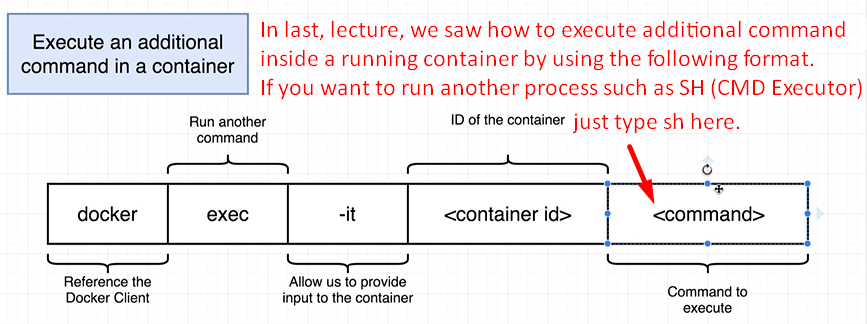
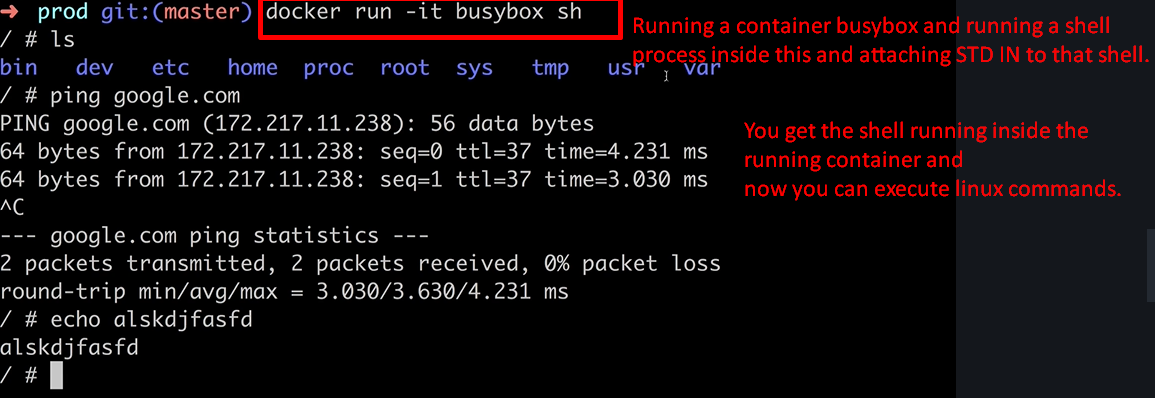
1. 
2. Start a new container out of busybox image. Run sh program inside that container. Then attach STDIN to that program.
3. We can run “docker run” along with –it flag that start up a shell immediately when a container first starts up.   
   Of course, if we start up a shell right when the container first starts that is going to displace or keep any other typical or default command from running. But sometimes, it’s quite useful to get an empty container with a shell inside of it to be able to poke around and not having any process running inside of it.
4. Command 🡺 docker run –it boxybox sh 🡸 Creates container without having any process. You might be thinking that but we’re creating the busybox container, so it’s process itself. Yes it is but we’re overriding the default initial command by specifying sh. We’re saying when container is created and at the time of running it, instead of executing its default command, execute my command sh.  
   Similarly, if running command 🡺 docker run –it redis sh🡸 It will also create a container but inside it, it will not run redis as we’re overriding the initial command with sh that is run to execute redis server (meaning default command to run redis server is overridden by my command sh).
5. The downside to using “docker run –it run sh”is that chances are you’re not going to be running any other process.   
   It’s little bit more common that you’re going to want to start up your container with a primary process of like maybe your web server or whatever it might be and then attach to it a running shell by using command 🡺 “docker exec –it <container\_id> sh”🡸 instead of sh, you can run some other process.