Title: Using docker image to store data files
Post Body:

I am new to docker and have just started exploring it. What I am trying to achieve is to publish a Docker image, which would just have some configuration files. These files would be shared across multiple projects.

So the image I want to publish is 'my-config-image'.

This is how the directory structure is:

```
/my-config-settings       /folder1/multiple files       /folder2/multiple files       file1.txt       file2.txt
```

After reading online and going through the docker quick start I have got some understanding.

I decided to use Create a simple base image using `scratch`. referred from (https://docs.docker.com/engine/userguide/eng-image/baseimages/)

This is how my Dockerfile looks.

```
FROM scratch ADD folder1 / folder2 /
```

This is the command I am using to publish an image.

```
/my-config-settings/docker build -t my-config-image . Sending build context to Docker daemon  5.12 kB Step 1 : FROM scratch  -
```

When I want to check if the image was created locally I am confirming it in this way by executing

```
docker images   REPOSITORY                                TAG              IMAGE ID          CREATED
```

My main query is how can I check if this is working, Also did this add the file1.txt and file2.txt into the image, if not how do I specify to add these files in the image.

Is there a way I can add all the files and directories recursively into the image I am trying to create?

Also how to I access this image, means how can I check if this image actually has all the folders and files. Is there a way to cd into this?

I am okay with suggestions and looking at another docker approach, if what I am trying to do doesn't make sense.

Thank you for reading.

Accepted Answer: None
Highest Rated Answer:

You're trying to address a configuration management problem with an application management approach. For shared configurations, you generally want to take the 'centralized location approach'. Basic examples of this would be a git repository or an S3 bucket. Both solutions have native document storage and can be appropriately shared between services with fine-grained access control.

A docker image isn't the docker approach to store/share configuration. By having an image, you can basically do two things:
1. Base other images off your initial image
2. Run a container

Given that your image here is just `Scratch` and files, there are no executables and nothing to run. The hello world example copied a directory with an actual script in it, which is why they ended up with a runnable container.

As a base image, configuration doesn't make sense coming before things like dependencies.

If you really want to use a docker tool for this, you're looking for docker volumes. That can persist a file system and is easy to share around different containers.
https://docs.docker.com/engine/userguide/containers/dockervolumes/