

DevOps Essential

Aman Shrestha

Associate Software Engineer

1. Launch an EC2 instance and login via ssh furthermore, install httpd (apache).

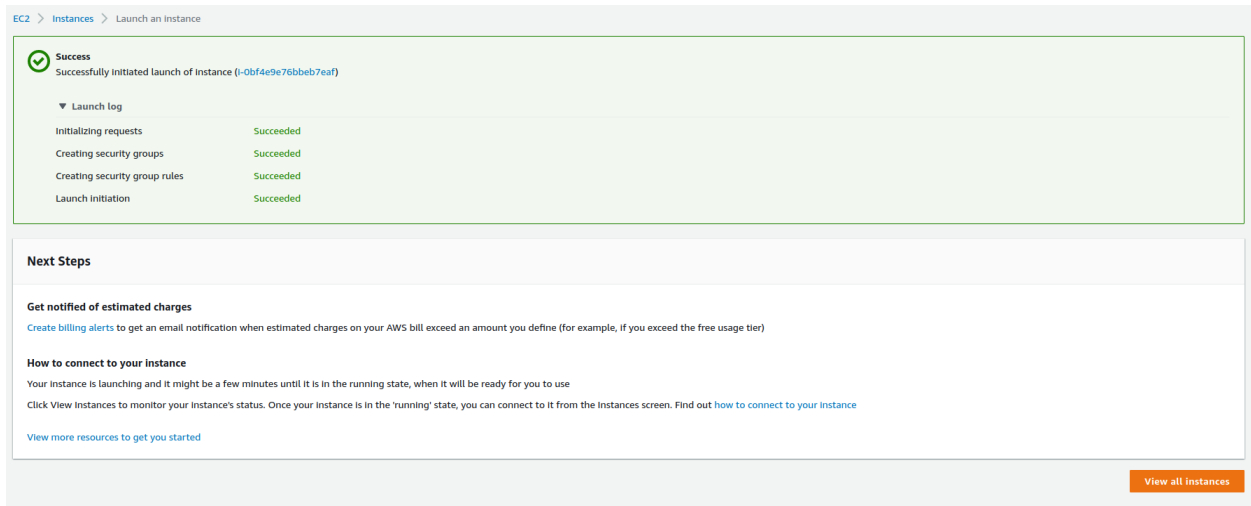


Fig: Successful EC2 instance launch

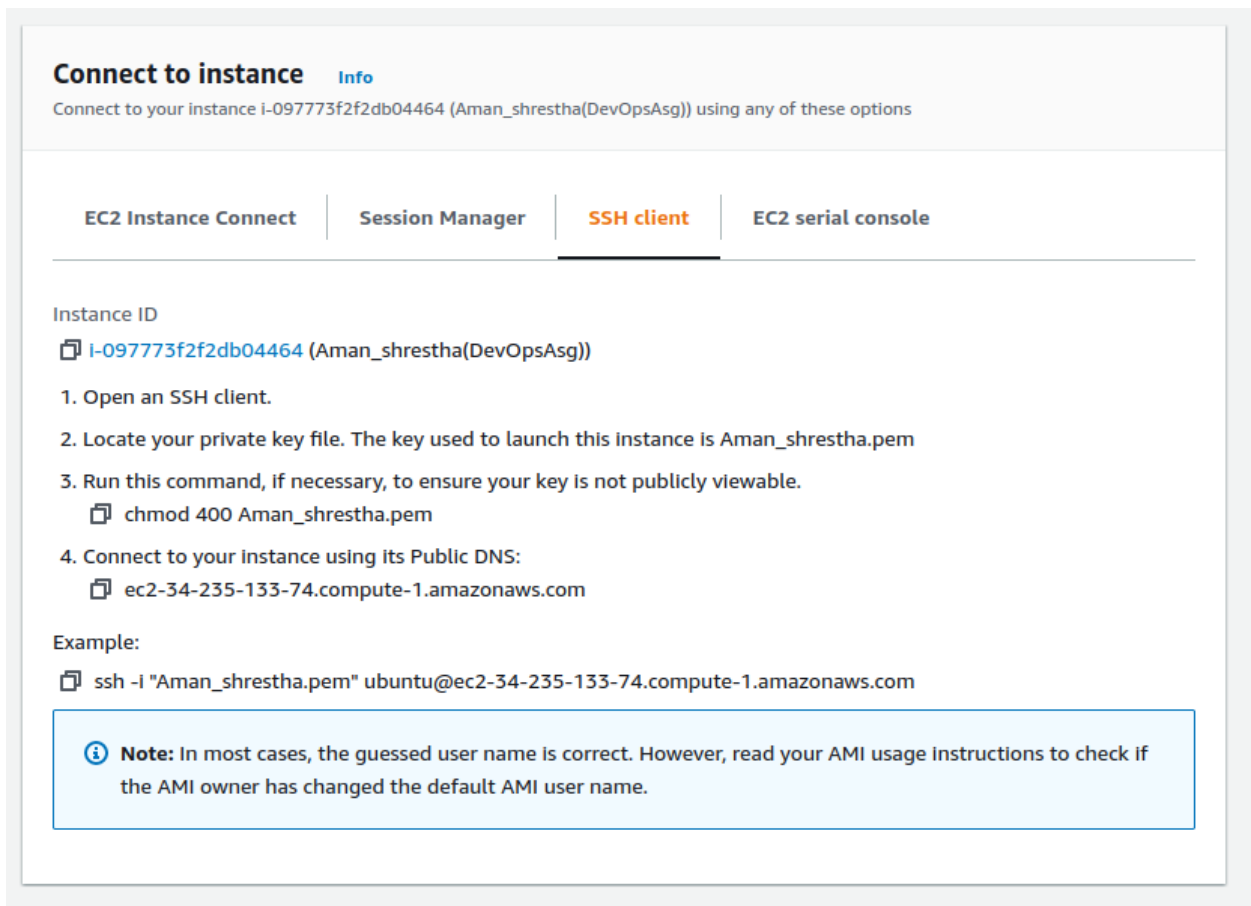


Fig: Connection instruction to an EC2 instance

```
ubuntu@ip-172-31-1-74: ~
aman@aman-Inspiron-15-3567: ~/Downloads$ chmod 400 Aman_shrestha.pem
aman@aman-Inspiron-15-3567:~/Downloads$ ll | grep Aman_shrestha.pem
-r----- 1 aman aman 1678 आसुत 10 12:22 Aman_shrestha.pem
aman@aman-Inspiron-15-3567:~/Downloads$ ssh -i "Aman_shrestha.pem" ubuntu@ec2-34-235-133-74.compute-1.amazonaws.com
The authenticity of host 'ec2-34-235-133-74.compute-1.amazonaws.com (34.235.133.74)' can't be established.
ECDSA key fingerprint is SHA256:nDVLVAMCFeGaQ456frGBbbMY7Wr+JOPJwgKpU240rVQ.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-34-235-133-74.compute-1.amazonaws.com,34.235.133.74' (ECDSA) to the list of known hosts.
Welcome to Ubuntu 22.04 LTS (GNU/Linux 5.15.0-1011-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

System information as of Wed Aug 10 06:51:14 UTC 2022

System load:  0.16015625      Processes:            99
Usage of /:   19.0% of 7.58GB  Users logged in:     0
Memory usage: 20%            IPv4 address for eth0: 172.31.1.74
Swap usage:   0%

0 updates can be applied immediately.

The list of available updates is more than a week old.
To check for new updates run: sudo apt update

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-172-31-1-74:~$
```

Fig: Logging to EC2 instance from SSH

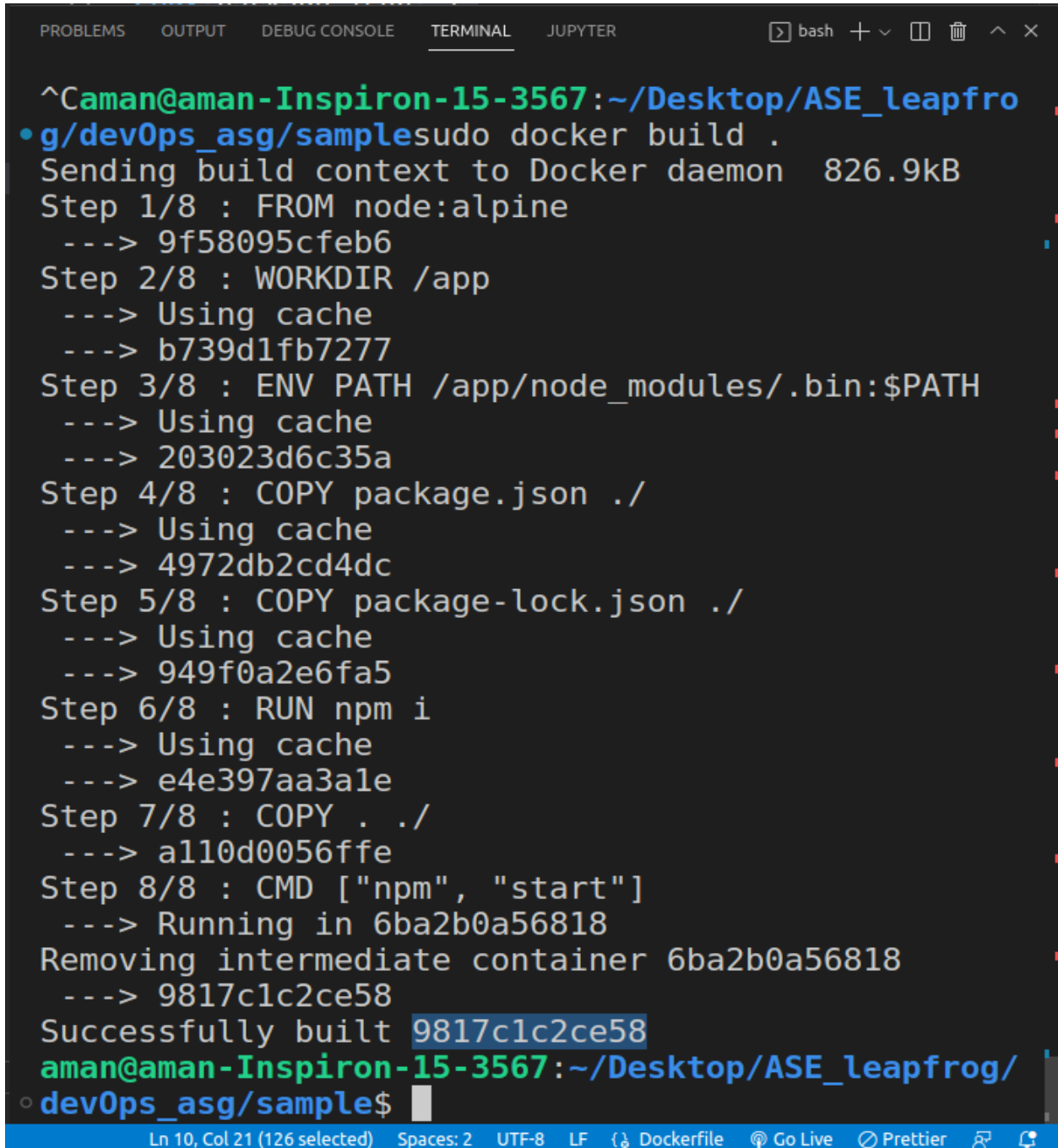
```
ubuntu@ip-172-31-1-74: ~  
ubuntu@ip-172-31-1-74:~$ sudo apt install apache2  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
The following additional packages will be installed:  
  apache2-bin apache2-data apache2-utils bzip2 libapr1 libaprutil1  
  libaprutil1-dbd-sqlite3 libaprutil1-ldap liblua5.3-0 mailcap mime-support  
  ssl-cert  
Suggested packages:  
  apache2-doc apache2-suexec-pristine | apache2-suexec-custom www-browser  
  bzip2-doc  
The following NEW packages will be installed:  
  apache2 apache2-bin apache2-data apache2-utils bzip2 libapr1 libaprutil1  
  libaprutil1-dbd-sqlite3 libaprutil1-ldap liblua5.3-0 mailcap mime-support  
  ssl-cert  
0 upgraded, 13 newly installed, 0 to remove and 0 not upgraded.  
Need to get 2135 kB of archives.  
After this operation, 8486 kB of additional disk space will be used.  
Do you want to continue? [Y/n] y  
Get:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 libapr1 am  
d64 1.7.0-8build1 [107 kB]
```

Fig: Installation of httpd (Apache)

```
ubuntu@ip-172-31-1-74: ~  
ubuntu@ip-172-31-1-74:/var/www/html$ sudo nano index.html  
ubuntu@ip-172-31-1-74:/var/www/html$ cd  
ubuntu@ip-172-31-1-74:~$ curl localhost  
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.or  
g/TR/xhtml1/DTD/xhtml1-transitional.dtd">  
<html xmlns="http://www.w3.org/1999/xhtml">  
  <head>  
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />  
    <title>Aman Shrestha's Website </title>  
  </head>  
  <body>  
    <div class="main_page">  
      <div class="page_header floating_element">  
        <h2>This section has been modified by aman shrestha </h2>  
  
      </div>  
      <h4>Hello World from the perspective of Aman Shrestha. </h4>  
    </body>  
</html>  
ubuntu@ip-172-31-1-74:~$
```

Fig: Changing default APACHE server response

2. Create a Dockerfile for node js and react and push the created docker images to the docker hub repository.

A screenshot of a terminal window with a dark background. The terminal shows the execution of the 'docker build .' command. It details the steps of the build process, including pulling the 'node:alpine' base image, setting the working directory to '/app', and installing dependencies with 'npm install'. The final output is 'Successfully built 9817c1c2ce58'. The terminal window has tabs for 'PROBLEMS', 'OUTPUT', 'DEBUG CONSOLE', 'TERMINAL', and 'JUPYTER'. The status bar at the bottom shows 'Ln 10, Col 21 (126 selected)', 'Spaces: 2', 'UTF-8', 'LF', and icons for 'Dockerfile', 'Go Live', 'Prettier', and other editor features.

```
^Caman@aman-Inspiron-15-3567:~/Desktop/ASE_leapfro
• g/devOps_asg/sample$ sudo docker build .
Sending build context to Docker daemon 826.9kB
Step 1/8 : FROM node:alpine
---> 9f58095cfeb6
Step 2/8 : WORKDIR /app
---> Using cache
---> b739d1fb7277
Step 3/8 : ENV PATH /app/node_modules/.bin:$PATH
---> Using cache
---> 203023d6c35a
Step 4/8 : COPY package.json ./
---> Using cache
---> 4972db2cd4dc
Step 5/8 : COPY package-lock.json ./
---> Using cache
---> 949f0a2e6fa5
Step 6/8 : RUN npm i
---> Using cache
---> e4e397aa3a1e
Step 7/8 : COPY . ./
---> a110d0056ffe
Step 8/8 : CMD ["npm", "start"]
---> Running in 6ba2b0a56818
Removing intermediate container 6ba2b0a56818
---> 9817c1c2ce58
Successfully built 9817c1c2ce58
aman@aman-Inspiron-15-3567:~/Desktop/ASE_leapfrog/
• devOps_asg/sample$
```

Fig: Docker image build

The above command searches for Dockerfile and builds an image for it returning image ID.

```
Successfully built 9817c1c2ce58
aman@aman-Inspiron-15-3567:~/Desktop/ASE_leapfrog/
devOps_asg/sample$ sudo docker run -p 3000:3000 04257d9cdd82

> sample@0.1.0 start
> react-scripts start

(node:25) [DEP_WEBPACK_DEV_SERVER_ON_AFTER_SETUP_MIDDLEWARE]
  DeprecationWarning: 'onAfterSetupMiddleware' option is depr
  ecated. Please use the 'setupMiddlewares' option.
  (Use `node --trace-deprecation ...` to show where the warnin
  g was created)
(node:25) [DEP_WEBPACK_DEV_SERVER_ON_BEFORE_SETUP_MIDDLEWARE]
  DeprecationWarning: 'onBeforeSetupMiddleware' option is de
  precated. Please use the 'setupMiddlewares' option.
Starting the development server...

Compiled successfully!

You can now view sample in the browser.

   Local:            http://localhost:3000
  On Your Network:  http://172.17.0.2:3000

Note that the development build is not optimized.
To create a production build, use npm run build.

webpack compiled successfully
Compiling...
Compiled successfully!
webpack compiled successfully
```

Fig: Run container using docker image at container port 3000

```
aman@aman-Inspiron-15-3567:~/Desktop/ASE_Leapfrog/devOps_as
o /sample$ sudo docker run -p 3000:3000 9817c1c2ce58

> sample@0.1.0 start

Running from container

Local:          http://localhost:3000
On Your Network: http://172.17.0.2:3000

Note that the development build is not optimized.
To create a production build, use npm run build.

webpack compiled successfully
Compiling...
Compiled successfully!
webpack compiled successfully
```

Fig: Running react app from container

The above snaps show we were successfully able to create an image and run it from the container itself in our local machine. Time to take it to the next level (Push it to docker hub). Let's go....

```
aman@aman-Inspiron-15-3567:~/Desktop/ASE_leapfrog/dev0ps_asg
• /sample$ sudo docker login
Authenticating with existing credentials...
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
aman@aman-Inspiron-15-3567:~/Desktop/ASE_leapfrog/dev0ps_asg
```


Fig: DockerHub login

```
aman@aman-Inspiron-15-3567:~/Desktop/ASE_leapfrog/dev0ps_asg
• /sample$ sudo docker build -t sample_app .
Sending build context to Docker daemon 826.9kB
Step 1/8 : FROM node:alpine
---> 9f58095cfeb6
Step 2/8 : WORKDIR /app
---> Using cache
---> b739d1fb7277
Step 3/8 : ENV PATH /app/node_modules/.bin:$PATH
---> Using cache
---> 203023d6c35a
Step 4/8 : COPY package.json ./
---> Using cache
---> 4972db2cd4dc
Step 5/8 : COPY package-lock.json ./
---> Using cache
---> 949f0a2e6fa5
Step 6/8 : RUN npm i
---> Using cache
---> e4e397aa3a1e
Step 7/8 : COPY . ./
---> Using cache
---> a110d0056ffe
Step 8/8 : CMD ["npm", "start"]
---> Using cache
---> 9817c1c2ce58
Successfully built 9817c1c2ce58
Successfully tagged sample_app:latest
aman@aman-Inspiron-15-3567:~/Desktop/ASE_leapfrog/dev0ps_asg
• /sample$
```


Fig: Tagging docker image as sample_app



```
netto world latest Feb 25, 2020 10 months ago 1
3.3kB
aman@aman-Inspiron-15-3567:~/Desktop/ASE_leapfrog/dev0ps_asg
• /sample$ sudo docker tag sample_app 9861008541/sample-image
aman@aman-Inspiron-15-3567:~/Desktop/ASE_leapfrog/dev0ps_asg
• /sample$ sudo docker push 9861008541/sample-image
Using default tag: latest
The push refers to repository [docker.io/9861008541/sample-i
mage]
a41fc4b32a5d: Pushed
0a66888cd477: Pushed
bba68474ff27: Pushed
9cef0df69bcf: Pushed
69e2ed38b4d0: Pushed
d01c433a71f9: Mounted from library/node
92ff8a47a573: Mounted from library/node
b158cee4c019: Mounted from library/node
4fc242d58285: Mounted from library/node
latest: digest: sha256:ddc0ce1ce3392cc1510cfdbfacbca0caf8927
4dc441a92d08379dd3bf1080487 size: 2204
aman@aman-Inspiron-15-3567:~/Desktop/ASE_leapfrog/dev0ps_asg
• /sample$
```

Fig: Pushing Image into DockerHub


 **9861008541 / sample-image**

Description



This repository does not have a description 

 Last pushed: 9 minutes ago

Tags and Scans

 VULNERABILITY SCANNING - DISABLED [Enable](#)

This repository contains 1 tag(s).

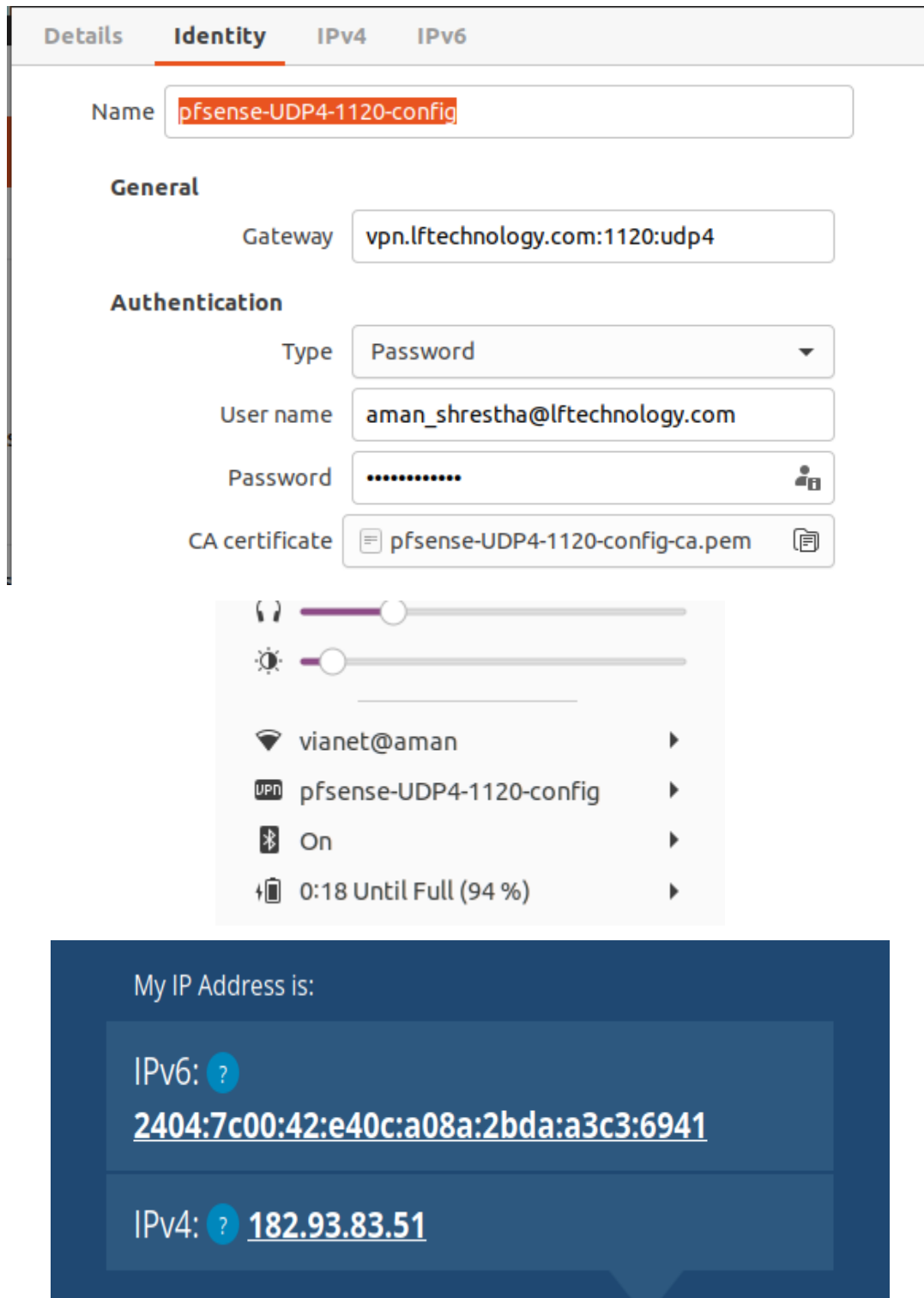
TAG	OS	PULLED	PUSHED
 latest		---	9 minutes ago

[See all](#)[Go to Advanced Image Management](#)

Fig: Output in docker hub

We can see the newly created repository in the docker hub was pushed a docker image of our application.

3. Setup office VPN and access self-learning platforms.



The image shows the pfSense configuration interface for a VPN. The 'Identity' tab is selected, showing the following configuration:

- Name:** pfsense-UDP4-1120-config
- General:**
 - Gateway:** vpn.lftechnology.com:1120:udp4
- Authentication:**
 - Type:** Password
 - User name:** aman_shrestha@lftechnology.com
 - Password:** [Redacted]
 - CA certificate:** pfsense-UDP4-1120-config-ca.pem

Below the configuration fields, a system status overlay is visible, showing:

- Headphones icon: [Volume slider]
- Sun icon: [Brightness slider]
- Wi-Fi icon: vianet@aman
- VPN icon: pfsense-UDP4-1120-config
- Bluetooth icon: On
- Battery icon: 0:18 Until Full (94 %)

At the bottom, a dark blue box displays the user's IP address information:

My IP Address is:

- IPv6: ? 2404:7c00:42:e40c:a08a:2bda:a3c3:6941
- IPv4: ? 182.93.83.51

The IPV4 address of 182.93.83.51 confirms connectivity to the office VPN.

4. List down common HTTP status codes(only 10) with their meaning.

1. 100 Continue

Only a part of the request has been received by the server, but as long as it has not been rejected, the client should continue with the request.

2. 200 OK

The request succeeded. The result meaning of "success" depends on the HTTP method (GET, HEAD, PUT, POST)

3. 201 Created

The request succeeded, and a new resource was created as a result. This is typically the response sent after POST requests or some PUT requests.

4. 301 Moved Permanently

The URL of the requested resource has been changed permanently. The new URL is given in the response

5. 400 Bad Request

The server cannot or will not process the request due to something that is perceived to be a client error (e.g., malformed request syntax, invalid request message framing, or deceptive request routing).

6. 401 Unauthorized

The clients are not authenticated to access the resources.

7. 403 Forbidden

The client does not have access rights to the content; that is, it is unauthorized, so the server is refusing to give the requested resource. Unlike 401 Unauthorized, the client's identity is known to the server.

8. 404 Not Found

The server can not find the requested resource. In the browser, this means the URL is not recognized. In an API, this can also mean that the endpoint is valid but the resource itself does not exist.

9. 405 Method Not Allowed

The request method is known by the server but is not supported by the target resource. For example, an API may not allow calling DELETE to remove a resource.

10.500 Internal Server Error

The request was not completed. The server met an unexpected condition.

5. Find network id and Broadcast for 150.10.20.30

Address Class	RANGE	Default Subnet Mask
A	1.0.0.0 to 126.255.255.255	255.0.0.0
B	128.0.0.0 to 191.255.255.255	255.255.0.0
C	192.0.0.0 to 223.255.255.255	255.255.255.0
D	224.0.0.0 to 239.255.255.255	Reserved for Multicasting
E	240.0.0.0 to 254.255.255.255	Experimental

Note: Class A addresses 127.0.0.0 to 127.255.255.255 cannot be used and is reserved for loopback testing.

Here, the first octet 150 falls between the range 128 and 191, this makes the above IP a class B IP address. The default Subnet mask for class B as shown in the table is 255.255.0.0

The binary notation for the Given Ip address

1001 0110. 0000 1010. 0001 0100. 0001 1110

Similarly, the binary notation for Subnet mask 255.255.0.0 is

1111 1111. 1111 1111. 0000 0000. 0000 0000

As we can see from the above bits representation, the first 16 bits of the subnet mask which are all 1's and corresponding bits in the IPV4 address belong to the network part. Likewise, the last 16 bits of the subnet mask which are all 0's and corresponding bits in the IPV4 address belong to the host part.

The above IP address can be denoted as 150.10.20.30/16 where /16 represents the number of network bits.

The Host 16 bits represents the number of devices within that network plus broadcast id and network id.

So, a total of $2^{16} - 2$ hosts are allowed within that particular network.

If we replace all the host bits within that IP address with 1's, we will get Broadcast id

The binary notation for Broadcast id

1001 0110. 0000 1010. 1111 1111. 1111 1111

Decimal Equivalent 150.10.255.255

Similarly, if we replace all the host bits within that IP address with 0's, we will get the network id

The binary notation for Network id

1001 0110. 0000 1010. 0000 0000. 0000 0000

Decimal Equivalent : 150.10.0.0

Conclusion

Network Id: 150.10.0.0/16

Broadcast Id: 150.10.255.255