Dockerizing a Plain HTML Page With Nginx

1. Launch EC2 instance for amazon linux machine.
2. Update using “**sudo yum update -y** ” and install nginx using **“sudo yum install nginx -y”**

```
[ec2-user@ip-10-0-0-233 var]$
[ec2-user@ip-10-0-0-233 var]$
[ec2-user@ip-10-0-0-233 var]$ sudo yum update -y
Amazon Linux 2023 Kernel Livepatch repository                                52 kB/s |  11 kB     00:00
Dependencies resolved.
Nothing to do.
Complete!
[ec2-user@ip-10-0-0-233 var]$ sudo yum install nginx -y
Last metadata expiration check: 0:00:44 ago on Sun Jan  5 18:51:08 2025.
Dependencies resolved.
==========================================================================================================
 Package                  Architecture     Version                        Repository              Size
==========================================================================================================
Installing:
 nginx                    x86_64           1:1.26.2-1.amzn2023.0.1        amazonlinux             33 k
Installing dependencies:
 generic-logos-httpd      noarch           18.0.0-12.amzn2023.0.3         amazonlinux             19 k
 gperftools-libs          x86_64           2.9.1-1.amzn2023.0.3           amazonlinux            308 k
 libunwind                x86_64           1.4.0-5.amzn2023.0.2           amazonlinux             66 k
 nginx-core               x86_64           1:1.26.2-1.amzn2023.0.1        amazonlinux            670 k
 nginx-filesystem         noarch           1:1.26.2-1.amzn2023.0.1        amazonlinux            9.9 k
 nginx-mimetypes          noarch           2.1.49-3.amzn2023.0.3          amazonlinux             21 k

Transaction Summary
==========================================================================================================
Install  7 Packages

Total download size: 1.1 M
Installed size: 3.6 M
Downloading Packages:
(1/7): libunwind-1.4.0-5.amzn2023.0.2.x86_64.rpm                             673 kB/s |  66 kB     00:00
(2/7): gperftools-libs-2.9.1-1.amzn2023.0.3.x86_64.rpm                       2.6 MB/s | 308 kB     00:00
(3/7): generic-logos-httpd-18.0.0-12.amzn2023.0.3.noarch.rpm                 147 kB/s |  19 kB     00:00
(4/7): nginx-1.26.2-1.amzn2023.0.1.x86_64.rpm                                900 kB/s |  33 kB     00:00
(5/7): nginx-core-1.26.2-1.amzn2023.0.1.x86_64.rpm                            21 MB/s | 670 kB     00:00
(6/7): nginx-filesystem-1.26.2-1.amzn2023.0.1.noarch.rpm                     469 kB/s | 9.9 kB     00:00
(7/7): nginx-mimetypes-2.1.49-3.amzn2023.0.3.noarch.rpm                      965 kB/s |  21 kB     00:00
----------------------------------------------------------------------------------------------------------
Total                                                                        4.9 MB/s | 1.1 MB     00:00
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing        :                                                                                   1/1
  Running scriptlet: nginx-filesystem-1:1.26.2-1.amzn2023.0.1.noarch                                   1/7
  Installing       : nginx-filesystem-1:1.26.2-1.amzn2023.0.1.noarch                                   1/7
  Installing       : nginx-mimetypes-2.1.49-3.amzn2023.0.3.noarch                                      2/7
  Installing       : libunwind-1.4.0-5.amzn2023.0.2.x86_64                                             3/7
  Installing       : gperftools-libs-2.9.1-1.amzn2023.0.3.x86_64                                       4/7
  Installing       : nginx-core-1:1.26.2-1.amzn2023.0.1.x86_64                                         5/7
  Installing       : generic-logos-httpd-18.0.0-12.amzn2023.0.3.noarch                                 6/7
  Installing       : nginx-1:1.26.2-1.amzn2023.0.1.x86_64                                              7/7
  Running scriptlet: nginx-1:1.26.2-1.amzn2023.0.1.x86_64                                              7/7
  Verifying        : generic-logos-httpd-18.0.0-12.amzn2023.0.3.noarch                                 1/7
  Verifying        : gperftools-libs-2.9.1-1.amzn2023.0.3.x86_64                                       2/7
  Verifying        : libunwind-1.4.0-5.amzn2023.0.2.x86_64                                             3/7
  Verifying        : nginx-1:1.26.2-1.amzn2023.0.1.x86_64                                              4/7
  Verifying        : nginx-core-1:1.26.2-1.amzn2023.0.1.x86_64                                         5/7
  Verifying        : nginx-filesystem-1:1.26.2-1.amzn2023.0.1.noarch                                   6/7
  Verifying        : nginx-mimetypes-2.1.49-3.amzn2023.0.3.noarch                                      7/7

Installed:
  generic-logos-httpd-18.0.0-12.amzn2023.0.3.noarch  gperftools-libs-2.9.1-1.amzn2023.0.3.x86_64     libunwind-1.4.0-5.amzn2023.0.2.x86_64        nginx-1:1.26.2-1.amzn2023.0.1.x86_64
  nginx-core-1:1.26.2-1.amzn2023.0.1.x86_64          nginx-filesystem-1:1.26.2-1.amzn2023.0.1.noarch  nginx-mimetypes-2.1.49-3.amzn2023.0.3.noarch

Complete!
```

3. Then enable/start the nginx service
   sudo systemctl start nginx
   sudo systemctl enable nginx

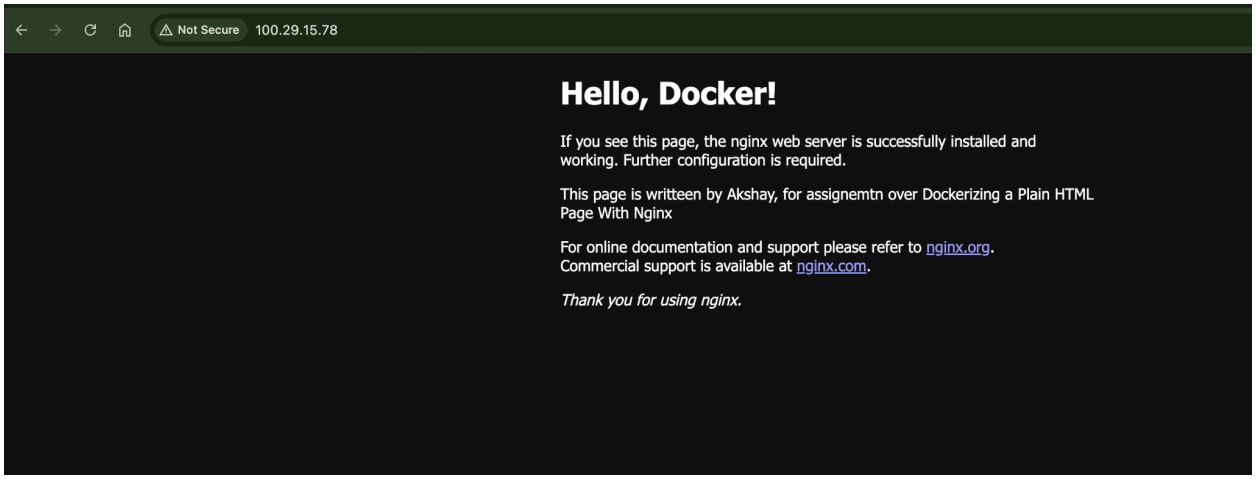
```
[ec2-user@ip-10-0-0-233 var]$
[ec2-user@ip-10-0-0-233 var]$ nginx -v
nginx version: nginx/1.26.2
[ec2-user@ip-10-0-0-233 var]$ sudo systemctl status nginx
○ nginx.service - The nginx HTTP and reverse proxy server
     Loaded: loaded (/usr/lib/systemd/system/nginx.service; disabled; preset: disabled)
     Active: inactive (dead)
[ec2-user@ip-10-0-0-233 var]$ sudo systemctl enable nginx
Created symlink /etc/systemd/system/multi-user.target.wants/nginx.service → /usr/lib/systemd/system/nginx.service.
[ec2-user@ip-10-0-0-233 var]$ sudo systemctl status nginx
○ nginx.service - The nginx HTTP and reverse proxy server
     Loaded: loaded (/usr/lib/systemd/system/nginx.service; enabled; preset: disabled)
     Active: inactive (dead)
[ec2-user@ip-10-0-0-233 var]$ sudo systemctl start nginx
[ec2-user@ip-10-0-0-233 var]$ sudo systemctl status nginx
● nginx.service - The nginx HTTP and reverse proxy server
     Loaded: loaded (/usr/lib/systemd/system/nginx.service; enabled; preset: disabled)
     Active: active (running) since Sun 2025-01-05 18:52:51 UTC; 2s ago
    Process: 25400 ExecStartPre=/usr/bin/rm -f /run/nginx.pid (code=exited, status=0/SUCCESS)
    Process: 25401 ExecStartPre=/usr/sbin/nginx -t (code=exited, status=0/SUCCESS)
    Process: 25402 ExecStart=/usr/sbin/nginx (code=exited, status=0/SUCCESS)
   Main PID: 25403 (nginx)
      Tasks: 3 (limit: 1058)
     Memory: 3.2M
        CPU: 58ms
     CGroup: /system.slice/nginx.service
             ├─25403 "nginx: master process /usr/sbin/nginx"
             ├─25404 "nginx: worker process"
             └─25405 "nginx: worker process"

Jan 05 18:52:51 ip-10-0-0-233.ec2.internal systemd[1]: Starting nginx.service - The nginx HTTP and reverse proxy server...
Jan 05 18:52:51 ip-10-0-0-233.ec2.internal nginx[25401]: nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
Jan 05 18:52:51 ip-10-0-0-233.ec2.internal nginx[25401]: nginx: configuration file /etc/nginx/nginx.conf test is successful
Jan 05 18:52:51 ip-10-0-0-233.ec2.internal systemd[1]: Started nginx.service - The nginx HTTP and reverse proxy server.
[ec2-user@ip-10-0-0-233 var]$ cd /usr/share/nginx/html/
[ec2-user@ip-10-0-0-233 html]$ ls
404.html  50x.html  icons  index.html  nginx-logo.png  poweredby.png
[ec2-user@ip-10-0-0-233 html]$ vi index.html
[ec2-user@ip-10-0-0-233 html]$ vi index.html
[ec2-user@ip-10-0-0-233 html]$ sudo vi index.html
```

4. Navigate to path ***cd /usr/share/nginx/html/index.html*** and edit as per requirement.

```
<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
<style>
html { color-scheme: light dark; }
body { width: 35em; margin: 0 auto;
font-family: Tahoma, Verdana, Arial, sans-serif; }
</style>
</head>
<body>
<h1>Hello, Docker!</h1>
<p>If you see this page, the nginx web server is successfully installed and
working. Further configuration is required.</p>

<p>This page is writteen by Akshay, for assignemtn over Dockerizing a Plain HTML Page With Nginx </p>

<p>For online documentation and support please refer to
<a href="http://nginx.org/">nginx.org</a>.<br/>
Commercial support is available at
<a href="http://nginx.com/">nginx.com</a>.</p>

<p><em>Thank you for using nginx.</em></p>
</body>
</html>
~
~
~
~
~
```

5. Made changes in nginx.conf file at path **/etc/nginx/nginx.conf**

Save and **Restart Nginx** to apply the changes:



```
[ec2-user@ip-10-0-0-233 nginx]$
[[ec2-user@ip-10-0-0-233 nginx]$ cat nginx.conf
# For more information on configuration, see:
#   * Official English Documentation: http://nginx.org/en/docs/
#   * Official Russian Documentation: http://nginx.org/ru/docs/

user nginx;
worker_processes auto;
error_log /var/log/nginx/error.log notice;
pid /run/nginx.pid;

# Load dynamic modules. See /usr/share/doc/nginx/README.dynamic.
include /usr/share/nginx/modules/*.conf;

events {
    worker_connections 1024;
}

http {
    log_format  main  '$remote_addr - $remote_user [$time_local] "$request" '
                      '$status $body_bytes_sent "$http_referer" '
                      '"$http_user_agent" "$http_x_forwarded_for"';

    access_log  /var/log/nginx/access.log  main;

    sendfile            on;
    tcp_nopush          on;
    keepalive_timeout   65;
    types_hash_max_size 4096;

    include             /etc/nginx/mime.types;
    default_type        application/octet-stream;

    # Load modular configuration files from the /etc/nginx/conf.d directory.
    # See http://nginx.org/en/docs/ngx_core_module.html#include
    # for more information.
    include /etc/nginx/conf.d/*.conf;

    server {
        listen       80;
        listen       [::]:80;
        server_name  localhost;
        root         /usr/share/nginx/html;
        index index.html;

        # Load configuration files for the default server block.
        include /etc/nginx/default.d/*.conf;

        error_page 404 /404.html;
        location = /404.html {
        }

        error_page 500 502 503 504 /50x.html;
        location = /50x.html {
        }
    }

}
```

Install Docker

sudo yum install docker -y
sudo systemctl start docker
sudo systemctl enable docker

```
[ec2-user@ip-10-0-0-233 ~]$
[ec2-user@ip-10-0-0-233 ~]$ sudo systemctl start docker
sudo systemctl enable docker
Created symlink /etc/systemd/system/multi-user.target.wants/docker.service → /usr/lib/systemd/system/docker.service.
[ec2-user@ip-10-0-0-233 ~]$
[ec2-user@ip-10-0-0-233 ~]$
[ec2-user@ip-10-0-0-233 ~]$ sudo systemctl status docker
● docker.service - Docker Application Container Engine
     Loaded: loaded (/usr/lib/systemd/system/docker.service; enabled; preset: disabled)
     Active: active (running) since Sun 2025-01-05 19:15:21 UTC; 10s ago
TriggeredBy: ● docker.socket
       Docs: https://docs.docker.com
   Main PID: 27640 (dockerd)
      Tasks: 10
     Memory: 39.9M
        CPU: 341ms
     CGroup: /system.slice/docker.service
             └─27640 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock --default-ulimit nofile=32768:65536

Jan 05 19:15:21 ip-10-0-0-233.ec2.internal systemd[1]: Starting docker.service - Docker Application Container Engine...
Jan 05 19:15:21 ip-10-0-0-233.ec2.internal dockerd[27640]: time="2025-01-05T19:15:21.122023759Z" level=info msg="Starting up"
Jan 05 19:15:21 ip-10-0-0-233.ec2.internal dockerd[27640]: time="2025-01-05T19:15:21.243803803Z" level=info msg="Loading containers: start."
Jan 05 19:15:21 ip-10-0-0-233.ec2.internal dockerd[27640]: time="2025-01-05T19:15:21.765834089Z" level=info msg="Loading containers: done."
Jan 05 19:15:21 ip-10-0-0-233.ec2.internal dockerd[27640]: time="2025-01-05T19:15:21.789770006Z" level=info msg="Docker daemon" commit=b08a51f containerd-
Jan 05 19:15:21 ip-10-0-0-233.ec2.internal dockerd[27640]: time="2025-01-05T19:15:21.789939224Z" level=info msg="Daemon has completed initialization"
Jan 05 19:15:21 ip-10-0-0-233.ec2.internal dockerd[27640]: time="2025-01-05T19:15:21.836001862Z" level=info msg="API listen on /run/docker.sock"
Jan 05 19:15:21 ip-10-0-0-233.ec2.internal systemd[1]: Started docker.service - Docker Application Container Engine.
[ec2-user@ip-10-0-0-233 ~]$
```

Add your current user to the Docker group:

>> sudo usermod -aG docker $USER

Check if your user is in the Docker group:

>> groups

```
[ec2-user@ip-10-0-0-233 ~]$
[ec2-user@ip-10-0-0-233 ~]$ sudo usermod -aG docker $USER
[ec2-user@ip-10-0-0-233 ~]$
[ec2-user@ip-10-0-0-233 ~]$ newgrp docker
[ec2-user@ip-10-0-0-233 ~]$ groups
docker adm wheel systemd-journal ec2-user
[ec2-user@ip-10-0-0-233 ~]$
```

Create Dockerfile.

```
FROM nginx:latest
RUN mkdir -p /app
WORKDIR /app
COPY nginx.conf /etc/nginx/nginx.conf
COPY index.html /usr/share/nginx/html/index.html
EXPOSE 80
CMD ["nginx", "-g", "daemon off;"]
```

Build it, using command:

docker build -t my-nginx-docker .

```
[ec2-user@ip-10-0-0-233 nginx-docker]$
[ec2-user@ip-10-0-0-233 nginx-docker]$ docker build -t my-nginx-docker .
[+] Building 0.3s (10/10) FINISHED
 => [internal] load build definition from Dockerfile
 => => transferring dockerfile: 280B
 => [internal] load metadata for docker.io/library/nginx:latest
 => [internal] load .dockerignore
 => => transferring context: 2B
 => [1/5] FROM docker.io/library/nginx:latest
 => [internal] load build context
 => => transferring context: 180B
 => CACHED [2/5] RUN mkdir -p /app
 => CACHED [3/5] WORKDIR /app
 => [4/5] COPY nginx.conf /etc/nginx/nginx.conf
 => [5/5] COPY index.html /usr/share/nginx/html/index.html
 => exporting to image
 => => exporting layers
 => => writing image sha256:b1dc1f189080ec77d4de567b8bd49c6c08f4b2ec0033b10d6c5120503d825bb3
 => => naming to docker.io/library/my-nginx-docker
[ec2-user@ip-10-0-0-233 nginx-docker]$
[ec2-user@ip-10-0-0-233 nginx-docker]$
[ec2-user@ip-10-0-0-233 nginx-docker]$
[ec2-user@ip-10-0-0-233 nginx-docker]$ ls
Dockerfile  index.html  nginx.conf
[ec2-user@ip-10-0-0-233 nginx-docker]$
```

Run and validate it.

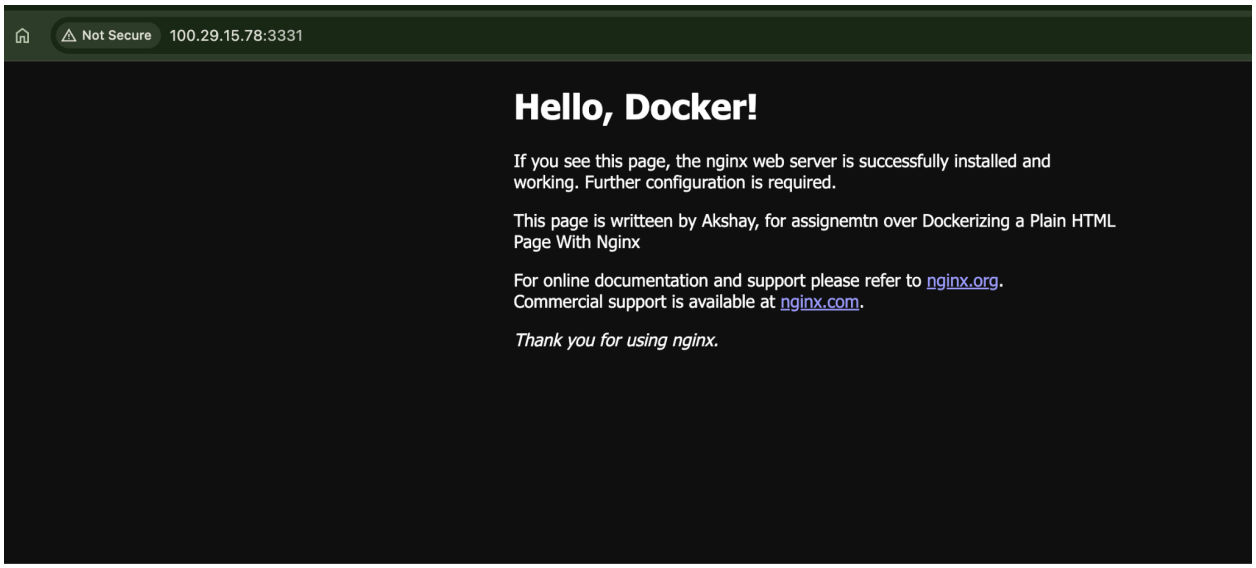
docker run -d --name nginx-2 -p 3331:80 my-nginx-docker

```
[ec2-user@ip-10-0-0-233 nginx-docker]$ docker run -d --name nginx-2 -p 3331:80 my-nginx-docker
356f8cfe5d054be414f0f65b35d66876c063b08d10a15e4ef74e67376f38e3e1
[ec2-user@ip-10-0-0-233 nginx-docker]$
[ec2-user@ip-10-0-0-233 nginx-docker]$
[ec2-user@ip-10-0-0-233 nginx-docker]$
[ec2-user@ip-10-0-0-233 nginx-docker]$ docker ps
CONTAINER ID   IMAGE             COMMAND                  CREATED          STATUS          PORTS                                       NAMES
356f8cfe5d05   my-nginx-docker   "/docker-entrypoint.…"   26 seconds ago   Up 25 seconds   0.0.0.0:3331->80/tcp, :::3331->80/tcp   nginx-2
[ec2-user@ip-10-0-0-233 nginx-docker]$
```

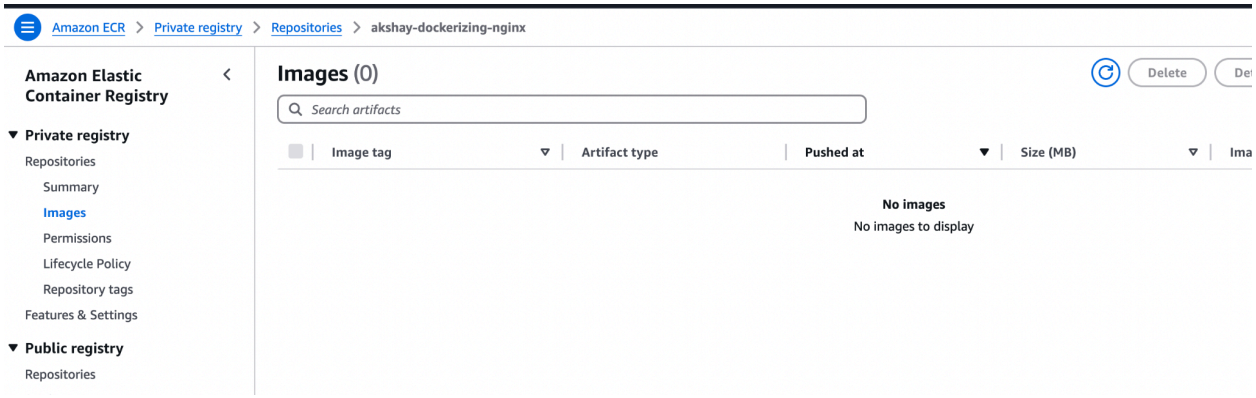
Validate the webpage using public IP address of the EC2 instance:

http://<your-ec2-public-ip>:3331

http://100.29.15.78:3331

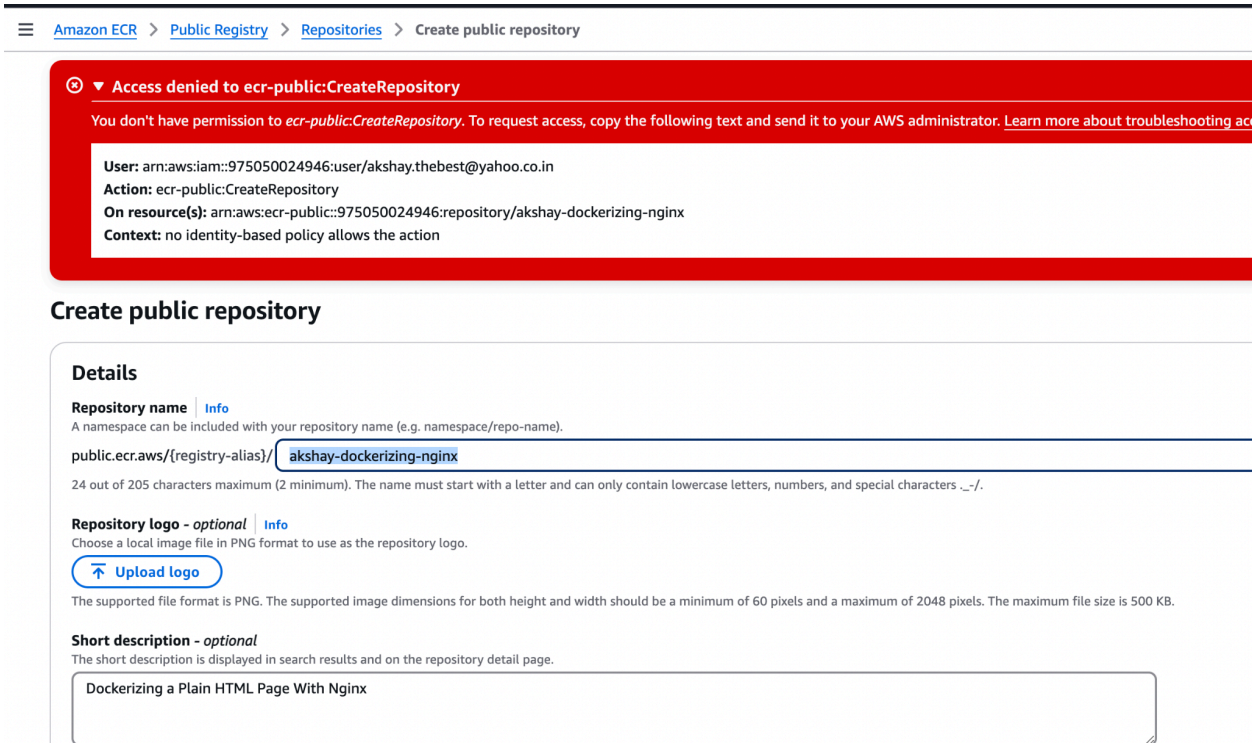


Create your ECR repository



Couldn’t create a public repo due to this error so created private

.



Push commands for akshay-dockerizing-nginx image to ECR:

1. 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 975050024946.dkr.ecr.us-east-1.amazonaws.com
```

2. Build your Docker image using the following command.

```
docker build -t akshay-dockerizing-nginx .
```

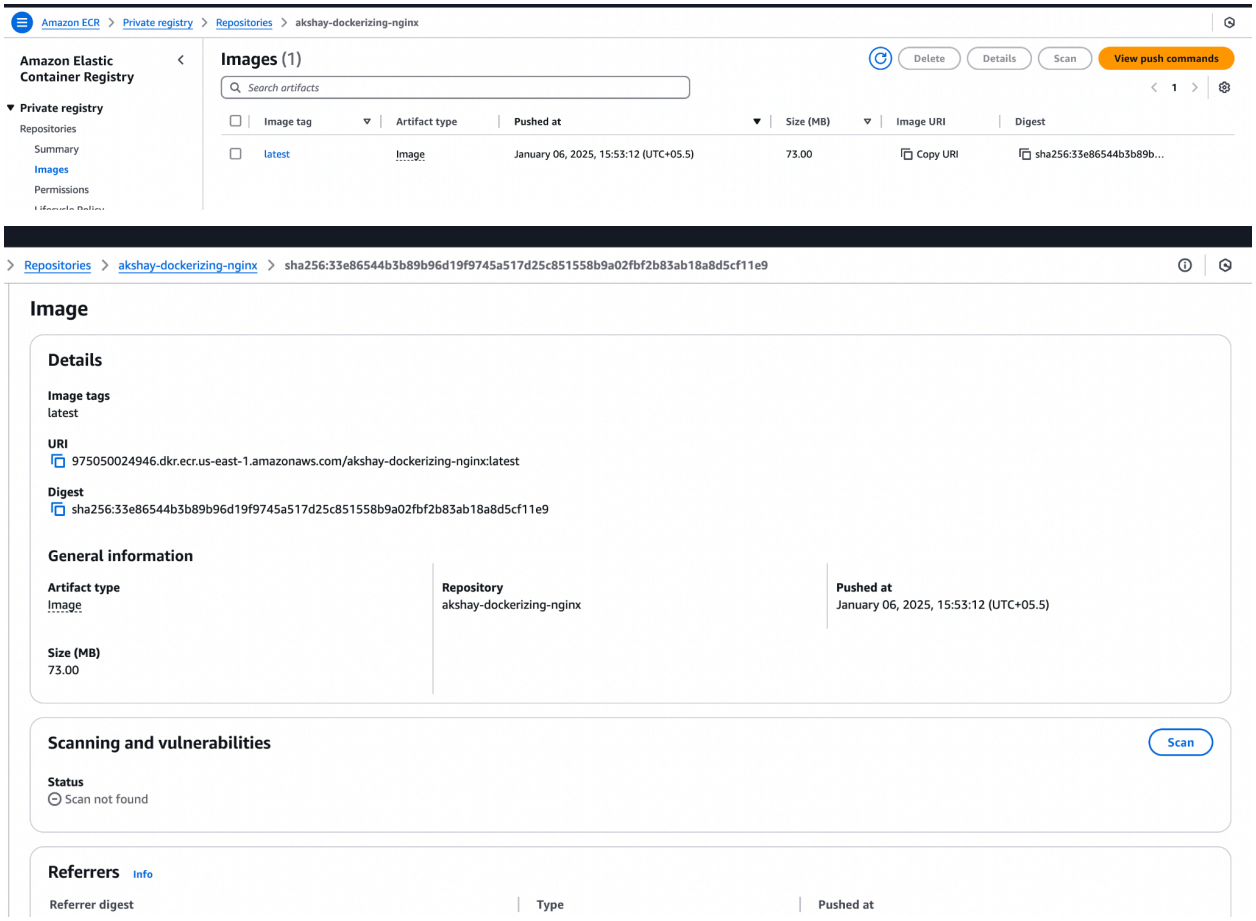
3. After the build completes, tag your image so you can push the image to this repository:

```
docker tag akshay-dockerizing-nginx:latest 975050024946.dkr.ecr.us-east-1.amazonaws.com/akshay-dockerizing-nginx:latest
```

4. Run the following command to push this image to your newly created AWS repository:

```
docker push 975050024946.dkr.ecr.us-east-1.amazonaws.com/akshay-dockerizing-nginx:latest
```

```
[ec2-user@ip-10-0-0-233 nginx-docker]$ docker build -t akshay-dockerizing-nginx .
[+] Building 0.1s (10/10) FINISHED
 => [internal] load build definition from Dockerfile
 => => transferring dockerfile: 280B
 => [internal] load metadata for docker.io/library/nginx:latest
 => [internal] load .dockerignore
 => => transferring context: 2B
 => [1/5] FROM docker.io/library/nginx:latest
 => [internal] load build context
 => => transferring context: 180B
 => CACHED [2/5] RUN mkdir -p /app
 => CACHED [3/5] WORKDIR /app
 => CACHED [4/5] COPY nginx.conf /etc/nginx/nginx.conf
 => CACHED [5/5] COPY index.html /usr/share/nginx/html/index.html
 => exporting to image
 => => exporting layers
 => => writing image sha256:b1dc1f189080ec77d4de567b8bd49c6c08f4b2ec0033b10d6c5120503d825bb3
 => => naming to docker.io/library/akshay-dockerizing-nginx
[ec2-user@ip-10-0-0-233 nginx-docker]$
[ec2-user@ip-10-0-0-233 nginx-docker]$ docker images
REPOSITORY                TAG       IMAGE ID       CREATED        SIZE
akshay-dockerizing-nginx  latest    b1dc1f189080   15 hours ago   192MB
my-nginx-docker           latest    b1dc1f189080   15 hours ago   192MB
nginx                     latest    f876bfc1cc63   5 weeks ago    192MB
[ec2-user@ip-10-0-0-233 nginx-docker]$
[ec2-user@ip-10-0-0-233 nginx-docker]$ docker tag akshay-dockerizing-nginx:latest 975050024946.dkr.ecr.us-east-1.amazonaws.com/akshay-dockerizin
[ec2-user@ip-10-0-0-233 nginx-docker]$ docker push 975050024946.dkr.ecr.us-east-1.amazonaws.com/akshay-dockerizing-nginx:latest
The push refers to repository [975050024946.dkr.ecr.us-east-1.amazonaws.com/akshay-dockerizing-nginx]
3a13e1522567: Pushed
7ea8d16e80f2: Pushed
5f70bf18a086: Pushed
74cba256c673: Pushed
af90855d8344: Pushed
ad206e285c61: Pushed
24aeff94f79e: Pushed
d567f5b4517e: Pushed
14a96b2ac595: Pushed
c4c8312766f1: Pushed
8b296f486960: Pushed
latest: digest: sha256:33e86544b3b89b96d19f9745a517d25c851558b9a02fbf2b83ab18a8d5cf11e9 size: 2605
[ec2-user@ip-10-0-0-233 nginx-docker]$
```


Amazon ECR > Private registry > Repositories > akshay-dockerizing-nginx
Amazon Elastic Container Registry
Private registry
Repositories
Summary
Images
Permissions
Images (1)
Search artifacts
Delete
Details
Scan
View push commands
Image tag
Artifact type
Pushed at
Size (MB)
Image URI
Digest
latest
Image
January 06, 2025, 15:53:12 (UTC+05.5)
73.00
Copy URI
sha256:33e86544b3b89b...
Repositories > akshay-dockerizing-nginx > sha256:33e86544b3b89b96d19f9745a517d25c851558b9a02fbf2b83ab18a8d5cf11e9
Image
Details
Image tags
latest
URI
975050024946.dkr.ecr.us-east-1.amazonaws.com/akshay-dockerizing-nginx:latest
Digest
sha256:33e86544b3b89b96d19f9745a517d25c851558b9a02fbf2b83ab18a8d5cf11e9
General information
Artifact type
Image
Repository
akshay-dockerizing-nginx
Pushed at
January 06, 2025, 15:53:12 (UTC+05.5)
Size (MB)
73.00
Scanning and vulnerabilities
Scan
Status
Scan not found
Referrers
Info
Referrer digest
Type
Pushed at