Audit Report

Project Name: Qoala Devops Internship Challenge

Project Date: 1st November 2024

Issues and Resolution

Various issues were encountered while building the provided dockerfiles and running the images. These issues ranged from syntax issues and misconfigurations to typo errors. A summary of these issues is specified below: -

File	Type	Issue	Resolution	Remarks
Python-app Dockerfile	Syntax	“eight thousand”	8000	Expose and mapped ports can only be passed as numerical
	Typo	/appp	/app	Correct working directory should be provided so that the flask file can be copied to the correct directory
	Typo	appy.py	app.py	Correct name of the flask file should be written
	Typo	netiface	netifaces	Correct package name should be provided
	Typo	pythn	python	If wrong command is provided then that can result in errors while running the image
Nginx Dockerfile	Typo	latests	latest	Latests is not the correct word to obtain the latest version of the base image we are using
	Typo	nginix	nginx	Correct file name should be used while copying a file
	Typo	htmll	html	Either the html file command can be commented out or a test html file inside a proper directory can be created to run on the nginx server
	Syntax	“eighty”	80	Expose and mapped ports can only be passed as numerical
	Typo	daemon of	daemon off	Correct command should be used else it can result in failure while running the image
Nginx Configuration File	Typo	worker-process	worker-processes	Correct the worker process syntax to be used for specifying the number of CPUs to be used. The command should also be terminated by “;”
	Typo	worker-connection	worker-connections	Correct the worker corrections command so that max clients can be handled correctly
	Typo	typess	types	The file extension for the mime file should be correct
	Typo	default_typ	default_type	The command should be written correctly to remove the configuration issues

Docker-compose.yml File	Version Warning	version: '3.8'	-	The version should not be specified if different versions of the same file are specified across the different files
	Syntax	"eighty:80"	80:80	Expose and mapped ports can only be passed as numerical
	Typo	nginx.confi	nginx.conf	The correct filename should be provided in the volume as specified in the dockerfile
	Syntax	"eight thousand"	8000	Expose and mapped ports can only be passed as numerical
	Typo	bridg	bridge	The correct network driver to be used should be specified
	Additional Parameters	options: compelex_option: value	-	Additional properties are not allowed in networks

Although the given files will build without any issues, but some additional changes are made to the given files so that we obtain the desired results: -

- Although IP address and port number are present, but additional proxy header is set to specify a username to differentiate the change between the flask and the nginx server.
- The port is mapped and configured for the python-app too in the docker-compose yaml file so that the flask app can work correctly. There is also not need the expose the port as the port is already exposed in the dockerfile for the python-app.
- A folder for nginx-logs is made locally and a volume for the same is introduced for the nginx service to obtain the access and error logs for the nginx server.

The files will now run without any issues, but some changes are made to the app.py file too so that we obtain better output: -

- The output obtained for the shows the mac address as 00:00:00:00:00:00. So the logging library is introduced to log the interface and its respective mac address to identify the issue. It is observed that the interface used for the mac address was the loopback one (lo) which does not have a physical address of its own. So conditional statements are used to output only if any other interface is used and finally the physical address of the device (eth0) is obtained.
- The second change was made to log the usernames of the python and nginx server to log the server changes.

Outputs

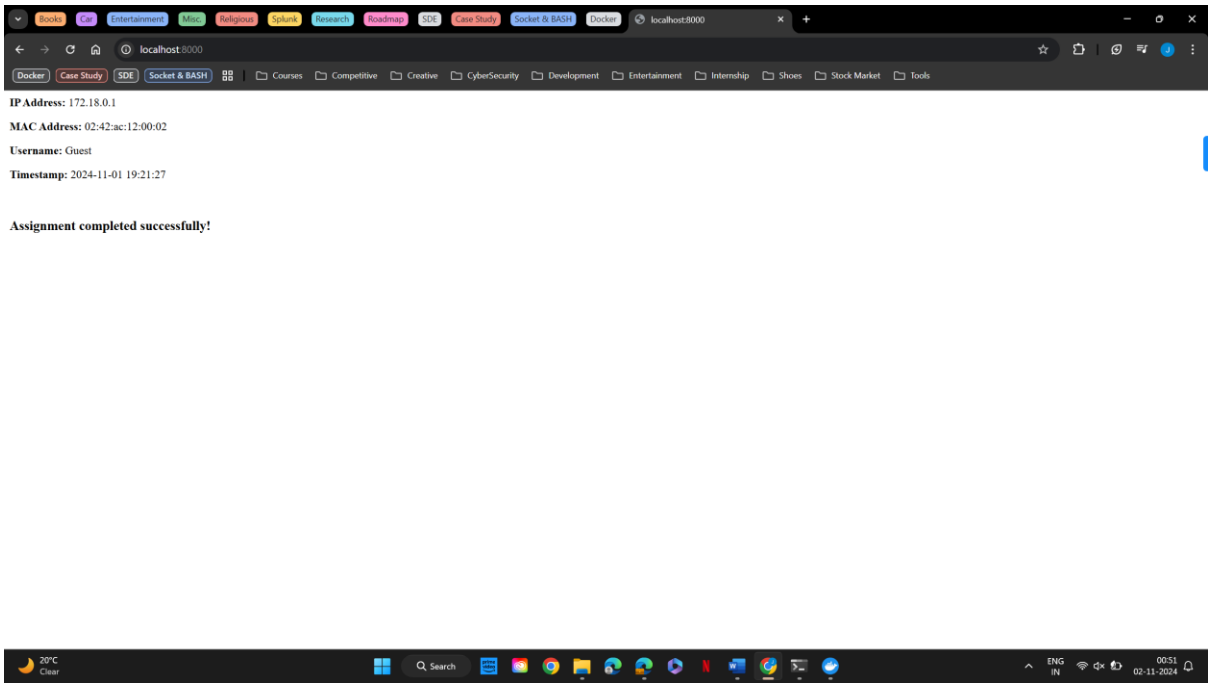

Fig. 1 – Running Application using Flask Web Framework on Port 8000

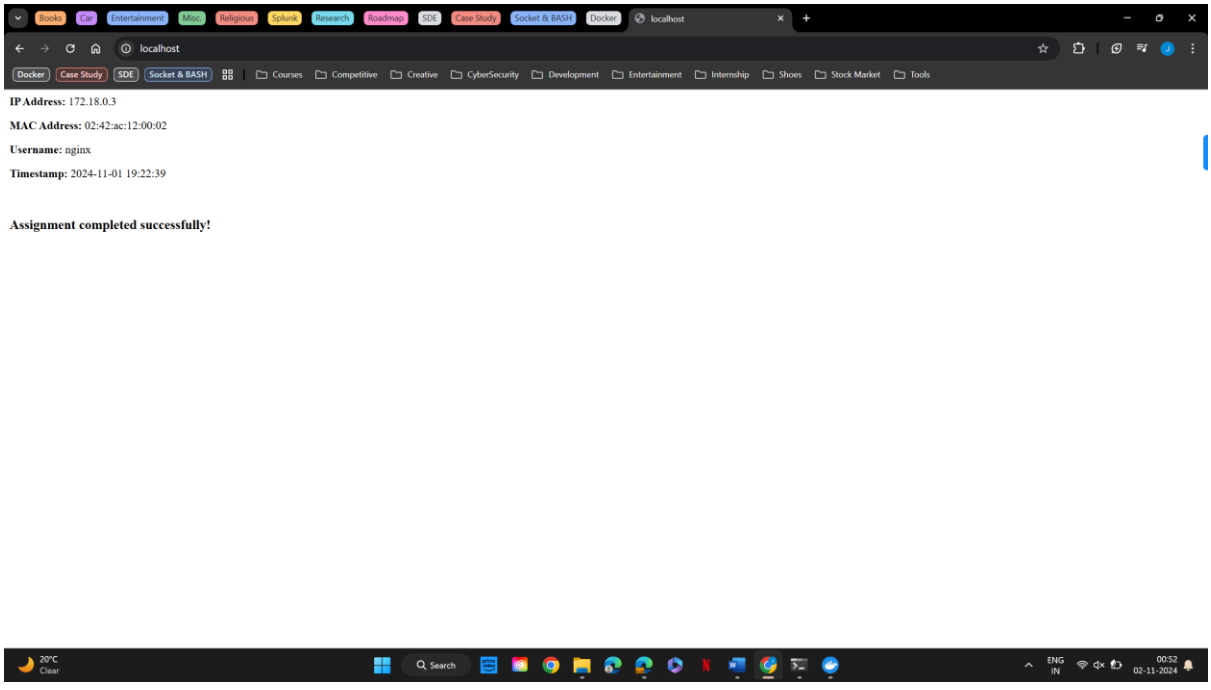

Fig. 2 – Running Application using Nginx Server on Port 80

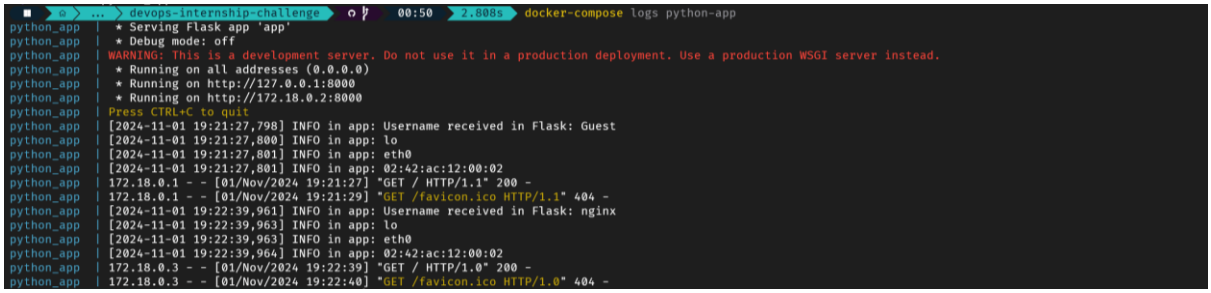

Fig. 3 – Docker-compose logs for the Python Application

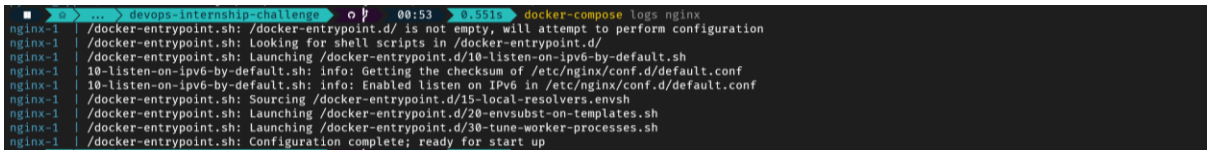

Fig. 4 – Docker-Compose logs for the Nginx Server

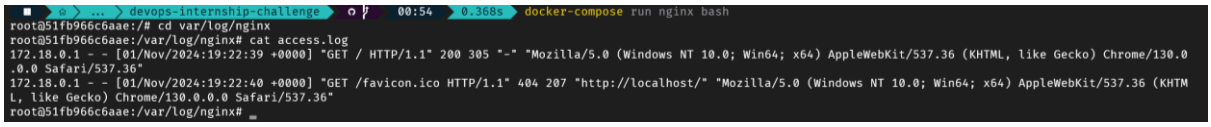

Fig. 5 – Access logs for the Nginx Server

BONUS

The application is deployed on AWS cloud server. It can be accessed using the following endpoints: -

Public IP: 3.110.256.207

Public DNS: ec2-3-110-156-207.ap-south-1.compute.amazonaws.com

Port: 80 (http)

Websites:

http://3.110.156.207/

OR

http://ec2-3-110-156-207.ap-south-1.compute.amazonaws.com/