Summary

Session No – 13

- **pgrep** is a command in Linux to check the process ID of the running program

```
[root@localhost ~]# pgrep  firefox
[root@localhost ~]# pgrep  firefox
2771
[root@localhost ~]#
```

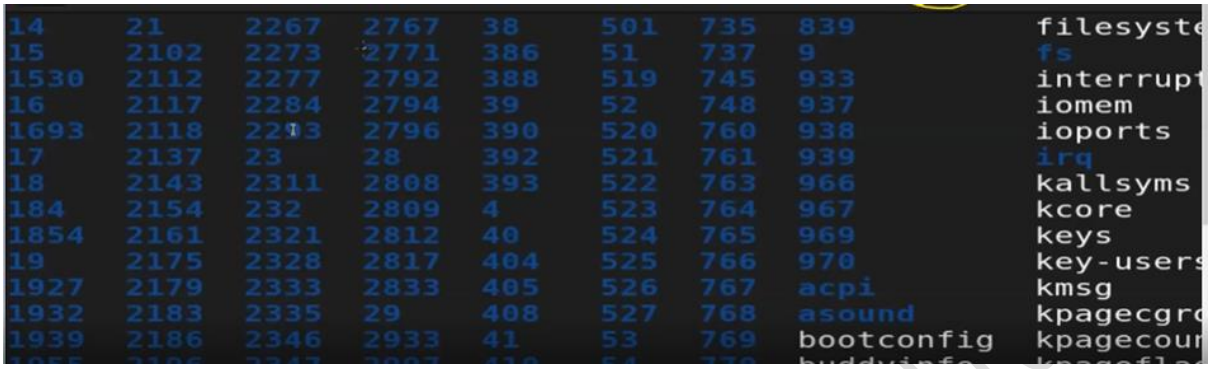
- **ps -aux** command will tell us all the processes running in the system

```
root        2179  0.0  0.0 450028  7152 ?        Ssl  21:03   0:00 /usr/libexec/gvfs-
root        2183  0.0  0.0 448260  6208 ?        Ssl  21:03   0:00 /usr/libexec/gvfs-
root        2186  0.0  0.4 894444 36852 ?        Ssl  21:03   0:00 /usr/libexec/goa-d
root        2196  0.0  0.1 530984  9312 ?        Ssl  21:03   0:00 /usr/libexec/goa-i
root        2202  0.1  0.4 270856 40344 ?        Ss   21:03   0:00 /usr/libexec/sssd/
root        2206  0.2  0.6 1144968 49868 ?       Ssl  21:03   0:00 /usr/libexec/evolu
root        2209  0.0  0.3 2796408 26376 ?       Ssl  21:03   0:00 /usr/bin/gjs /usr/
root        2210  0.0  0.0 161696  7208 ?        Ssl  21:03   0:00 /usr/libexec/at-sp
root        2221  0.0  0.1 522160  8472 ?        Ssl  21:03   0:00 /usr/libexec/gsd-a
root        2222  0.1  0.3 748880 31968 ?        Ssl  21:03   0:00 /usr/libexec/gsd-c
root        2235  0.0  0.2 594360 18652 ?        Ssl  21:03   0:00 /usr/libexec/gsd-d
root        2238  0.0  0.0 524528  7228 ?        Ssl  21:03   0:00 /usr/libexec/gsd-h
root        2254  0.1  0.4 599720 32640 ?        Ssl  21:03   0:00 /usr/libexec/gsd-k
root        2263  0.1  0.4 744160 33428 ?        Ssl  21:03   0:00 /usr/libexec/gsd-m
```

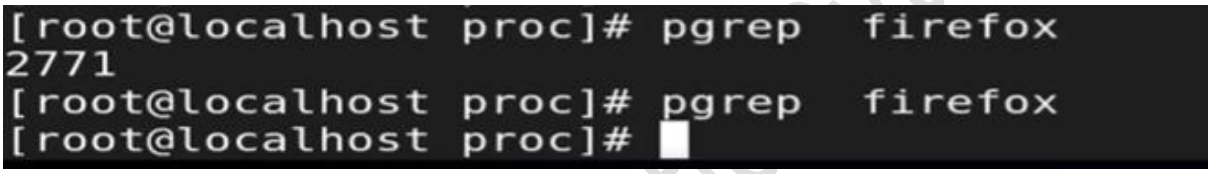
- In one operating system, there are lots of processes running and inside the operating system, there is one main program known as Kernel. The duty of the kernel is to start the process and give the ram
- Process ID at the memory level or physical level is just a directory
- Whenever we run a program it starts the process and behind the scene, the kernel gives every process a unique id which means go to the ram and create one directory, and anything related to the process is stored in that directory
- The entire information of the ram is linked to the **/proc** directory

```
[root@localhost ~]#
[root@localhost ~]#
[root@localhost ~]# cd /proc/
[root@localhost proc]#
```

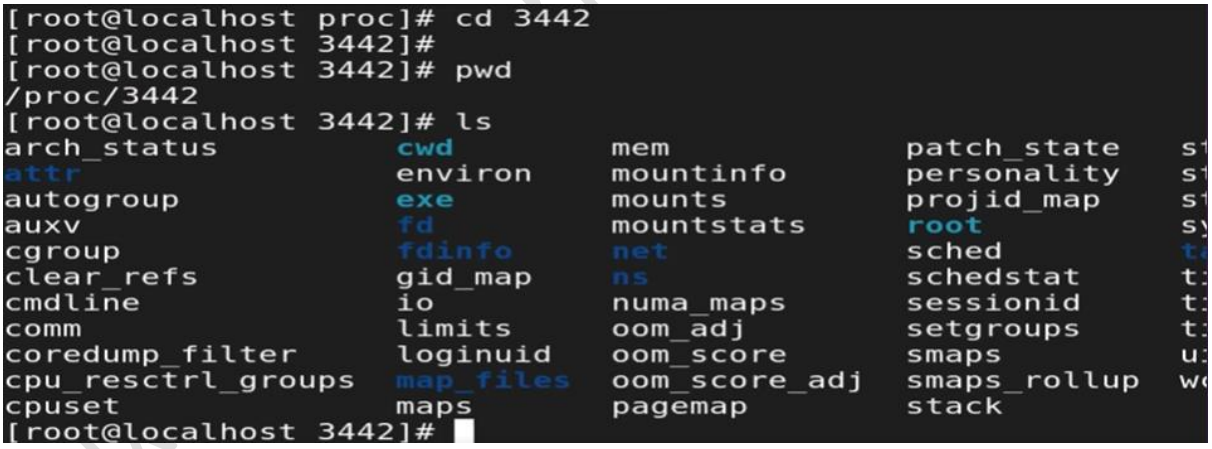
- In the /proc directory whatever we see in blue colour is a folder or directory and anything in white colour is a file



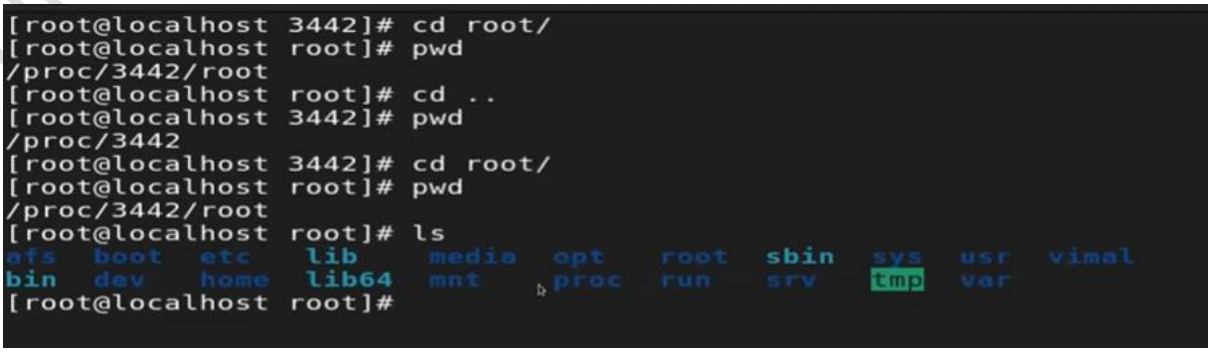
- As soon as we stop the program the process id folder will be deleted from the /proc directory



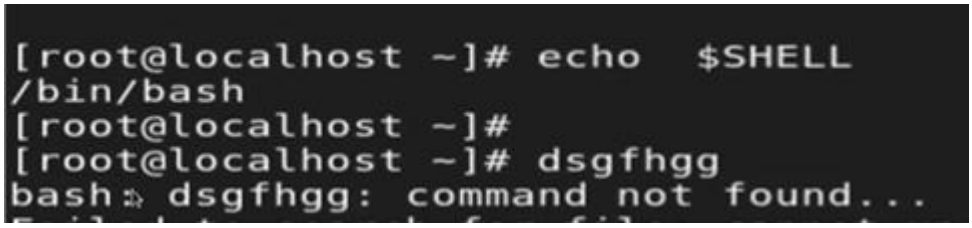
- This means the PID is just a folder or directory in the ram
- All the information about the process in the RAM we can find inside the process ID folder in the " / " proc directory



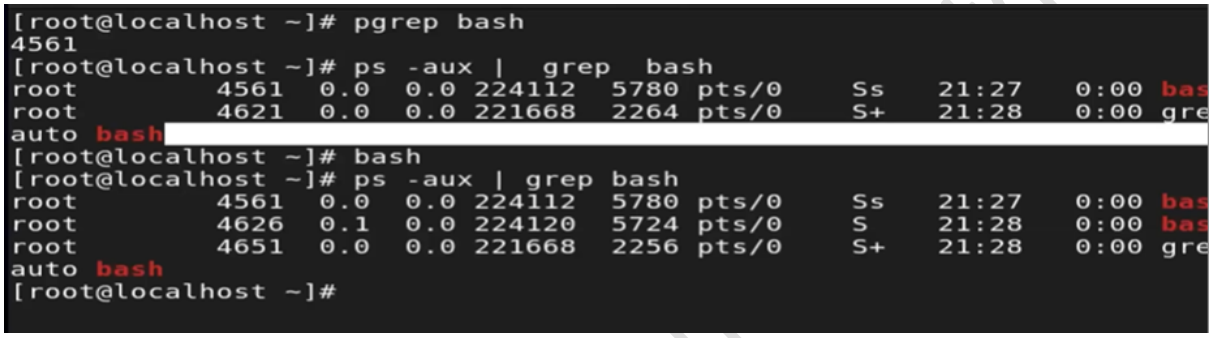
- **root** file is given to all the processes and it is linked with the " **/** " drive. It means any process has access to / drive



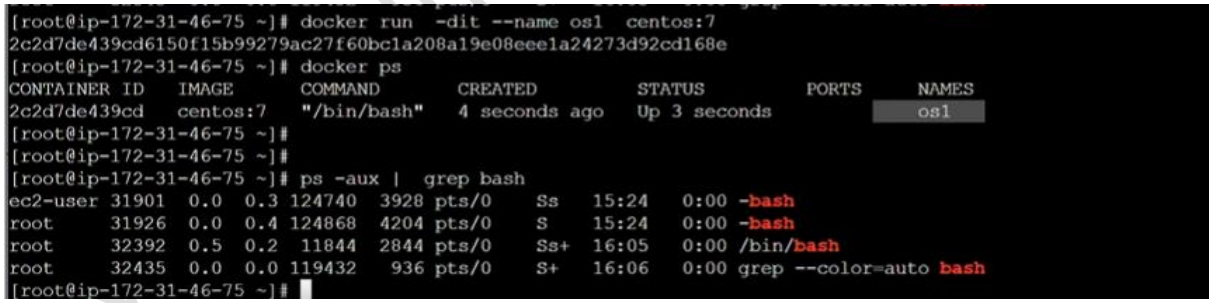
- In the command prompt, we can run the commands because the bash shell or bash program is running



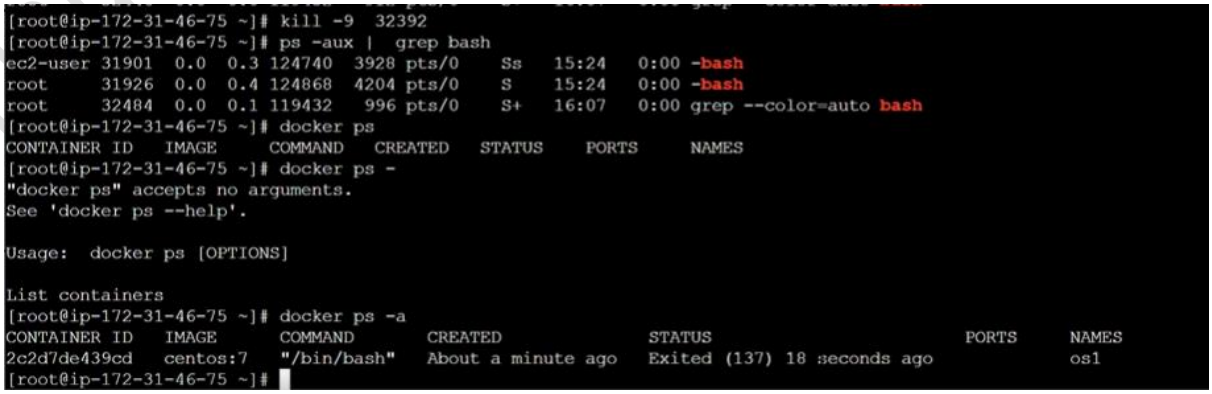
- We can see the process ID of the bash & we can run the two different processes ie bash process in the bash



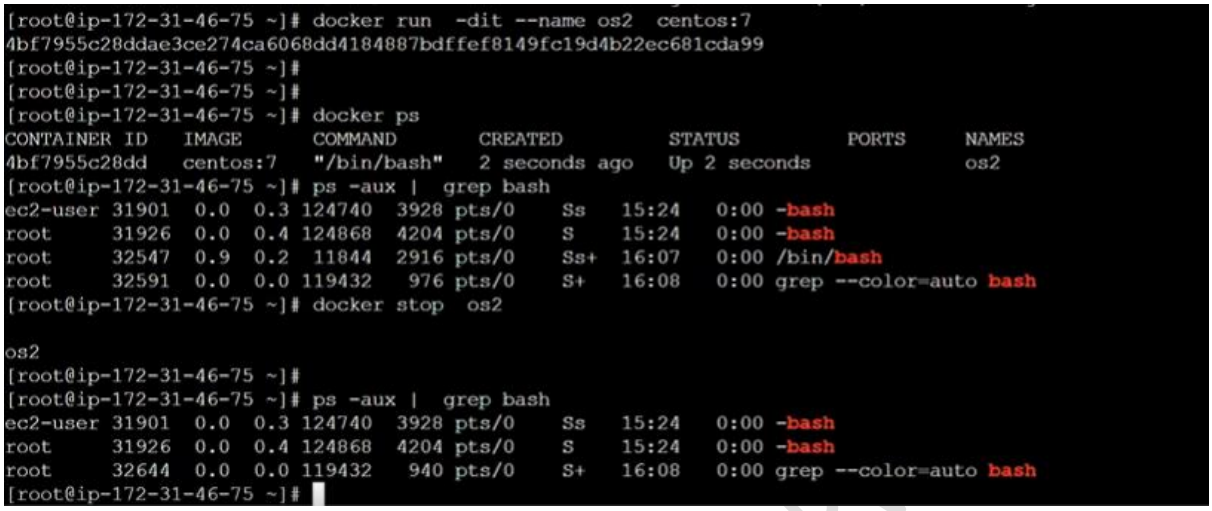
- Users can only interact with the process
- Inside one OS we can run multiple processes & it has the capability to launch the process inside the process
- Docker behind the scene launching the process in that we can launch multiple processes that is the reason docker is superfast



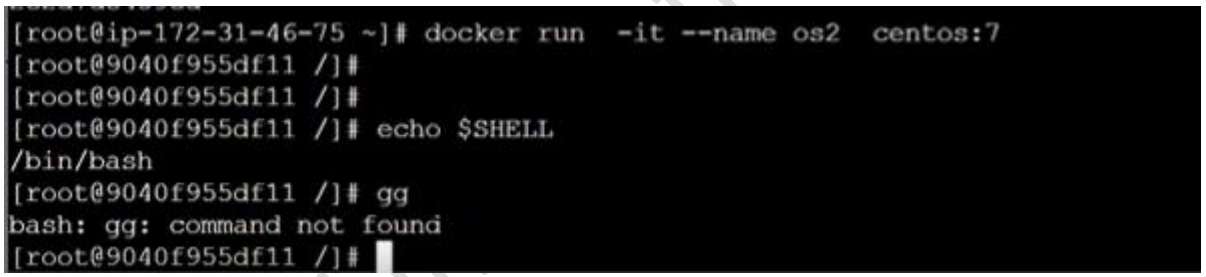
- If we kill the process the container will stop



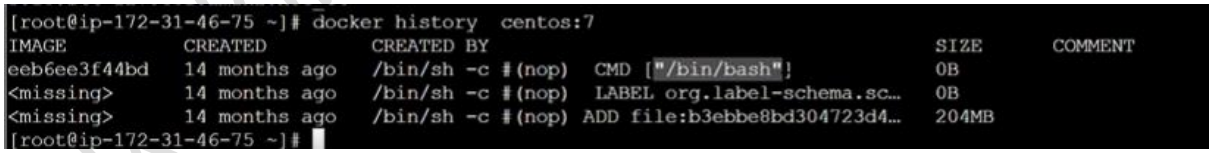
- If we stop the container behind the scene, it is stopping the process only



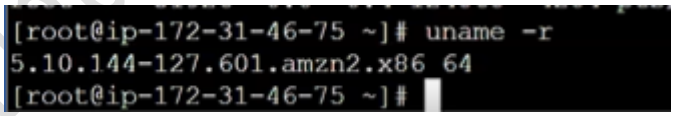
- Whenever we launch the container in interactive mode docker gives us the terminal because behind the scene it is running the bash shell



- In the image with the help of the CMD or ENTRYPOINT keyword, we can decide which process to run & the entire container will be the process

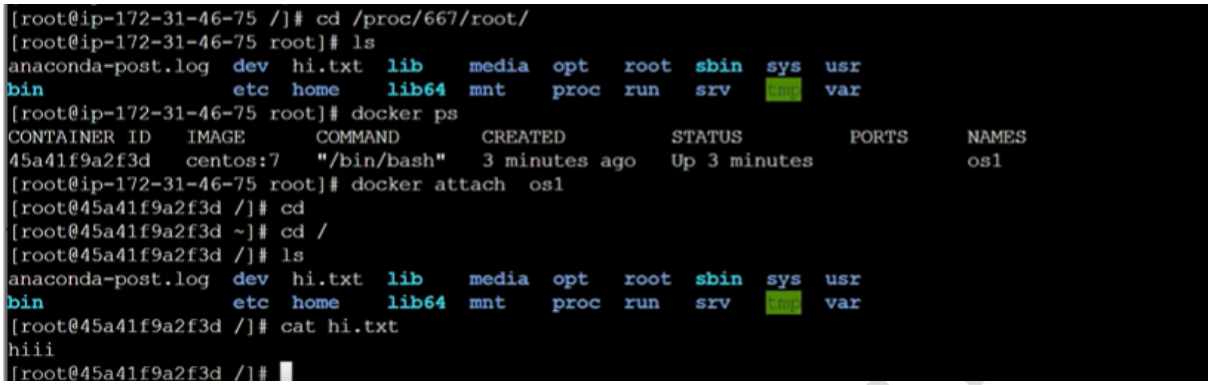


- Kernel version 2.6 and above supports nested process concept



- Whenever we start the process, every process has a unique id and a unique folder known as root which is linked to the / drive

- In the container, the root directory is linked with the " / " drive and the data in the / drive is coming from the image, not from the base system



- Data in / drive is coming from the image that is the reason if a container is running & we tried to remove the container, it gives the error