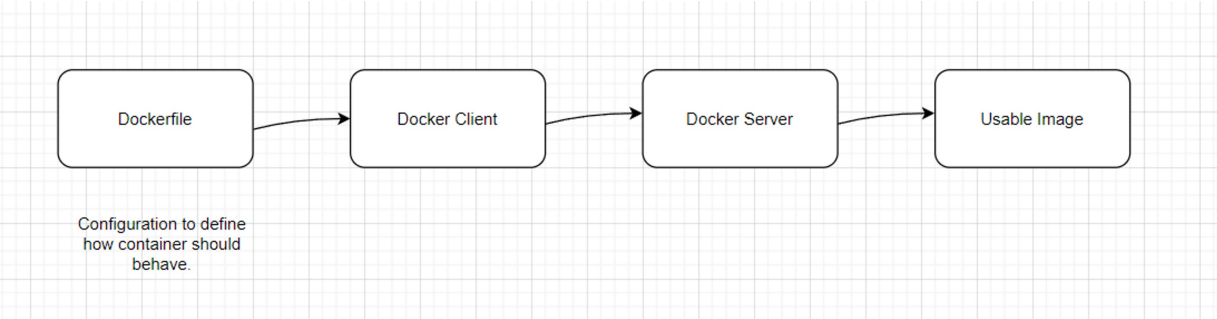
Building Custom Images through Docker Server

13 December 2021 17:49

Creating Docker Image:



Flow of creating a Dockefile:

1. Specify a base Image.
2. Run some command to install additional program.
3. Specify a command to run on container startup.

Creating Custom DockerFile:

Goal: Creating a Image that runs a Redis Server.

1. Create a Dockerfile.

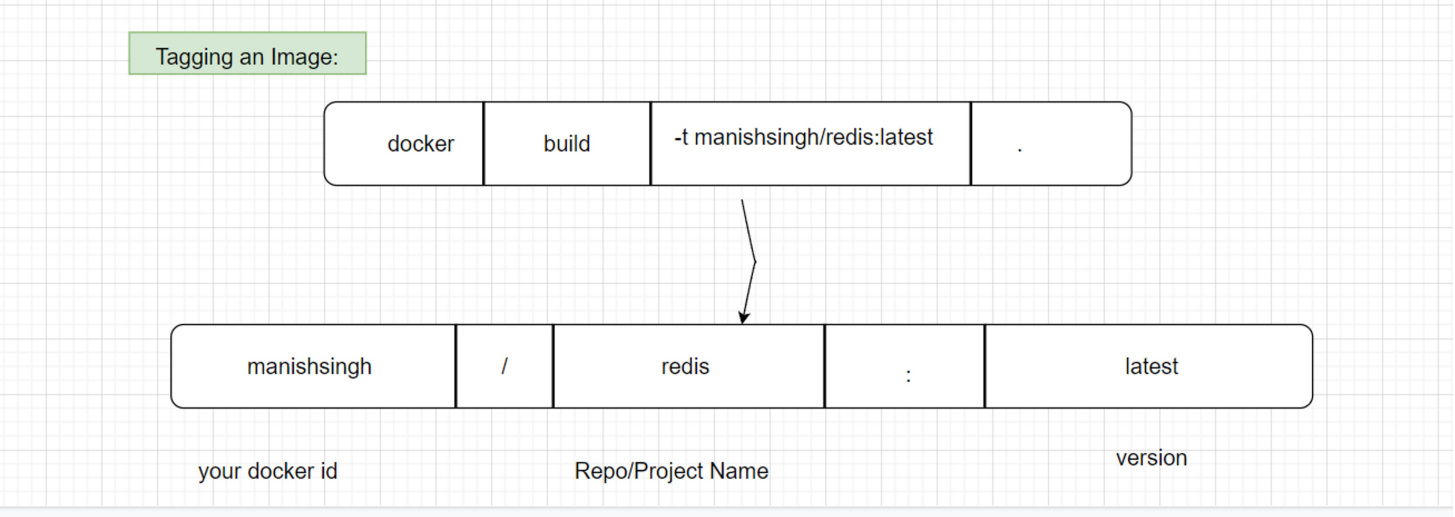
```
# Use an existing docker image as Base
FROM alpine
# Download and install a dependency
RUN apk add --update redis
# Tell the image what to do when it starts
#as a container
CMD [ "redis-server" ]
```

2. Navigate to the folder where you have saved the Dockerfile and run the below command.

docker build .

```
# Series of Steps happened in background:
# 1. Download alpine image
# ----------------------------------------
# 2. Get Image from the previous Steps
# 3. Create a container out of it.
# 4. Run "apk add --update redis" in it.
# 5. Take snapshot of that container's FS.
# 6. Shutdown that temporary container.
# 7. Get Image ready for the next instruction.
# ----------------------------------------
# 8.  Get image from the last step.
# 9. Create a container out of it.
# 10. Tell container it should run 'redis-server' when started.
# 11. Shutdown that temporary container.
# 12. Get Image ready for next instruction.
# 13. No more steps !
# 14. Output is the image generated from previous step.
```

Tag an Image:



we will be running a command to create a new image using docker commit with this command:

docker commit -c 'CMD ["redis-server"]' CONTAINERID

If you are a Windows user you may get an error like "**/bin/sh: [redis-server]: not found**" or **"No Such Container"**

Instead, try running the command like this:

docker commit -c **"CMD 'redis-server'"** CONTAINERID