**How do you structure your Go projects? Explain the advantages and disadvantages?**

**What is Docker?**

* Docker is a tool designed to make it easier to create, deploy, and run applications by using containers. Containers allow a developer to package up an application with all of the parts it needs, such as libraries and other dependencies, and ship it all out as one package

**What is docker file?**

* Docker uses Docker file to build the image itself. This mean A Docker File is a simple text file with instructions on how to build your images.

**What is docker compose?**

* Compose is a tool for defining and running multi-container Docker applications. With Compose, you use a YAML file to configure your application’s services. Then, with a single command, you create and start all the services from your configuration

Using Compose is basically a three-step process.

* Define your app's environment with a Dockerfile so it can be reproduced anywhere.
* Define the services that make up your app in docker-compose.yml so they can be run together in an isolated environment.
* Lastly, run docker-compose up and Compose will start and run your entire app

**FROM**

The base image for building a new image. This command must be on top of the dockerfile

**MAINTAINER**

This is optional, it contains the name of the maintainer of the image.

**RUN**

Used to execute a command during the build process of the docker image.

**ADD**

Copy a file from the host machine to the new docker image. There is an option to use an URL for the file, docker will then download that file to the destination directory.

**ENV**

Define an environment variable.

**CMD**

Used for executing commands when we build a new container from the docker image.

**ENTRYPOINT**

Define the default command that will be executed when the container is running.

**WORKDIR**

This is directive for CMD command to be executed.

**USER**

Set the user or UID for the container created with the image.

**VOLUME**

Enable access/linked directory between the container and the host machine.

**EXPOSE**

The EXPOSE command tells Docker that the image will listen on the specified ports when running

**Build image từ Dockerfile :**

Docker build -t my-go-app .

docker run -p 8080:8081 -it my-go-app

* p 8080:8081 - This exposes our application which is running on port 8081 within our container on http://localhost:8080 on our local machine.
* -it - This flag specifies that we want to run this image in interactive mode with a tty for this container process.
* my-go-app - This is the name of the image that we want to run in a container.

upload image: [DOCKER HUB](https://registry.hub.docker.com/?source=post_page---------------------------)

docker-compose down –volumes

Để kết thúc các services đang chạy, sử dụng lệnh:

$ docker-compose stop

Và để xóa hoàn toàn container và data volume sử dụng bởi Redis container:

$ docker-compose down --volumes

[Docker Hub](https://hub.docker.com/), the container registry from which we pulled the golang image earlier, offers a feature called [Automated Builds](http://docs.docker.com/docker-hub/builds/) that builds images from a GitHub or BitBucket repository.