* Initially, the Docker container will be in the **created** state.
* Then the Docker container goes into the running state when the Docker **run** command is used.
* The Docker **kill** command is used to kill an existing Docker container.
* The Docker **pause** command is used to pause an existing Docker container.
* The Docker **stop** command is used to pause an existing Docker container.
* The Docker **run** command is used to put a container back from a **stopped** state to a **running** state.

#This is a sample Image

FROM gudditi/webapp:v4

MAINTAINER gudditinaganjaneyulu@gmail.com

RUN service nginx start

CMD [“echo”,”Image created”]

The following points need to be noted about the above file −

* The first line "#This is a sample Image" is a comment. You can add comments to the Docker File with the help of the **#** command
* The next line has to start with the **FROM** keyword. It tells docker, from which base image you want to base your image from. In our example, we are creating an image from the **ubuntu** image.
* The next command is the person who is going to maintain this image. Here you specify the **MAINTAINER** keyword and just mention the email ID.
* The **RUN** command is used to run instructions against the image. In our case, we first update our Ubuntu system and then install the nginx server on our **ubuntu** image.
* The last command is used to display a message to the user.

docker build -t name:resume .

Options

* **-t** − is to mention a tag to the image
* **ImageName** − This is the name you want to give to your image.
* **TagName** − This is the tag you want to give to your image.
* **.**  − The directory where the Docker File is present.

docker run -dit -v resume:/usr/share/nginx/html -p 80:80 gudditi/webapp:resume /bin/bash

ghp\_eknqgKFLPN55jmt1Jh8pojKz0IOQ8z2g7OrR

Usage: docker network connect [OPTIONS] NETWORK CONTAINER

<https://kapeli.com/cheat_sheets/Dockerfile.docset/Contents/Resources/Documents/index>

web:

image: gudditi/webapp:resume

volumes:

- resume:/usr/share/nginx/html

ports:

- "80:80"

environment:

- NGINX\_PORT=80