# Docker - Create & Manage volumes

## **Tutorial**

Expand guided task

This exercise will get you to create and manage volumes in Docker.

You will be able to see that data can be persisted after a container is destroyed and how a single volume can be used across multiple containers at the same time.

NGINX will serve as another good tool demonstrate this, we will create a simple webpage for NGINX to serve and store it on the volume.

If the NGINX container is stopped and removed, when it is created again the same webpage will be served.

If another NGINX server is created with the volume, the same webpage will be accessible from that instance of NGINX as well.

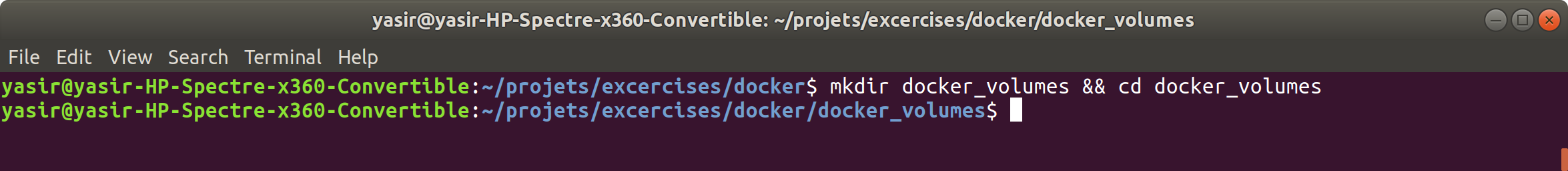
**Step 1: Create a new directory**

Create a new directory called docker\_volumes, execute the following command for this:

**$ mkdir docker\_volumes**

Change the directory by executing:

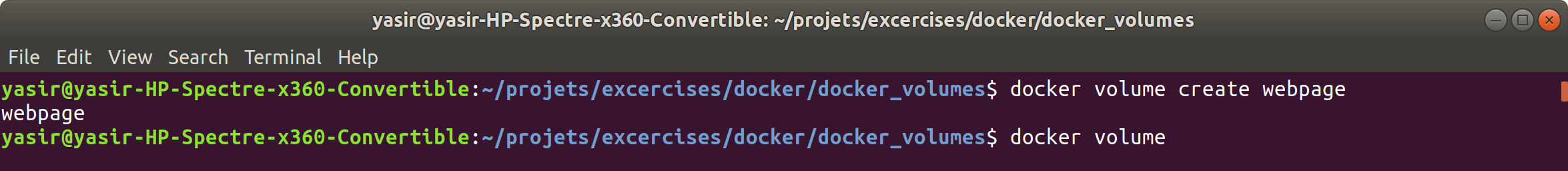
**$ cd docker\_volumes**

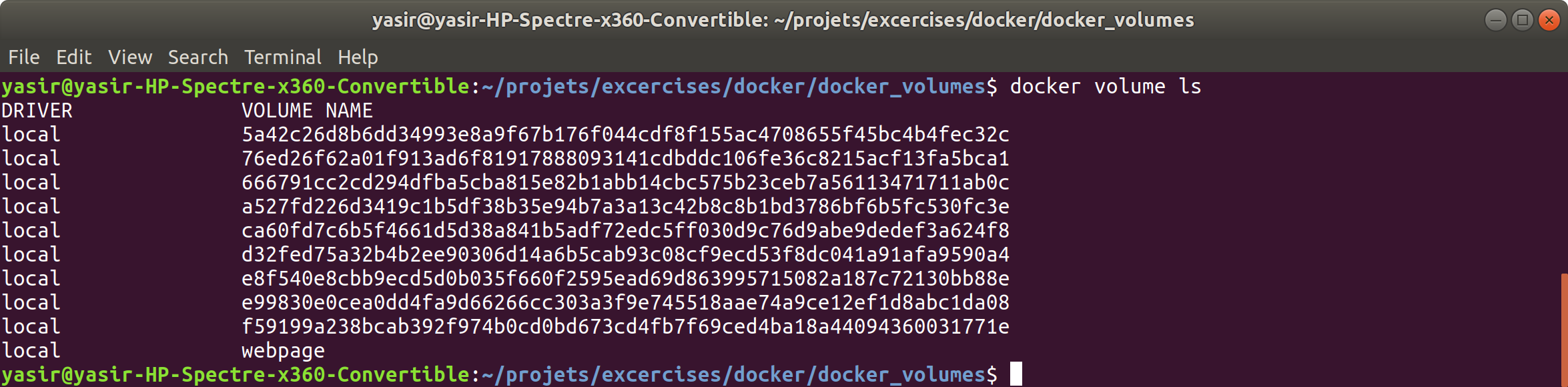
****

**Step 2: Creating a volume**

Create a new volume called webpage, the command to do this is:

**$ docker volume create webpage**

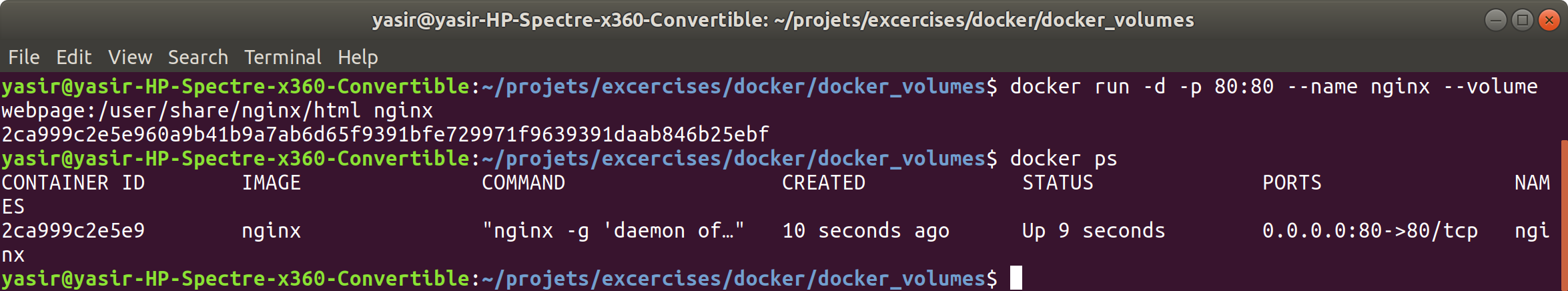
****

****

**Step 3: Create an NGINX Container**

Create an NGINX container, expose port **80**, mount the webpage volume to **/usr/share/nginx/html**, execute the following command:

**$ docker run -d -p 80:80 --name nginx --volume webpage:/usr/share/nginx/html nginx**

****

**Step 4: Create a Webpage**

Let’s make a change to the default NGINX home page, so it is our page.

You will need to connect to the container that you created and edit the index.html file, replacing the entire contents with our one.

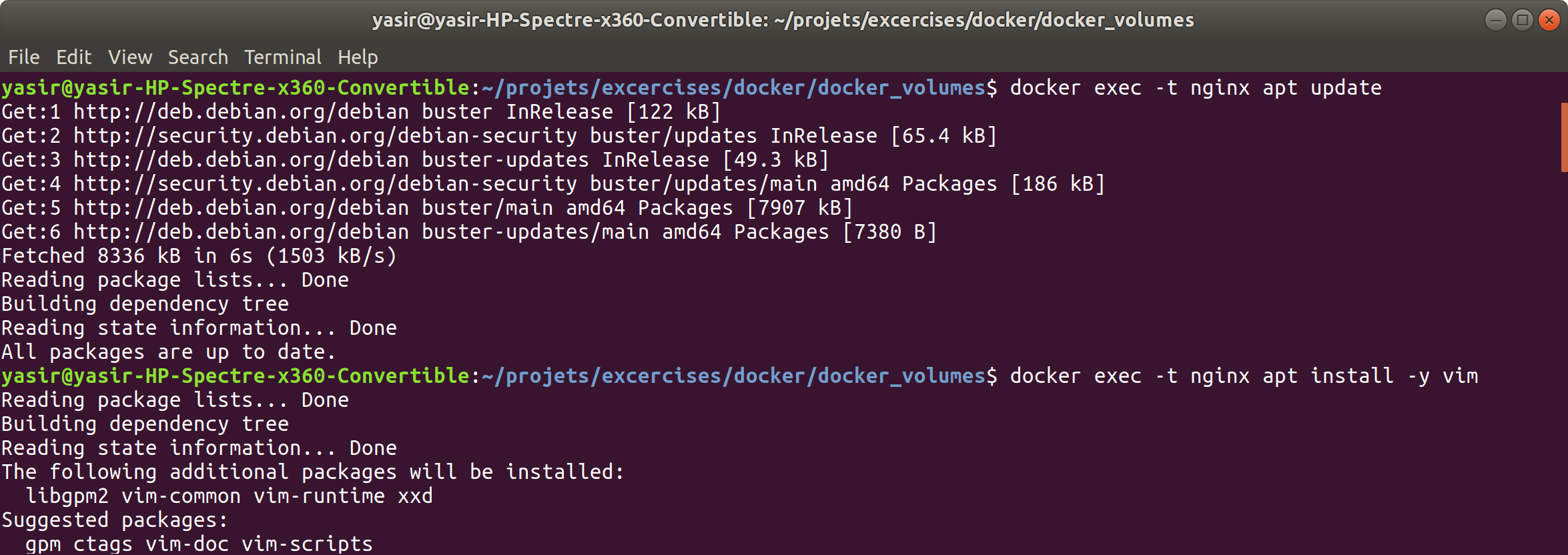
For that however we need a text editor such as vim or nano.

The latest NGINX Docker image is based on Debian so you can use the apt package manager to install one of these.

Install dependencies for text editor:

**$ docker exec -it nginx apt update**

**$ docker exec -it nginx apt install -y nano**



Open the index.html file with a nano text editor by executing:

**$ docker exec -it nginx nano /usr/share/nginx/html/index.html**

Place the following into the file, use **SHIFT + INSERT**:

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<title>NGINX</title>

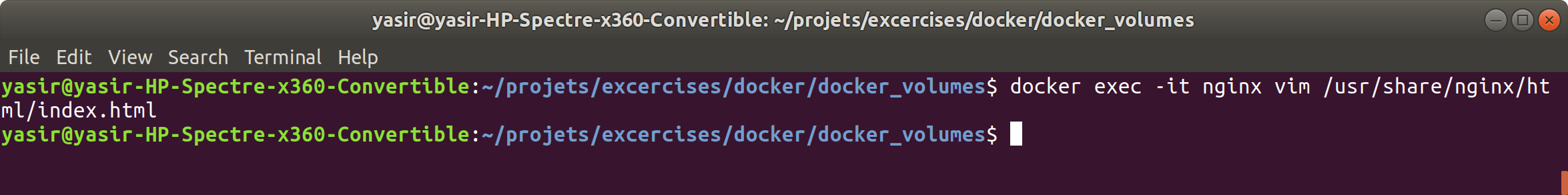
</head>

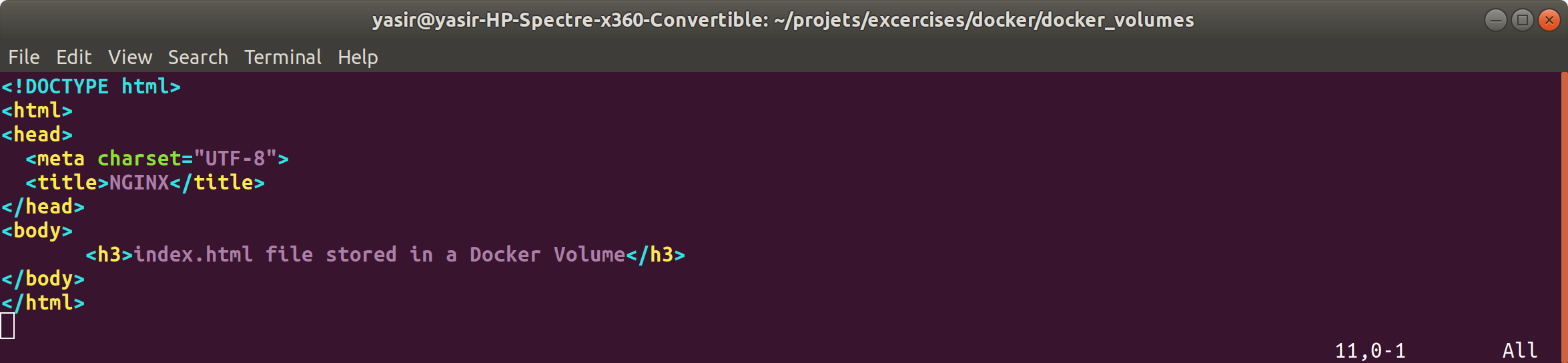
<body>

<h3>index.html file stored in a Docker Volume</h3>

</body>

</html>





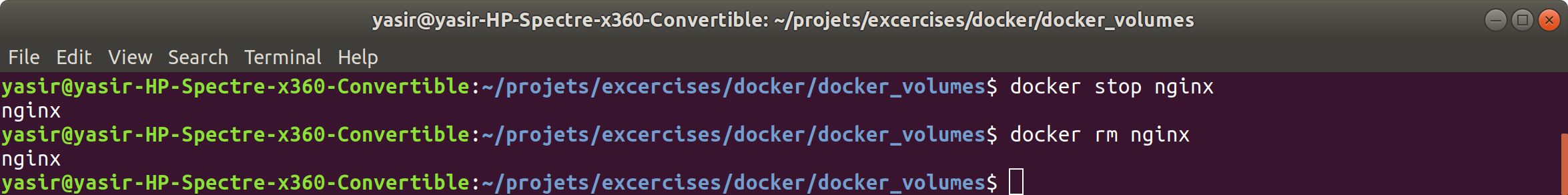
**Step 5: Destroy and recreate the container**

Stop the container

**$ docker stop nginx**

Remove the container

**$ docker rm nginx**

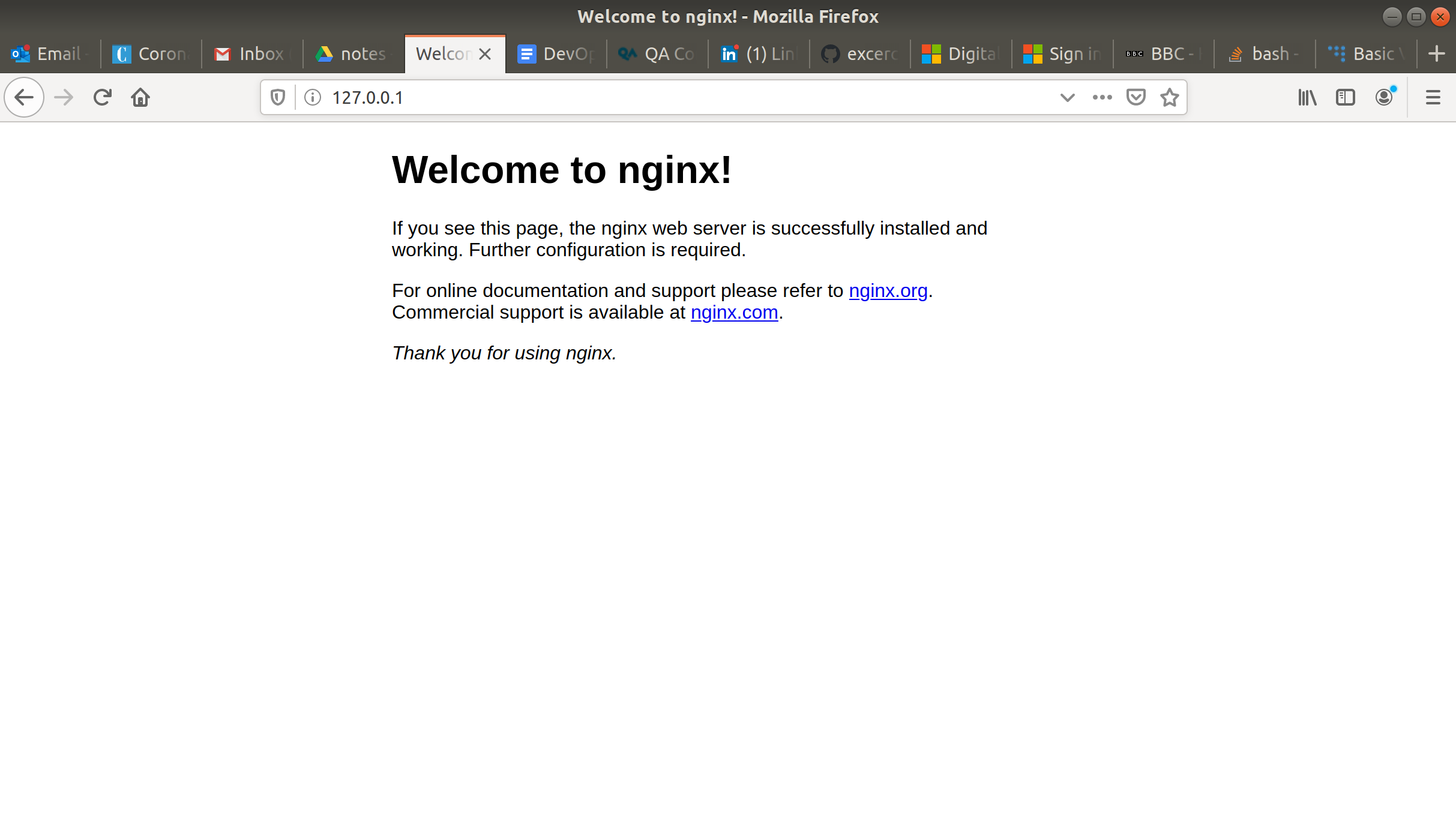
****

**Step 5: Recreate the container**

Recreate the container, expose port **80**, mount the webpage volume to **/usr/share/nginx/html**, execute the following command:

**$ docker run -d -p 80:80 --name nginx --volume webpage:/usr/share/nginx/html nginx**

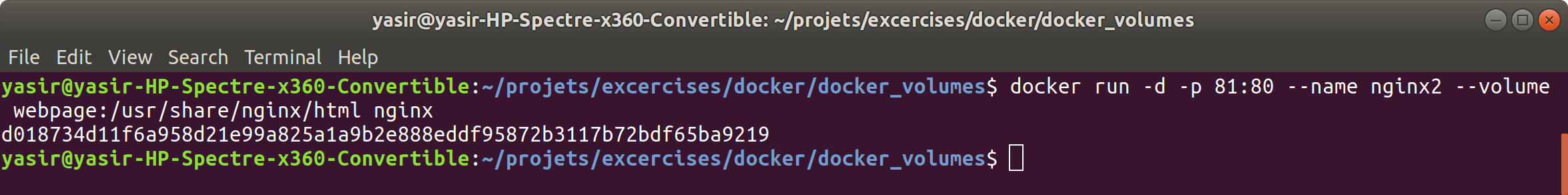
As you can see the webpage is still the same as it's coming from the volume rather than the default NGINX page like you would have expected.

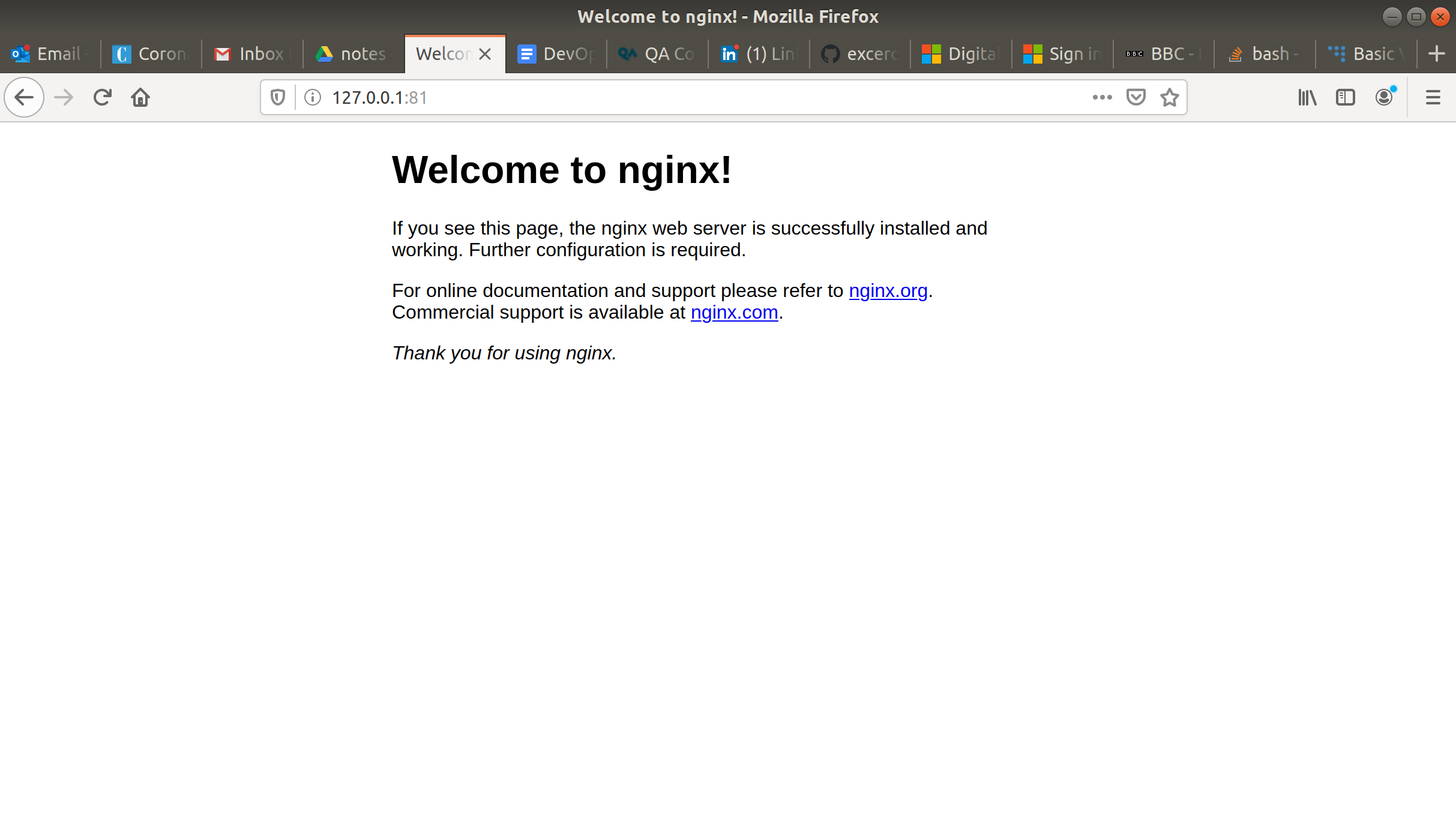
****

**Step 5: Start Another NGINX Container**

Create another NGINX container using the same volume configurations and publish it to a different port on your host, when you connect to that instance of NGINX you will see your index.html there as well.

**$ docker run -d -p 81:80 --name nginx2 --volume webpage:/usr/share/nginx/html nginx**

****

****

**Step 6: Make a Change to the Webpage**

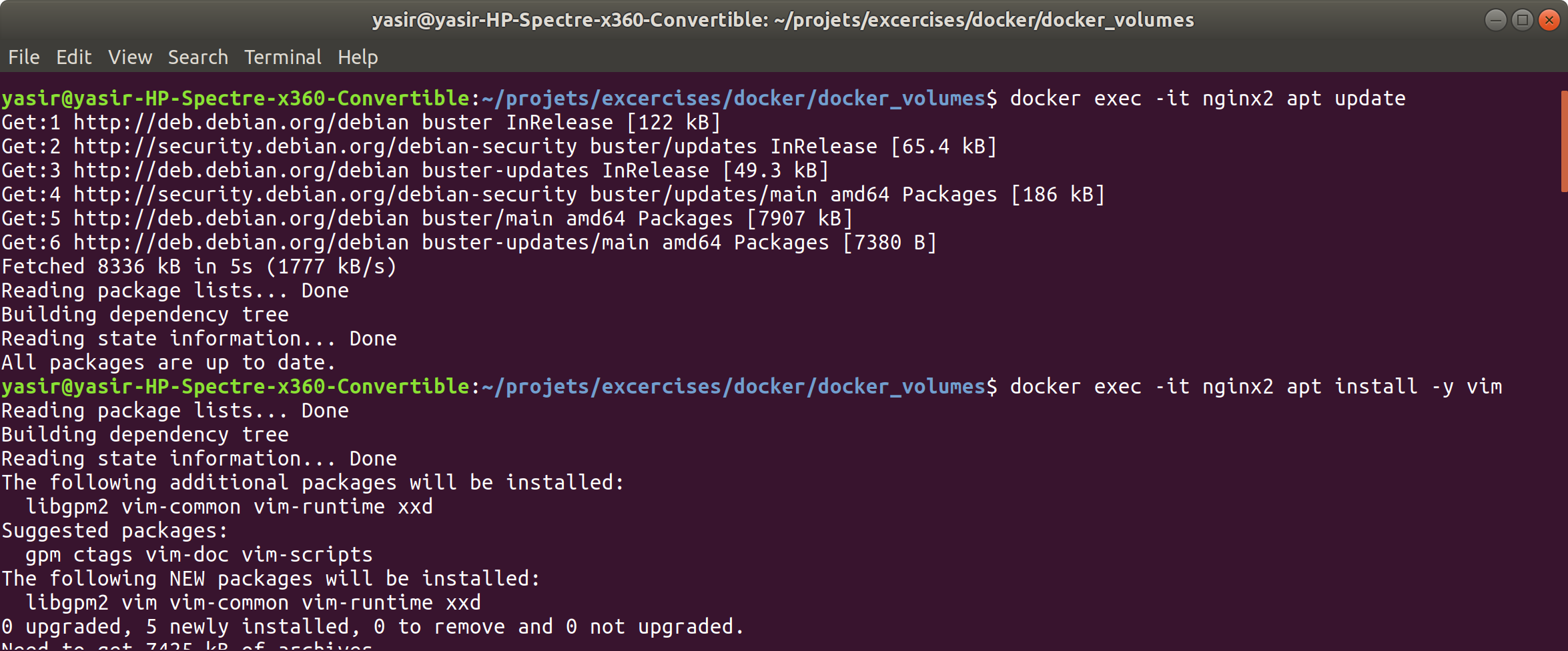
Connect to the second NGINX container that you created, install a text editor and make a change to the **/usr/share/nginx/html/index.html** file inside the <h3></h3> tags.

The changes you make should be reflected on both of the containers when you make a HTTP request to them.

Install dependencies for text editor:

**$ docker exec -it nginx2 apt update**

**$ docker exec -it nginx2 apt install -y nano**



Open the index.html file with a nano text editor by executing:

**$ docker exec -it nginx2 nano /usr/share/nginx/html/index.html**

Place the following into the file, use **SHIFT + INSERT**:

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<title>NGINX</title>

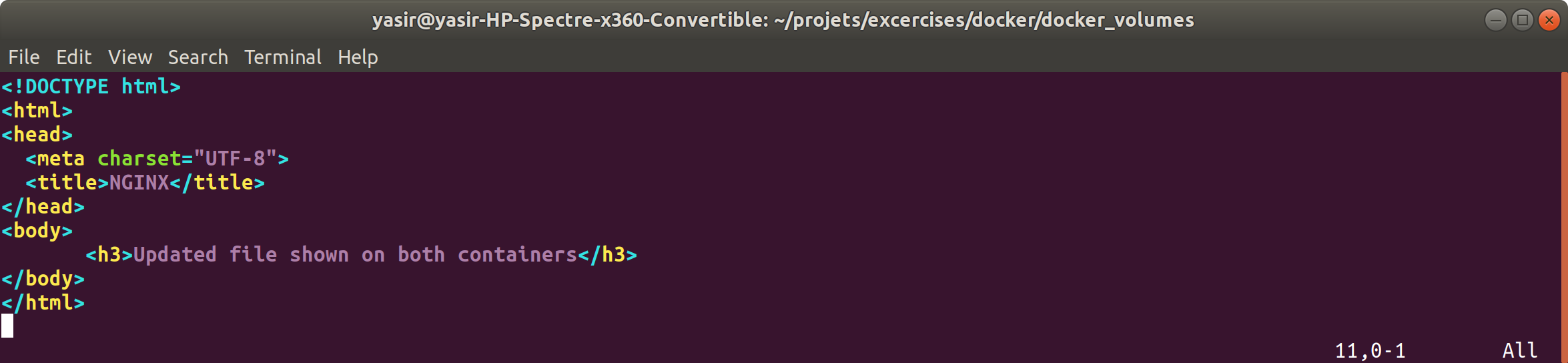
</head>

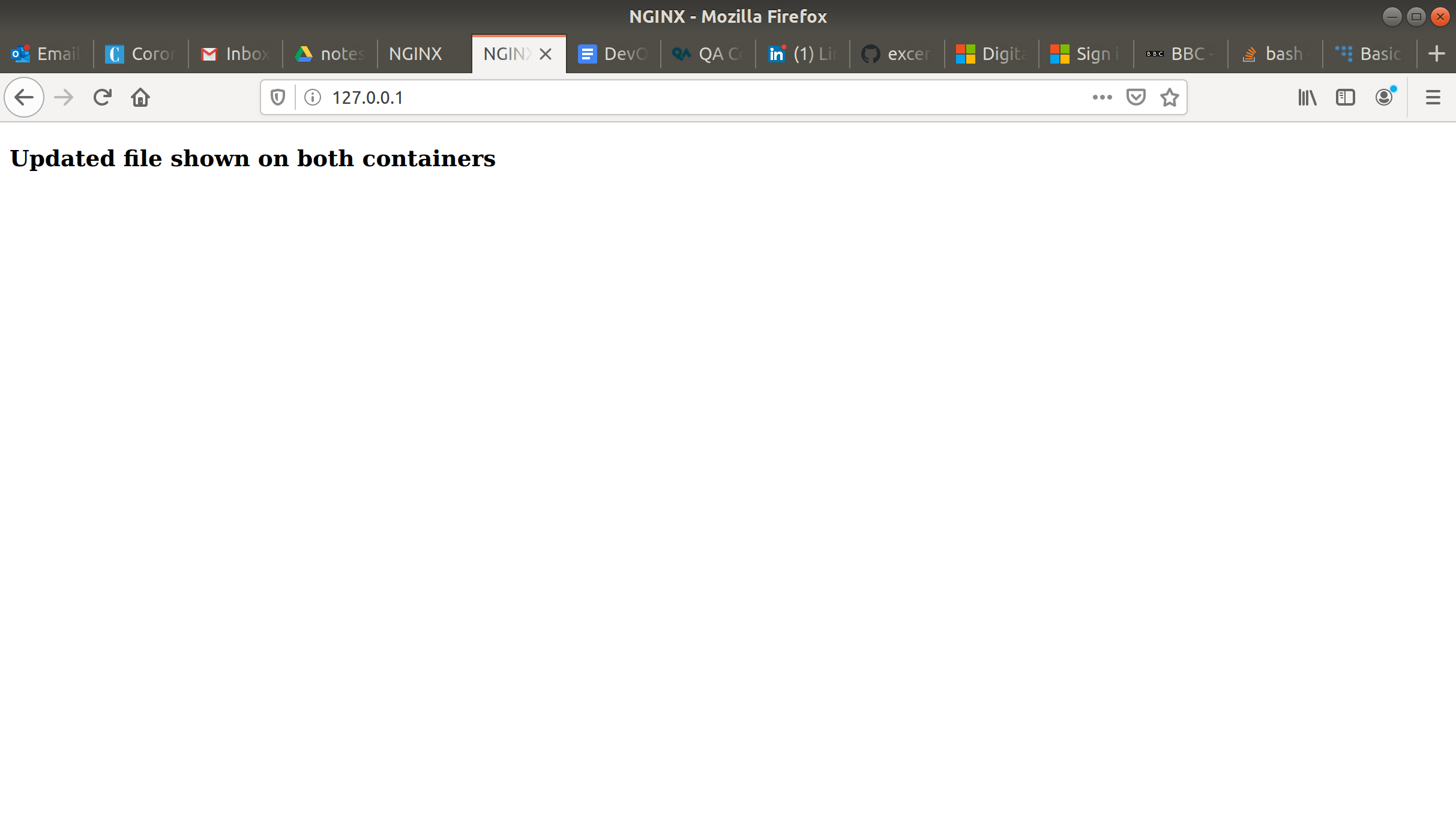
<body>

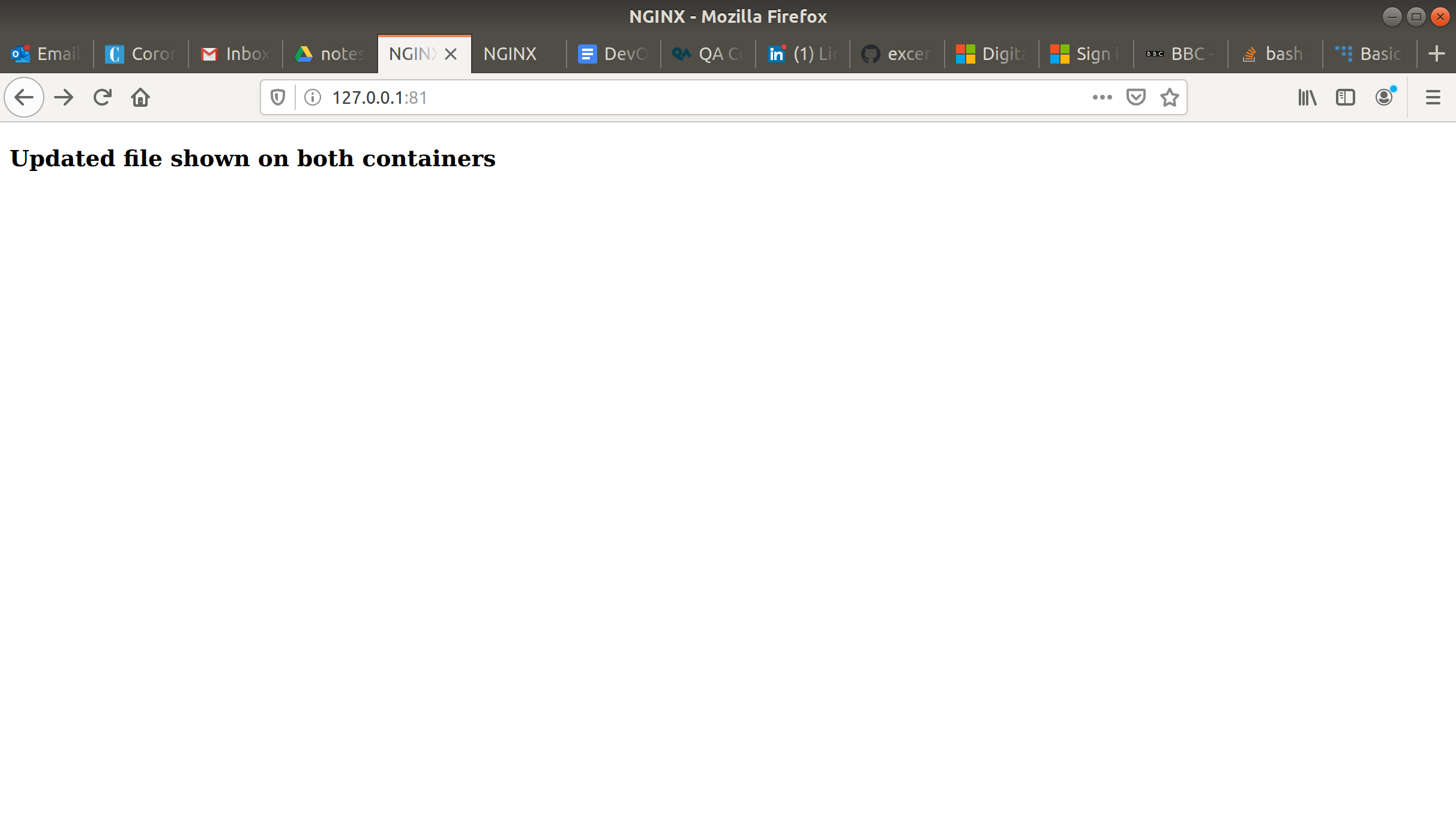
<h3>Updated file shown on both containers</h3>

</body>

</html>







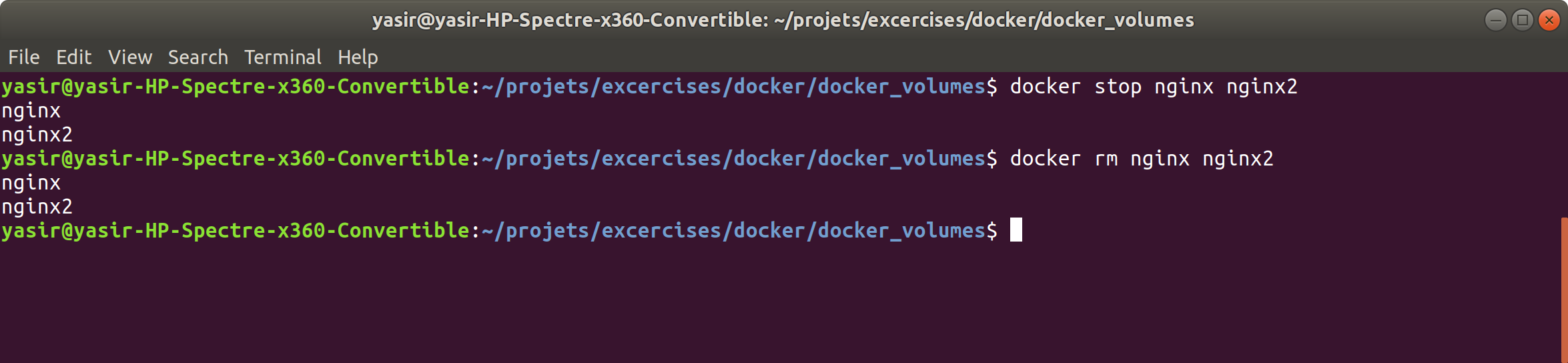
**Step 7: Stop and remove container**

Stop the containers:

**$ docker stop nginx nginx2**

Remove containers:

**$ docker rm nginx nginx2**

****

**Step 8: Remove NGINX image**

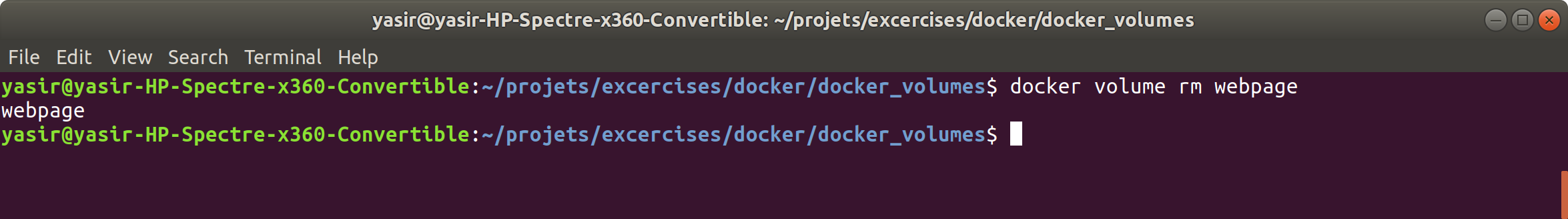
Remove the NGINX image:

**$ docker rmi nginx**

**Step 9: Remove volume**

Remove volume:

**$ docker volume rm webpage**

****