

# ECR

-> created a dockerfile with ubuntu as base image,nginx server installed and added few static html pages

FROM ubuntu:latest

## RUN apt-get update

```
RUN apt-get install -y nginx
```

COPY first.html /var/www/html/first.html

COPY second.html /var/www/html/second.html

COPY third.html /var/www/html/third.html

COPY fourth.html /var/www/html/fourth.html

COPY fifth.html /var/www/html/fifth.html

EXPOSE 80

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

->

```

~\ ##### Amazon Linux 2023
~ \ #####\
~ \###|
~ \#/ https://aws.amazon.com/linux/amazon-linux-2023
~ V~' '->
~~
~~_./
~/m/'

Last login: Thu Apr 13 05:04:54 2023 from 18.206.107.28
[ec2-user@ip-172-31-84-6 ~]$ ls
communicate ecrassign forhttpd
[ec2-user@ip-172-31-84-6 ~]$ cd ecrassign/
[ec2-user@ip-172-31-84-6 ecrassign]$ ls
Dockerfile fifth.html first.html fourth.html second.html third.html
[ec2-user@ip-172-31-84-6 ecrassign]$ docker ps
CONTAINER ID   IMAGE      COMMAND                  CREATED        STATUS        PORTS                               NAMES
62bcf076b973   assign2    "httpd-foreground"       18 hours ago   Up 18 hours   80/tcp, 0.0.0.0:81->81/tcp, :::81->81/tcp   jovial_raman
7e91da39e1be   assign1    "nginx -g 'daemon off..." 18 hours ago   Up 18 hours   0.0.0.0:80->80/tcp, :::80->80/tcp          clever_neumann
[ec2-user@ip-172-31-84-6 ecrassign]$ nano Dockerfile
[ec2-user@ip-172-31-84-6 ecrassign]$ ls
Dockerfile fifth.html first.html fourth.html second.html third.html

```

i-0b5931fcc6f9ef3f4 (docker2)

PublicIPs: 18.212.21.224 PrivateIPs: 172.31.84.6

## Dockerfile and HTML files

```
GNU nano 5.8 second.html
<!DOCTYPE html>
<html>
<head>
  <title>AWS Services</title>
</head>
<body>
  <h1>AWS Services</h1>
  <p>Here are some popular AWS services:</p>

  <h2>Amazon S3</h2>
  
  <p>Amazon S3 is a simple storage service that offers industry-leading durability, availability, and scalability. </p>

  <h2>Amazon EC2</h2>
  
  <p>Amazon EC2 provides resizable compute capacity in the cloud, making it easy to scale applications up and down as needed. </p>

  <h2>Amazon RDS</h2>

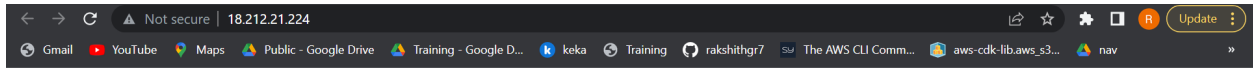
  [ Read 23 lines ]
^G Help      ^O Write Out  ^W Where Is   ^K Cut        ^T Execute    ^C Location   M-U Undo     M-A Set Mark  M-I To Bracket
^X Exit      ^R Read File  ^N Replace    ^U Paste      ^J Justify    ^_ Go To Line M-E Redo     M-G Copy     ^Q Where Was
```

i-0b5931fcc6f9ef3f4 (docker2) X

PublicIPs: 18.212.21.224 PrivateIPs: 172.31.84.6

## Basic HTML page

- then build docker -docker build . -t assign1
- docker run -p 80:80 -d assign1

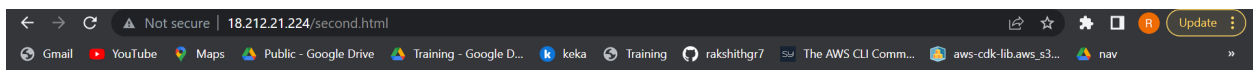


## Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to [nginx.org](http://nginx.org).  
Commercial support is available at [nginx.com](http://nginx.com).

*Thank you for using nginx.*



## AWS Services

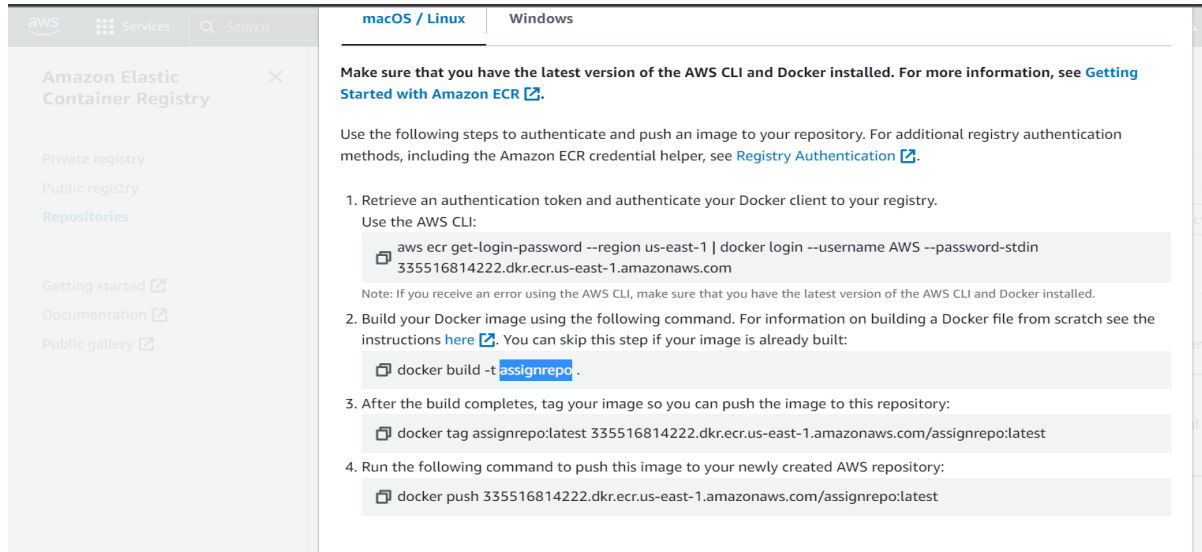
Here are some popular AWS services:

### Amazon S3



- Nginx server is up and running on public ip 18.212.21.224
- We can see one of the HTML page up and running

Created a ECR repository named-assignpro- then pushed this newly created image to it



The screenshot shows the AWS CLI documentation for pushing an image to ECR. It includes instructions for macOS/Linux and Windows, and provides the following steps:

- Retrieve an authentication token and authenticate your Docker client to your registry. Use the AWS CLI:

```
aws ecr get-login-password --region us-east-1 | docker login --username AWS --password-stdin 335516814222.dkr.ecr.us-east-1.amazonaws.com
```

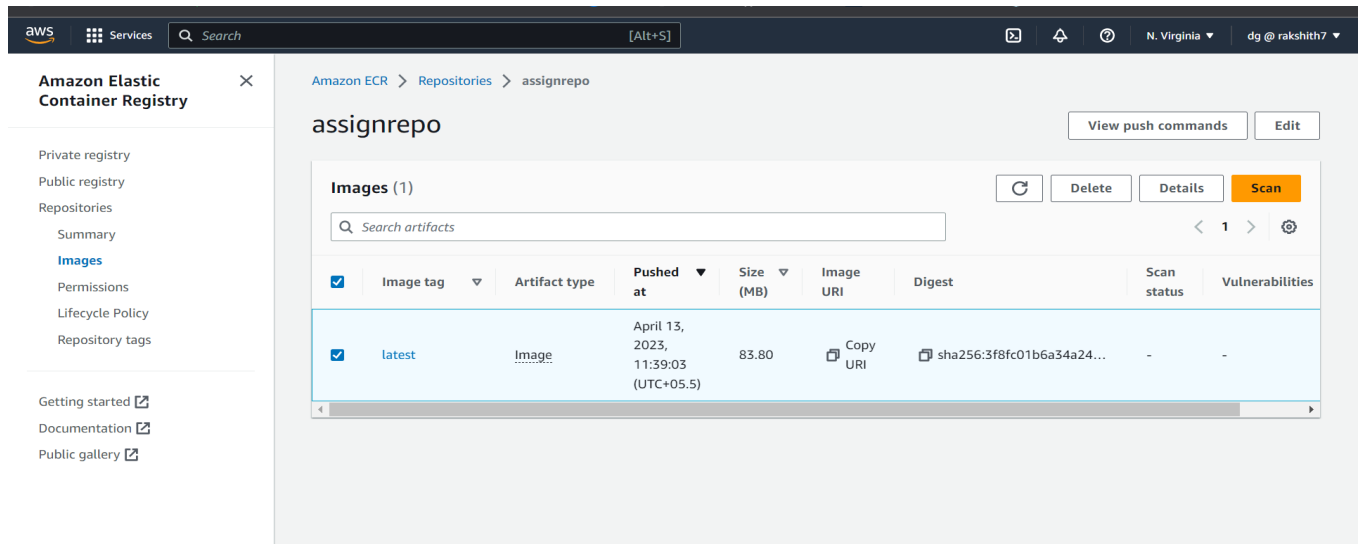
Note: If you receive an error using the AWS CLI, make sure that you have the latest version of the AWS CLI and Docker installed.
- Build your Docker image using the following command. For information on building a Docker file from scratch see the instructions [here](#). You can skip this step if your image is already built:

```
docker build -t assignrepo .
```
- After the build completes, tag your image so you can push the image to this repository:

```
docker tag assignrepo:latest 335516814222.dkr.ecr.us-east-1.amazonaws.com/assignrepo:latest
```
- Run the following command to push this image to your newly created AWS repository:

```
docker push 335516814222.dkr.ecr.us-east-1.amazonaws.com/assignrepo:latest
```

## Steps for pushing the image



The screenshot shows the AWS ECR console for the 'assignrepo' repository. The 'Images (1)' section displays a table with the following data:

	Image tag	Artifact type	Pushed at	Size (MB)	Image URI	Digest	Scan status	Vulnerabilities
<input checked="" type="checkbox"/>	latest	Image	April 13, 2023, 11:39:03 (UTC+05.5)	83.80	Copy URI	sha256:3f8fc01b6a34a24...	-	-

## Image in repository

## 2. Build another image httpd and configure it with 4-5 static web pages and Should expose the service on port 80.

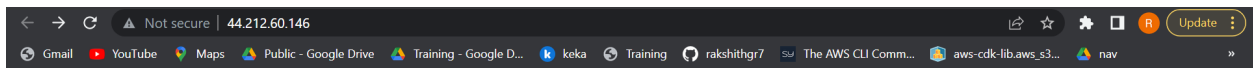
Dockerfile-

```
FROM httpd:latest
COPY first.html /usr/local/apache2/htdocs/first.html
COPY second.html /usr/local/apache2/htdocs/second.html
COPY third.html /usr/local/apache2/htdocs/third.html
EXPOSE 80
CMD ["httpd-foreground"]
```

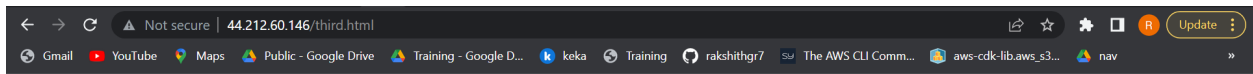
Then build

```
docker build . -t assign2
docker run -p 80:80 -d assign2
```

- Then tried accessing it with public ip-44.212.60.146
- And server was up and running



**It works!**



## Contact Us

If you have any questions or comments, please get in touch with us using the information below:

### Address

123 Main Street  
Anytown, banglore

### Phone

999897758768

### Email

[info@example.com](mailto:info@example.com)

### Social Media

- [Facebook](#)
- [Twitter](#)
- [Instagram](#)

One of the page