

PRACTICAL 04

LOCAL IMAGE RESOLUTION REDUCTION USING DOCKER

Common Commands:

run	Create and run a new container from an image
exec	Execute a command in a running container
ps	List containers
build	Build an image from a Dockerfile
pull	Download an image from a registry
push	Upload an image to a registry
images	List images
login	Authenticate to a registry
logout	Log out from a registry
search	Search Docker Hub for images
version	Show the Docker version information
info	Display system-wide information

Management Commands:

builder	Manage builds
buildx*	Docker Buildx
compose*	Docker Compose
container	Manage containers
context	Manage contexts
image	Manage images
manifest	Manage Docker image manifests and manifest lists
network	Manage networks
plugin	Manage plugins
system	Manage Docker
trust	Manage trust on Docker images
volume	Manage volumes

Swarm Commands:

Commands:

attach	Attach local standard input, output, and error streams to a running container
commit	Create a new image from a container's changes
cp	Copy files/folders between a container and the local filesystem
create	Create a new container
diff	Inspect changes to files or directories on a container's filesystem
events	Get real time events from the server
export	Export a container's filesystem as a tar archive
history	Show the history of an image
import	Import the contents from a tarball to create a filesystem image
inspect	Return low-level information on Docker objects
kill	Kill one or more running containers
load	Load an image from a tar archive or STDIN
logs	Fetch the logs of a container
pause	Pause all processes within one or more containers
port	List port mappings or a specific mapping for the container
rename	Rename a container
restart	Restart one or more containers
rm	Remove one or more containers
rmi	Remove one or more images
save	Save one or more images to a tar archive (streamed to STDOUT by default)
start	Start one or more stopped containers
stats	Display a live stream of container(s) resource usage statistics
stop	Stop one or more running containers
tag	Create a tag TARGET_IMAGE that refers to SOURCE_IMAGE
top	Display the running processes of a container
unpause	Unpause all processes within one or more containers
update	Update configuration of one or more containers
wait	Block until one or more containers stop, then print their exit codes

Global Options:

--config string Location of client config files (default "/home/lenovo/.docker")
-c, --context string Name of the context to use to connect to the daemon (overrides DOCKER_HOST env var and default context set with "docker context use")
-D, --debug Enable debug mode
-H, --host list Daemon socket to connect to
-l, --log-level string Set the logging level ("debug", "info", "warn", "error", "fatal") (default "info")

--tls Use TLS; implied by --tlsverify
--tlscacert string Trust certs signed only by this CA (default "/home/lenovo/.docker/ca.pem")
--tlscert string Path to TLS certificate file (default "/home/lenovo/.docker/cert.pem")
--tlskey string Path to TLS key file (default "/home/lenovo/.docker/key.pem")
--tlsverify Use TLS and verify the remote
-v, --version Print version information and quit

Run 'docker COMMAND --help' for more information on a command.

For more help on how to use Docker, head to <https://docs.docker.com/go/guides/>

[+] Building 26.8s (9/9) FINISHED docker:default

=> [internal] load build definition from Dockerfile 0.1s
=> => transferring dockerfile: 163B 0.0s
=> [internal] load metadata for docker.io/library/python:3.12-slim 3.5s
=> [internal] load .dockerignore 0.0s
=> => transferring context: 2B 0.0s

=> [1/4] FROM docker.io/library/python:3.12-

slim@sha256:85824326bc4ae27a1abb5bc0dd9e08847aa5fe73d8a 16.0s

=> => resolve docker.io/library/python:3.12-

slim@sha256:85824326bc4ae27a1abb5bc0dd9e08847aa5fe73d8af 0.0s

=> =>

sha256:a08295c4819f91ab6bbf02a62f2d70b87a3b76ac79fecf5260228f41dc4aa1c4

1.75kB / 1.75kB 0.0s

=> =>

sha256:3d1d8f39b99d1a90422205ebfa504475db03aabe1ae4f937fea5d4ed1076ad8
d

5.51kB / 5.51kB 0.0s

=> =>

sha256:8a628cdd7ccc83e90e5a95888fcb0ec24b991141176c515ad101f12d6433eb
96

28.23MB / 28.23MB 11.1s

=> =>

sha256:6cca951a45d16183402c0ea0d763285dff4024af76a75119fb405268bc35c35
9

3.51MB / 3.51MB 1.7s

=> =>

sha256:b3b4f30c6749b72e9b25139714a21ee8c3c5689b213a023c7b42be2634cfc3
20

13.66MB / 13.66MB 10.1s

=> =>

sha256:85824326bc4ae27a1abb5bc0dd9e08847aa5fe73d8afb593b1b45b7cb4180f5
7

9.13kB / 9.13kB 0.0s

=> =>

sha256:e9ddbe7a005f6950194f9ec89ccff2c86729491901f767d8b03f8062de3daf5
249B

/ 249B 2.2s

=> => extracting

sha256:8a628cdd7ccc83e90e5a95888fcb0ec24b991141176c515ad101f12d6433eb
96

2.7s

=> => extracting

sha256:6cca951a45d16183402c0ea0d763285dff4024af76a75119fb405268bc35c35

9

0.4s

=> => extracting

sha256:b3b4f30c6749b72e9b25139714a21ee8c3c5689b213a023c7b42be2634cfc3

20

1.2s

=> => extracting

sha256:e9ddbe7a005f6950194f9ec89ccff2c86729491901f767d8b03f8062de3daf5

0.0s

=> [internal] load build context 0.2s

=> => transferring context: 2.68MB 0.1s

=> [2/4] WORKDIR /app 0.4s

=> [3/4] COPY . . 0.1s

=> [4/4] RUN pip install --no-cache-dir -r requirements.txt 6.2s

=> exporting to image 0.3s

=> => exporting layers 0.3s

=> => writing image

sha256:5547b89ca47bf85b7d3628032df3aa4d136c96538dc66ad1cd677a45f2f06e4

2

0.0s

=> => naming to docker.io/library/image-resizer 0.0s

Image resized to (755, 1007) and saved to output/resized.jpg

MAIN.PY CODE :-

```
from PIL import Image
import os

input_path = 'images/input.jpg'
output_path = 'output/resized.jpg'

img = Image.open(input_path)

# Reduce resolution (example: 25% of original)
new_size = (img.width // 4, img.height // 4)
resized_img = img.resize(new_size)

# Save the reduced image
os.makedirs(os.path.dirname(output_path), exist_ok=True)
resized_img.save(output_path)

print(f"Image resized to {new_size} and saved to {output_path}")
```