

Docker Image Upload & Container Creation((AWS Linux / EC2)):

Correct way (step by step on EC2)

Login into your Amazon Ec2 instances using cmd

Step 1: Create a working directory

```
mkdir nginx-docker
```

```
cd nginx-docker
```

Step 2: Create the Dockerfile

```
vi Dockerfile
```

Press **i** (insert mode) and paste:

```
FROM nginx
```

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

Save & exit:

```
ESC → :wq → Enter
```

Step 3: Create index.html

```
vi index.html
```

Paste:

```
<!DOCTYPE html>
```

```
<html>  
<head>  
  <title>Docker on EC2</title>  
</head>  
<body>  
  <h1>Hello from Nginx inside Docker 🚀</h1>  
  <p>Running on AWS EC2</p>  
</body>  
</html>
```

Save & exit.



Step 4: Build the Docker image

```
docker build -t my-nginx-image .
```

You should see steps like:

Step 1/2 : FROM nginx

Step 2/2 : COPY index.html ...

Successfully built ...



Step 5: Run the container

```
docker run -d -p 80:80 --name my-nginx my-nginx-image
```

Step 6: Access from browser

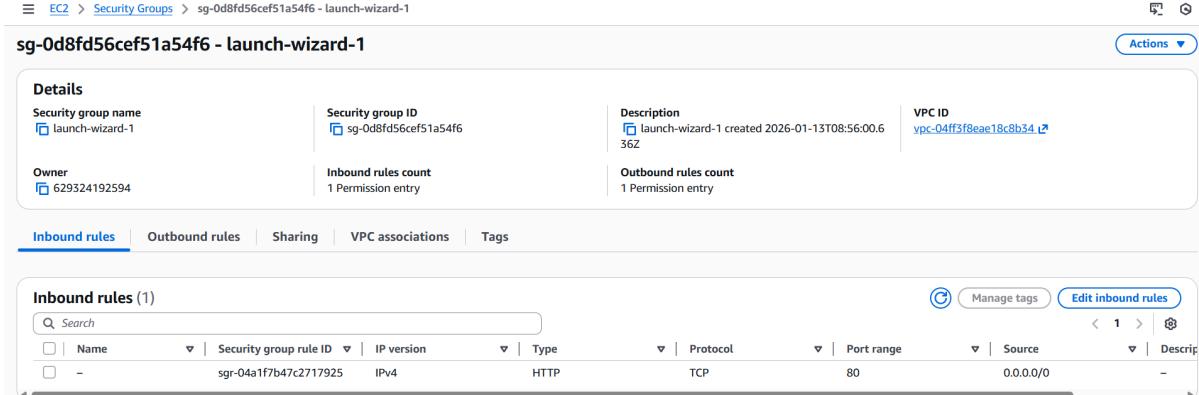
Open your browser and visit:

http://<EC2-Public-IP> i.e **http://13.53.198.16/**

 Make sure EC2 **Security Group** allows:

- **Inbound → HTTP → Port 80 → 0.0.0.0/0**

If not click on Edit inbound rules



The screenshot shows the AWS Security Groups console. At the top, the navigation path is EC2 > Security Groups > sg-0d8fd56cef51a54f6 - launch-wizard-1. Below the title, there are four main sections: Details, Inbound rules, Outbound rules, Sharing, VPC associations, and Tags. The Details section displays the security group's name (sg-0d8fd56cef51a54f6), ID, owner (629324192594), and various counts for inbound and outbound rules. The Inbound rules section is currently selected and shows one rule: a TCP port 80 rule from 0.0.0.0/0 to the security group itself. The rule is named 'sgr-04a1f7b47c2717925' and is associated with IPv4 and HTTP.

OUTPUT:

```
[ec2-user@ip-172-31-20-246 ~]$ ls
Dockerfile
[ec2-user@ip-172-31-20-246 ~]$ mkdir nginx-docker
[ec2-user@ip-172-31-20-246 ~]$ cd nginx-docker
[ec2-user@ip-172-31-20-246 nginx-docker]$ nano Dockerfile
[ec2-user@ip-172-31-20-246 nginx-docker]$ nano index.html
[ec2-user@ip-172-31-20-246 nginx-docker]$ docker build -t my-nginx-image
ERROR: "docker buildx build" requires exactly 1 argument.
See 'docker buildx build --help'.

Usage: docker buildx build [OPTIONS] PATH | URL | -
Start a build
[ec2-user@ip-172-31-20-246 nginx-docker]$ docker build -t my-nginx-image .
[+] Building 0.3s (7/7) FINISHED                                            docker:default
=> [internal] load build definition from Dockerfile                      0.0s
=> => transferring dockerfile: 157B                                         0.0s
=> [internal] load metadata for docker.io/library/nginx:latest           0.0s
=> [internal] load .dockerrcignore                                       0.0s
=> => transferring context: 2B                                           0.0s
=> [internal] load build context                                         0.1s
=> => transferring context: 2548                                         0.0s
=> [1/2] FROM docker.io/library/nginx:latest                           0.1s
=> [2/2] COPY index.html /usr/share/nginx/html/index.html             0.0s
=> exporting to image                                                 0.0s
=> => exporting layers                                              0.0s
=> => writing image sha256:a3eadfec852575b6cd4d7d5ed895ff6c25e0497827d8723c215b7c33d8a3a1a 0.0s
=> => naming to docker.io/library/my-nginx-image                         0.0s
[ec2-user@ip-172-31-20-246 nginx-docker]$ docker run -d -p 80:80 --name my-nginx my-nginx-image
40be5ed37d427260ce4accbcd853ff80b7b8e3e5bc532005d1e901d020322155
[ec2-user@ip-172-31-20-246 nginx-docker]$ docker ps
CONTAINER ID   IMAGE      COMMAND       CREATED     STATUS      PORTS
NAMES
40be5ed37d42   my-nginx-image   "/docker-entrypoint..."   8 seconds ago   Up 7 seconds   0.0.0.0:80->80/tcp
[ec2-user@ip-172-31-20-246 nginx-docker]$
```



Hello from NGINX inside Docker

Running on AWS EC2