

Name -Anmol yadav

SAP ID -500083814

EXPERIMENT 10

AIM: Working with .dockerignore

Perform Docker Link and .dockerignore based scenario on Katacoda.

Working with .dockerignore

Step 1 - Docker Ignore

To prevent sensitive files or directories from being included by mistake in images, you can add a file named *.dockerignore*.

Example

A Dockerfile copies the working directory into the Docker Image. As a result, this would include potentially sensitive information such as a passwords file which we'd want to manage outside the image. View the Dockerfile with `cat Dockerfile`

Build the image with `docker build -t password .` Look at the output using `docker run password ls /app` This will include the passwords file.

```

anmol@LAPTOP-T9RMRNSD MINGW64 ~/docker_lab10 (main)
$ pwd
/c/Users/anmol/docker_lab10

anmol@LAPTOP-T9RMRNSD MINGW64 ~/docker_lab10 (main)
$ docker build -t exp_10:1.0 .
#1 [internal] load build definition from Dockerfile
#1 sha256:9c9003974b6acbbddad12eff7eba3105e73594b52b4f41e502e3ed815b5e5ba2
#1 transferring dockerfile: 91B done
#1 DONE 0.0s

#2 [internal] load .dockerignore
#2 sha256:2d4033529ad5dec27af5bd482c659f1ca78d37649ad7afbeba51ff085abe5d50
#2 transferring context: 2B done
#2 DONE 0.0s

#3 [internal] load metadata for docker.io/library/alpine:latest
#3 sha256:d4fb25f5b5c00defc20ce26f2efc4e288de8834ed5aa59dff877b495ba88fda6
#3 DONE 0.0s

#4 [1/4] FROM docker.io/library/alpine
#4 sha256:665ba8b2cdc0cb0200e2a42a6b3c0f8f684089f4cd1b81494fbb9805879120f7
#4 DONE 0.0s

#6 [internal] load build context
#6 sha256:6c96f43cba48b326d661821cfe24d56dec12e1e616d79197a9c000261ca3063c
#6 transferring context: 124B done
#6 DONE 0.0s

#5 [2/4] RUN mkdir /app
#5 sha256:aaae3c5bf6b6be6b00f1f6fe67ae40908ec4a41aa3e5d90cfe772fe3893d68ad
#5 CACHED

#7 [3/4] COPY . /app
#7 sha256:e7b04dae751fbbc443a27a02aFc0f458ee5dbc205d7917fdcf695b16f69509f3
#7 DONE 0.0s

#8 [4/4] WORKDIR /app
#8 sha256:0f94d28c357cfb6ff38ccf29c684e5ae670ed357cf25c54a50f308057abb0138
#8 DONE 0.0s

#9 exporting to image
#9 sha256:e8c613e07b0b7ff33893b694f7759a10d42e180f2b4dc349fb57dc6b71dcab00
#9 exporting layers 0.0s done
#9 writing image sha256:94752cb6dd49b84624e59b2df6769265cf35742f688bc0f06e16c23e8ee21191 done
#9 naming to docker.io/library/exp_10:1.0 done
#9 DONE 0.0s

Use 'docker scan' to run Snyk tests against images to find vulnerabilities and learn how to fix them

```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
exp_10	1.0	94752cb6dd49	38 seconds ago	5.54MB
lab_10	1.2	f84f3877b6d2	8 hours ago	5.54MB
<none>	<none>	26b25fc44e68	8 hours ago	5.84MB
lab_10	1.0	ed1b392a19c1	8 hours ago	5.54MB
anmol6758/r214220178	latest	dad8ca893c9e	3 weeks ago	200MB
r214220178	latest	dad8ca893c9e	3 weeks ago	200MB
anmol6758/my-roll-no-0178	latest	94e38a63a397	3 weeks ago	142MB
my-roll-no-0178	latest	94e38a63a397	3 weeks ago	142MB

```


anmol@LAPTOP-T9RMRNSD MINGW64 ~/docker_lab10 (main)
$ docker build -t exp_10:1.1 .
#1 [internal] load build definition from Dockerfile
#1 sha256:633ef000b487aa4e5b11d0f4b52b961f58b06297bc8f1d33eecd19cb1f29e3dd
#1 transferring dockerfile: 31B done
#1 DONE 0.0s

#2 [internal] load .dockerignore
#2 sha256:11d018a0056a9fb7b91fcb4e1e1010324a2d1ff029f7c2712f6af0933101acdf
#2 transferring context: 2B done
#2 DONE 0.0s

#3 [internal] load metadata for docker.io/library/alpine:latest
#3 sha256:d4fb25f5b5c00defc20ce26f2efc4e288de8834ed5aa59dff877b495ba88fda6
#3 DONE 0.0s

#4 [1/4] FROM docker.io/library/alpine
#4 sha256:665ba8b2cdc0cb0200e2a42a6b3c0f8f684089f4cd1b81494fbb9805879120f7
#4 DONE 0.0s

```


 .dockerignore

 final.PNG


 free2.PNG


 realloc.PNG

 Dockerfile


 finaloutput.PNG


 freeinc.PNG


 realloc3.PNG

 example 2.PNG


 fork.PNG

 password.txt

 reallocinc2.PNG

 example.PNG

 free1.PNG

 realloc output.PNG

```
C:\Users\anmol>docker run exp_10:1.0 ls /app
Dockerfile
password.txt
```

```
C:\Users\anmol>docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
exp_10	1.1	f3ace12cdc89	31 seconds ago	6.41MB
exp_10	1.0	94752cb6dd49	8 minutes ago	5.54MB
lab_10	1.2	f84f3877b6d2	8 hours ago	5.54MB
<none>	<none>	26b25fc44e68	8 hours ago	5.84MB
lab_10	1.0	ed1b392a19c1	8 hours ago	5.54MB
anmol16758/e714220178	latest	da08ca893c9a	3 weeks ago	200MB

```
C:\Users\anmol>docker run exp_10:1.1 ls /app
Dockerfile
Screenshot 2022-08-27 175716.jpg
example 2.PNG
example.PNG
final.PNG
finaloutput.PNG
fork.PNG
free1.PNG
free2.PNG
freeinc.PNG
password.txt
polynesia-3021072_1920.jpg
realloc output.PNG
realloc.PNG
realloc3.PNG
reallocinc2.PNG
```

Ignore File

The following command would include passwords.txt in our *.dockerignore* file and ensure that it didn't accidentally end up in a container. The *.dockerignore* file would be stored in source control and share with the team to ensure that everyone is consistent.

```
echo passwords.txt >> .dockerignore
```

The ignore file supports directories and Regular expressions to define the restrictions, very similar to *.gitignore*. This file can also be used to improve build times which we'll investigate in the next step.

Build the image, because of the Docker Ignore file it shouldn't include the passwords file.

```
docker build -t nopassword .
```

Look at the output using `docker run nopassword ls /app`

Protip

If you need to use the passwords as part of a *RUN* command then you need to copy, execute and delete the files as part of a single RUN command. Only the final state of the Docker container is persisted inside the image.

Step 2 - Docker Build Context

The `.dockerignore` file can ensure that sensitive details are not included in a Docker Image. However they can also be used to improve the build time of images.

In the environment, a 100M temporary file has been created. This file is never used by the `Dockerfile`. When you execute a build command, Docker sends the entire path contents to the Engine for it to calculate which files to include. As a result sending the 100M file is unrequired and creates a slower build.

You can see the 100M impact by executing following the command.

```
docker build -t large-file-context .
```

In the next step, we'll demonstrate how to improve the performance of the build.

Protip

It's wise to ignore `.git` directories along with dependencies that are downloaded/built within the image such as `node_modules`. These are never used by the application running within the Docker Container and just add overhead to the build process.

```
anmol@LAPTOP-T9RMRNSD MINGW64 ~/docker_lab10 (main)
$ echo password.txt >> .dockerignore

anmol@LAPTOP-T9RMRNSD MINGW64 ~/docker_lab10 (main)
$ docker build -t exp_10:1.2 .
#1 [internal] load build definition from Dockerfile
#1 sha256:628aad80061e6e41833dd1a287fdf0b172f2c328461dc244c5d2de07fc01ed2d
#1 transferring dockerfile: 31B done
#1 DONE 0.0s

#2 [internal] load .dockerignore
#2 sha256:bcadcb74dea47d4e6883da80be655b9561f5dae5d9bbf18a7224cfcc522f6b65
#2 transferring context: 53B done
#2 DONE 0.0s

#3 [internal] load metadata for docker.io/library/alpine:latest
#3 sha256:d4fb25f5b5c00defc20ce26f2efc4e288de8834ed5aa59dff877b495ba88fda6
#3 DONE 0.0s
```

```
password.txt
*.PNG
```

```
C:\Users\anmol>docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
exp_10	1.2	ebfc46bfe53f	19 seconds ago	6.41MB
exp_10	1.1	f3ace12cdc89	4 minutes ago	6.41MB
exp_10	1.0	94752cb0dd49	12 minutes ago	5.54MB
lab_10	1.2	f84f3877b6d2	8 hours ago	5.54MB
<none>	<none>	26b25fc44c68	8 hours ago	5.84MB
lab_10	1.0	ed1b392a19c1	8 hours ago	5.54MB
214220178	latest	dad8ca893c9e	3 weeks ago	200MB

```
C:\Users\anmol>docker run exp_10:1.2 ls /app
Dockerfile
example.2.PNG
example.PNG
final.PNG
finaloutput.PNG
fork.PNG
free1.PNG
free2.PNG
freeinc.PNG
password.txt
realloc output.PNG
realloc.PNG
realloc3.PNG
reallocinc2.PNG
```

```
C:\Users\anmol>docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
exp_10	1.3	d2fc2468a954	12 seconds ago	5.54MB
exp_10	1.2	58847e95c9ba	About a minute ago	5.68MB
<none>	<none>	ebfc46bfe53f	6 minutes ago	6.41MB
exp_10	1.1	f3ace12cdc89	11 minutes ago	6.41MB
exp_10	1.0	94752cb6dd49	19 minutes ago	5.54MB
lab_10	1.2	f84f3877b6d2	8 hours ago	5.54MB
<none>	<none>	26b25fc44e68	8 hours ago	5.84MB
lab_10	1.0	ed1b392a19c1	8 hours ago	5.54MB
anmol6758/r214220178	latest	dad8ca893c9e	3 weeks ago	200MB
r214220178	latest	dad8ca893c9e	3 weeks ago	200MB
anmol6758/my-roll-no-0178	latest	94e38a63a397	3 weeks ago	142MB
my-roll-no-0178	latest	94e38a63a397	3 weeks ago	142MB
<none>	<none>	acc246ed59de	3 weeks ago	203MB
mysql	latest	43fcfca0776d	4 weeks ago	449MB
nginx	latest	2d389e545974	5 weeks ago	142MB

```
anmol@LAPTOP-T9RMRNSD MINGW64 ~/docker_lab10 (main)
$ docker build -t exp_10:1.3 .
#1 [internal] load build definition from Dockerfile
#1 sha256:b341c026d24dc301acac7727d7a821f4947e880cc53a19242e7c597f8eda2111
#1 transferring dockerfile: 31B done
#1 DONE 0.0s

#2 [internal] load .dockerignore
#2 sha256:69e70e0aaafc331763c0378830d4613ac6cab2adae674443207513de5d3feb89
#2 transferring context: 59B done
#2 DONE 0.0s

#3 [internal] load metadata for docker.io/library/alpine:latest
#3 sha256:d4fb25f5b5c00defc20ce26f2efc4e288de8834ed5aa59dfff877b495ba88fda6
#3 DONE 0.0s

#4 [1/4] FROM docker.io/library/alpine
#4 sha256:665b9b32d1940290c3e43c6b3e68568408051e11b814045b4080587013067
```

```
C:\Users\anmol>docker run exp_10:1.3 ls /app
Dockerfile
```

```
C:\Users\anmol>
```