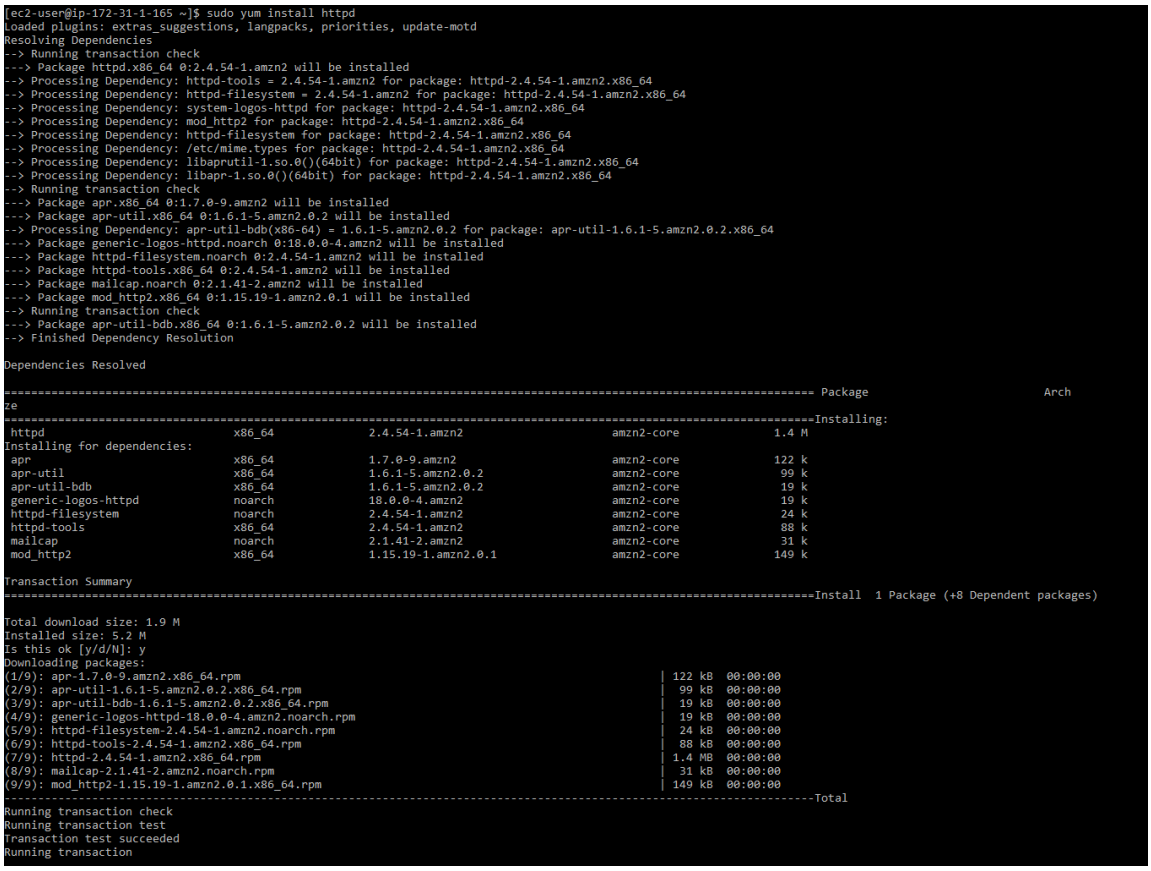
1. Launch an EC2 instance and login via ssh furthermore, install httpd(apache).



```
[ec2-user@ip-172-31-1-165 ~]$ sudo yum install httpd
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
Resolving Dependencies
--> Running transaction check
---> Package httpd.x86_64 0:2.4.54-1.amzn2 will be installed
--> Processing Dependency: httpd-tools = 2.4.54-1.amzn2 for package: httpd-2.4.54-1.amzn2.x86_64
--> Processing Dependency: httpd-filesystem = 2.4.54-1.amzn2 for package: httpd-2.4.54-1.amzn2.x86_64
--> Processing Dependency: system-logos-httpd for package: httpd-2.4.54-1.amzn2.x86_64
--> Processing Dependency: mod_http2 for package: httpd-2.4.54-1.amzn2.x86_64
--> Processing Dependency: httpd-filesystem for package: httpd-2.4.54-1.amzn2.x86_64
--> Processing Dependency: /etc/mime.types for package: httpd-2.4.54-1.amzn2.x86_64
--> Processing Dependency: libaprutil-1.so.0()(64bit) for package: httpd-2.4.54-1.amzn2.x86_64
--> Processing Dependency: libapr-1.so.0()(64bit) for package: httpd-2.4.54-1.amzn2.x86_64
--> Running transaction check
---> Package apr.x86_64 0:1.7.0-9.amzn2 will be installed
---> Package apr-util.x86_64 0:1.6.1-5.amzn2.0.2 will be installed
--> Processing Dependency: apr-util-bdb(x86-64) = 1.6.1-5.amzn2.0.2 for package: apr-util-1.6.1-5.amzn2.0.2.x86_64
---> Package generic-logos-httpd.noarch 0:18.0.0-4.amzn2 will be installed
---> Package httpd-filesystem.noarch 0:2.4.54-1.amzn2 will be installed
---> Package httpd-tools.x86_64 0:2.4.54-1.amzn2 will be installed
---> Package mailcap.noarch 0:2.1.41-2.amzn2 will be installed
---> Package mod_http2.x86_64 0:1.15.19-1.amzn2.0.1 will be installed
--> Running transaction check
---> Package apr-util-bdb.x86_64 0:1.6.1-5.amzn2.0.2 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

================================================================================ Package                        Arch
ze
================================================================================Installing:
 httpd                          x86_64              2.4.54-1.amzn2                amzn2-core              1.4 M
Installing for dependencies:
 apr                            x86_64              1.7.0-9.amzn2                 amzn2-core              122 k
 apr-util                       x86_64              1.6.1-5.amzn2.0.2             amzn2-core               99 k
 apr-util-bdb                   x86_64              1.6.1-5.amzn2.0.2             amzn2-core               19 k
 generic-logos-httpd            noarch              18.0.0-4.amzn2                amzn2-core               19 k
 httpd-filesystem               noarch              2.4.54-1.amzn2                amzn2-core               24 k
 httpd-tools                    x86_64              2.4.54-1.amzn2                amzn2-core               88 k
 mailcap                        noarch              2.1.41-2.amzn2                amzn2-core               31 k
 mod_http2                      x86_64              1.15.19-1.amzn2.0.1           amzn2-core              149 k

Transaction Summary
================================================================================Install  1 Package (+8 Dependent packages)

Total download size: 1.9 M
Installed size: 5.2 M
Is this ok [y/d/N]: y
Downloading packages:
(1/9): apr-1.7.0-9.amzn2.x86_64.rpm                                          | 122 kB  00:00:00
(2/9): apr-util-1.6.1-5.amzn2.0.2.x86_64.rpm                                 |  99 kB  00:00:00
(3/9): apr-util-bdb-1.6.1-5.amzn2.0.2.x86_64.rpm                             |  19 kB  00:00:00
(4/9): generic-logos-httpd-18.0.0-4.amzn2.noarch.rpm                         |  19 kB  00:00:00
(5/9): httpd-filesystem-2.4.54-1.amzn2.noarch.rpm                            |  24 kB  00:00:00
(6/9): httpd-tools-2.4.54-1.amzn2.x86_64.rpm                                 |  88 kB  00:00:00
(7/9): httpd-2.4.54-1.amzn2.x86_64.rpm                                       | 1.4 MB  00:00:00
(8/9): mailcap-2.1.41-2.amzn2.noarch.rpm                                     |  31 kB  00:00:00
(9/9): mod_http2-1.15.19-1.amzn2.0.1.x86_64.rpm                              | 149 kB  00:00:00
--------------------------------------------------------------------------------Total
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
```

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

Dockerfile

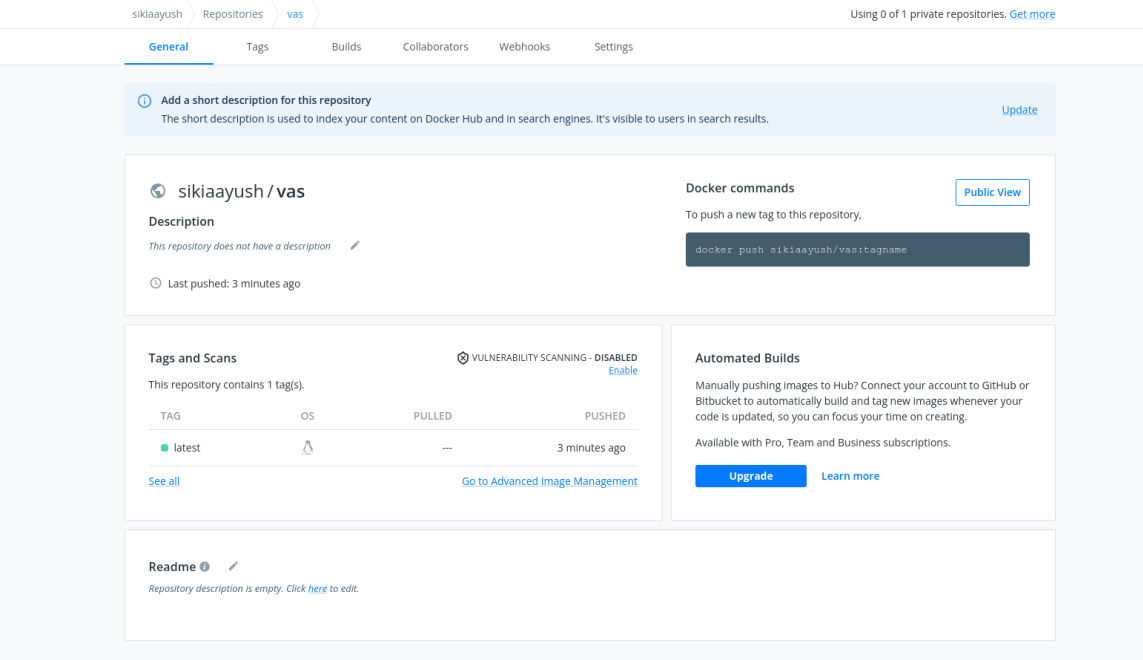
```
# pull official base image
FROM node:alpine

# set working directory
WORKDIR /app

# add `/app/node_modules/.bin` to $PATH
ENV PATH /app/node_modules/.bin:$PATH

# install app dependencies
```

```
COPY package.json ./
RUN npm install --silent
RUN npm install react-scripts@3.4.1 -g --silent

# add app
COPY . ./

# start app
CMD ["npm", "start"]
```

Commands

1. docker login
2. docker build -t vas:dev .
3. docker tag vas:dev sikiaayush/vas
4. docker push sikiaayush/vas

Image on Repo



3. List down common HTTP status codes (only 10) with means.

Status Code 200

Status 200 is the standard “OK” status code for a successful HTTP request. The response that is returned is dependent on the request. For example, for a GET request, the response will be included in the message body. For a PUT/POST request, the response will include the resource that contains the result of the action.

Status Code 201

Status Code 201 – This is the status code that confirms that the request was successful and, as a result, a new resource was created. Typically, this is the status code sent after a POST/PUT request.

Status Code 204

Status Code 204 – This status code confirms that the server has fulfilled the request but does not need to return information. Examples of this status code include delete requests or if a request was sent via a form,

and the response should not cause the form to be refreshed or for a new page to load.

Status Code 304

Status Code 304 – This is the status code used for browser caching. If the response has not been modified, the client/user can continue to use the same response/cached version. For example, a browser can request if a resource has been modified since a specific time. If it hasn't, the status code, 304 is sent. If it has been modified, a status code 200 is sent, along with the resource.

Status Code 400

The server cannot understand and process a request due to a client error. Missing data, domain validation, and invalid formatting are some examples that cause the status code 400 to be sent.

Status Code 401

This status code request occurs when authentication is required but has failed or not been provided.

Status Code 403

Status Code 403 – Similar to status code 401, a status code 403 happens when a valid request is sent, but the server refuses to accept it. This happens if a client/user requires permission or may need an account to access the resource. Unlike a status code 401, authentication will not apply here.

Status Code 404

The most common status code the average user will see. A status code 404 occurs when the request is valid, but the resource cannot be found on the server. Even though these are grouped in the Client Errors "bucket," they are often due to improper URL redirection.

Status Code 409

A status code 409 is sent when a request conflicts with the current state of the resource. This is usually an issue with simultaneous updates or versions that conflict.

Status Code 410

The resource requested is no longer available and will not be available again. Learn about network error 410.

4. Find network id and Broadcast for 150.10.20.30.

a. Network Address: 150.10.20.0
b. Broadcast Address 150.10.20.255

5. Setup office vpn and access self learning platforms.