Day 20 : Docker Cheat-sheet

This is #90DaysofDevops challenge under the guidance of Shubham Londhe sir.

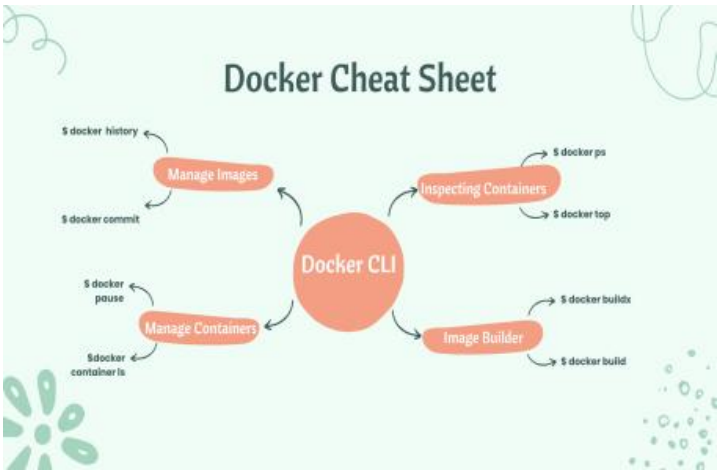
Day 20 TASK

check this for task:

https://github.com/LondheShubham153/90DaysOfDevOps/blob/master/2023/day20/tasks.md

Docker

Docker is a platform that enables developers to easily build, ship, and run applications as containers. Containers are lightweight, portable, and self-contained environments that can run on any platform. Docker uses a client-server architecture, with the Docker client communicating with the Docker daemon to build, run, and manage containers.



Docker makes it easy to package an application and all its dependencies into a single container, ensuring that it will run consistently across different environments. It also allows for easy scaling, as multiple instances of the same container can be run simultaneously to handle increased traffic.

Build

The build command is used for building images from a Docker file.

Commands

- To build an image from the Docker file and tag it:

```
docker build -t myapp :1.0
```

- To list all the images that are locally stored:

```
docker images
```

- To delete an image from the Docker Store:

```
docker rmi alpine: 3.3
```

Run

The run command is used for creating a container from a specified image.

Commands

- To create and run a command:

```
docker run --name container_name docker_image
```

Flags used:

- -d: To detach a container on start
- -rm: To remove a container once it stops
- -p: To publish the host IP and the host port to the container port
- -v: To define and share the volume across containers
- –read-only: To set to the read-only permission

Ship

Docker gives us the capability of shipping our application container anywhere, on any platform.

Commands

- To pull an image from the registry:

```
Docker pull alpine:3.4
```

- To retag a local image with a new image name:

```
Docker tag alpine:3.4 myrepo/ myalpine:3.4
```

- To log in to the registry:

```
Docker login my.registry.com:8000
```

- To push an image to the registry:

```
Docker push myrepo/ myalpine:3.4
```

Clean up

Deleting the unused containers and images.

Commands

- To clean an unused image:

```
Docker image prune
```

- To remove an image that is not used in a container:

```
Docker image prune -a
```

- To prune the entire system:

```
Docker system prune
```

- To leave a swarm:

```
Docker swarm leave
```

- To remove a swarm:

```
Docker stack rm stack_name
```

- To kill all running containers:

```
Docker kill $ (docker ps -q)
```

- To delete all stopped containers:

```
docker rm $(docker ps -a -q)
```

- To delete all images:

```
docker rmi $(docker images -q)
```

Interaction with a Container

Commands

- Pull an image from a Docker registry:

```
docker pull <image-name>
```

- Run a container from an image:

```
docker run <image-name>
```

- Run a container in detached mode:

```
docker run -d <image-name>
```

- Run a container with a specific name:

```
docker run --name <container-name> <image-name>
```

- Bind mount a directory from the host machine to the container:

```
docker run -v <host-directory>:<container-directory> <image-name>
```

- Expose a port from the container to the host machine:

```
docker run -p <host-port>:<container-port> <image-name>
```

If this post was helpful, please do follow and click the clap

_Thank you for reading

_Rajani