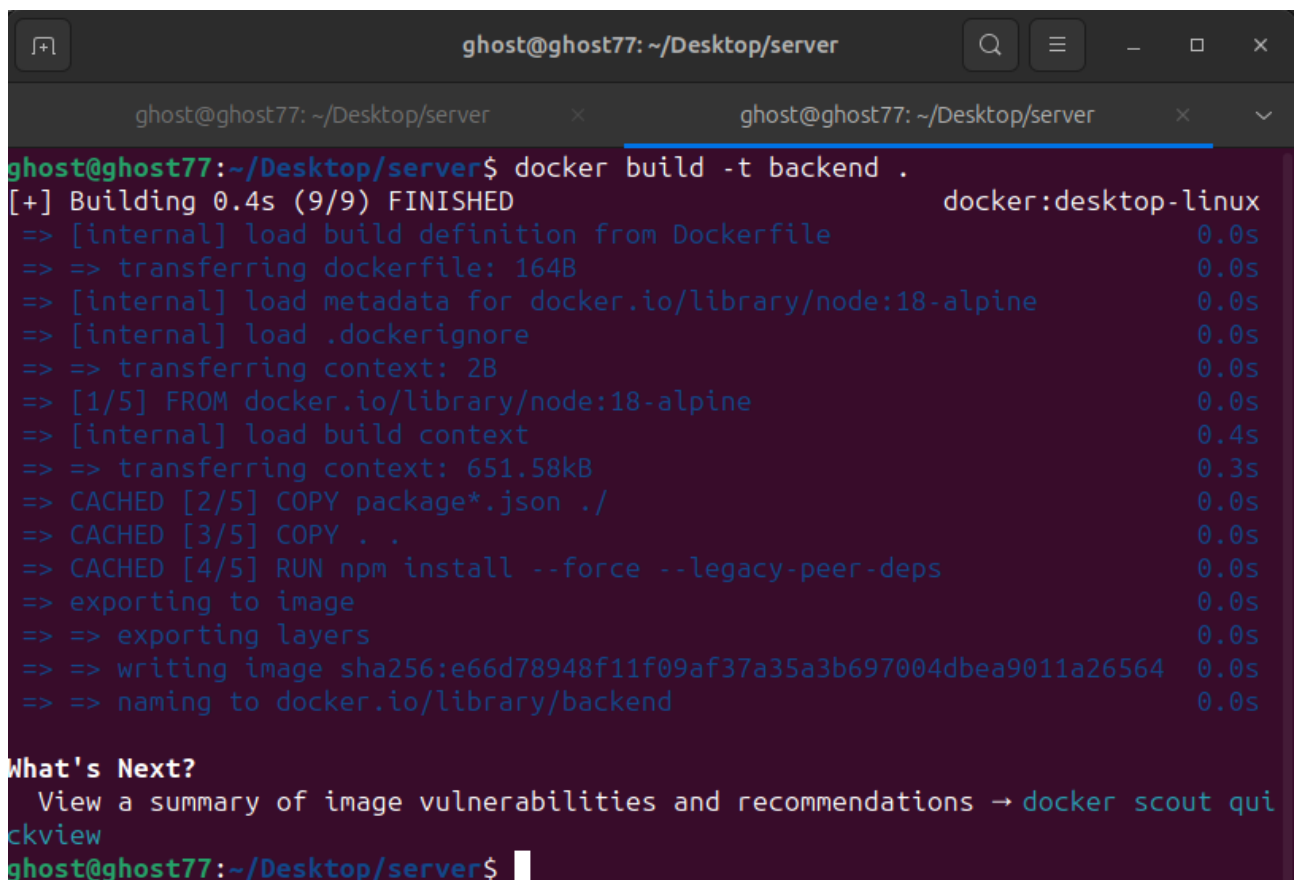


5. Push and Pull your own image with pre-installations to/from repository

To push and pull your own Docker image with pre-installations to and from a repository, follow these steps:

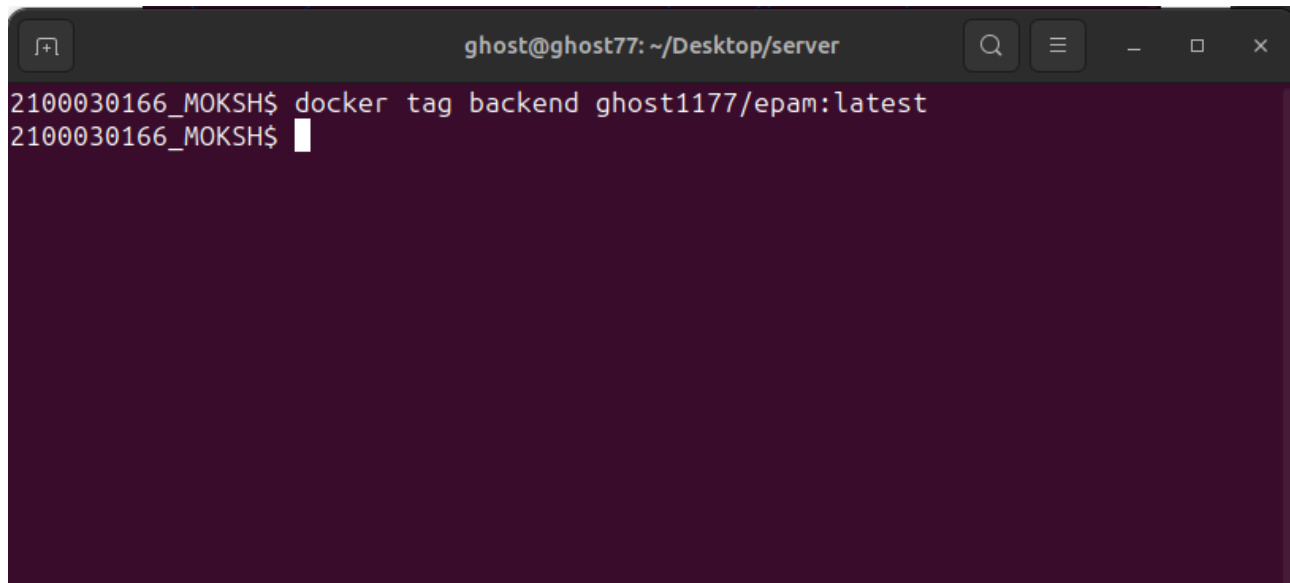
1. Create a Dockerfile: Start by creating a Dockerfile that describes the configuration of your Docker image. This file specifies the base image, any additional configurations or dependencies, and the commands needed to set up your application. You can use a text editor to create a file named "Dockerfile" (with no file extension) in your project directory.
2. Build the Docker image: Open a terminal or command prompt and navigate to the directory containing your Dockerfile. Use the `docker build` command to build the Docker image.

A terminal window titled 'ghost@ghost77: ~/Desktop/server' showing the execution of 'docker build -t backend .' in the directory ~/Desktop/server. The build process is shown with progress bars and timing for each step, including loading the Dockerfile, metadata, context, and building the image layers. The final output shows the image is named 'docker.io/library/backend'. Below the build output, there is a section titled 'What's Next?' with a link to 'docker scout quickview'.

```
ghost@ghost77: ~/Desktop/server$ docker build -t backend .
[+] Building 0.4s (9/9) FINISHED                                docker:desktop-linux
=> [internal] load build definition from Dockerfile              0.0s
=> => transferring dockerfile: 164B                             0.0s
=> [internal] load metadata for docker.io/library/node:18-alpine 0.0s
=> [internal] load .dockerignore                                0.0s
=> => transferring context: 2B                                   0.0s
=> [1/5] FROM docker.io/library/node:18-alpine                  0.0s
=> [internal] load build context                                0.4s
=> => transferring context: 651.58kB                             0.3s
=> CACHED [2/5] COPY package*.json ./                            0.0s
=> CACHED [3/5] COPY . .                                         0.0s
=> CACHED [4/5] RUN npm install --force --legacy-peer-deps      0.0s
=> exporting to image                                           0.0s
=> => exporting layers                                           0.0s
=> => writing image sha256:e66d78948f11f09af37a35a3b697004dbea9011a26564 0.0s
=> => naming to docker.io/library/backend                       0.0s

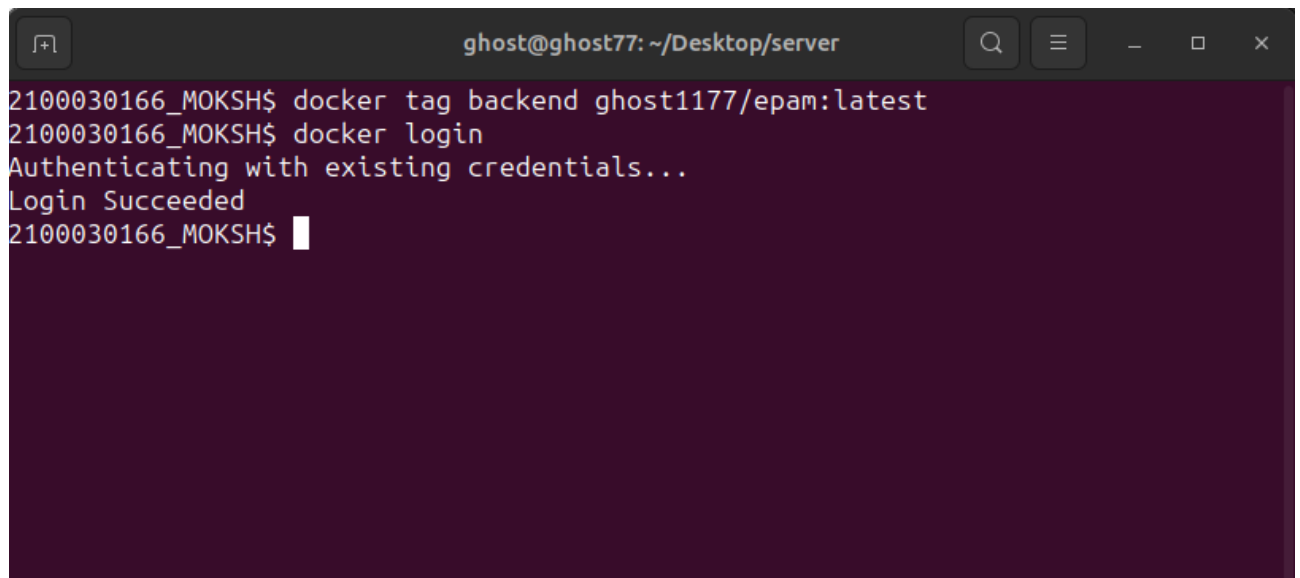
What's Next?
View a summary of image vulnerabilities and recommendations → docker scout quickview
ghost@ghost77:~/Desktop/server$
```

- Tag the image: After successfully building the image, you need to tag it with a repository name. Use the `docker tag` command to assign a repository name to your image. The command syntax is as follows:

A terminal window with a dark purple background. The title bar shows 'ghost@ghost77: ~/Desktop/server'. The prompt is '2100030166_MOKSH\$'. The command 'docker tag backend ghost1177/epam:latest' has been entered and executed. A cursor is visible on the line following the command.

```
ghost@ghost77: ~/Desktop/server
2100030166_MOKSH$ docker tag backend ghost1177/epam:latest
2100030166_MOKSH$
```

- Authenticate Docker: Before pushing the image to Docker Hub, you need to authenticate Docker to your Docker Hub account. Use the `docker login` command and provide your Docker Hub credentials when prompted:

A terminal window with a dark purple background. The title bar shows 'ghost@ghost77: ~/Desktop/server'. The prompt is '2100030166_MOKSH\$'. The command 'docker login' has been entered and executed. The output shows 'Authenticating with existing credentials...' and 'Login Succeeded'. A cursor is visible on the line following the command.

```
ghost@ghost77: ~/Desktop/server
2100030166_MOKSH$ docker tag backend ghost1177/epam:latest
2100030166_MOKSH$ docker login
Authenticating with existing credentials...
Login Succeeded
2100030166_MOKSH$
```

- Push the image: Once authenticated, you can push the tagged image to your Docker Hub repository using the `docker push` command. The command syntax is as follows:

```
ghost@ghost77: ~/Desktop/server
2100030166_MOKSH$ docker tag backend ghost1177/epam:latest
2100030166_MOKSH$ docker login
Authenticating with existing credentials...
Login Succeeded
2100030166_MOKSH$ docker push ghost1177/epam:latest
The push refers to repository [docker.io/ghost1177/epam]
ed654b282edb: Pushed
65a191582db4: Pushed
f0dc3ec7a7f9: Mounted from ghost1177/mern
efdc30a9b9c0: Mounted from ghost1177/mern
a7ede14bfe13: Mounted from ghost1177/mern
d49996f6d982: Mounted from ghost1177/mern
d4fc045c9e3a: Mounted from ghost1177/mern
latest: digest: sha256:91de7231a3771cd5385618ad772986ef0c0778c597203d4f7441d7dfbad0b95a size: 1790
```

- Pull the image: To pull the image from the repository, use the `docker pull` command followed by the repository name and tag. The command syntax is as follows:

```
ghost@ghost77: ~/Desktop/server
2100030166_MOKSH$ docker login
Authenticating with existing credentials...
Login Succeeded
2100030166_MOKSH$ docker push ghost1177/epam:latest
The push refers to repository [docker.io/ghost1177/epam]
ed654b282edb: Pushed
65a191582db4: Pushed
f0dc3ec7a7f9: Mounted from ghost1177/mern
efdc30a9b9c0: Mounted from ghost1177/mern
a7ede14bfe13: Mounted from ghost1177/mern
d49996f6d982: Mounted from ghost1177/mern
d4fc045c9e3a: Mounted from ghost1177/mern
latest: digest: sha256:91de7231a3771cd5385618ad772986ef0c0778c597203d4f7441d7dfbad0b95a size: 1790
2100030166_MOKSH$ docker pull ghost1177/epam:latest
latest: Pulling from ghost1177/epam
Digest: sha256:91de7231a3771cd5385618ad772986ef0c0778c597203d4f7441d7dfbad0b95a
Status: Image is up to date for ghost1177/epam:latest
docker.io/ghost1177/epam:latest

What's Next?
View a summary of image vulnerabilities and recommendations → docker scout quickview ghost1177/epam:latest
2100030166_MOKSH$
```