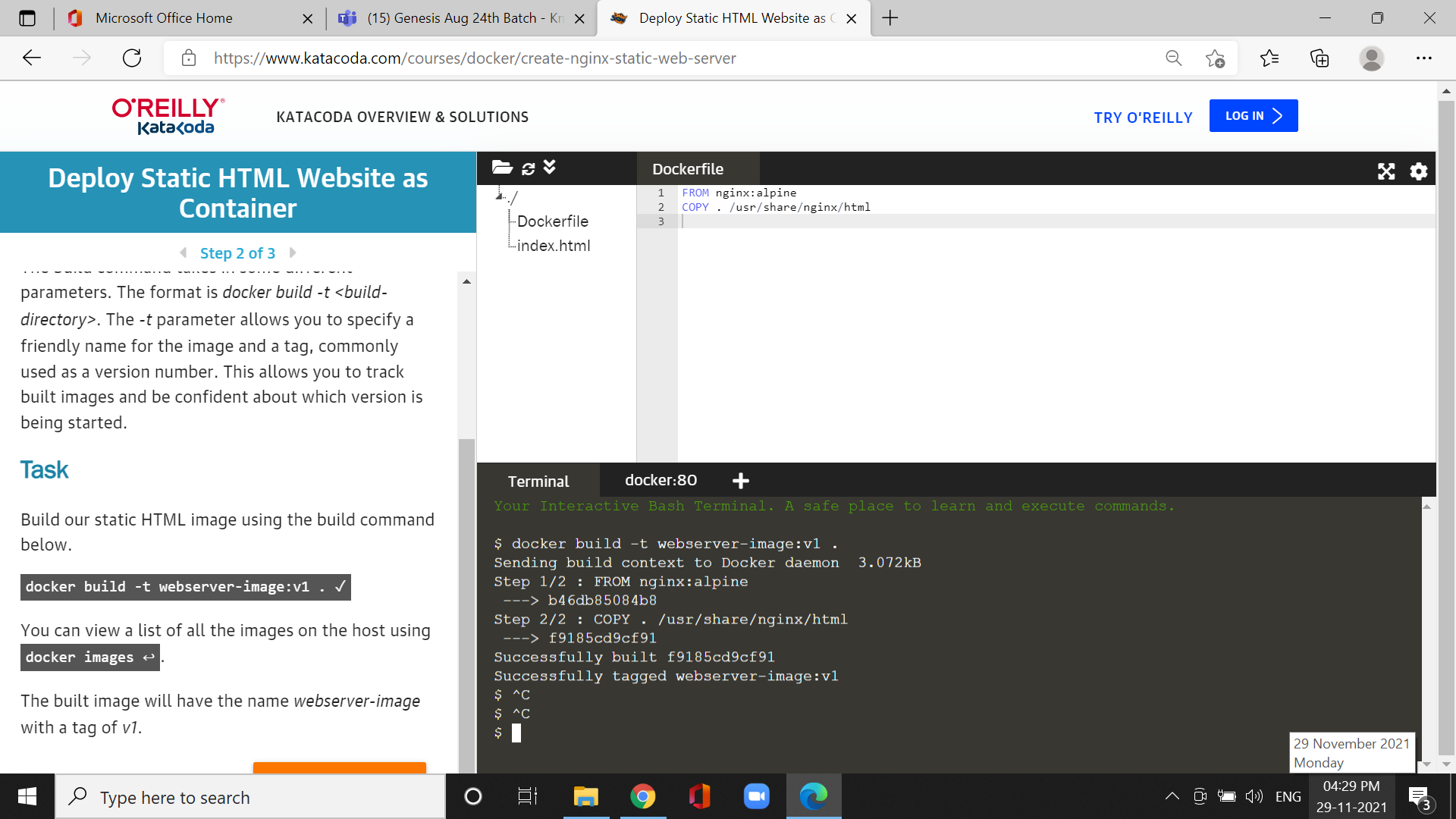
\*The second line copies the content of the current directory into a particular location inside the container.

**Step2: Build docker image**

\*The Dockerfile is used by the Docker CLI *build* command.

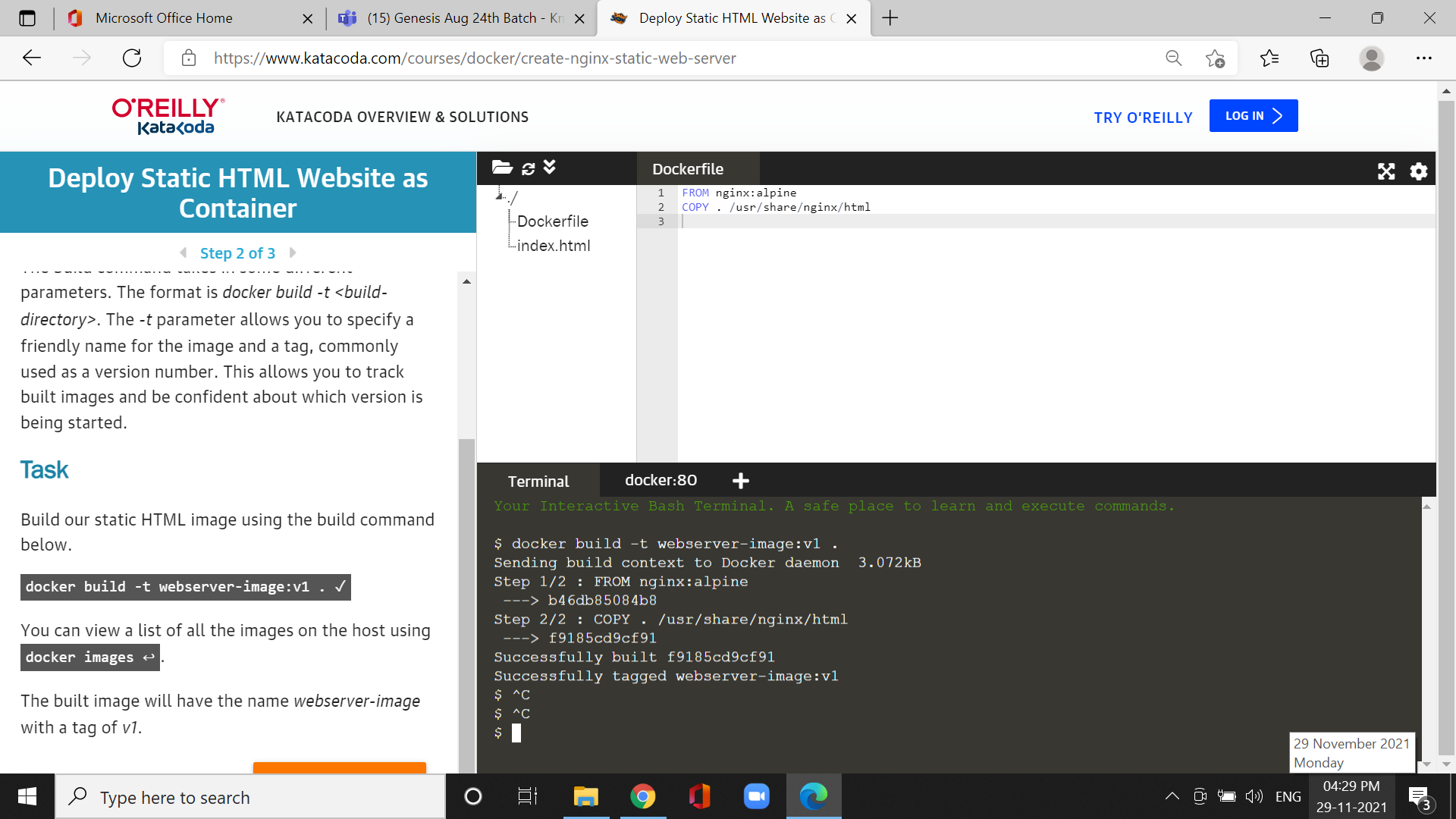
\*The *build* command executes each instruction within the Dockerfile.

Cmd: docker build -t webserver-image:v1



\*Build our static HTML image using the build command

Cmd: docker image



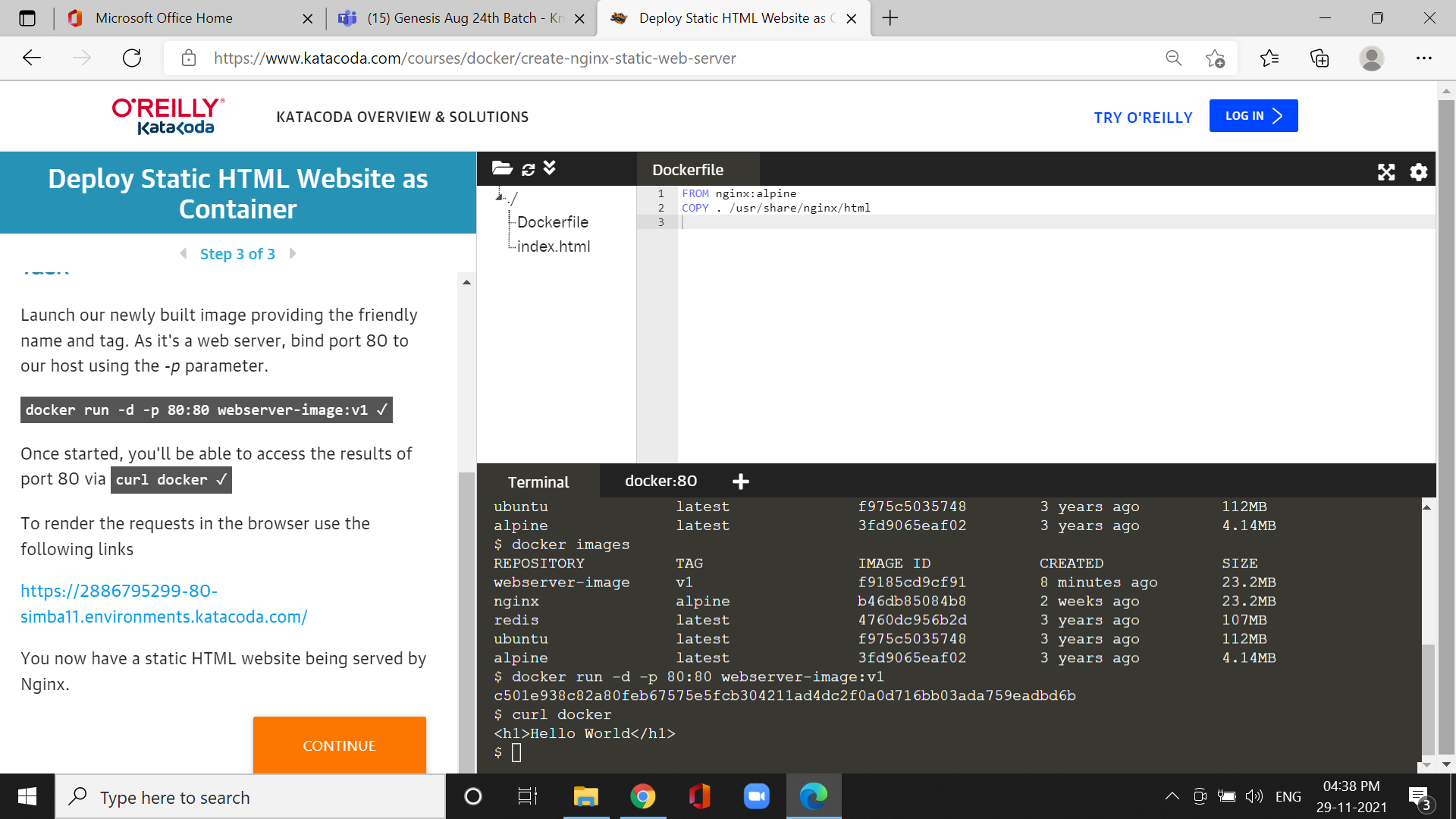
\*it can used for a list of all the images on the host.

**Step 3: Run**

\*built image providing the friendly name and tag.

\* As it's a web server, bind port 80 to our host using the *-p* parameter.

Cmd: docker run -d -p 80:80 webserver-image:v1 &cmd: curl docker



\*To access the results of port 80 via.