

## Docker Task -3

### Launching an EC2 Instance

The screenshot shows the AWS Cloud Console interface for launching an EC2 instance. The top navigation bar includes tabs for Docker Task -3 - Google Docs, Untitled document - Google Doc, Launch an instance | EC2 | us-east-1, and Docker Home. The main content area is titled "Launch an instance" and displays the following steps:

- Name and tags**: A text input field containing "Docker-vm" with a "Add additional tags" link.
- Application and OS Images (Amazon Machine Image)**: A search bar and a grid of recent AMI icons (Amazon Linux, macOS, Ubuntu, Windows, Red Hat, SUSE Linux, Debian). A "Browse more AMIs" button is available.
- Amazon Machine Image (AMI)**: Details for the selected "Amazon Linux 2023 AMI": ami-08b5b3a93ed654d19 (64-bit (x86), uefi-preferred) / ami-0eae2a0fc13b15fce (64-bit (Arm), uefi). It notes "Free tier eligible".
- Description**: A detailed description of Amazon Linux 2023, stating it is a modern, general purpose Linux-based OS optimized for AWS.
- Architecture**: Set to "64-bit (x86)".
- Boot mode**: Set to "uefi-preferred".
- AMI ID**: ami-08b5b3a93ed654d19
- Publish Date**: 2025-03-04
- Username**: ec2-user
- Verified provider**: A green button.
- Instance type**: Set to "t2.micro". A table provides details: Family: t2, 1 vCPU, 1 GiB Memory, Current generation: true. Pricing: On-Demand Windows base pricing: 0.0162 USD per Hour, On-Demand Ubuntu Pro base pricing: 0.0134 USD per Hour, On-Demand SUSE base pricing: 0.0116 USD per Hour, On-Demand RHEL base pricing: 0.026 USD per Hour, On-Demand Linux base pricing: 0.0116 USD per Hour. Options include "Free tier eligible", "All generations", and "Compare instance types".
- Summary**: Shows 1 instance selected. Configuration details: Software Image (AMI) - Amazon Linux 2023 AMI, Virtual server type (instance type) - t2.micro, Firewall (security group) - New security group, Storage (volumes) - 1 volume(s) - 8 GiB. Buttons include "Cancel", "Launch instance" (highlighted in orange), and "Preview code".

The bottom of the screen shows the Windows taskbar with various pinned icons and the date/time as 09-03-2025.

The screenshot shows the AWS CloudShell interface with a browser window open to the EC2 Instances Launch wizard. The wizard is configured to launch one instance of the t2.micro type with an Amazon Linux 2023.6.2 AMI. The instance will be named 'LINUX-KEY' and will be associated with a new security group called 'launch-wizard-1'. The security group allows SSH traffic from anywhere. The summary section shows the instance details and provides 'Launch Instance' and 'Preview code' buttons.

**Key pair (login)** [Info](#)  
You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.  
**Key pair name - required**  
LINUX-KEY [Create new key pair](#)

**Network settings** [Info](#)  
**Network** [Info](#)  
vpc-07f2dc389716a7f01  
**Subnet** [Info](#)  
No preference (Default subnet in any availability zone)  
**Auto-assign public IP** [Info](#)  
Enable  
Additional charges apply when outside of [free tier](#) allowance  
**Firewall (security groups)** [Info](#)  
A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.  
 Create security group  Select existing security group

We'll create a new security group called 'launch-wizard-1' with the following rules:  
 Allow SSH traffic from Anywhere  
Helps you connect to your instance

**Summary**  
Number of instances [Info](#)  
1  
**Software Image (AMI)**  
Amazon Linux 2023 AMI 2023.6.2...[read more](#)  
ami-08b5b3a93ed654d19  
**Virtual server type (instance type)**  
t2.micro  
**Firewall (security group)**  
New security group  
**Storage (volumes)**  
1 volume(s) - 8 GiB

[Cancel](#) [Launch Instance](#) [Preview code](#)

The screenshot shows the AWS CloudShell interface with a browser window open to the EC2 Instances Launch wizard. The configuration is identical to the first screenshot, except for the security group rules. In addition to allowing SSH traffic from anywhere, it also allows HTTPS and HTTP traffic from the internet. A warning message at the bottom of the wizard advises against allowing all IP addresses and recommends setting security group rules to allow access from known IP addresses only.

**Key pair (login)** [Info](#)  
You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.  
**Key pair name - required**  
LINUX-KEY [Create new key pair](#)

**Network settings** [Info](#)  
**Network** [Info](#)  
vpc-07f2dc389716a7f01  
**Subnet** [Info](#)  
No preference (Default subnet in any availability zone)

We'll create a new security group called 'launch-wizard-1' with the following rules:  
 Allow SSH traffic from Anywhere  
Helps you connect to your instance  
 Allow HTTPS traffic from the internet  
To set up an endpoint, for example when creating a web server  
 Allow HTTP traffic from the internet  
To set up an endpoint, for example when creating a web server

**Summary**  
Number of instances [Info](#)  
1  
**Software Image (AMI)**  
Amazon Linux 2023 AMI 2023.6.2...[read more](#)  
ami-08b5b3a93ed654d19  
**Virtual server type (instance type)**  
t2.micro  
**Firewall (security group)**  
New security group  
**Storage (volumes)**  
1 volume(s) - 8 GiB

[Cancel](#) [Launch Instance](#) [Preview code](#)

⚠️ Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

The screenshot shows the AWS EC2 Instances Launch an instance page. A green success message at the top states: "Successfully initiated launch of instance (i-0b94b615672fa3999)". Below this, a "Launch log" section is visible. Under "Next Steps", there are four cards: "Create billing and free tier usage alerts", "Connect to your instance", "Connect an RDS database", and "Create EBS snapshot policy". At the bottom, there are links for "Manage detailed monitoring", "Create Load Balancer", "Create AWS budget", and "Manage CloudWatch alarms". The browser status bar indicates the URL is us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#LaunchInstances.

The screenshot shows the AWS EC2 Instances page. On the left, a navigation menu includes "EC2", "Instances", "Images", and "Elastic Block Store". The main area displays a table titled "Instances (1) Info" with one row: "Docker-vm" (Instance ID: i-0b94b615672fa3999, Instance state: Running, Instance type: t2.micro, Status check: Initializing). Below the table, a "Select an instance" dropdown is open. The browser status bar indicates the URL is us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#Instances:instancetype=i-0b94b615672fa3999.

**Instance summary for i-0b94b615672fa3999 (Docker-vm)**

**Public IPv4 address:** 174.129.75.51 | [open address](#)

**Private IP4 addresses:** 172.31.89.37

**Public IPv4 DNS:** ec2-174-129-75-51.compute-1.amazonaws.com | [open address](#)

**Private IP DNS name (IPv4 only):** ip-172-31-89-37.ec2.internal

**Instance type:** t2.micro

**VPC ID:** vpc-07f2dc389716a7f01

**Subnet ID:** subnet-0e07ae45177b4d083

**Instance ARN:** arn:aws:ec2:us-east-1:905418201986:instance/i-0b94b615672fa3999

**Elastic IP addresses:** -

**AWS Compute Optimizer finding:** Opt-in to AWS Compute Optimizer for recommendation

**Auto Scaling Group name:** -

**Managed:** false

## Connect to the EC2 instance

**Connect to instance**

Connect to your instance i-0b94b615672fa3999 (Docker-vm) using any of these options

**EC2 Instance Connect**   **Session Manager**   **SSH client** (selected)   **EC2 serial console**

**Instance ID:** i-0b94b615672fa3999 (Docker-vm)

- Open an SSH client.
- Locate your private key file. The key used to launch this instance is **LINUX-KEY.pem**.
- Run this command, if necessary, to ensure your key is not publicly viewable.  
chmod 400 "LINUX-KEY.pem"
- Connect to your instance using its Public DNS:  
ec2-174-129-75-51.compute-1.amazonaws.com

**Command copied**

```
ssh -i "LINUX-KEY.pem" ec2-user@ec2-174-129-75-51.compute-1.amazonaws.com
```

**Note:** In most cases, the guessed username is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI username.

Cancel

```
ssh -i "LINUX-KEY.pem" ec2-user@ec2-174-129-75-51.compute-1.amazonaws.com
```

```
ec2-user@ip-172-31-89-37:~  
Microsoft Windows [Version 10.0_26100.3323]  
(c) Microsoft Corporation. All rights reserved.  
C:\Users\mouni\Downloads>ssh -i "LINUX-KEY.pem" ec2-user@ec2-174-129-75-51.compute-1.amazonaws.com  
The authenticity of host 'ec2-174-129-75-51.compute-1.amazonaws.com (174.129.75.51)' can't be established.  
ED25519 key fingerprint is SHA256:mP6fhCpxvxv0UgGpp2RloqBpJmlW7v/SillysH6bQ.  
This key is not known by any other names.  
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes  
Warning: Permanently added 'ec2-174-129-75-51.compute-1.amazonaws.com' (ED25519) to the list of known hosts.  
[ec2-user@ip-172-31-89-37 ~]$
```



## Update package manager: \$ sudo yum update -y

```
ec2-user@ip-172-31-89-37:~  
Microsoft Windows [Version 10.0_26100.3323]  
(c) Microsoft Corporation. All rights reserved.  
C:\Users\mouni\Downloads>ssh -i "LINUX-KEY.pem" ec2-user@ec2-174-129-75-51.compute-1.amazonaws.com  
The authenticity of host 'ec2-174-129-75-51.compute-1.amazonaws.com (174.129.75.51)' can't be established.  
ED25519 key fingerprint is SHA256:mP6fhCpxvxv0UgGpp2RloqBpJmlW7v/SillysH6bQ.  
This key is not known by any other names.  
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes  
Warning: Permanently added 'ec2-174-129-75-51.compute-1.amazonaws.com' (ED25519) to the list of known hosts.  
[ec2-user@ip-172-31-89-37 ~]$ sudo yum update -y  
Amazon Linux 2023 Kernel Livepatch repository  
Dependencies resolved.  
Nothing to do.  
Complete!  
[ec2-user@ip-172-31-89-37 ~]$
```



## Install Docker

```
$ sudo yum install docker
```

```
[ec2-user@ip-172-31-89-37 ~]$ Select ec2-user@ip-172-31-89-37:~  
Amazon Linux 2023 Kernel Livepatch repository  
Dependencies resolved.  
Nothing to do.  
Complete! [ec2-user@ip-172-31-89-37 ~]$ sudo yum install docker  
Last metadata expiration check: 0:02:16 ago on Sun Mar 9 17:24:46 2025.  
Dependencies resolved.  
=====  


| Package                  | Architecture | Version               | Repository  | Size  |
|--------------------------|--------------|-----------------------|-------------|-------|
| Installing:              |              |                       |             |       |
| docker                   | x86_64       | 25.0.8-1.amzn2023.0.1 | amazonlinux | 44 M  |
| Installing dependencies: |              |                       |             |       |
| containedr               | x86_64       | 1.7.25-1.amzn2023.0.1 | amazonlinux | 36 M  |
| iptables-libs            | x86_64       | 1.8.8-3.amzn2023.0.2  | amazonlinux | 401 k |
| iptables-nft             | x86_64       | 1.8.8-3.amzn2023.0.2  | amazonlinux | 183 k |
| libcgroup                | x86_64       | 3.0-1.amzn2023.0      | amazonlinux | 75 k  |
| libnetfilter_conntrack   | x86_64       | 1.0.8-2.amzn2023.0.2  | amazonlinux | 58 k  |
| libnetfilter_tlink       | x86_64       | 1.0.1-19.amzn2023.0.2 | amazonlinux | 30 k  |
| libnftnl                 | x86_64       | 1.2.2-2.amzn2023.0.2  | amazonlinux | 84 k  |
| pigz                     | x86_64       | 2.5-1.amzn2023.0.3    | amazonlinux | 83 k  |
| runc                     | x86_64       | 1.2.4-1.amzn2023.0.1  | amazonlinux | 3.4 M |


| Transaction Summary                                            |                         |  |
|----------------------------------------------------------------|-------------------------|--|
| Install                                                        | 10 Packages             |  |
| Total download size: 84 M                                      |                         |  |
| Installed size: 319 M                                          |                         |  |
| Is this ok [y/N]: y                                            |                         |  |
| Downloading Packages:                                          |                         |  |
| (1/10): iptables-libs-1.8.8-3.amzn2023.0.2.x86_64.rpm          | 7.6 MB/s   401 kB 00:00 |  |
| (2/10): iptables-nft-1.8.8-3.amzn2023.0.2.x86_64.rpm           | 3.5 MB/s   183 kB 00:00 |  |
| (3/10): libcgroup-3.0-1.amzn2023.0.1.x86_64.rpm                | 2.5 MB/s   75 kB 00:00  |  |
| (4/10): libnetfilter_conntrack-1.0.8-2.amzn2023.0.2.x86_64.rpm | 2.5 MB/s   58 kB 00:00  |  |
| (5/10): libnetfilter_tlink-1.0.1-19.amzn2023.0.2.x86_64.rpm    | 823 kB/s   30 kB 00:00  |  |
| (6/10): libnftnl-1.2.2-2.amzn2023.0.2.x86_64.rpm               | 2.3 MB/s   84 kB 00:00  |  |
| (7/10): pigz-2.5-1.amzn2023.0.3.x86_64.rpm                     | 2.6 MB/s   83 kB 00:00  |  |
| (8/10): runc-1.2.4-1.amzn2023.0.1.x86_64.rpm                   | 15 MB/s   3.4 MB 00:00  |  |
| (9/10): docker-25.0.8-1.amzn2023.0.1.x86_64.rpm                | 32 MB/s   44 MB 00:01   |  |
| (10/10): containedr-1.7.25-1.amzn2023.0.1.x86_64.rpm           | 22 MB/s   36 MB 00:01   |  |
| Total:                                                         | 51 MB/s   84 MB 00:01   |  |
| Running transaction check                                      |                         |  |
| Transaction check succeeded.                                   |                         |  |
| Running transaction test                                       |                         |  |
| Transaction test succeeded.                                    |                         |  |
| Running transaction                                            |                         |  |

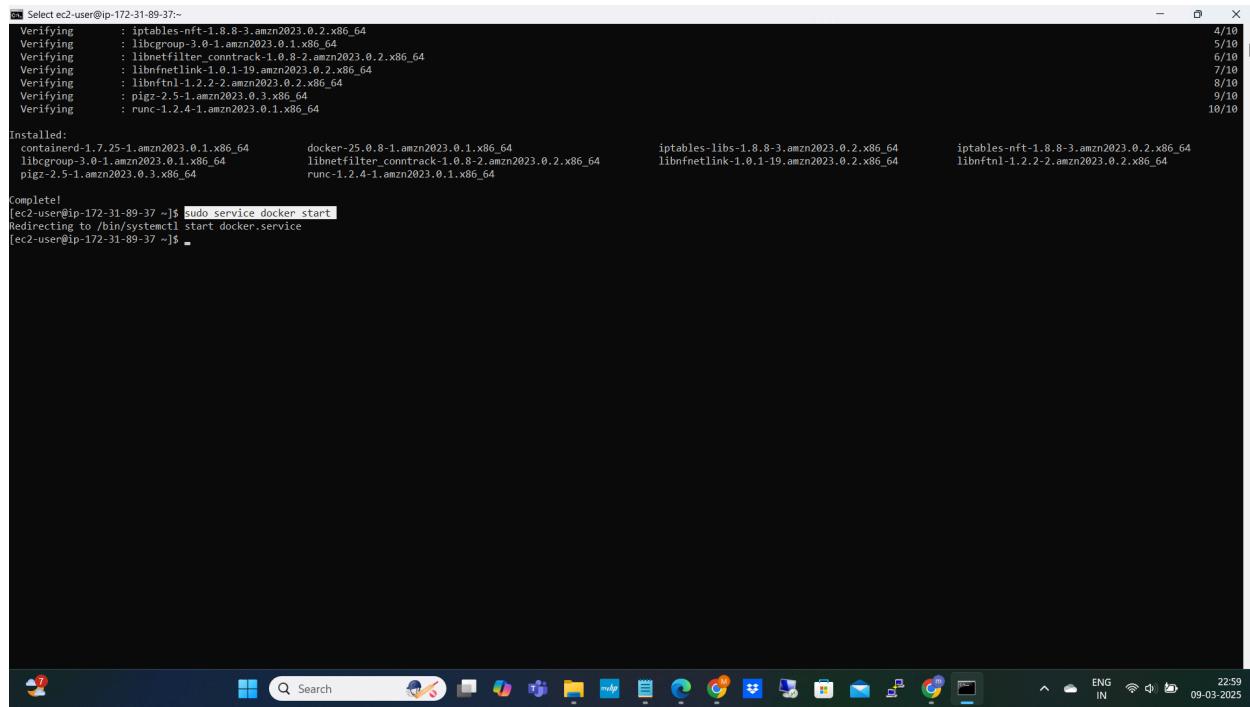

```

```
[ec2-user@ip-172-31-89-37 ~]$ Select ec2-user@ip-172-31-89-37:~  
(5/10): libnftnl-1.2.2-2.amzn2023.0.2.x86_64.rpm  
(6/10): libnftnl-1.2.2-2.amzn2023.0.2.x86_64.rpm  
(7/10): pigz-2.5-1.amzn2023.0.3.x86_64.rpm  
(8/10): runc-1.2.4-1.amzn2023.0.1.x86_64.rpm  
(9/10): docker-25.0.8-1.amzn2023.0.1.x86_64.rpm  
(10/10): containedr-1.7.25-1.amzn2023.0.1.x86_64.rpm  
  
Total:  
Running transaction check  
Transaction check succeeded.  
Running transaction test  
Transaction test succeeded.  
Running transaction  
  Preparing :  
    Installing : runc-1.2.4-1.amzn2023.0.1.x86_64  
    Installing : containedr-1.7.25-1.amzn2023.0.1.x86_64  
    Running scriptlet: containedr-1.7.25-1.amzn2023.0.1.x86_64  
    Installing : pigz-2.5-1.amzn2023.0.3.x86_64  
    Installing : libnftnl-1.2.2-2.amzn2023.0.2.x86_64  
    Installing : libnftnl-1.0.1-19.amzn2023.0.2.x86_64  
    Installing : libnetfilter_conntrack-1.0.8-2.amzn2023.0.2.x86_64  
    Installing : iptables-libs-1.8.8-3.amzn2023.0.2.x86_64  
    Installing : iptables-nft-1.8.8-3.amzn2023.0.2.x86_64  
    Running scriptlet: runc-1.2.4-1.amzn2023.0.1.x86_64  
    Installing : libcgroup-3.0-1.amzn2023.0.1.x86_64  
    Running scriptlet: docker-25.0.8-1.amzn2023.0.1.x86_64  
    Installing : docker-25.0.8-1.amzn2023.0.1.x86_64  
    Running scriptlet: docker-25.0.8-1.amzn2023.0.1.x86_64  
    Created symlink /etc/systemd/system/sockets.target.wants/docker.socket → /usr/lib/systemd/system/docker.socket.  
  
Verifying : containedr-1.7.25-1.amzn2023.0.1.x86_64  
Verifying : docker-25.0.8-1.amzn2023.0.1.x86_64  
Verifying : iptables-libs-1.8.8-3.amzn2023.0.2.x86_64  
Verifying : iptables-nft-1.8.8-3.amzn2023.0.2.x86_64  
Verifying : libcgroup-3.0-1.amzn2023.0.1.x86_64  
Verifying : libnftnl-1.0.1-19.amzn2023.0.2.x86_64  
Verifying : libnetfilter_conntrack-1.0.8-2.amzn2023.0.2.x86_64  
Verifying : libnftnl-1.2.2-2.amzn2023.0.2.x86_64  
Verifying : pigz-2.5-1.amzn2023.0.3.x86_64  
Verifying : runc-1.2.4-1.amzn2023.0.1.x86_64  
  
Installed:  
containedr-1.7.25-1.amzn2023.0.1.x86_64  
libcgroup-3.0-1.amzn2023.0.1.x86_64  
pigz-2.5-1.amzn2023.0.3.x86_64  
  
Complete! [ec2-user@ip-172-31-89-37 ~]$  
  
[ec2-user@ip-172-31-89-37 ~]$ Select ec2-user@ip-172-31-89-37:~  
Amazon Linux 2023 Kernel Livepatch repository  
Dependencies resolved.  
Nothing to do.  
Complete! [ec2-user@ip-172-31-89-37 ~]$
```



## To start Docker:

```
$ sudo service docker start
```



```
[ec2-user@ip-172-31-89-37 ~]$ sudo service docker start
Redirecting to /bin/systemctl start docker.service
[ec2-user@ip-172-31-89-37 ~]$
```

## Install Docker Compose:

i.Download latest version of docker compose

```
$ sudo curl -L
```

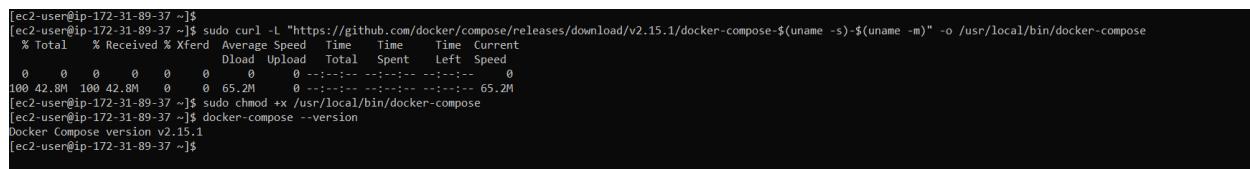
```
"https://github.com/docker/compose/releases/download/v2.15.1/docker-compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose
```

Give the executable permissions to the docker compose file

```
$ sudo chmod +x /usr/local/bin/docker-compose
```

Check the Installation:

```
$ docker-compose --version
```

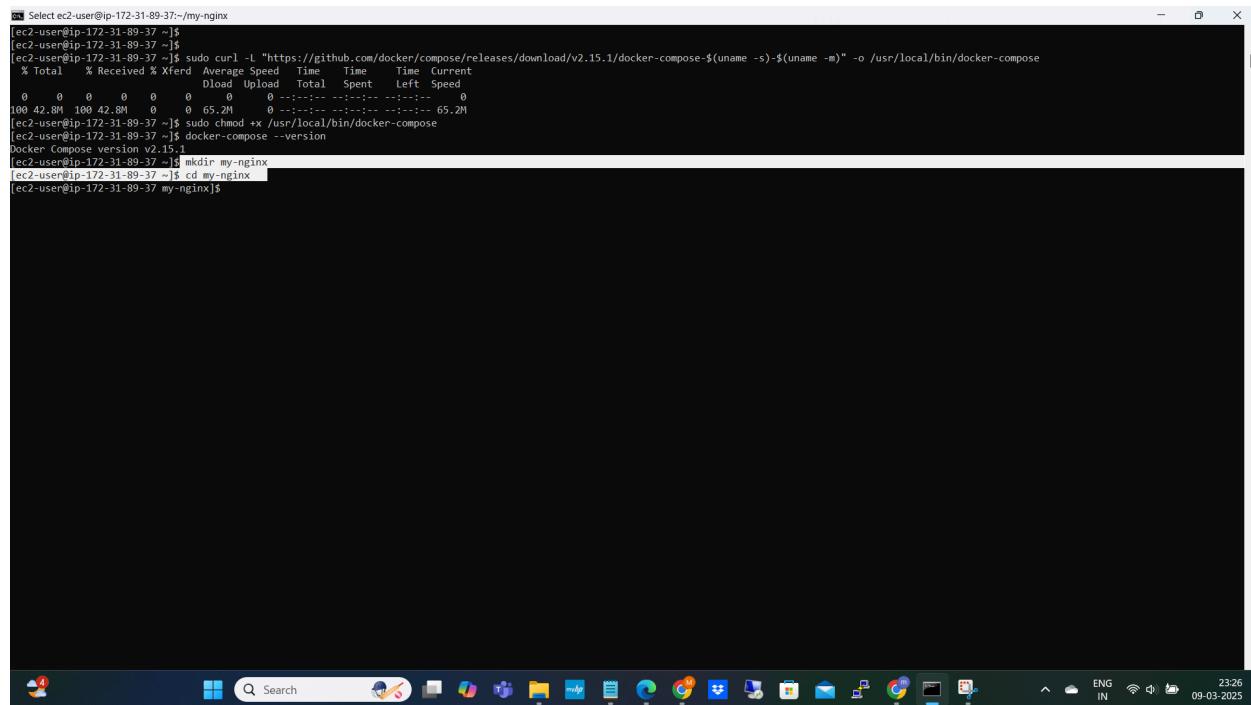


```
[ec2-user@ip-172-31-89-37 ~]$
[ec2-user@ip-172-31-89-37 ~]$ sudo curl -L "https://github.com/docker/compose/releases/download/v2.15.1/docker-compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose
% Total % Received % Xferd Average Speed Time Time Current
          Dload Upload Total Spent Left Speed
0     0    0    0    0     0   0:--:--:--:--:--:--:--:--:--:0
100 42.8M 100 42.8M 0     0 65.2M 0:--:--:--:--:--:--:65.2M
[ec2-user@ip-172-31-89-37 ~]$ sudo chmod +x /usr/local/bin/docker-compose
[ec2-user@ip-172-31-89-37 ~]$ docker-compose --version
Docker Compose version v2.15.1
[ec2-user@ip-172-31-89-37 ~]$
```

## Create a directory for the Nginx:

```
$ mkdir my-nginx
```

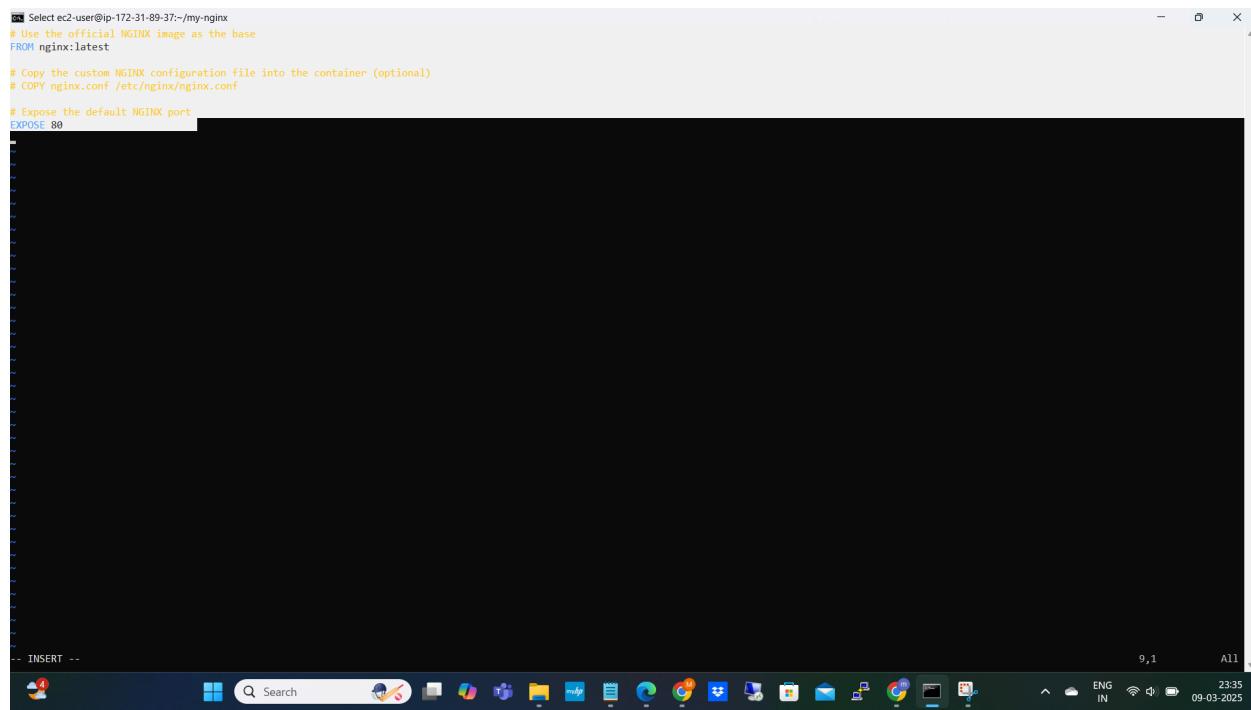
```
$ cd my-nginx
```



```
[ec2-user@ip-172-31-89-37:~/my-nginx]$ Select ec2-user@ip-172-31-89-37:~/my-nginx
[ec2-user@ip-172-31-89-37 ~]$ [ec2-user@ip-172-31-89-37 ~]$ sudo curl -L "https://github.com/docker/compose/releases/download/v2.15.1/docker-compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose
  % Total    % Received   % Xferd  Average Speed   Time   Time  Current
          Dload  Upload   Total Spent  Left  Speed
0     0      0      0      0      0      0      0      0      0      0      0      0
100 42.8M  100 42.8M  0     0      65.2M
[ec2-user@ip-172-31-89-37 ~]$ sudo chmod +x /usr/local/bin/docker-compose
[ec2-user@ip-172-31-89-37 ~]$ docker-compose --version
Docker Compose version v2.15.1
[ec2-user@ip-172-31-89-37 ~]$ mkdir my-nginx
[ec2-user@ip-172-31-89-37 ~]$ cd my-nginx
[ec2-user@ip-172-31-89-37 my-nginx]$
```

## Create a Dockerfile

```
$ Vi Dockerfile
```



```
[ec2-user@ip-172-31-89-37:~/my-nginx]$ Select ec2-user@ip-172-31-89-37:~/my-nginx
# Use the official NGINX image as the base
FROM nginx:latest

# Copy the custom NGINX configuration file into the container (optional)
# COPY nginx.conf /etc/nginx/nginx.conf

# Expose the default NGINX port
EXPOSE 80

```

**Build the custom docker nginx image:**  
**\$ docker build -t my-nginx:latest .**

## Create Docker-Compose.yml file

```
Select ec2-user@ip-172-31-89-37:~/my-nginx
version: '3'

services:
  nginx:
    image: my-nginx:latest # Use the custom NGINX image
    container_name: my_nginx
    ports:
      - "8080:80" # Bind port 80 of the container to port 8080 on your host
    volumes:
      - ./nginx_data:/var/opt/nginx # Bind mount volume
    restart: always

-- INSERT --
```

**Deploy the image using Docker-compose with mount path '/var/opt/nginx location'**  
\$ docker-compose up -d

```
[ec2-user@ip-172-31-89-37:~/my-nginx] $ Select ec2-user@ip-172-31-89-37:~/my-nginx
ERROR: permission denied while trying to connect to the Docker daemon socket at unix:///var/run/docker.sock: Get "http://%2Fvar%2Frun%2Fdocker.sock/_ping": dial unix /var/run/docker.sock: connect: permission denied
[ec2-user@ip-172-31-89-37 my-nginx]$ sudo docker build -t my-nginx:latest .
[*] Building 4.8s (5/5) FINISHED
  => [internal] load build definition from Dockerfile
  => transferring dockerfile: 316B
  => [internal] load metadata for docker.io/library/nginx:latest
  => [internal] load .dockernignore
  =>> transferring context: 2B
[[/1]] FROM docker.io/library/nginx:latest@sha256:9d6b58feebd2bd3c56ab5853333d627c6e281011cf6050fa4bcf2072e9496
  => > resolve docker.io/library/nginx:latest@sha256:9d6b58feebd2bd3c56ab5853333d627c6e281011cf6050fa4bcf2072e9496
  => sha256:052e0094c026c9eddc4eab7a64cc0033c31608087d4ff1132c4e93e8f7b 8.58KB / 8.58KB
  => sha256:b6f9acace214dc236308030999111f07d1ba08a30812a62fd05a6ad108e274 43.99MB / 43.99MB
  => sha256:286d1806e638478a06177a7312045e10d91dse944:8a001240614:839c44 2.459KB / 2.459KB
  => sha256:7c63259811a8c44f6d08e81a9f5363eeef575f362a8995ce6084f8753c 28.29MB / 28.29MB
  => sha256:94e4f92d512d0415c2441f73d2ba0ad2026407408e242700824f3c 6.62MB / 6.62MB
  => sha256:0014f92d512d0415c2441f73d2ba0ad2026407408e242700824f3c 95.7B / 95.7B
  => sha256:0014f92d512d0415c2441f73d2ba0ad2026407408e242700824f3c 95.7B / 95.7B
  => sha256:21ad5a6a8956bd21bb0bb5147c30ad98a0768c365155773535e7649b77d 406B / 406B
  => sha256:94e4f92d512d0415c2441f73d2ba0ad2026407408e242700824f3c 1.21kB / 1.21kB
  => sha256:103f59c13e9f200431b555073c5e88bf7db5dd2c5e4d714a10b994a430e98a3 1.40kB / 1.40kB
  =>> extracting sha256:1f9acae214dc23630803099011f1fd71ba06a3033f0a62f036a60101e274
  =>> extracting sha256:513c3649b0148039e04c7f320b5a99e24eac18a271f056b81241d6730
  =>> extracting sha256:d0149f205d32d416c7b9e4d8244f14f73d3d2eau120264b749e342700824f3c
  =>> extracting sha256:9ud21ad5a6a8956bd21bb0bb5147c30ad98a0768c36515577554d7649bc74d
  =>> extracting sha256:94e4fa0e7c2e422ccacc7ac0579147ea0d62b6b228287a30f1400bb43d37088
  =>> extracting sha256:103f59c13e9f200431b555073c5e88bf7db5dd2c5e4d714a10b994a430e98a3
  =>> exporting to image
  =>> exporting layers
  =>> writing Image sha256:f0465558c2b4d53717e99c98d5a2cace134ceb609ff908329680085d0a4
[ec2-user@ip-172-31-89-37 my-nginx]$ vi docker-compose.yml
[ec2-user@ip-172-31-89-37 my-nginx]$ [ New] 12L, 298B written
[ec2-user@ip-172-31-89-37 my-nginx]$ sudo docker-compose up -d
[*] Running 2/2
# Network my-nginx_default Created
# Container my_nginx Started
[ec2-user@ip-172-31-89-37 my-nginx]$ -
```

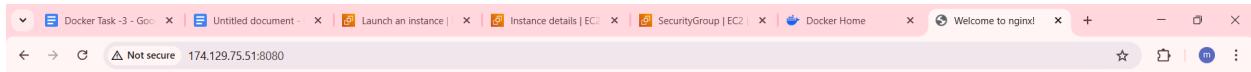
**Check is the nginx container running:**

```
$ sudo docker ps
```

## Verify the deployment done

Access the nginx webpage through the 8080 port

<http://174.129.75.51:8080/>



### 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.

## Verify the mount path

\$ sudo docker inspect my-nginx

```
>Select ec2-user@ip-172-31-89-37:~/my-nginx
{
    "Mounts": [
        {
            "Type": "bind",
            "Source": "/home/ec2-user/my-nginx/nginx_data",
            "Destination": "/var/opt/nginx",
            "Mode": "rw",
            "RW": true,
            "Propagation": "rprivate"
        }
    ],
    "Config": {
        "Hostname": "0b7ec8c49af",
        "Domainname": "",
        "User": "",
        "AttachStdin": false,
        "AttachStdout": true,
        "AttachStderr": true,
        "ExposedPorts": {
            "80/tcp": {}
        },
        "Tty": false,
        "OpenStdin": false,
        "StdinOnce": false,
        "Env": [
            "PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin",
            "NGINX_VERSION=1.27.4",
            "NJS_VERSION=0.8.9",
            "NJS_RELEASE=1-bookworm",
            "PKG_RELEASE=1-bookworm"
        ]
    }
}
```



**Push the custom Docker image to Docker Hub**

**Login to the Docker hub**

**\$ sudo docker login**

```
[ec2-user@ip-172-31-89-37 ~]$ Select ec2-user@ip-172-31-89-37:~/.my-nginx
Error response from daemon: Get "https://registry-1.docker.io/v2/": unauthorized: incorrect username or password
[ec2-user@ip-172-31-89-37 my-nginx]$ sudo docker login
Log in with your Docker ID or email address to push and pull images from Docker Hub. If you don't have a Docker ID, head over to https://hub.docker.com/ to create one.
You can log in with your password or a Personal Access Token (PAT). Using a limited-scope PAT grants better security and is required for organizations using SSO. Learn more at https://docs.docker.com/go/access-tokens/
Username: mounika1112
Password:
WARNING! Your password will be stored unencrypted in /root/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store

Login Succeeded
[ec2-user@ip-172-31-89-37 my-nginx]$
```

**Tag the image for Docker hub**

**\$ docker tag my-nginx:latest mounika1112/my-nginx:latest**

**Push the image to Docker hub**

**\$ docker push mounika1112/my-nginx:latest**

```
[ec2-user@ip-172-31-89-37 ~]$ Select ec2-user@ip-172-31-89-37:~/.my-nginx
Login Succeeded
[ec2-user@ip-172-31-89-37 my-nginx]$ docker tag my-nginx:latest mounika1112/my-nginx:latest
permission denied while trying to connect to the Docker daemon socket at unix:///var/run/docker.sock: Post "http://%2Fvar%2Frun%2Fdocker.sock/v1.44/images/my-nginx:latest/tag?repo=mounika1112%2Fmy-nginx&tag=latest"
st": dial unix /var/run/docker.sock: connect: permission denied
[ec2-user@ip-172-31-89-37 my-nginx]$ sudo docker tag my-nginx:latest mounika1112/my-nginx:latest
[ec2-user@ip-172-31-89-37 my-nginx]$ sudo docker push mounika1112/my-nginx:latest
The push refers to repository [docker.io/mounika1112/my-nginx]
55e9644f21c3: Mounted from library/nginx
7d22e2347c12: Mounted from library/nginx
f0d5b15f290e: Mounted from library/nginx
79393a2a2a1a: Mounted from library/nginx
cabe05c000e: Mounted from library/nginx
c69632c459ae: Mounted from library/nginx
5f1ee22ff5e: Mounted from library/nginx
latest: digest: sha256:41733a06511b916545a9806be6db5d98cf15c22fd8d386ed271363b66b2998b size: 1778
[ec2-user@ip-172-31-89-37 my-nginx]$
```

## Verify the image uploaded in Docker hub

The screenshot shows a web browser window with multiple tabs open, including Docker Task -3 - GoCo, Untitled document, Launch an instance, Instance details | EC2, SecurityGroup | EC2, Docker Hub Container, Welcome to nginx!, and others. The main content area is the Docker Hub homepage at [hub.docker.com](https://hub.docker.com). The navigation bar includes 'dockerhub', 'Explore', 'Repositories', 'Organizations', and 'Usage'. A search bar says 'Search Docker Hub' with a 'ctrl+K' keyboard shortcut. On the left, a dropdown shows 'mounika1112' and a search bar for 'Search by repository name'. Below is a table for the repository 'mounika1112/my-nginx':

Name	Last Pushed	Contains	Visibility	Scout
mounika1112/my-nginx	2 minutes ago	IMAGE	Public	Inactive

Below the table, it says '1-1 of 1'. To the right, there's a diagram of three interconnected icons (blue hexagon, red circle, green triangle) with dashed lines between them. Below the diagram are two links: 'Create an organization' and 'Create and manage users and grant access to your repositories.' The bottom of the screen shows the Windows taskbar with various pinned icons and the system tray.