**Docker for Application Packaging**

**Summary**

The appearance of containers enabled organizations to ship products using a lightweight mechanism, that would make the most of available infrastructure. There are plenty of tools used to containerize services, however, Docker has set the industry standards for many years.

To containerize an application using Docker, 3 main components are distinguished: Dockerfiles, Docker images, and Docker registries. Let's explore each component in a bit more detail!

**Dockerfile**

A Dockerfile is a set of instructions used to create a Docker image. Each instruction is an operation used to package the application, such as installing dependencies, compile the code, or impersonate a specific user. A Docker image is composed of multiple layers, and each layer is represented by an instruction in the Dockerfile. All layers are cached and if an instruction is modified, then during the build process only the changed layer will be rebuild. As a result, building a Docker image using a Dockerfile is a lightweight and quick process.

To construct a Dockerfile, it is necessary to use the pre-defined instructions, such as:

To construct a Dockerfile, it is necessary to use the pre-defined instructions, such as:

FROM - to set the base image

RUN - to execute a command

COPY & ADD - to copy files from host to the container

CMD - to set the default command to execute when the container starts

EXPOSE - to expose an application port

Below is an example of a Dockerfile that targets to package a Python hello-world application:

# set the base image. Since we're running

# a Python application a Python base image is used

FROM python:3.8

# set a key-value label for the Docker image

LABEL maintainer="Katie Gamanji"

# copy files from the host to the container filesystem.

# For example, all the files in the current directory

# to the `/app` directory in the container

COPY . /app

# defines the working directory within the container

WORKDIR /app

# run commands within the container.

# For example, invoke a pip command

# to install dependencies defined in the requirements.txt file.

RUN pip install -r requirements.txt

# provide a command to run on container start.

# For example, start the `app.py` application.

CMD [ "python", "app.py" ]

Once a Dockerfile is constructed, these instructions are used to build a **Docker image**. A Docker image is a read-only template that enables the creation of a runnable instance of an application. In a nutshell, a Docker image provides the execution environment for an application, including any essential code, config files, and dependencies.

A Docker image can be built from an existing Dockerfile using the docker build command. Below is the syntax for this command: