Connecting to container shell

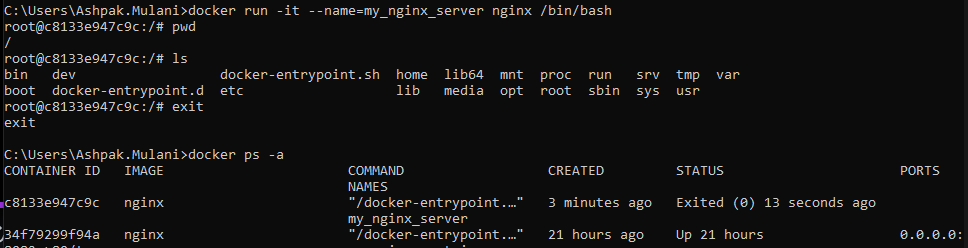
#### Foreground container

Connecting to container shell in production is not really common but sometime we need to connect and run some commands inside container manually and see what is happening.

In below command we are running nginx container in interactive nondetached mode and we are sending it a command to run bash so that we have some prompt to work inside container.

docker run -it --name=my\_nginx\_server nginx /bin/bash

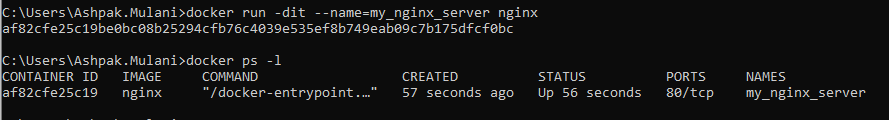
as shown below container gave us bash prompt to use where we can run commands inside container. Once we exit from container using exit command then container stops execution.



#### Background container

Above example shows that we are running container in foreground and not in detached mode in background. What if we want to connect or run command in container running in background??

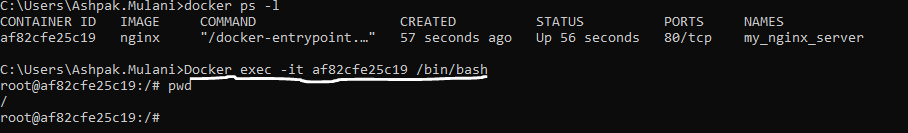
Let’s start the container in detached mode first. Since we are starting container in detached mode it doesn’t make sense to give it a command to run shell in same docker run command since we cant see it.



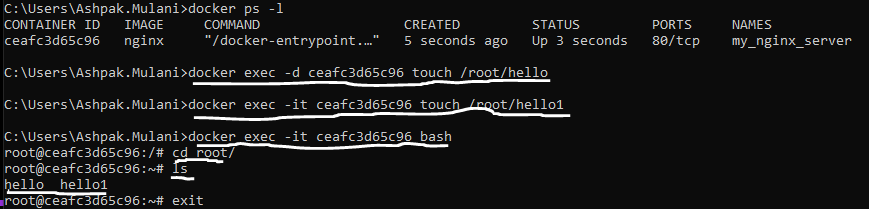
see it Now let’s execute a shell command on running container and ask docker to run shell prompt interactively using -it

Docker exec -it af82cfe25c19 /bin/bash

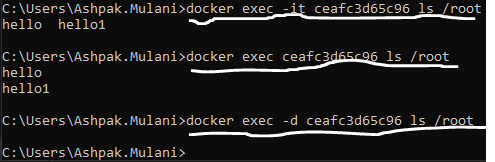
We get the bash interactive shell from inside docker as shown below



shown below instead of bin/bash in above command we can give any command what we want to execute like Docker exec -it af82cfe25c19 touch /root/hello this will create hello file in root directory. One thing to note here even if we run this command (touch) in detached (-d) or in interactive (-it) mode we won’t get any shell prompt in interactive mode because command just does its job and close. If we want to open shell prompt then we can run it though ‘bash’ or ‘/bin/bash’ command

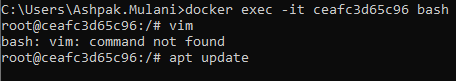


Below commands are just to show that we can get docker output in return if commands what we are providing for execution to container returns anything. Example as shown below if we provide ls /root command to execute to docker then it returns list of files from then folder to us exception is if we use -d (detached) option then we wont get any output as commands will be returning output in background and we can’t see it.



Install requires apps in docker

Docker images contains very minimal resources, so we mostly need to install our required apps and services on base images. Ex. as shown below container doesn’t even have vim editor, so we can install it using ‘apt’ linux installer since it’s a linux based image.



we can run ‘apt update’ command to update the installer and then ‘apt install -y vim’ installs the vim editor in container. *Note,* this change is being done in container only and not in image so if we run new container using image it won’t have these changes.

If we have to change the image and create new image then we can use docker build with DockerFile as discussed in section 2.