Dockerfile - Persistence and Layers

We will prove that without caution, docker images return a lot of information.

# Setting environment variables

* Create a directory “secret0” and use this for this exercise.
* Create a Dockerfile.
  + The Dockerfile uses the alpine image.
  + It sets 2 environment variables: PASSWORD and USER.
  + PASSWORD gets the value “password” and USER gets the value “admin”.
* Build the image, and name it “alpine\_secret0”.
* Run a container based on this image.
* There is no output, but run the container again interactively, with a TTY and see if the environment variables are present.

Dockerfile:

| FROM alpine:latest  LABEL author="Rodzers Usackis <rodzers.usackis@student.kdg.be>"  ENV USER admin  ENV PASSWORD password |
| --- |

Run the container based on this image:

| rodzers\_usackis@cloudshell:~/secret0$ docker build -t alpine\_secret0 .  Sending build context to Docker daemon 2.048kB  Step 1/4 : FROM alpine:latest  ---> 0ac33e5f5afa  Step 2/4 : LABEL author="Rodzers Usackis <rodzers.usackis@student.kdg.be>"  ---> Using cache  ---> 767c1a156da2  Step 3/4 : ENV USER admin  ---> Using cache  ---> 39286f863cd4  Step 4/4 : ENV PASSWORD password  ---> Using cache  ---> fcc5b6ca3fd7  Successfully built fcc5b6ca3fd7  Successfully tagged alpine\_secret0:latest  rodzers\_usackis@cloudshell:~/secret0$ docker container run -it fcc  / # env  USER=admin  HOSTNAME=1c036d5678be  SHLVL=1  HOME=/root  TERM=xterm  PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin  PWD=/  PASSWORD=password  / # |
| --- |

**SAMPLE OUTPUT:**

| **/ # env**  USER=admin  HOSTNAME=a00088c121d4  SHLVL=1  HOME=/root  TERM=xterm  PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin  PWD=/  PASSWORD=password  **/ #** |
| --- |

# Create a second image

* Create a directory “secret1” and use this for this exercise.
* Create a Dockerfile.
  + The Dockerfile uses the alpine\_secret0 image created in the previous step.
  + It uses the CMD instruction to show the contents of the environment variables PASSWORD and USER.

DockerFile:

| FROM alpine:latest  LABEL author="Rodzers Usackis <rodzers.usackis@student.kdg.be>"  ENV PASSWORD password  ENV USER admin  CMD echo PASSWORD is $PASSWORD && echo USER is $USER |
| --- |

Build the image and name it “alpine\_secret1”. Run it as a container.

Run the container based on this image:

| rodzers\_usackis@cloudshell:~/secret1$ docker build -t alpine\_secret1 .  Sending build context to Docker daemon 2.048kB  Step 1/5 : FROM alpine:latest  ---> 0ac33e5f5afa  Step 2/5 : LABEL author="Rodzers Usackis <rodzers.usackis@student.kdg.be>"  ---> Using cache  ---> 767c1a156da2  Step 3/5 : ENV PASSWORD password  ---> Using cache  ---> 3b73360415bf  Step 4/5 : ENV USER admin  ---> Using cache  ---> 6efaf02e078a  Step 5/5 : CMD echo PASSWORD is $PASSWORD && echo USER is $USER  ---> Using cache  ---> 177029792297  Successfully built 177029792297  Successfully tagged alpine\_secret1:latest  rodzers\_usackis@cloudshell:~/secret1$ docker run 177  PASSWORD is password  USER is admin |
| --- |

**SAMPLE OUTPUT:**

| **$ docker run alpine\_secret1**  PASSWORD is password  USER is admin |
| --- |

# Updating the 2nd image with a “secret” file

Place a file in the “secret1” directory, called secret.txt.

Record some *secret* text in the file.

Update the Dockerfile in “secret1” so that it also copies the secret.txt file to the root of the image.

Build the image and name it “alpine\_secret2”.

Run the image and check via the interactive shell that the file is present.

DockerFile:

| FROM alpine:latest  LABEL author="Rodzers Usackis <rodzers.usackis@student.kdg.be>"  ENV PASSWORD password  ENV USER admin  COPY "secret.txt" "./"  CMD echo PASSWORD is $PASSWORD && echo USER is $USER |
| --- |

Run the container based on this image:

| rodzers\_usackis@cloudshell:~/secret1$ docker build -t alpine\_secret2 .  Sending build context to Docker daemon 3.072kB  Step 1/6 : FROM alpine:latest  ---> 0ac33e5f5afa  Step 2/6 : LABEL author="Rodzers Usackis <rodzers.usackis@student.kdg.be>"  ---> Using cache  ---> 767c1a156da2  Step 3/6 : ENV PASSWORD password  ---> Using cache  ---> 3b73360415bf  Step 4/6 : ENV USER admin  ---> Using cache  ---> 6efaf02e078a  Step 5/6 : COPY "secret.txt" "./"  ---> Using cache  ---> 2141745a718b  Step 6/6 : CMD echo PASSWORD is $PASSWORD && echo USER is $USER  ---> Using cache  ---> 8597fa2c9d8e  Successfully built 8597fa2c9d8e  Successfully tagged alpine\_secret2:latest  rodzers\_usackis@cloudshell:~/secret1$ docker run -it alpine\_secret2 "/bin/sh"  / # cat secret.txt  This is some very secret text!  / # exit |
| --- |

**SAMPLE OUTPUT:**

| **$ docker run -it alpine\_secret2 “/bin/sh”**  / # cat secret.txt  This is some very secret text!  / # |
| --- |

OR: In the previous sample output the container continues running unless you terminate the container manually. An alternative way is to run the container, display the text and let the container exit immediately afterwards. How is the ‘docker run’ instruction (don’t modify the Dockerfile)?

| rodzers\_usackis@cloudshell:~/secret1$ docker run alpine\_secret2 cat secret.txt  This is some very secret text! |
| --- |

# Create image #3: delete “secret.txt”

Create a “secret2” directory.

Create a Dockerfile that is based on the alpine\_secret2 image.

This Dockerfile should have instructions to remove the secret.txt file within the image. Hint: use a RUN instruction.

DockerFile:

| FROM alpine\_secret2  LABEL author="Rodzers Usackis <rodzers.usackis@student.kdg.be>"  RUN rm ./secret.txt |
| --- |

Build the image and tag it as “alpine\_secret3”. Run the image in a container (non-interactive, no TTY).

Check with an ls command whether the file has gone.

| rodzers\_usackis@cloudshell:~/secret2$ docker run alpine\_secret\_3 ls  bin  dev  etc  home  lib  media  mnt  opt  proc  root  run  sbin  srv  sys  tmp  usr  var |
| --- |

Nevertheless, the secret.txt file is still present in the lower layers!

Can you find a way to retrieve the file on disk, from the alpine\_secret3 image?

Hints:

1. Use the history command on the latest image.
2. Check in which layer (image) the file still exists (= layer before removing the file).
3. Inspect this layer on the local disk to retrieve the file system layer in which it exists.
4. Go to the filesystem and the correct directory.
5. Retrieve/show the file.

List your solution:

| A)  rodzers\_usackis@cloudshell:~/secret2$ docker image history alpine\_secret\_3  “””  IMAGE CREATED CREATED BY SIZE COMMENT  1aa80296239c 7 minutes ago /bin/sh -c rm ./secret.txt 0B  ca3be88e3276 15 minutes ago /bin/sh -c #(nop) LABEL author=Rodzers Usac… 0B  8597fa2c9d8e 26 minutes ago /bin/sh -c #(nop) CMD ["/bin/sh" "-c" "echo… 0B  2141745a718b 26 minutes ago /bin/sh -c #(nop) COPY file:fae5d023c136eace… 31B  6efaf02e078a About an hour ago /bin/sh -c #(nop) ENV USER=admin 0B  3b73360415bf About an hour ago /bin/sh -c #(nop) ENV PASSWORD=password 0B  767c1a156da2 2 hours ago /bin/sh -c #(nop) LABEL author=Rodzers Usac… 0B  “””  B) The file still exists in 2141745a718b  C) rodzers\_usackis@cloudshell:~/solution$ docker image inspect 214  “””  [  {  "Id": "sha256:2141745a718b275f1f8b8dfc340bc4984022248bed615a62f28e7569f0cec2f2",  "RepoTags": [],  "RepoDigests": [],  "Parent": "sha256:6efaf02e078a2d9b20140ed959950a4120d4a1ed403fa69d0a7801256e369ca9",  "Comment": "",  "Created": "2022-05-23T02:26:06.676933758Z",  "Container": "",  "ContainerConfig": {  "Hostname": "",  "Domainname": "",  "User": "",  "AttachStdin": false,  "AttachStdout": false,  "AttachStderr": false,  "Tty": false,  "OpenStdin": false,  "StdinOnce": false,  "Env": [  "PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin",  "PASSWORD=password",  "USER=admin"  ],  "Cmd": [  "/bin/sh",  "-c",  "#(nop) COPY file:fae5d023c136eace67b81286a3c94bf3a312cb08406f6f7bb7d9eea149d1de7e in ./ "  ],  "Image": "sha256:6efaf02e078a2d9b20140ed959950a4120d4a1ed403fa69d0a7801256e369ca9",  "Volumes": null,  "WorkingDir": "",  "Entrypoint": null,  "OnBuild": null,  "Labels": {  "author": "Rodzers Usackis <rodzers.usackis@student.kdg.be>"  }  },  "DockerVersion": "20.10.14",  "Author": "",  "Config": {  "Hostname": "",  "Domainname": "",  "User": "",  "AttachStdin": false,  "AttachStdout": false,  "AttachStderr": false,  "Tty": false,  "OpenStdin": false,  "StdinOnce": false,  "Env": [  "PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin",  "PASSWORD=password",  "USER=admin"  ],  "Cmd": [  "/bin/sh"  ],  "Image": "sha256:6efaf02e078a2d9b20140ed959950a4120d4a1ed403fa69d0a7801256e369ca9",  "Volumes": null,  "WorkingDir": "",  "Entrypoint": null,  "OnBuild": null,  "Labels": {  "author": "Rodzers Usackis <rodzers.usackis@student.kdg.be>"  }  },  "Architecture": "amd64",  "Os": "linux",  "Size": 5574995,  "VirtualSize": 5574995,  "GraphDriver": {  "Data": {  "LowerDir": "/var/lib/docker/overlay2/923364e928b9936786b8848d42b1aaca63e094d386a5e2292401eacdce637860/diff",  "MergedDir": "/var/lib/docker/overlay2/dcd3e1bab4a5ca88d623f50bc36bd07b7db786e8dc69771cbce5f50356806ecc/merged",  "UpperDir": "/var/lib/docker/overlay2/dcd3e1bab4a5ca88d623f50bc36bd07b7db786e8dc69771cbce5f50356806ecc/diff",  "WorkDir": "/var/lib/docker/overlay2/dcd3e1bab4a5ca88d623f50bc36bd07b7db786e8dc69771cbce5f50356806ecc/work"  },  "Name": "overlay2"  },  "RootFS": {  "Type": "layers",  "Layers": [  "sha256:4fc242d58285699eca05db3cc7c7122a2b8e014d9481f323bd9277baacfa0628",  "sha256:a561da7073ca284db6d0cddc627dbd7d77695fbcce519aa4bf4ce8ae0f51a217"  ]  },  "Metadata": {  "LastTagTime": "0001-01-01T00:00:00Z"  }  }  ]  “””  rodzers\_usackis@cloudshell:~/secret2$ sudo find /var/lib/docker/overlay2/dcd3e1bab4a5ca88d623f50bc36bd07b7db786e8dc69771cbce5f50356806ecc/diff  /var/lib/docker/overlay2/dcd3e1bab4a5ca88d623f50bc36bd07b7db786e8dc69771cbce5f50356806ecc/diff  /var/lib/docker/overlay2/dcd3e1bab4a5ca88d623f50bc36bd07b7db786e8dc69771cbce5f50356806ecc/diff/secret.txt  E)  rodzers\_usackis@cloudshell:~$ mkdir solution  rodzers\_usackis@cloudshell:~$ sudo cp /var/lib/docker/overlay2/dcd3e1bab4a5ca88d623f50bc36bd07b7db786e8dc69771cbce5f50356806ecc/diff/secret.txt ./solution/  rodzers\_usackis@cloudshell:~$ ls solution/  secret.txt  > File copied to my local directory. |
| --- |