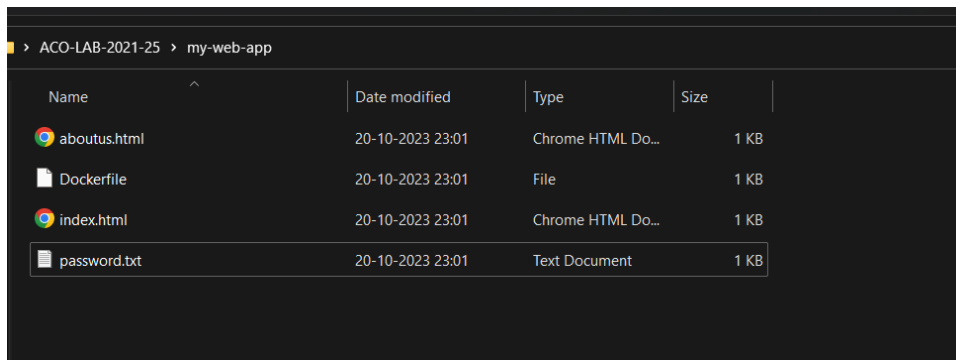


Lab Exercise 2– .dockerignore in Docker

Below is a lab exercise that can help you understand and practice using the .dockerignore file:

Create a Sample Project:

- Create a simple project directory with various types of files, including source code, configuration files, and build artifacts.

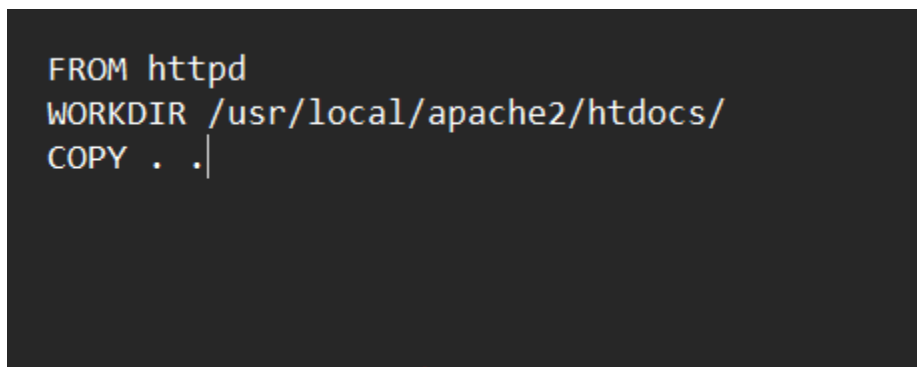


The screenshot shows a file explorer window with the path 'ACO-LAB-2021-25 > my-web-app'. It displays a table of files and folders:

Name	Date modified	Type	Size
aboutus.html	20-10-2023 23:01	Chrome HTML Do...	1 KB
Dockerfile	20-10-2023 23:01	File	1 KB
index.html	20-10-2023 23:01	Chrome HTML Do...	1 KB
password.txt	20-10-2023 23:01	Text Document	1 KB

Create a Dockerfile:

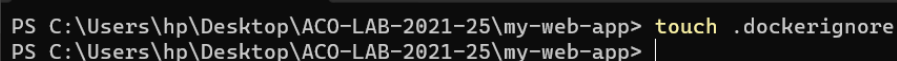
- Write a Dockerfile that specifies the build steps for your application, including any necessary dependencies.



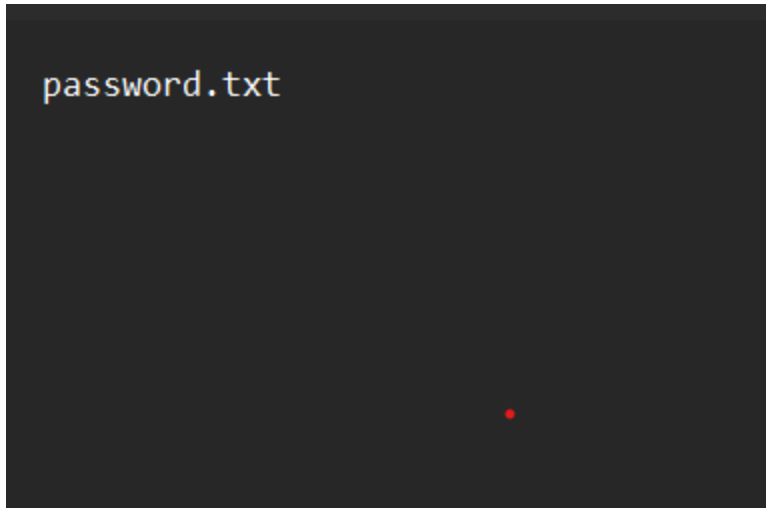
```
FROM httpd
WORKDIR /usr/local/apache2/htdocs/
COPY . .|
```

Create a .dockerignore File:

- Create a .dockerignore file in the root directory of your project.
- Add specific files or directories to the .dockerignore file that you do not want to be included in the Docker image during the build process.



```
PS C:\Users\hp\Desktop\ACO-LAB-2021-25\my-web-app> touch .dockerignore
PS C:\Users\hp\Desktop\ACO-LAB-2021-25\my-web-app> |
```



Build the Docker Image:

- Use the docker build command to build the Docker image from the Dockerfile.
- Observe the build context and check which files are being included in the Docker image.

```
PS C:\Users\hp\Desktop\AC0-LAB-2021-25\my-web-app> docker build -t my-image:0.1 .
[*] Building 15.1s (9/9) FINISHED
=> [internal] load .dockerignore
=> => transferring context: 52B
=> [internal] load build definition from Dockerfile
=> => transferring dockerfile: 93B
=> [internal] load metadata for docker.io/library/httpd:latest
=> [auth] library/httpd:pull token for registry-1.docker.io
=> [1/3] FROM docker.io/library/httpd@sha256:ed6db4a8c394d075c9c59a3dbd61a3818cd302d9948057f1e19046e5bffc027
=> => resolve docker.io/library/httpd@sha256:ed6db4a8c394d075c9c59a3dbd61a3818cd302d9948057f1e19046e5bffc027
=> => sha256:073cbcfef6634b5131786873a7a92a3b3bda43672e5830126dfa94352649358d 4.28MB / 4.28MB
=> => sha256:ed6db4a8c394d075c9c59a3dbd61a3818cd302d9948057f1e19046e5bffc027 1.86kB / 1.86kB
=> => sha256:2f1ec45327a35711f293cf543c2e0efadfd44bc71e8081dcd76a558b99778095 1.37kB / 1.37kB
=> => sha256:75a48b16cd565cdaff0cfcb3462c292e56108a22b9733c0a04a9dclcbd7a774 9.09kB / 9.09kB
=> => sha256:c20157372e943d84bb5a0624e80395697de1f41ecd54b3bcead2b03bb6b13fe8 176B / 176B
=> => sha256:a378f10b321842c3042cdeff4f6997f34f4cb21f2eff27704b7f6193ab7b5fea 29.15MB / 29.15MB
=> => sha256:5f75f8a17f406c926d42ef7e5e0803a23b150c46f77cda9c83bff10db3d68f2f 31.40MB / 31.40MB
=> => sha256:39fc6f8c5be2c5b90e4d08812608080bc921cb4c1db44c7a55e0b638914f0842f 301B / 301B
=> => extracting sha256:a378f10b321842c3042cdeff4f6997f34f4cb21f2eff27704b7f6193ab7b5fea 2.3s
=> => extracting sha256:c20157372e943d84bb5a0624e80395697de1f41ecd54b3bcead2b03bb6b13fe8 0.0s
=> => extracting sha256:073cbcfef6634b5131786873a7a92a3b3bda43672e5830126dfa94352649358d 0.2s
=> => extracting sha256:5f75f8a17f406c926d42ef7e5e0803a23b150c46f77cda9c83bff10db3d68f2f 1.5s
=> => extracting sha256:39fc6f8c5be2c5b90e4d08812608080bc921cb4c1db44c7a55e0b638914f0842f 0.0s
=> [internal] load build context
=> => transferring context: 482B
=> [2/3] WORKDIR /usr/local/apache2/htdocs/
=> [3/3] COPY . .
=> exporting to image
=> => exporting layers
=> => writing image sha256:ee26eedad720cf60849c7f9bf5c7c387ecb2bb5e292a2636a704b39afc82b929
=> => naming to docker.io/library/my-image:0.1

PS C:\Users\hp\Desktop\AC0-LAB-2021-25\my-web-app> docker run -it my-image:0.1 bash
root@ba1ce1ca795e:/usr/local/apache2/htdocs# ls
Dockerfile aboutus.html index.html
root@ba1ce1ca795e:/usr/local/apache2/htdocs#
```

Modify .dockerignore and Rebuild:

- Modify the .dockerignore file to exclude different types of files or directories from the Docker image.
- Rebuild the Docker image and observe the changes in the build context.

password.txt
aboutus.html

```
PS C:\Users\hp\Desktop\ACQ-LAB-2021-25\my-web-app> docker build -t img .
[*] Building 1.8s (8/8) FINISHED
=> [internal] load .dockerignore
=> => transferring context: 80B
=> [internal] load build definition from Dockerfile
=> => transferring dockerfile: 93B
=> [internal] load metadata for docker.io/library/httpd:latest
=> [1/3] FROM docker.io/library/httpd@sha256:ed6db4a8c394d075c9c59a3bdc61a3818cd302d9948057f1e19046e5bffe027
=> [internal] load build context
=> => transferring context: 109B
=> CACHED [2/3] WORKDIR /usr/local/apache2/htdocs/
=> [3/3] COPY . .
=> exporting to image
=> => exporting layers
=> => writing image sha256:528ebf75f7c674f2738f065d97486ae4572597e8958b22041c3c21fb4e181a21
=> => naming to docker.io/library/img
docker:default
0.1s
0.0s
0.1s
0.0s
1.1s
0.0s
0.0s
0.0s
0.0s
0.2s
0.2s
0.2s
0.0s
0.0s

What's Next?
View a summary of image vulnerabilities and recommendations → docker scout quickview
PS C:\Users\hp\Desktop\ACQ-LAB-2021-25\my-web-app> docker run -it img bash
root@c8ba05d9f1b6:/usr/local/apache2/htdocs# ls
Dockerfile index.html
root@c8ba05d9f1b6:/usr/local/apache2/htdocs# df -h
Filesystem      1K-blocks    Used Available Use% Mounted on
overlay          1055762868 1578252 1000481144   1% /
proc              0            0           0    - /proc
tmpfs             65536         0        65536    0% /dev
devpts            0            0           0    - /dev/pts
sysfs             0            0           0    - /sys
tmpfs            1841720         0      1841720    0% /sys/fs/cgroup
cpuset           0            0           0    - /sys/fs/cgroup/cpuset
cpu              0            0           0    - /sys/fs/cgroup/cpu
cpuacct          0            0           0    - /sys/fs/cgroup/cpuacct
blkio            0            0           0    - /sys/fs/cgroup/blkio
memory          0            0           0    - /sys/fs/cgroup/memory
devices          0            0           0    - /sys/fs/cgroup/devices
freezer          0            0           0    - /sys/fs/cgroup/freezer
net_cls          0            0           0    - /sys/fs/cgroup/net_cls
```

Test the Docker Image:

- Run the Docker image as a container using the docker run command.
- Verify that the container works as expected, despite certain files being excluded from the Docker image.

```
PS C:\Users\hp\Desktop\AC0-LAB-2021-25\my-web-app> docker build -t img .
[*] Building 1.8s (8/8) FINISHED
=> [internal] load .dockerignore
=> => transferring context: 808B
=> [internal] load build definition from Dockerfile
=> => transferring dockerfile: 93B
=> [internal] load metadata for docker.io/library/httpd:latest
=> [1/3] FROM docker.io/library/httpd@sha256:ed6db4a8c394d075c9c59a3dbd61a3818cd382d9948057f1e19046e5bfffec027
=> [internal] load build context
=> => transferring context: 189B
=> CACHED [2/3] WORKDIR /usr/local/apache2/htdocs/
=> [3/3] COPY . .
=> exporting to image
=> => exporting layers
=> => writing image sha256:528ebf75f7c674f2738f065d97486ae4572597e8958b22041c3c21fb4e181a21
=> => naming to docker.io/library/img

What's Next?
View a summary of image vulnerabilities and recommendations → docker scout quickview
PS C:\Users\hp\Desktop\AC0-LAB-2021-25\my-web-app> docker run -it img bash
root@c8ba05d9f1b6:/usr/local/apache2/htdocs# ls
Dockerfile index.html
root@c8ba05d9f1b6:/usr/local/apache2/htdocs# df -h
Filesystem      1K-blocks    Used Available Use% Mounted on
overlay          1055762868 1578252 1000481144    1% /
proc              0            0           0    - /proc
tmpfs             65536         0        65536    0% /dev
devpts            0            0           0    - /dev/pts
sysfs             0            0           0    - /sys
tmpfs            1841720         0      1841720    0% /sys/fs/cgroup
cpuset            0            0           0    - /sys/fs/cgroup/cpuset
cpu               0            0           0    - /sys/fs/cgroup/cpu
cpuct             0            0           0    - /sys/fs/cgroup/cpuacct
blkio             0            0           0    - /sys/fs/cgroup/blkio
memory           0            0           0    - /sys/fs/cgroup/memory
devices           0            0           0    - /sys/fs/cgroup/devices
freezer           0            0           0    - /sys/fs/cgroup/freezer
net_cls           0            0           0    - /sys/fs/cgroup/net_cls
```

Optimize the .dockerignore File:

- Experiment with different patterns and wildcards in the .dockerignore file to optimize the build context and reduce the image size.

Check Image Size and Contents:

- Use the docker image ls and docker image inspect commands to check the size of the Docker image and its contents.

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
img	latest	528ebf75f7c6	3 minutes ago	168MB
new-image	latest	1b791b120d8d	5 minutes ago	168MB
my-image	0.1	ee26eedad720	12 minutes ago	168MB

Update .dockerignore for CI/CD:

- Modify the .dockerignore file to optimize the build context and reduce build times in a continuous integration and continuous deployment (CI/CD) pipeline.

Documentation and Best Practices:

- Document your findings and the best practices for using the .dockerignore file in your projects.

Through this exercise, you'll gain a better understanding of how to use the .dockerignore file effectively to optimize Docker image builds and reduce image sizes. Adjust the exercise based on your specific project requirements.