

DOCKER

A Docker is a set of platform as a service (PAAS) products that use OS level virtualization to deliver software in packages called containers. Docker is an open-source platform that enables developers and system administrators to build, ship, and run applications in containers. Containers are lightweight, portable, and self-sufficient environments that include all the dependencies an application needs to run.

Key Features of Docker

1. **Containerization:** Packages applications and their dependencies into isolated containers.
2. **Portability:** Containers can run consistently across different environments (local, development, or production).
3. **Efficiency:** Containers share the host operating system kernel, making them more resource-efficient than virtual machines.
4. **Speed:** Containers are lightweight, start quickly, and require fewer resources.
5. **Version Control:** Docker enables versioning for container images, allowing easy updates and rollbacks.

Created a EC2 instance and launch instance

The screenshot displays the AWS Management Console interface for the 'ap-south-1' region. The left-hand navigation pane shows the 'Instances' section selected. The main content area, titled 'Instances (1/1) Info', shows a table with one instance: 'Docker' with ID 'i-0d3c0cd5bb5537f5f', in a 'Running' state, using the 't2.micro' instance type. Below the table, the 'Details' tab for the selected instance is active, showing an 'Instance summary' with fields for Instance ID, Public IPv4 address (13.201.222.0), Private IPv4 addresses (172.31.3.99), and Instance state. The bottom of the screen shows a Windows taskbar with various application icons and system information like '25°C Cloudy' and '11:20 PM 18-12-2024'.

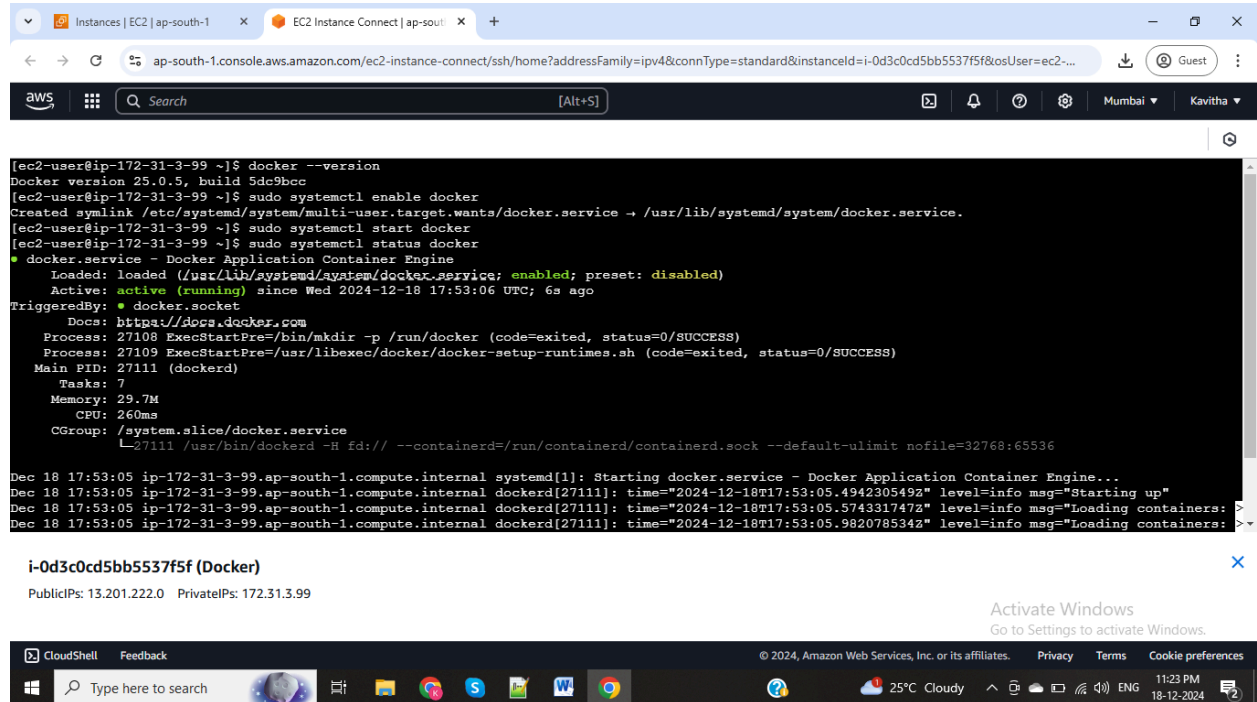
Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability
Docker	i-0d3c0cd5bb5537f5f	Running	t2.micro	Initializing	View alarms +	ap-south-11

i-0d3c0cd5bb5537f5f (Docker)

Instance summary

Instance ID	Public IPv4 address	Private IPv4 addresses
i-0d3c0cd5bb5537f5f	13.201.222.0 open address	172.31.3.99
IPv6 address	Instance state	Public IPv4 DNS

Install docker on EC2 instance – enable, start and check status



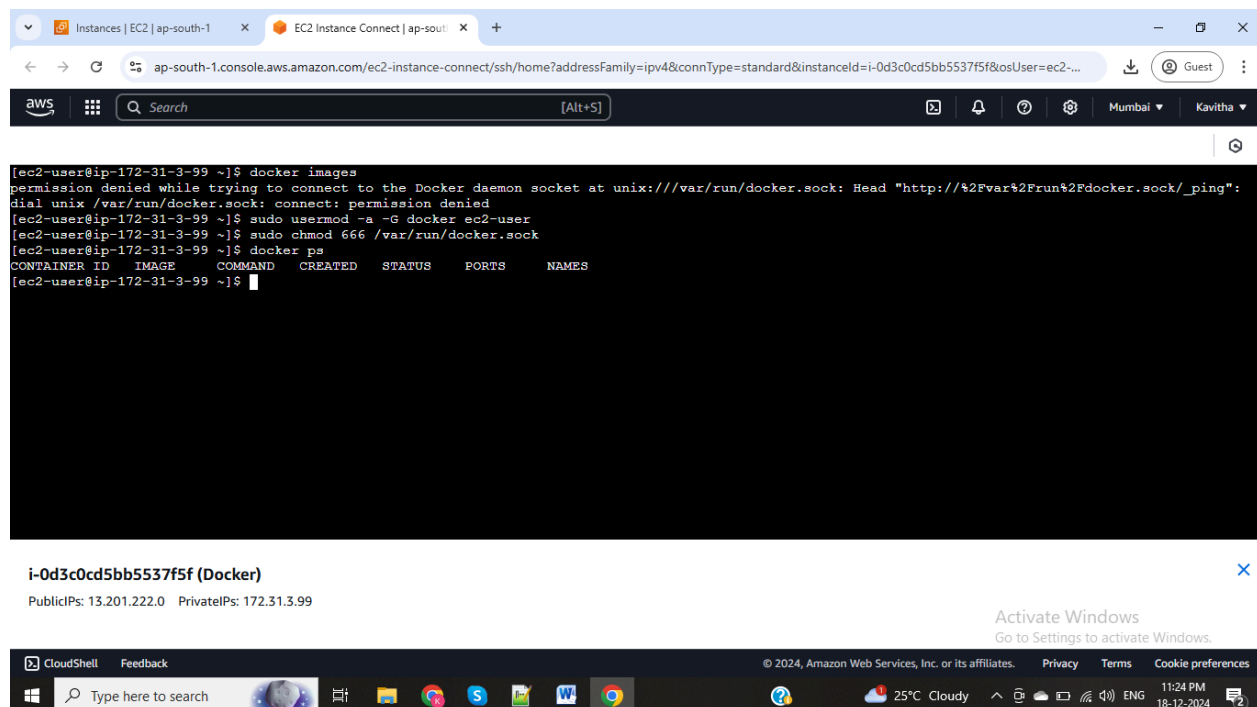
The screenshot shows the AWS Management Console for an EC2 instance named 'ap-south-1'. The terminal window displays the following commands and output:

```
[ec2-user@ip-172-31-3-99 ~]$ docker --version
Docker version 25.0.5, build 5dc9bcc
[ec2-user@ip-172-31-3-99 ~]$ sudo systemctl enable docker
Created symlink /etc/systemd/system/multi-user.target.wants/docker.service → /usr/lib/systemd/system/docker.service.
[ec2-user@ip-172-31-3-99 ~]$ sudo systemctl start docker
[ec2-user@ip-172-31-3-99 ~]$ 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 Wed 2024-12-18 17:53:06 UTC; 6s ago
     TriggeredBy: ● docker.socket
     Docs: https://docs.docker.com
    Process: 27108 ExecStartPre=/bin/mkdir -p /run/docker (code=exited, status=0/SUCCESS)
    Process: 27109 ExecStartPre=/usr/libexec/docker/docker-setup-runtimes.sh (code=exited, status=0/SUCCESS)
   Main PID: 27111 (dockerd)
      Tasks: 7
     Memory: 29.7M
        CPU: 260ms
    CGroup: /system.slice/docker.service
            └─27111 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock --default-ulimit nofile=32768:65536

Dec 18 17:53:05 ip-172-31-3-99.ap-south-1.compute.internal systemd[1]: Starting docker.service - Docker Application Container Engine...
Dec 18 17:53:05 ip-172-31-3-99.ap-south-1.compute.internal dockerd[27111]: time="2024-12-18T17:53:05.494230549Z" level=info msg="Starting up"
Dec 18 17:53:05 ip-172-31-3-99.ap-south-1.compute.internal dockerd[27111]: time="2024-12-18T17:53:05.574331747Z" level=info msg="Loading containers: >
Dec 18 17:53:05 ip-172-31-3-99.ap-south-1.compute.internal dockerd[27111]: time="2024-12-18T17:53:05.982078534Z" level=info msg="Loading containers: >
```

The instance ID is **i-0d3c0cd5bb5537f5f (Docker)**. Public IP: 13.201.222.0, Private IP: 172.31.3.99.

Add ec2-user with docker in groups --> `sudo usermod -a -G docker ec2-user` --> `sudo chmod 666 /var/run/docker.sock`. Now check docker command whether it works.



The screenshot shows the AWS Management Console for the same EC2 instance. The terminal window displays the following commands and output:

```
[ec2-user@ip-172-31-3-99 ~]$ docker images
permission denied while trying to connect to the Docker daemon socket at unix:///var/run/docker.sock: Head "http://%2Fvar%2Frun%2Fdocker.sock/_ping":
dial unix /var/run/docker.sock: connect: permission denied
[ec2-user@ip-172-31-3-99 ~]$ sudo usermod -a -G docker ec2-user
[ec2-user@ip-172-31-3-99 ~]$ sudo chmod 666 /var/run/docker.sock
[ec2-user@ip-172-31-3-99 ~]$ docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS   NAMES
[ec2-user@ip-172-31-3-99 ~]$
```

The instance ID is **i-0d3c0cd5bb5537f5f (Docker)**. Public IP: 13.201.222.0, Private IP: 172.31.3.99.

Login hub.docker.com and search images then nginx and ubuntu pulled from docker hub.

The screenshot shows the AWS Management Console interface. The top navigation bar includes the AWS logo, a search bar, and user information (Mumbai, Kavitha). The main content area displays a terminal window with the following commands and output:

```
[ec2-user@ip-172-31-3-99 ~]$ docker images
permission denied while trying to connect to the Docker daemon socket at unix:///var/run/docker.sock: Head "http://%2Fvar%2Frun%2Fdocker.sock/_ping":
dial unix /var/run/docker.sock: connect: permission denied
[ec2-user@ip-172-31-3-99 ~]$ sudo usermod -a -G docker ec2-user
[ec2-user@ip-172-31-3-99 ~]$ sudo chmod 666 /var/run/docker.sock
[ec2-user@ip-172-31-3-99 ~]$ docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS              PORTS              NAMES
[ec2-user@ip-172-31-3-99 ~]$ docker pull nginx
Using default tag: latest
latest: Pulling from library/nginx
bc0965b23a04: Pull complete
650ee30bbe5e: Pull complete
8cc1569e58f5: Pull complete
362f35df001b: Pull complete
13e320bf29cd: Pull complete
7b50399908e1: Pull complete
57b64962dd44: Pull complete
Digest: sha256:fb197595ebe76b9c0c14ab68159fd3c08bd067ec62300583543f0ebda353b5be
Status: Downloaded newer image for nginx:latest
docker.io/library/nginx:latest
[ec2-user@ip-172-31-3-99 ~]$ docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
nginx                latest             66f8bdd3810c       3 weeks ago        192MB
[ec2-user@ip-172-31-3-99 ~]$
```

Below the terminal output, the Docker image details for 'i-0d3c0cd5bb5537f5f' are shown, including Public and Private IP addresses. The bottom of the screenshot shows the Windows taskbar with various application icons and system tray information.

List the images with the help of docker images command and created containers from the docker image using --> docker run -itd --name nginx-kavi (image id)

The screenshot shows the AWS Management Console interface. The top navigation bar includes the AWS logo, a search bar, and user information (Mumbai, Kavitha). The main content area displays a terminal window with the following commands and output:

```
[ec2-user@ip-172-31-3-99 ~]$ docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
nginx                latest             66f8bdd3810c       3 weeks ago        192MB
ubuntu              latest             b1d9df8ab815       4 weeks ago        78.1MB
httpd               latest             494b2b45fd74       5 months ago       147MB
[ec2-user@ip-172-31-3-99 ~]$ docker ps -a
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS              PORTS              NAMES
886101e759d2       b1d9df8ab815       "/bin/bash"        About a minute ago  Up About a minute  80/tcp             ubuntu-kavi
4bcbf1af0865       66f8bdd3810c       "/docker-entrypoint..."  5 minutes ago      Up 5 minutes       80/tcp             quizzical_gagarin
[ec2-user@ip-172-31-3-99 ~]$ docker run -itd --name nginx-kavi 66f8bdd3810c
2909fbc19ab8a92c39cf2430908e1936e273a58290f9d7a84ab43012a7610721
[ec2-user@ip-172-31-3-99 ~]$ docker ps -a
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS              PORTS              NAMES
2909fbc19ab8       66f8bdd3810c       "/docker-entrypoint..."  4 seconds ago      Up 3 seconds       80/tcp             nginx-kavi
886101e759d2       b1d9df8ab815       "/bin/bash"        3 minutes ago      Up 3 minutes       80/tcp             ubuntu-kavi
4bcbf1af0865       66f8bdd3810c       "/docker-entrypoint..."  6 minutes ago      Up 6 minutes       80/tcp             quizzical_gagarin
[ec2-user@ip-172-31-3-99 ~]$
```

Below the terminal output, the Docker image details for 'i-0d3c0cd5bb5537f5f' are shown, including Public and Private IP addresses. The bottom of the screenshot shows the Windows taskbar with various application icons and system tray information.

Go into the container --> **docker exec -it (ubuntu container id) /bin/bash** -> **apt update -y**
-> **apt install git -y** and check git --version and for come outside the container use ->
control p+q

```
aws
Instances | EC2 | ap-south-1
EC2 Instance Connect | ap-south-1
ap-south-1.console.aws.amazon.com/ec2-instance-connect/ssh/home?addressFamily=ipv4&connType=standard&instanceId=i-0d3c0cd5bb5537f5f&osUser=ec2-...
Search [Alt+S] Mumbai Kavitha

[ec2-user@ip-172-31-3-99 ~]$ docker images
REPOSITORY TAG IMAGE ID CREATED SIZE
nginx latest 66f8bdd3810c 3 weeks ago 192MB
docker-task v1 b1d9df8ab815 4 weeks ago 78.1MB
ubuntu latest b1d9df8ab815 4 weeks ago 78.1MB
httpd latest 494b2b45fd74 5 months ago 147MB
[ec2-user@ip-172-31-3-99 ~]$ docker ps -a
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
2909fbc19ab8 66f8bdd3810c "/docker-entrypoint..." 16 minutes ago Up 16 minutes 80/tcp nginx-kavi
886101e759d2 b1d9df8ab815 "/bin/bash" 20 minutes ago Up 20 minutes ubuntu-kavi
4bcbf1af0865 66f8bdd3810c "/docker-entrypoint..." 23 minutes ago Up 23 minutes 80/tcp quizzical_gagarin
[ec2-user@ip-172-31-3-99 ~]$ docker exec -it 886101e759d2 /bin/bash
root@886101e759d2:/# git --version
git version 2.43.0
root@886101e759d2:/#
```

i-0d3c0cd5bb5537f5f (Docker)

PublicIPs: 13.201.222.0 PrivateIPs: 172.31.3.99

Activate Windows
Go to Settings to activate Windows.



Created a Image from the container to push docker hub: **Docker commit -m "demo" -a "name" (container id) docker hub username**

```
aws
Instances | EC2 | ap-south-1
EC2 Instance Connect | ap-south-1
ap-south-1.console.aws.amazon.com/ec2-instance-connect/ssh/home?addressFamily=ipv4&connType=standard&instanceId=i-0d3c0cd5bb5537f5f&osUser=ec2-...
Search [Alt+S] Mumbai Kavitha

REPOSITORY TAG IMAGE ID CREATED SIZE
nginx latest 66f8bdd3810c 3 weeks ago 192MB
docker-task v1 b1d9df8ab815 4 weeks ago 78.1MB
ubuntu latest b1d9df8ab815 4 weeks ago 78.1MB
httpd latest 494b2b45fd74 5 months ago 147MB
[ec2-user@ip-172-31-3-99 ~]$ docker ps -a
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
2909fbc19ab8 66f8bdd3810c "/docker-entrypoint..." 16 minutes ago Up 16 minutes 80/tcp nginx-kavi
886101e759d2 b1d9df8ab815 "/bin/bash" 20 minutes ago Up 20 minutes ubuntu-kavi
4bcbf1af0865 66f8bdd3810c "/docker-entrypoint..." 23 minutes ago Up 23 minutes 80/tcp quizzical_gagarin
[ec2-user@ip-172-31-3-99 ~]$ docker exec -it 886101e759d2 /bin/bash
root@886101e759d2:/# git --version
git version 2.43.0
root@886101e759d2:/# read escape sequence
[ec2-user@ip-172-31-3-99 ~]$ docker commit -m "demo" -a ubuntu-kavi 886101e759d2 kavi1122
sha256:9b1bf8742d209627c6e2541326f394c19fc906c4ec2ff96c262f38c517e48952
[ec2-user@ip-172-31-3-99 ~]$ docker images
REPOSITORY TAG IMAGE ID CREATED SIZE
kavi1122 latest 9b1bf8742d20 54 seconds ago 206MB
nginx latest 66f8bdd3810c 3 weeks ago 192MB
docker-task v1 b1d9df8ab815 4 weeks ago 78.1MB
ubuntu latest b1d9df8ab815 4 weeks ago 78.1MB
httpd latest 494b2b45fd74 5 months ago 147MB
[ec2-user@ip-172-31-3-99 ~]$
```

i-0d3c0cd5bb5537f5f (Docker)

PublicIPs: 13.201.222.0 PrivateIPs: 172.31.3.99

Activate Windows
Go to Settings to activate Windows.



Login hub.docker.com → Created a new repository

nginx - Official Image x +

hub.docker.com/repository/create?namespace=kavi1122

New More Docker. Easy Access. New Streamlined Plans. Learn more. →

dockerhub Explore Repositories Organizations Usage

Repositories / Create

Using 0 of 1 private repositories.

Create repository

Namespace: kavi1122

Repository Name: docker-task

Short description

A short description to identify your repository. If the repository is public, this description is used to index your content on Docker Hub and in search engines, and is visible to users in search results.

Visibility

Using 0 of 1 private repositories. [Get more](#)

☒ Public Appears in Docker Hub search results

☐ Private Only visible to you

[Cancel](#) [Create](#)

Pushing images

You can push a new image to this repository using the CLI:

```
docker tag local-image:tagname new-repo:tagname
docker push new-repo:tagname
```

Make sure to replace `tagname` with your desired image repository tag.

Activate Windows
Go to Settings to activate Windows.

Check docker images → docker tag (image id) (reponame):tag - created a new image

```
[ec2-user@ip-172-31-3-99 ~]$ docker images
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE
nginx         latest    66f8bdd3810c   3 weeks ago    192MB
ubuntu        latest    b1d9df8ab815   4 weeks ago    78.1MB
httpd         latest    494b2b45fd74   5 months ago   147MB

[ec2-user@ip-172-31-3-99 ~]$ docker ps -a
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS          NAMES
2909fbc19ab8   66f8bdd3810c   "/docker-entrypoint..." 2 minutes ago  Up 2 minutes  80/tcp         nginx-kavi
886101e759d2   b1d9df8ab815   "/bin/bash"              5 minutes ago  Up 5 minutes  80/tcp         ubuntu-kavi
4bcbf1af0865   66f8bdd3810c   "/docker-entrypoint..." 9 minutes ago  Up 9 minutes  80/tcp         quizzical_gagarin

[ec2-user@ip-172-31-3-99 ~]$ docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS          NAMES
2909fbc19ab8   66f8bdd3810c   "/docker-entrypoint..." 4 minutes ago  Up 4 minutes  80/tcp         nginx-kavi
886101e759d2   b1d9df8ab815   "/bin/bash"              7 minutes ago  Up 7 minutes  80/tcp         ubuntu-kavi
4bcbf1af0865   66f8bdd3810c   "/docker-entrypoint..." 11 minutes ago Up 11 minutes  80/tcp         quizzical_gagarin

[ec2-user@ip-172-31-3-99 ~]$ docker tag b1d9df8ab815 docker-task:v1
[ec2-user@ip-172-31-3-99 ~]$ docker images
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE
nginx         latest    66f8bdd3810c   3 weeks ago    192MB
docker-task    v1        b1d9df8ab815   4 weeks ago    78.1MB
ubuntu        latest    b1d9df8ab815   4 weeks ago    78.1MB
httpd         latest    494b2b45fd74   5 months ago   147MB

[ec2-user@ip-172-31-3-99 ~]$
```

i-0d3c0cd5bb5537f5f (Docker)

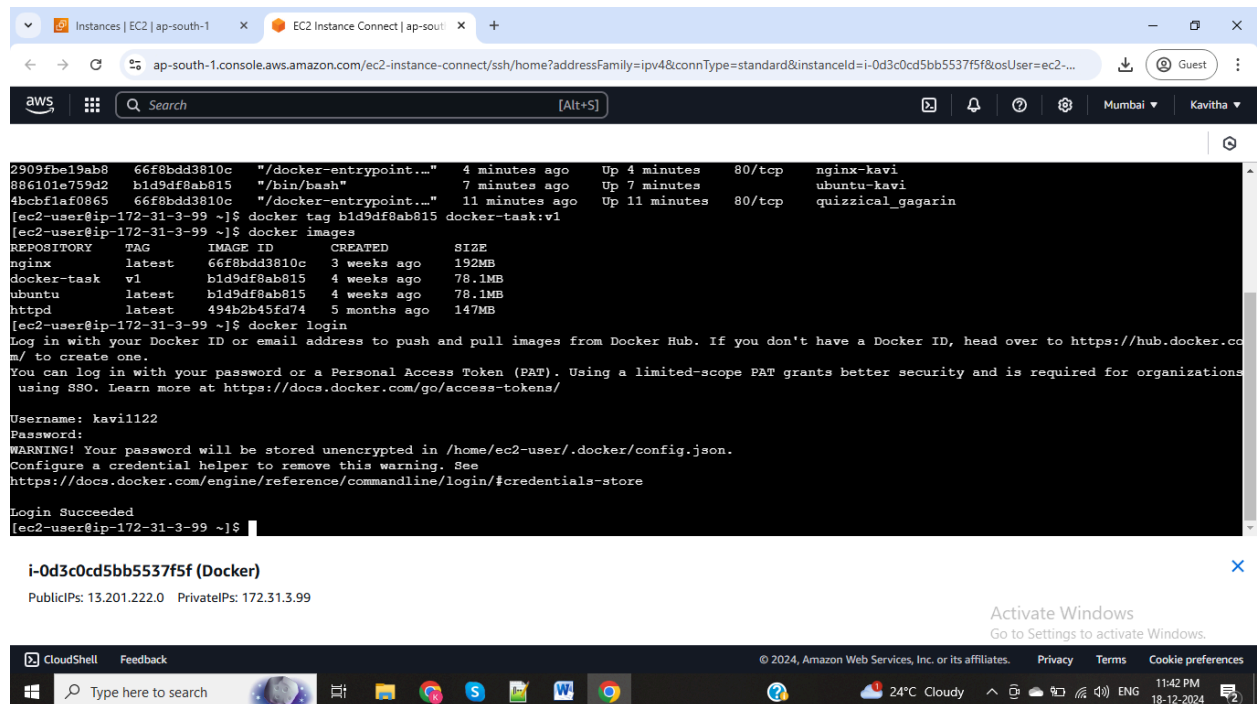
PublicIPs: 13.201.222.0 PrivateIPs: 172.31.3.99

Activate Windows
Go to Settings to activate Windows.

CloudShell Feedback

© 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

For pushing the image to docker hub for this give → docker login and provide credentials and docker login succeeded



The screenshot shows a terminal window within the AWS CloudShell interface. The terminal output includes the following commands and results:

```
2909f8e19ab8 66f8bdd3810c "/docker-entrypoint..." 4 minutes ago Up 4 minutes 80/tcp nginx-kavi
886101e759d2 b1d9df8ab815 "/bin/bash" 7 minutes ago Up 7 minutes ubuntu-kavi
4bc6f1af0865 66f8bdd3810c "/docker-entrypoint..." 11 minutes ago Up 11 minutes quizzical_gagarin
[ec2-user@ip-172-31-3-99 ~]$ docker tag b1d9df8ab815 docker-task:v1
[ec2-user@ip-172-31-3-99 ~]$ docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
nginx	latest	66f8bdd3810c	3 weeks ago	192MB
docker-task	v1	b1d9df8ab815	4 weeks ago	78.1MB
ubuntu	latest	b1d9df8ab815	4 weeks ago	78.1MB
htptd	latest	494b2b45fd74	5 months ago	147MB

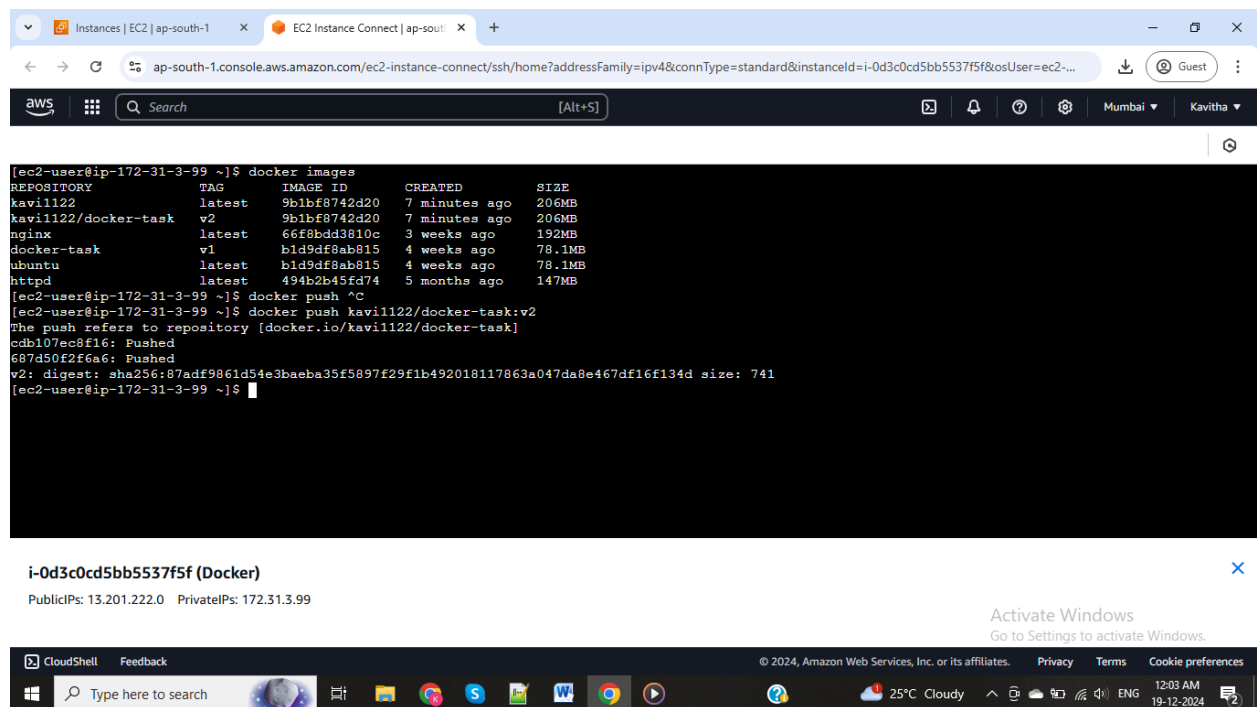
```
[ec2-user@ip-172-31-3-99 ~]$ docker login
Log in with your Docker ID or email address to push and pull images from Docker Hub. If you don't have a Docker ID, head over to https://hub.docker.com/ to create one.
You can log in with your password or a Personal Access Token (PAT). Using a limited-scope PAT grants better security and is required for organizations using SSO. Learn more at https://docs.docker.com/go/access-tokens/
Username: kavil122
Password:
WARNING! Your password will be stored unencrypted in /home/ec2-user/.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-172-31-3-99 ~]$
```

Below the terminal output, the Docker instance details are shown:

i-0d3c0cd5bb5537f5f (Docker)
PublicIPs: 13.201.222.0 PrivateIPs: 172.31.3.99

Now, give docker push command: docker push repository name:tag → image successfully pushed to Docker Hub



The screenshot shows a terminal window within the AWS CloudShell interface. The terminal output includes the following commands and results:

```
[ec2-user@ip-172-31-3-99 ~]$ docker images
REPOSITORY          TAG          IMAGE ID          CREATED          SIZE
kavil122             latest       9b1bf8742d20     7 minutes ago   206MB
kavil122/docker-task v2          9b1bf8742d20     7 minutes ago   206MB
nginx                latest       66f8bdd3810c     3 weeks ago     192MB
docker-task          v1          b1d9df8ab815     4 weeks ago     78.1MB
ubuntu               latest       b1d9df8ab815     4 weeks ago     78.1MB
htptd                latest       494b2b45fd74     5 months ago    147MB
[ec2-user@ip-172-31-3-99 ~]$ docker push ^C
[ec2-user@ip-172-31-3-99 ~]$ docker push kavil122/docker-task:v2
The push refers to repository [docker.io/kavil122/docker-task]
c8b107ec8f16: Pushed
687450f2f6a6: Pushed
v2: digest: sha256:87adf9861d54e3baeba35f5897f29f1b492018117863a047da8e467df16f134d size: 741
[ec2-user@ip-172-31-3-99 ~]$
```

Below the terminal output, the Docker instance details are shown:

i-0d3c0cd5bb5537f5f (Docker)
PublicIPs: 13.201.222.0 PrivateIPs: 172.31.3.99

Successfully image pushed into Docker Hub and output

The screenshot shows the Docker Hub repository page for `kavi1122/docker-task`. The page includes a header with navigation tabs (General, Tags, Builds, Collaborators, Webhooks, Settings) and a sub-header with 'Add a description' and 'Add a category' links. The main content area is divided into three sections: 'Tags' (showing a single tag 'v2'), 'Automated builds' (with an 'Upgrade' button), and 'Repository overview' (with an 'INCOMPLETE' status). The 'Tags' section contains a table with columns: Tag, OS, Type, Pulled, and Pushed. The 'Automated builds' section includes a description of the feature and a link to 'Read more about automated builds'. The 'Repository overview' section has a link to 'Activate Windows'.

Tag	OS	Type	Pulled	Pushed
v2	Linux	Image	---	a few seconds ago

For Deleting all the containers and images → `docker rm $(docker ps -a -q) →`
`docker rmi $(docker images -a)`

The screenshot shows the AWS CloudShell terminal interface. The terminal displays the following commands and output:

```
[ec2-user@ip-172-31-3-99 ~]$ docker ps -a
CONTAINER ID        IMAGE               COMMAND                  CREATED            STATUS              PORTS               NAMES
2909f8e19ab8        66f8bdd3810c       "/docker-entrypoint..." 29 minutes ago     Exited (0) 15 seconds ago           nginx-kavi
886101e759d2        b1d9df8ab815       "/bin/bash"              33 minutes ago     Exited (137) 5 seconds ago          ubuntu-kavi

[ec2-user@ip-172-31-3-99 ~]$ docker rm $(docker ps -a -q)
2909f8e19ab8
886101e759d2

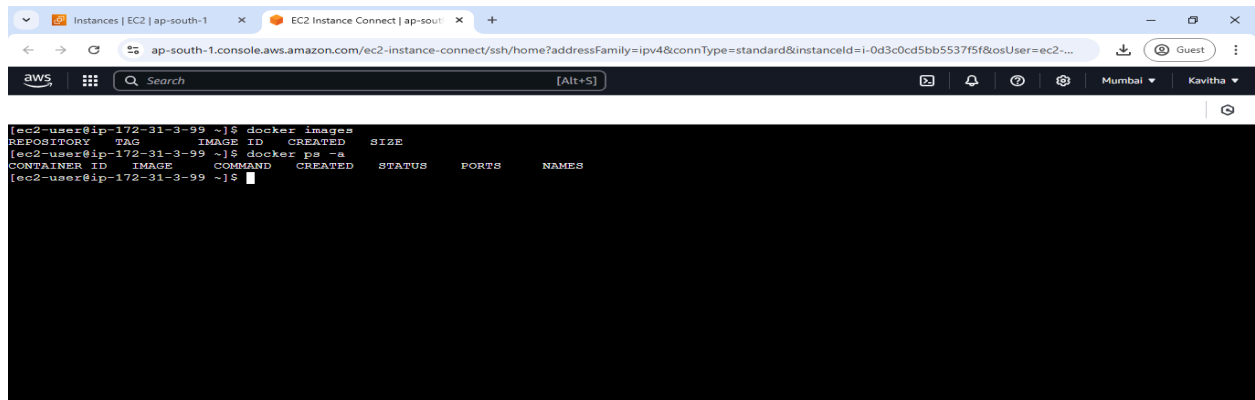
[ec2-user@ip-172-31-3-99 ~]$ docker images
REPOSITORY          TAG                 IMAGE ID            CREATED            SIZE
kavi1122             latest             9b1bf8742d20       10 minutes ago    206MB
kavi1122/docker-task v2                 9b1bf8742d20       10 minutes ago    206MB
nginx                latest             66f8bdd3810c       3 weeks ago       192MB
docker-task          v1                 b1d9df8ab815       4 weeks ago       78.1MB
ubuntu               latest             b1d9df8ab815       4 weeks ago       78.1MB
httpd                latest             494b2b45fd74       5 months ago      147MB

[ec2-user@ip-172-31-3-99 ~]$ docker rmi $(docker images -a)
Untagged: kavi1122:latest
Untagged: kavi1122/docker-task:v2
Untagged: kavi1122/docker-task@sha256:87adf9861d54e3baeba35f5897f29f1b492018117863a047da8e467df16f134d
Deleted: sha256:9b1bf8742d209627c6e2541326f394c19fc906c4ec2ff96c262f38c517e48952
Deleted: sha256:e0cc15f189e1a491dc500d56729b26bba699f031b92b7e6db20046cec53005b
Untagged: nginx:latest
Untagged: nginx@sha256:fb197595ebe76b9c0c14ab68159fd3c08bd067ec62300583543f0ebda353b5be
Deleted: sha256:66f8bdd3810c96dc5c28aec39583af731b34a2cd99471530f53c8794ed5b423e
```

The terminal output shows the successful deletion of all containers and images. The final output is:

```
Untagged: kavi1122:latest
Untagged: kavi1122/docker-task:v2
Untagged: kavi1122/docker-task@sha256:87adf9861d54e3baeba35f5897f29f1b492018117863a047da8e467df16f134d
Deleted: sha256:9b1bf8742d209627c6e2541326f394c19fc906c4ec2ff96c262f38c517e48952
Deleted: sha256:e0cc15f189e1a491dc500d56729b26bba699f031b92b7e6db20046cec53005b
Untagged: nginx:latest
Untagged: nginx@sha256:fb197595ebe76b9c0c14ab68159fd3c08bd067ec62300583543f0ebda353b5be
Deleted: sha256:66f8bdd3810c96dc5c28aec39583af731b34a2cd99471530f53c8794ed5b423e
```


Deleted containers and images output

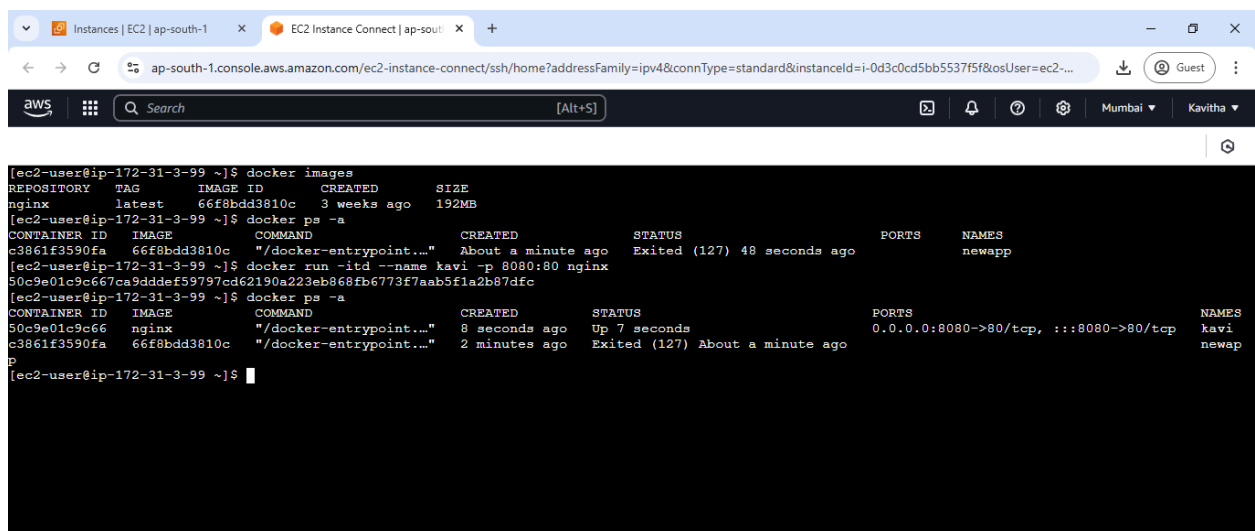


```
[ec2-user@ip-172-31-3-99 ~]$ docker images
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE
[ec2-user@ip-172-31-3-99 ~]$ docker ps -a
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS          NAMES
[ec2-user@ip-172-31-3-99 ~]$
```

Inside the Container we have to access the application

Pull nginx image from dockerhub and create the container with port number for run the application in new web server or outside the server.

Docker run -it --name kavi -p 8080:80 (image name)



```
[ec2-user@ip-172-31-3-99 ~]$ docker images
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE
nginx         latest    66f8bdd3810c   3 weeks ago    192MB
[ec2-user@ip-172-31-3-99 ~]$ docker ps -a
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS          NAMES
c3861f3590fa   66f8bdd3810c   "/docker-entrypoint..." About a minute ago   Exited (127) 48 seconds ago           newapp
[ec2-user@ip-172-31-3-99 ~]$ docker run -itd --name kavi -p 8080:80 nginx
50c9e01c9c667ca9dddef59797cd62190a223eb868fb6773f7aab5f1a2b87dfc
[ec2-user@ip-172-31-3-99 ~]$ docker ps -a
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS          NAMES
50c9e01c9c66   nginx     "/docker-entrypoint..." 8 seconds ago   Up 7 seconds   0.0.0.0:8080->80/tcp, :::8080->80/tcp   kavi
c3861f3590fa   66f8bdd3810c   "/docker-entrypoint..." 2 minutes ago   Exited (127) About a minute ago           newapp
[ec2-user@ip-172-31-3-99 ~]$
```

i-0d3c0cd5bb5537f5f (Docker)

PublicIPs: 13.201.222.0 PrivateIPs: 172.31.3.99

Activate Windows
Go to Settings to activate Windows.



To access the application edit the inbound rules in security group and add 8080 port

The screenshot shows the AWS Management Console interface for editing inbound rules of a security group. The breadcrumb navigation indicates the path: EC2 > Security Groups > sg-0759b4735db15d895 - launch-wizard-1 > Edit inbound rules. The main heading is 'Edit inbound rules' with an 'Info' link. Below it, a note states: 'Inbound rules control the incoming traffic that's allowed to reach the instance.'

The 'Inbound rules' section contains a table with the following columns: Security group rule ID, Type, Protocol, Port range, Source, and Description - optional. There are two rules listed:

Security group rule ID	Type	Protocol	Port range	Source	Description - optional
sg-02e3011aaa650328c	SSH	TCP	22	Cu...	
-	Custom TCP	TCP	8080	An...	

Below the table is an 'Add rule' button. A warning message at the bottom states: 'Rules with source of 0.0.0.0/0 or ::/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.'

For checking whether it works use EC2 instance public ip address with port 8080

The screenshot shows a web browser window with the URL '13.201.222.0:8080'. The page title is 'Welcome to nginx!'. The main content of the page reads: '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. Commercial support is available at nginx.com. Thank you for using nginx.'

Pull httpd image from Docker Hub and create a new container with new port number for access the application in container with new port number

Instances | EC2 | ap-south-1

EC2 Instance Connect | ap-south-1

ap-south-1.console.aws.amazon.com/ec2-instance-connect/ssh/home?addressFamily=ipv4&connType=standard&instanceId=i-0d3c0cd5bb5537f5f&osUser=ec2-...

aws

Search [Alt+S]

Mumbai Kavitha

```
[ec2-user@ip-172-31-3-99 ~]$ docker images
REPOSITORY TAG IMAGE ID CREATED SIZE
nginx latest 66f8bdd3810c 3 weeks ago 192MB
httpd latest 494b2b45fd74 5 months ago 147MB

[ec2-user@ip-172-31-3-99 ~]$ docker ps -a
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
50c9e01c9c66 nginx "/docker-entrypoint..." 3 minutes ago Up 3 minutes 0.0.0.0:8080->80/tcp, :::8080->80/tcp kavi
c3861f3590fa 66f8bdd3810c "/docker-entrypoint..." 5 minutes ago Exited (127) 5 minutes ago newapp

[ec2-user@ip-172-31-3-99 ~]$ docker run -itd --name demo -p 8081:80 httpd
9bbdfcfc4445a6a8e9d9cfbf0dd4253e9da8ea7da9d3f4c7dc66bb1a261e1fb

[ec2-user@ip-172-31-3-99 ~]$
```

i-0d3c0cd5bb5537f5f (Docker)

PublicIPs: 13.201.222.0 PrivateIPs: 172.31.3.99

Activate Windows
Go to Settings to activate Windows.

CloudShell Feedback

Type here to search

© 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

12:15 AM 19-12-2024

Edited the inbound rules to run the application in outside the web server

ModifyInboundSecurityGroupR...

EC2 Instance Connect | ap-south-1

ap-south-1.console.aws.amazon.com/ec2/home?region=ap-south-1#ModifyInboundSecurityGroupRules:securityGroupId=sg-0759b4735db15d895

aws

Search [Alt+S]

Mumbai Kavitha

EC2

Security Groups

sg-0759b4735db15d895 - launch-wizard-1

Edit inbound rules

Edit inbound rules

Inbound rules control the incoming traffic that's allowed to reach the instance.

Security group rule ID	Type	Protocol	Port range	Source	Description - optional	
sg-r01e04ba49ccdc2fc2	Custom TCP	TCP	8080	Cu...		Delete
sg-r02e3011aaa650328c	SSH	TCP	22	Cu...		Delete
-	Custom TCP	TCP	8081	An...	0.0.0.0/0	Delete

Add rule

Activate Windows
Go to Settings to activate Windows.

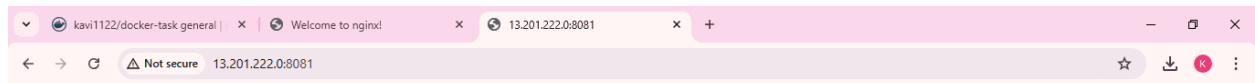
CloudShell Feedback

Type here to search

© 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

12:16 AM 19-12-2024

Access the application with public ip with port 8081



It works!

