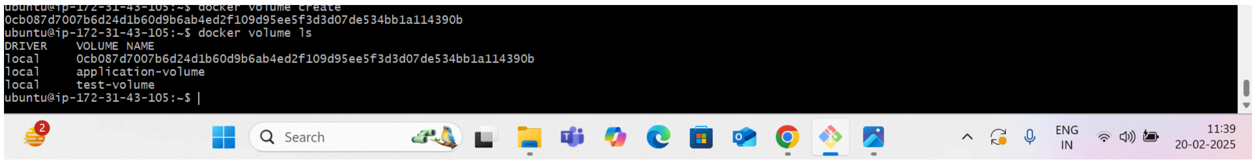
Creating a volume

Creating a volume without name

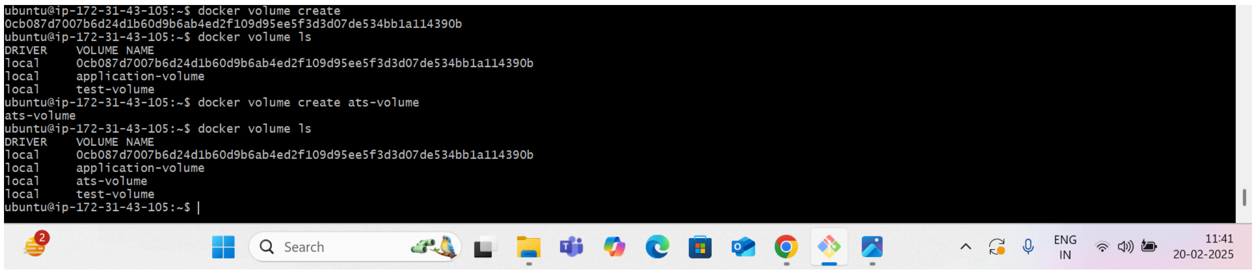
Command: docker volume create



```
ubuntu@ip-172-31-43-105:~$ docker volume create
0cb087d7007b6d24d1b60d9b6ab4ed2f109d95ee5f3d3d07de534bb1a114390b
ubuntu@ip-172-31-43-105:~$ docker volume ls
DRIVER    VOLUME NAME
local     0cb087d7007b6d24d1b60d9b6ab4ed2f109d95ee5f3d3d07de534bb1a114390b
local     application-volume
local     test-volume
ubuntu@ip-172-31-43-105:~$ |
```

Creating volume with name

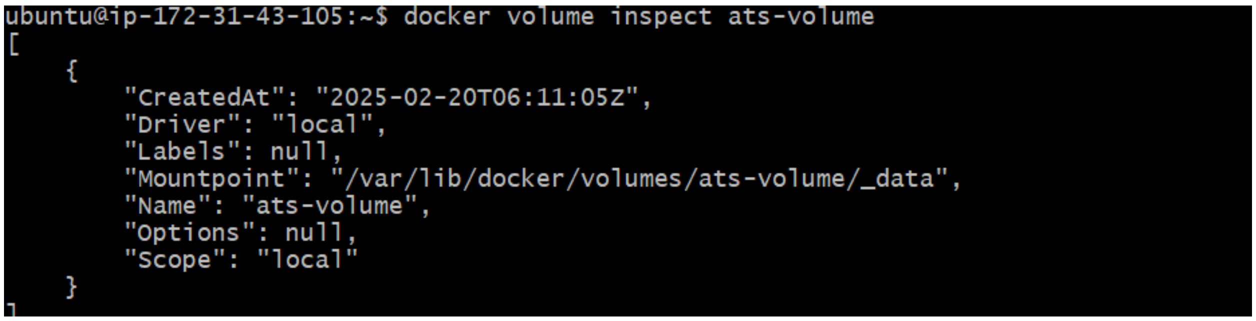
Command: docker volume create ats-volume



```
ubuntu@ip-172-31-43-105:~$ docker volume create
0cb087d7007b6d24d1b60d9b6ab4ed2f109d95ee5f3d3d07de534bb1a114390b
ubuntu@ip-172-31-43-105:~$ docker volume ls
DRIVER    VOLUME NAME
local     0cb087d7007b6d24d1b60d9b6ab4ed2f109d95ee5f3d3d07de534bb1a114390b
local     application-volume
local     test-volume
ubuntu@ip-172-31-43-105:~$ docker volume create ats-volume
ats-volume
ubuntu@ip-172-31-43-105:~$ docker volume ls
DRIVER    VOLUME NAME
local     0cb087d7007b6d24d1b60d9b6ab4ed2f109d95ee5f3d3d07de534bb1a114390b
local     application-volume
local     ats-volume
local     test-volume
ubuntu@ip-172-31-43-105:~$ |
```

Inspect the ats-volume to check the path of the volume

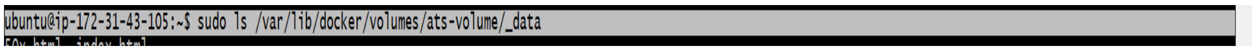
Command: docker volume inspect ats-volume



```
ubuntu@ip-172-31-43-105:~$ docker volume inspect ats-volume
[
    {
        "CreatedAt": "2025-02-20T06:11:05Z",
        "Driver": "local",
        "Labels": null,
        "Mountpoint": "/var/lib/docker/volumes/ats-volume/_data",
        "Name": "ats-volume",
        "Options": null,
        "Scope": "local"
    }
]
```

Check the path, inside the volume do we have any files inside path

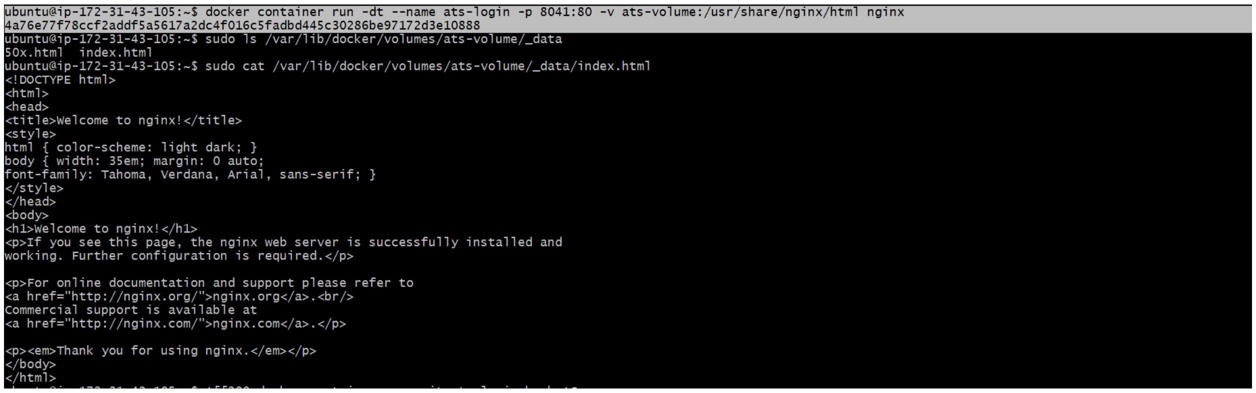
Command: sudo ls /var/lib/docker/volumes/ats-volume/_data



```
ubuntu@ip-172-31-43-105:~$ sudo ls /var/lib/docker/volumes/ats-volume/_data
50x.html  index.html
```

Now create a container using volume = ats-volume

Command: docker container run -dt --name ats-login -p 8041:80 -v ats-volume:/usr/share/nginx/html nginx



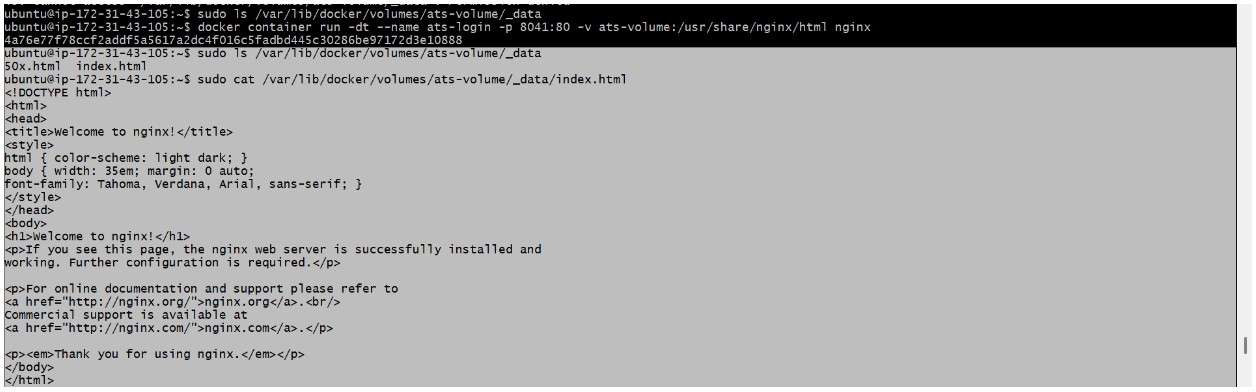
```
ubuntu@ip-172-31-43-105:~$ docker container run -dt --name ats-login -p 8041:80 -v ats-volume:/usr/share/nginx/html nginx
4a76e77f78ccf2addf5a5617a2dc4f016c5fadbd445c30286be97172d3e10888
ubuntu@ip-172-31-43-105:~$ sudo ls /var/lib/docker/volumes/ats-volume/_data
50x.html  index.html
ubuntu@ip-172-31-43-105:~$ sudo cat /var/lib/docker/volumes/ats-volume/_data/index.html
<!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>Welcome to nginx!</h1>
<p>If you see this page, the nginx web server is successfully installed and
working. Further configuration is required.</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>
```

Now check the volume path is there any files by using below command

Command: sudo cat /var/lib/docker/volumes/ats-volume/_data/index.html



```
ubuntu@ip-172-31-43-105:~$ sudo ls /var/lib/docker/volumes/ats-volume/_data
ubuntu@ip-172-31-43-105:~$ docker container run -dt --name ats-login -p 8041:80 -v ats-volume:/usr/share/nginx/html nginx
4a76e77f78ccf2addf5a5617a2dc4f016c5fadbd445c30286be97172d3e10888
ubuntu@ip-172-31-43-105:~$ sudo ls /var/lib/docker/volumes/ats-volume/_data
50x.html  index.html
ubuntu@ip-172-31-43-105:~$ sudo cat /var/lib/docker/volumes/ats-volume/_data/index.html
<!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>Welcome to nginx!</h1>
<p>If you see this page, the nginx web server is successfully installed and
working. Further configuration is required.</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>
```

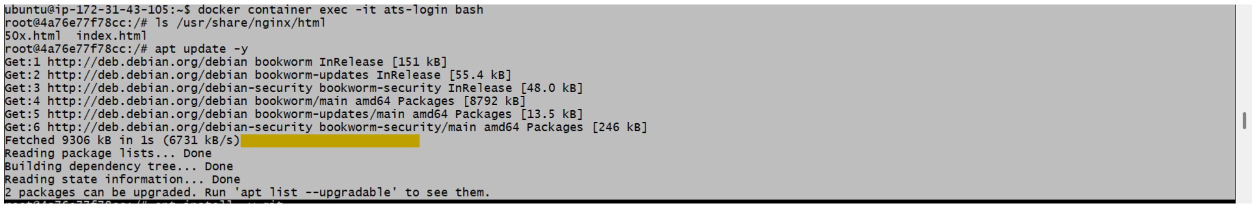
Now go inside the container by using below command:

Command: docker container exec -it ats-login bash

Command: ls /usr/share/nginx/html

Update the system with below command:

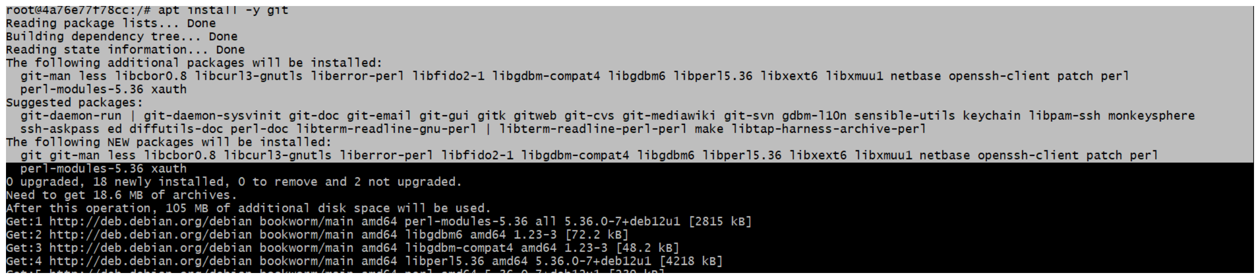
Command: apt update -y



```
ubuntu@ip-172-31-43-105:~$ docker container exec -it ats-login bash
root@4a76e77f78cc:/# ls /usr/share/nginx/html
50x.html  index.html
root@4a76e77f78cc:/# apt update -y
Get:1 http://deb.debian.org/debian bookworm InRelease [151 kB]
Get:2 http://deb.debian.org/debian bookworm-updates InRelease [55.4 kB]
Get:3 http://deb.debian.org/debian-security bookworm-security InRelease [48.0 kB]
Get:4 http://deb.debian.org/debian bookworm/main amd64 Packages [8792 kB]
Get:5 http://deb.debian.org/debian bookworm-updates/main amd64 Packages [13.5 kB]
Get:6 http://deb.debian.org/debian-security bookworm-security/main amd64 Packages [246 kB]
Fetched 9306 kB in 1s (6731 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
2 packages can be upgraded. Run 'apt list --upgradable' to see them.
```

To Install git use below command:

Command: apt install -y git

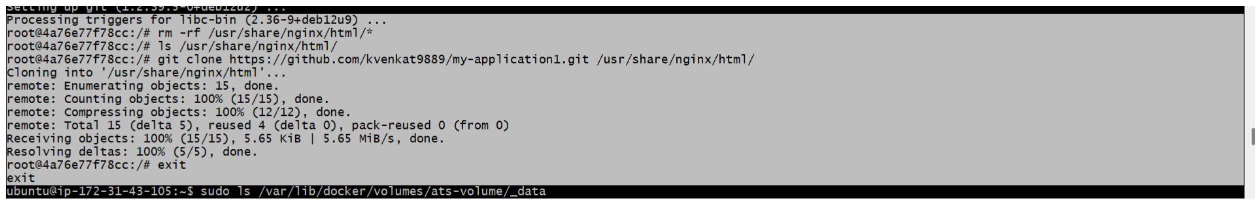


```
root@4a76e77f78cc:/# apt install -y git
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  git-man less libcbor0.8 libcurl3-gnutls liberror-perl libfido2-1 libgdbm-compat4 libgdbm6 libperl5.36 libxext6 libxmuu1 netbase openssh-client patch perl
  perl-modules-5.36 xauth
Suggested packages:
  git-daemon-run | git-daemon-sysvinit git-doc git-email git-gui gitk gitweb git-cvs git-mediawiki git-svn gdbm-l10n sensible-utils keychain libpam-ssh monkeysphere
  ssh-askpass ed diffutils-doc perl-doc libterm-readline-gnu-perl | libterm-readline-perl-perl make libtap-harness-archive-perl
The following NEW packages will be installed:
  git git-man less libcbor0.8 libcurl3-gnutls liberror-perl libfido2-1 libgdbm-compat4 libgdbm6 libperl5.36 libxext6 libxmuu1 netbase openssh-client patch perl
  perl-modules-5.36 xauth
0 upgraded, 18 newly installed, 0 to remove and 2 not upgraded.
Need to get 18.6 MB of archives.
After this operation, 105 MB of additional disk space will be used.
Get:1 http://deb.debian.org/debian bookworm/main amd64 perl-modules-5.36 all 5.36.0-7+deb12u1 [2815 kB]
Get:2 http://deb.debian.org/debian bookworm/main amd64 libgdbm6 amd64 1.23-3 [72.2 kB]
Get:3 http://deb.debian.org/debian bookworm/main amd64 libgdbm-compat4 amd64 1.23-3 [48.2 kB]
Get:4 http://deb.debian.org/debian bookworm/main amd64 libperl5.36 amd64 5.36.0-7+deb12u1 [4218 kB]
```

Now list the path with below command:
Command: ls /usr/share/nginx/html

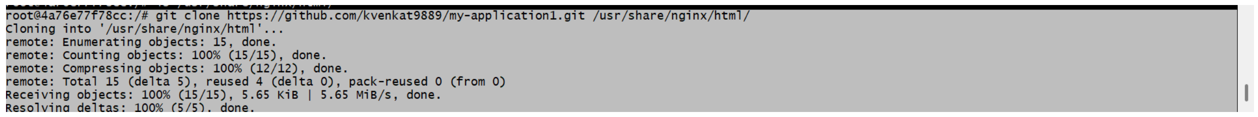
Remove the files which are inside nginx path with below command:
Command: rm -rf /usr/share/nginx/html/*

Now list the path and check do you find any files after deleting ?
Command: ls /usr/share/nginx/html



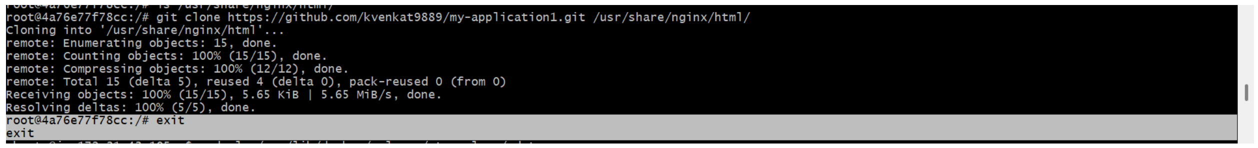
```
Processing triggers for libc-bin (2.36-9+deb12u9) ...
root@4a76e77f78cc:/# rm -rf /usr/share/nginx/html/*
root@4a76e77f78cc:/# ls /usr/share/nginx/html/
root@4a76e77f78cc:/# git clone https://github.com/kvenkat9889/my-application1.git /usr/share/nginx/html/
Cloning into '/usr/share/nginx/html'...
remote: Enumerating objects: 15, done.
remote: Counting objects: 100% (15/15), done.
remote: Compressing objects: 100% (12/12), done.
remote: Total 15 (delta 5), reused 4 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (15/15), 5.65 KiB | 5.65 MiB/s, done.
Resolving deltas: 100% (5/5), done.
root@4a76e77f78cc:/# exit
exit
ubuntu@ip-172-31-43-105:~$ sudo ls /var/lib/docker/volumes/ats-volume/_data
```

Now clone the Application URL code files from github with below command:
Command: git clone https://github.com/kvenkat9889/my-application1.git /usr/share/nginx/html/



```
root@4a76e77f78cc:/# git clone https://github.com/kvenkat9889/my-application1.git /usr/share/nginx/html/
Cloning into '/usr/share/nginx/html'...
remote: Enumerating objects: 15, done.
remote: Counting objects: 100% (15/15), done.
remote: Compressing objects: 100% (12/12), done.
remote: Total 15 (delta 5), reused 4 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (15/15), 5.65 KiB | 5.65 MiB/s, done.
Resolving deltas: 100% (5/5), done.
```

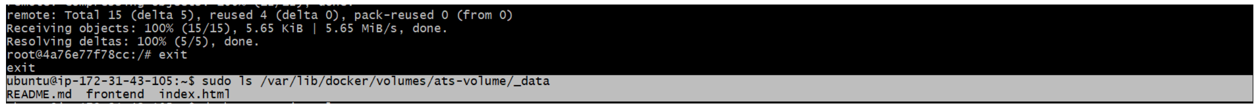
Now exit from container
Command: exit



```
root@4a76e77f78cc:/# git clone https://github.com/kvenkat9889/my-application1.git /usr/share/nginx/html/
Cloning into '/usr/share/nginx/html'...
remote: Enumerating objects: 15, done.
remote: Counting objects: 100% (15/15), done.
remote: Compressing objects: 100% (12/12), done.
remote: Total 15 (delta 5), reused 4 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (15/15), 5.65 KiB | 5.65 MiB/s, done.
Resolving deltas: 100% (5/5), done.
root@4a76e77f78cc:/# exit
exit
```

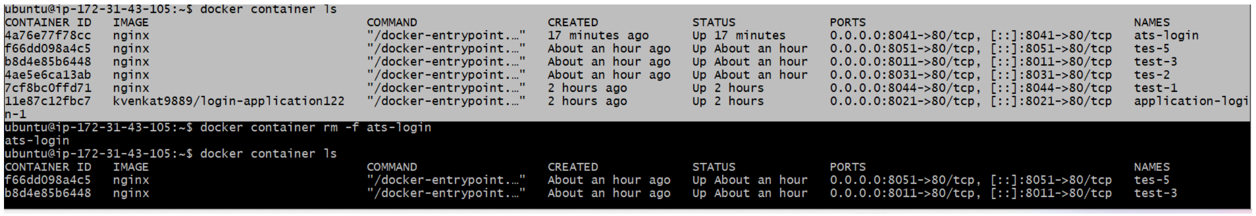
Now you are back to host, and lets check the path of the volume with below command
Command: sudo ls /var/lib/docker/volumes/ats-volume/_data

Note: Here you can see the URL code files



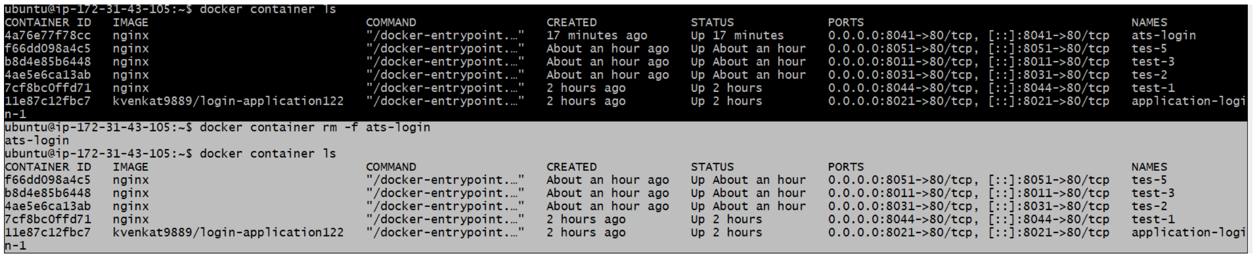
```
remote: Total 15 (delta 5), reused 4 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (15/15), 5.65 KiB | 5.65 MiB/s, done.
Resolving deltas: 100% (5/5), done.
root@4a76e77f78cc:/# exit
exit
ubuntu@ip-172-31-43-105:~$ sudo ls /var/lib/docker/volumes/ats-volume/_data
README.md  frontend  index.html
```

Now check the docker containers with ls command
Command: docker container ls



```
ubuntu@ip-172-31-43-105:~$ docker container ls
CONTAINER ID   IMAGE                              COMMAND                  CREATED             STATUS              PORTS                                     NAMES
4a76e77f78cc   nginx                              "/docker-entrypoint.…"   17 minutes ago      Up 17 minutes       0.0.0.0:8041->80/tcp, [::]:8041->80/tcp   ats-login
f66dd098a4c5   nginx                              "/docker-entrypoint.…"   About an hour ago   Up About an hour    0.0.0.0:8051->80/tcp, [::]:8051->80/tcp   tes-5
b8d4e85b6448   nginx                              "/docker-entrypoint.…"   About an hour ago   Up About an hour    0.0.0.0:8011->80/tcp, [::]:8011->80/tcp   test-3
4ae5e6ca13ab   nginx                              "/docker-entrypoint.…"   About an hour ago   Up About an hour    0.0.0.0:8031->80/tcp, [::]:8031->80/tcp   tes-2
7cf8bc0ffd71   nginx                              "/docker-entrypoint.…"   2 hours ago         Up 2 hours          0.0.0.0:8044->80/tcp, [::]:8044->80/tcp   test-1
11e87c12fbc7   kvenkat9889/login-application122   "/docker-entrypoint.…"   2 hours ago         Up 2 hours          0.0.0.0:8021->80/tcp, [::]:8021->80/tcp   application-logi
n-1
ubuntu@ip-172-31-43-105:~$ docker container rm -f ats-login
ats-login
ubuntu@ip-172-31-43-105:~$ docker container ls
CONTAINER ID   IMAGE                              COMMAND                  CREATED             STATUS              PORTS                                     NAMES
f66dd098a4c5   nginx                              "/docker-entrypoint.…"   About an hour ago   Up About an hour    0.0.0.0:8051->80/tcp, [::]:8051->80/tcp   tes-5
b8d4e85b6448   nginx                              "/docker-entrypoint.…"   About an hour ago   Up About an hour    0.0.0.0:8011->80/tcp, [::]:8011->80/tcp   test-3
```

Now delete the container which you have created ats-login container

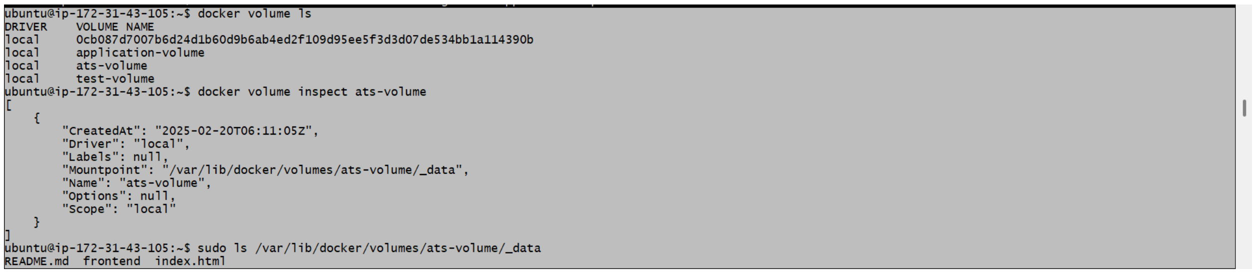


```
ubuntu@ip-172-31-43-105:~$ docker container ls
CONTAINER ID   IMAGE                              COMMAND                  CREATED             STATUS              PORTS                                     NAMES
4a76e77f78cc   nginx                              "/docker-entrypoint.…"   17 minutes ago      Up 17 minutes       0.0.0.0:8041->80/tcp, [::]:8041->80/tcp   ats-login
f66dd098a4c5   nginx                              "/docker-entrypoint.…"   About an hour ago   Up About an hour    0.0.0.0:8051->80/tcp, [::]:8051->80/tcp   tes-5
b8d4e85b6448   nginx                              "/docker-entrypoint.…"   About an hour ago   Up About an hour    0.0.0.0:8011->80/tcp, [::]:8011->80/tcp   test-3
4ae5e6ca13ab   nginx                              "/docker-entrypoint.…"   About an hour ago   Up About an hour    0.0.0.0:8031->80/tcp, [::]:8031->80/tcp   tes-2
7cf8bc0ffd71   nginx                              "/docker-entrypoint.…"   2 hours ago         Up 2 hours          0.0.0.0:8044->80/tcp, [::]:8044->80/tcp   test-1
11e87c12fbc7   kvenkat9889/login-application122   "/docker-entrypoint.…"   2 hours ago         Up 2 hours          0.0.0.0:8021->80/tcp, [::]:8021->80/tcp   application-logi
n-1
ubuntu@ip-172-31-43-105:~$ docker container rm -f ats-login
ats-login
ubuntu@ip-172-31-43-105:~$ docker container ls
CONTAINER ID   IMAGE                              COMMAND                  CREATED             STATUS              PORTS                                     NAMES
f66dd098a4c5   nginx                              "/docker-entrypoint.…"   About an hour ago   Up About an hour    0.0.0.0:8051->80/tcp, [::]:8051->80/tcp   tes-5
b8d4e85b6448   nginx                              "/docker-entrypoint.…"   About an hour ago   Up About an hour    0.0.0.0:8011->80/tcp, [::]:8011->80/tcp   test-3
4ae5e6ca13ab   nginx                              "/docker-entrypoint.…"   About an hour ago   Up About an hour    0.0.0.0:8031->80/tcp, [::]:8031->80/tcp   tes-2
7cf8bc0ffd71   nginx                              "/docker-entrypoint.…"   2 hours ago         Up 2 hours          0.0.0.0:8044->80/tcp, [::]:8044->80/tcp   test-1
11e87c12fbc7   kvenkat9889/login-application122   "/docker-entrypoint.…"   2 hours ago         Up 2 hours          0.0.0.0:8021->80/tcp, [::]:8021->80/tcp   application-logi
n-1
```

After deleting the containers also still you created volume are available to check use below command
Command: docker volumes ls

Now inspect the volume to take volume path
Command: docker volume inspect ats-volume

Now list the path of the volume and check Application code files are available or not.
Command: sudo ls /var/lib/docker/volumes/ats-volume/_data
Output: README.md frontend index.html

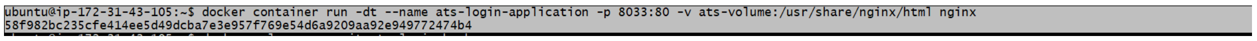


```
ubuntu@ip-172-31-43-105:~$ docker volume ls
DRIVER    VOLUME NAME
local     0cb087d7007b6d24d1b60d9b6ab4ed2f109d95ee5f3d3d07de534bb1a114390b
local     application-volume
local     ats-volume
local     test-volume
ubuntu@ip-172-31-43-105:~$ docker volume inspect ats-volume
[
    {
        "CreatedAt": "2025-02-20T06:11:05Z",
        "Driver": "local",
        "Labels": null,
        "Mountpoint": "/var/lib/docker/volumes/ats-volume/_data",
        "Name": "ats-volume",
        "Options": null,
        "Scope": "local"
    }
]
ubuntu@ip-172-31-43-105:~$ sudo ls /var/lib/docker/volumes/ats-volume/_data
README.md  frontend  index.html
```

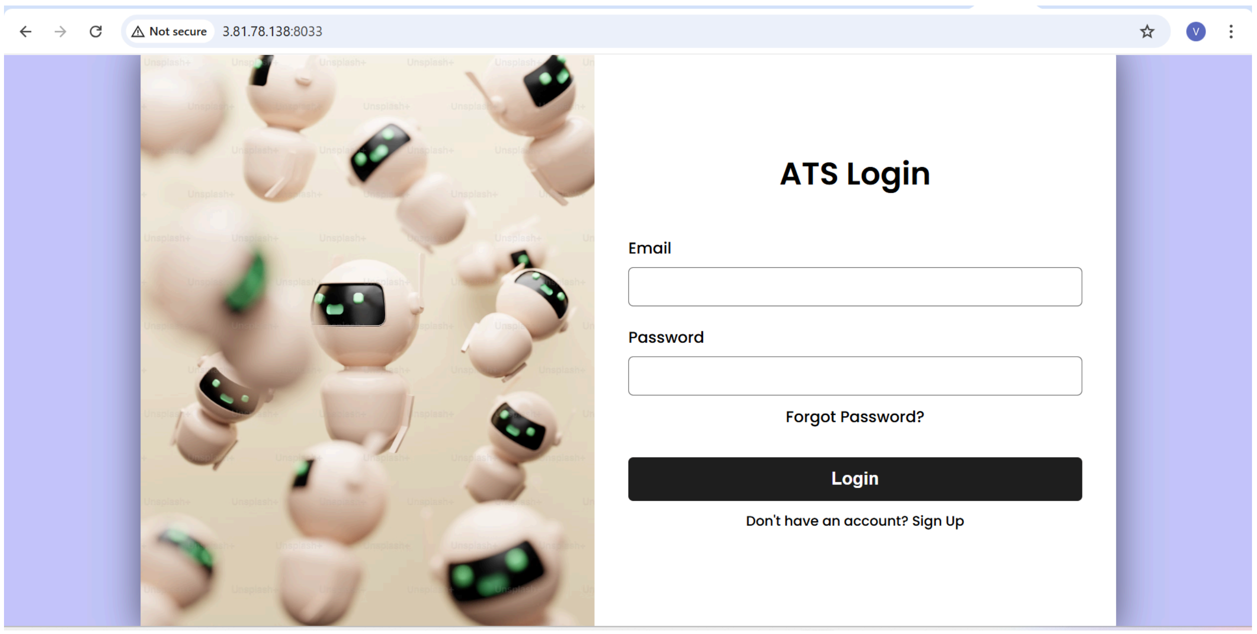
Note : By using volume ats-volume you can create multiple containers to load the same application.

For Example : create a one more container with same volume (ats-volume) with port no and go to browser and check the application is loading or not

Command: docker container run -dt --name ats-login-application -p 8033:80 -v ats-volume:/usr/share/nginx/html nginx

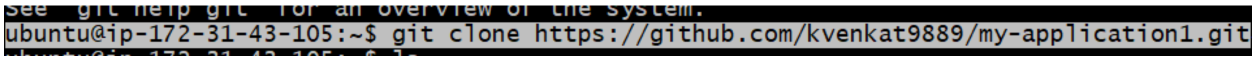


```
ubuntu@ip-172-31-43-105:~$ docker container run -dt --name ats-login-application -p 8033:80 -v ats-volume:/usr/share/nginx/html nginx
58f982bc235cfe414ee5d49dcba7e3e957f769e54d6a9209aa92e949772474b4
```

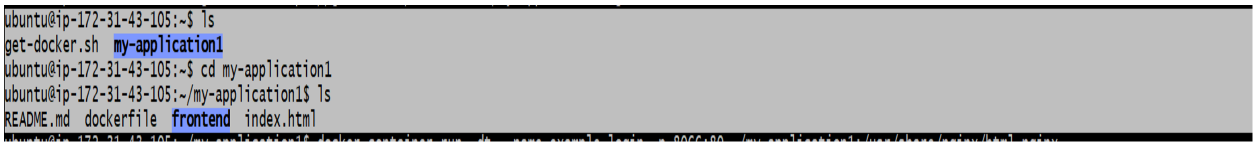


HOST VOLUMES

On Host clone the Application URL From github
Command: git clone https://github.com/kvenkat9889/my-application1.git

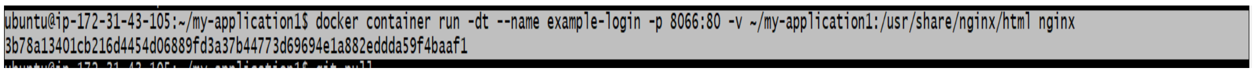


```
ubuntu@ip-172-31-43-105:~$ git clone https://github.com/kvenkat9889/my-application1.git
```



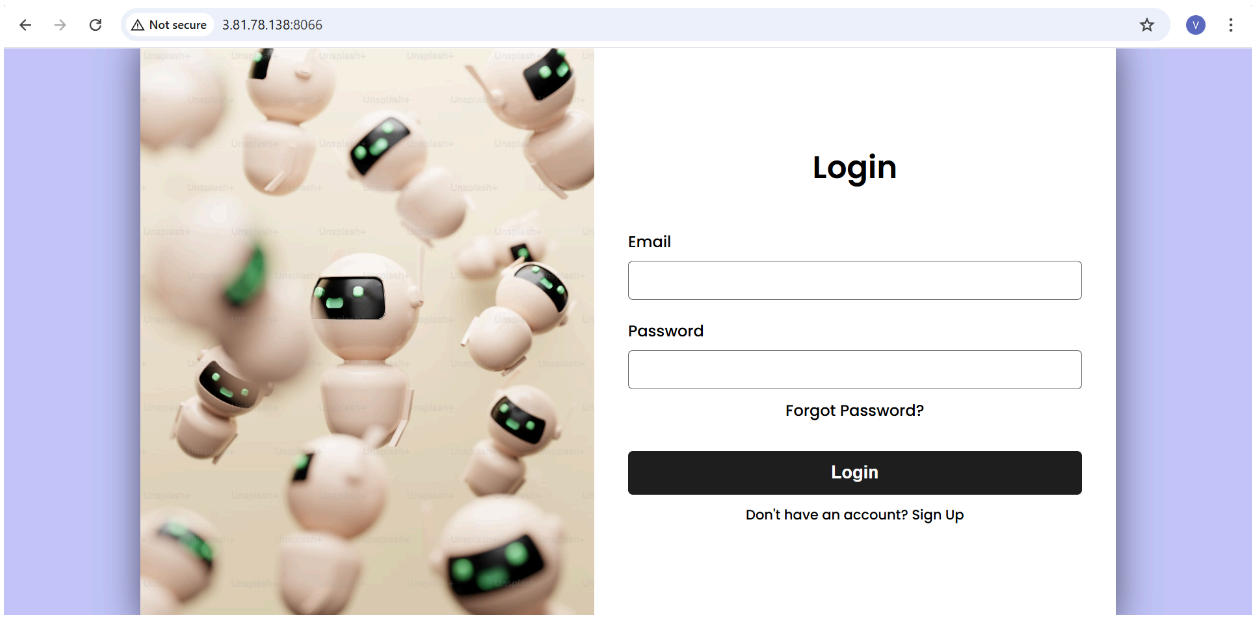
```
ubuntu@ip-172-31-43-105:~$ ls
get-docker.sh  my-application1
ubuntu@ip-172-31-43-105:~$ cd my-application1
ubuntu@ip-172-31-43-105:~/my-application1$ ls
README.md  dockerfile  frontend  index.html
```

Now create a container on host level with nginx path
Command: docker container run -dt --name example-login -p 8066:80 -v ~/my-application1:/usr/share/nginx/html nginx



```
ubuntu@ip-172-31-43-105:~/my-application1$ docker container run -dt --name example-login -p 8066:80 -v ~/my-application1:/usr/share/nginx/html nginx
3b78a13401cb216d4454d06889fd3a37b44773d69694e1a882eddda59f4baaf1
```

On port 8066 page is available


Not secure 3.81.78.138:8066

Login

Email

Password

Forgot Password?

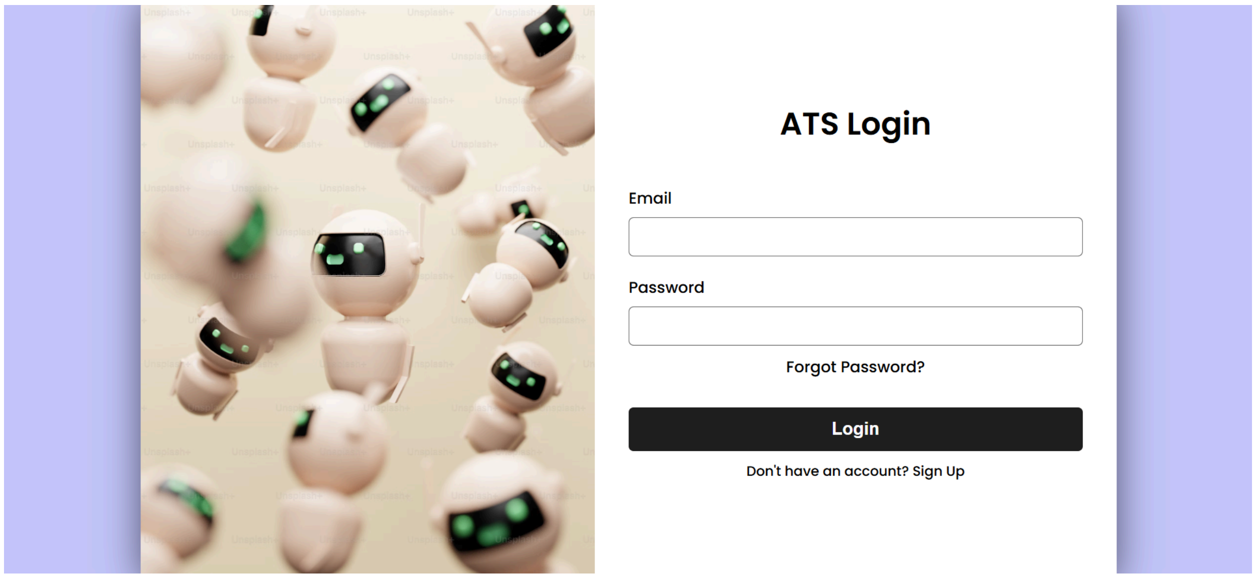
Login

Don't have an account? Sign Up

Now developer made changes in code and that code should be reflected in our sever with a single command:

Command: git pull

Changes made page **login** to **ATS Login**


ATS Login

Email

Password

Forgot Password?

Login

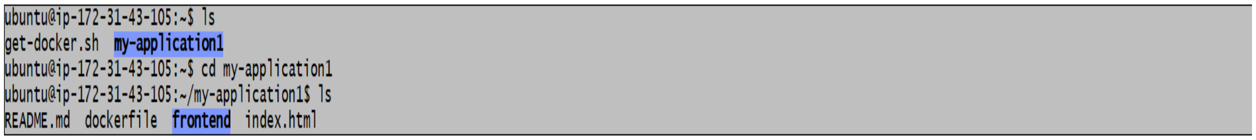
Don't have an account? Sign Up

This is how host volumes will work

CUSTOMIZED IMAGE

Command: ls
Output: get-docker.sh my-application1

Command: cd my-application1
Command: ls
Output: README.md frontend index.html



```
ubuntu@ip-172-31-43-105:~$ ls
get-docker.sh  my-application1
ubuntu@ip-172-31-43-105:~$ cd my-application1
ubuntu@ip-172-31-43-105:~/my-application1$ ls
README.md  dockerfile  frontend  index.html
```

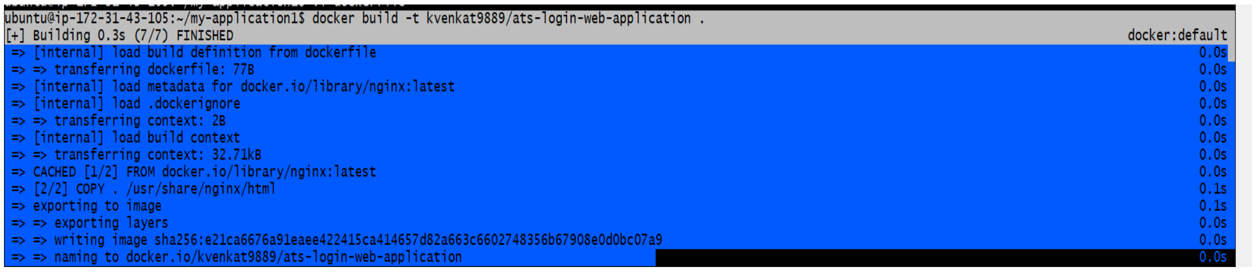
Now create docker file with vi mode usinmg below command:
Command: vi dockerfile

Command: ls
Output: README.md **dockerfile** frontend index.html

After creating the dockerfile run this below command to build your own image
Command: docker build -t kvenkat9889/ats-login-web-application **.**

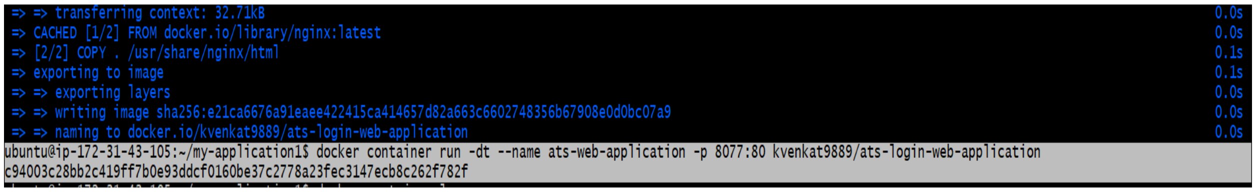
Note: create a docker hub account
- Take username of docker hub (example- kvenkat9889)

Now create a own image name you want, with giving any name
- (example - ats-login-web-application)



```
ubuntu@ip-172-31-43-105:~/my-application1$ docker build -t kvenkat9889/ats-login-web-application .
[+] Building 0.3s (7/7) FINISHED                                                          docker:default
 => [internal] load build definition from dockerfile                                                0.0s
 => => transferring dockerfile: 77B                                                                 0.0s
 => [internal] load metadata for docker.io/library/nginx:latest                                     0.0s
 => [internal] load .dockerignore                                                                   0.0s
 => => transferring context: 2B                                                                     0.0s
 => [internal] load build context                                                                   0.0s
 => => transferring context: 32.71kB                                                                0.0s
 => CACHED [1/2] FROM docker.io/library/nginx:latest                                                0.0s
 => [2/2] COPY . /usr/share/nginx/html                                                              0.1s
 => exporting to image                                                                              0.1s
 => => exporting layers                                                                             0.0s
 => => writing image sha256:e21ca6676a91eaee422415ca414657d82a663c6602748356b67908e0d0bc07a9        0.0s
 => => naming to docker.io/kvenkat9889/ats-login-web-application                                    0.0s
```

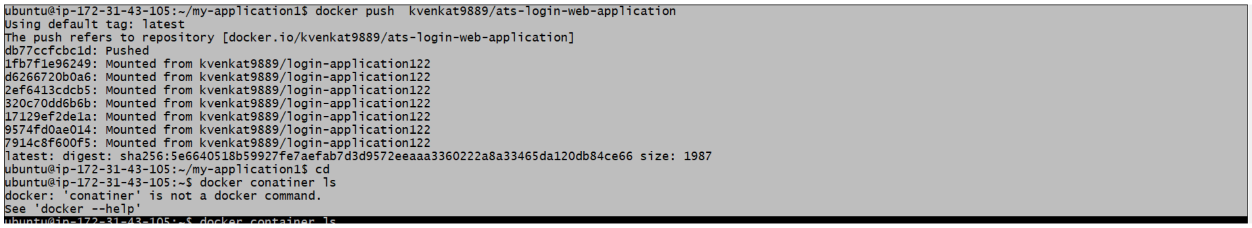
Now create a new container with port no
Command: docker container run -dt --name ats-web-application -p 8077:80 kvenkat9889/ats-login-web-application



```
 => => transferring context: 32.71kB                                                                0.0s
 => CACHED [1/2] FROM docker.io/library/nginx:latest                                                0.0s
 => [2/2] COPY . /usr/share/nginx/html                                                              0.1s
 => exporting to image                                                                              0.1s
 => => exporting layers                                                                             0.0s
 => => writing image sha256:e21ca6676a91eaee422415ca414657d82a663c6602748356b67908e0d0bc07a9        0.0s
 => => naming to docker.io/kvenkat9889/ats-login-web-application                                    0.0s
ubuntu@ip-172-31-43-105:~/my-application1$ docker container run -dt --name ats-web-application -p 8077:80 kvenkat9889/ats-login-web-application
c94003c28bb2c419ff7b0e93ddcf0160be37c2778a23fec3147ecb8c262f782f
```

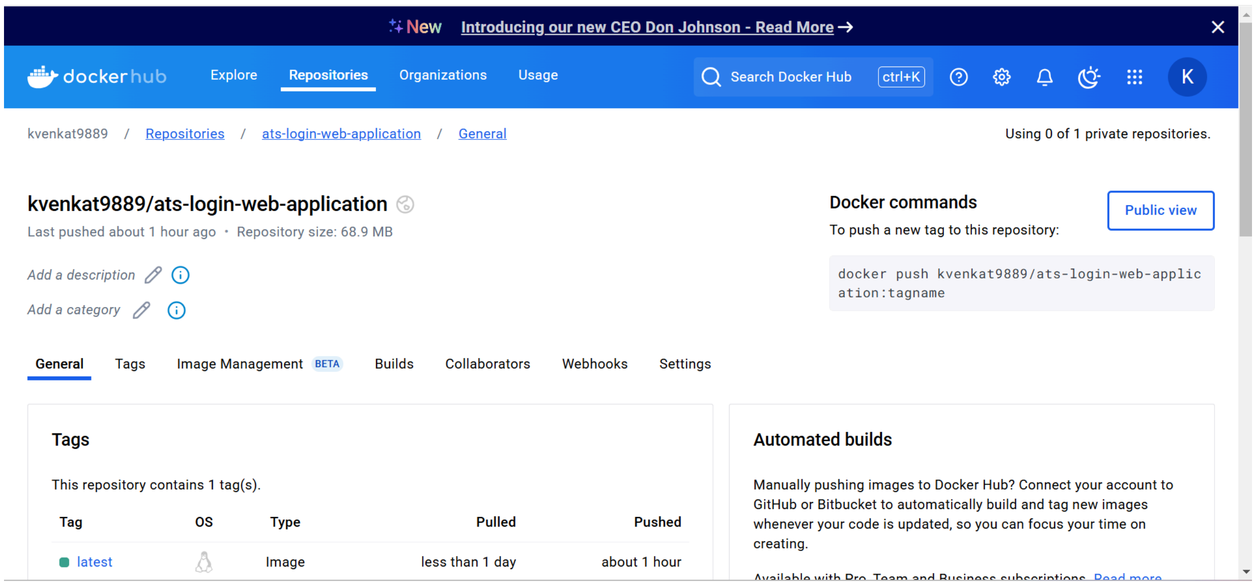
Browse and Check the URl:ipaddress:8077

Now push image to docker hub
Command: docker push kvenkat9889/ats-login-web-application



```
ubuntu@ip-172-31-43-105:~/my-application1$ docker push  kvenkat9889/ats-login-web-application
Using default tag: latest
The push refers to repository [docker.io/kvenkat9889/ats-login-web-application]
db77ccfcbc1d: Pushed
1fb7f1e96249: Mounted from kvenkat9889/login-application122
d6266720b0a6: Mounted from kvenkat9889/login-application122
2ef6413cdcb5: Mounted from kvenkat9889/login-application122
320c70dd6b6b: Mounted from kvenkat9889/login-application122
17129ef2de1a: Mounted from kvenkat9889/login-application122
9574fd0ae014: Mounted from kvenkat9889/login-application122
7914c8f600f5: Mounted from kvenkat9889/login-application122
latest: digest: sha256:5e6640518b59927fe7aefab7d3d9572eeaaa3360222a8a33465da120db84ce66 size: 1987
ubuntu@ip-172-31-43-105:~/my-application1$ cd
ubuntu@ip-172-31-43-105:~$ docker conatiner ls
docker: 'conatiner' is not a docker command.
See 'docker --help'
ubuntu@ip-172-31-43-105:~$ docker container ls
```

Go back you Docker hub now check the image is available or not



New Introducing our new CEO Don Johnson - Read More →

kvenkat9889 / Repositories / ats-login-web-application / General

Using 0 of 1 private repositories.

kvenkat9889/ats-login-web-application

Last pushed about 1 hour ago • Repository size: 68.9 MB

Add a description

Add a category

Docker commands

To push a new tag to this repository:

```
docker push kvenkat9889/ats-login-web-applic
ation:tagname
```

Tags

This repository contains 1 tag(s).

Tag	OS	Type	Pulled	Pushed
latest		Image	less than 1 day	about 1 hour

Automated builds

Manually pushing images to Docker Hub? Connect your account to GitHub or Bitbucket to automatically build and tag new images whenever your code is updated, so you can focus your time on creating.

Available with Pro, Team and Business subscriptions. Read more