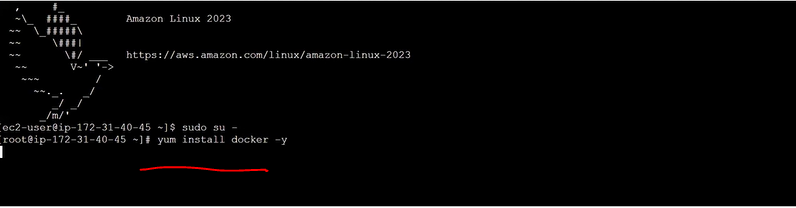
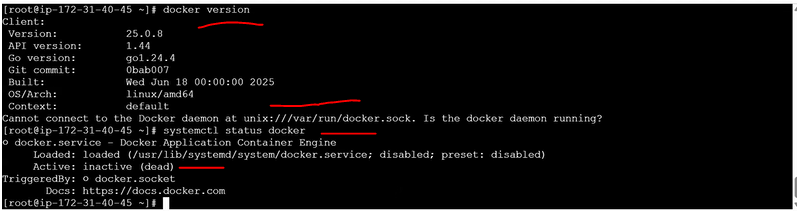
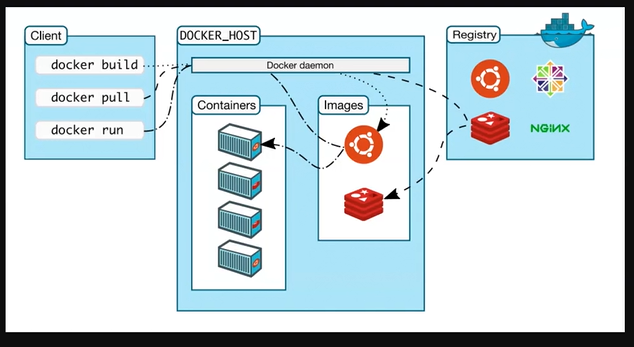
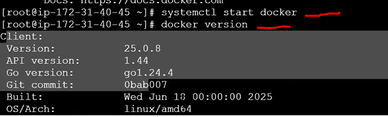
1. Create an EC2 Instance than allow SG Inbound Rule for all traffic. Than switch to root and install Docker as below command.



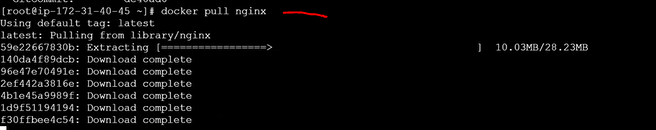
1. To check docker version:- docker --version
2. When we will run docker version than Than it will show docker client details and it will show as docker demon is inactive means docker server is inactive. Than we will active first than we will pull image. When we will pull image or build as docker command than it will check first if docker demon server is connected or not . If its not connected than it will fail otherwise it will connect to decker demon and check is there local image is there if not than it will connect to docker hub for pull the image and save in local than send that image to docker client.







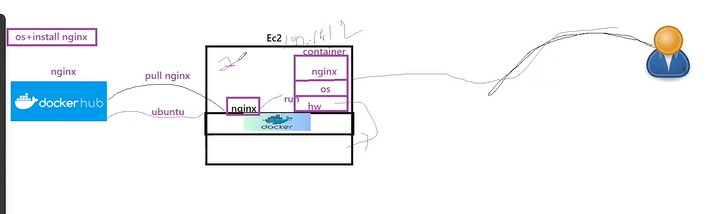
1. Now if we want to pull the any image from docker hub. Than we can.



1. If you will not specify any version than it will pull the latest version. If you want to check than you can go to docker hub and check.
2. How to Check what are the docker images are present.

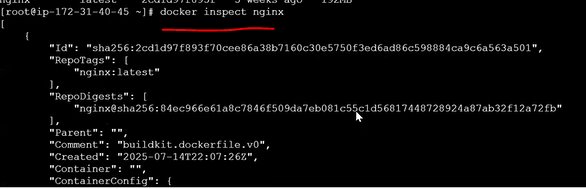


1. How to create a container format so that we will run the image.

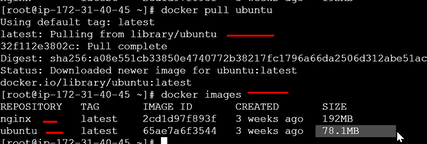


When we will run an image than its automatically run inside container.

1. To know more information on the image.



1. Now I am pulling another image from docker hub and checking how many images are.



If I will take nginx than I can run the image/process but if I took Ubuntu than its OS. We can’t run that iamge.

1. When we will run an image than it will run as process. Than its 2 commands for run as **it and dt .**
2. **It(Interative Terminal) :-** Mean interactive mode which connect direct to the container. It called foreground process. Here it will run than once we give exist than it will detached.
3. **Dt(Detach Terminal):**- Means Its not interactive mode. Its not connect directly to the container. Its run in background. It called a background process. Container run than detached it. But process can run and not stop it.
4. **/bin/bash:**- It’s a process. Why we are giving because if a small processing running in container than container will not stop. If no process running in container than container will exis. That’s why we are giving /bin/bash in run command to keep a continuous process in container so that it will not stop. Once we started than its came inside into to image as container. Now I am inside the Ubuntu container.



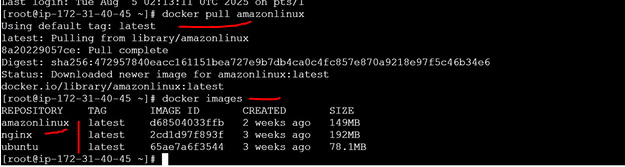
1. If we will exit than it will exist from container. Also process is stopped. Which process is stoped as /bin/bash



1. To check if any containers are running or not.



1. Now I am taking amazon linux image.



1. To check how many process are running in EC2 linux not in docker linux.



1. Now see In EC2 amazon linux 119 process are running in background but if we will login into docker linux image and see how many process are running.
2. Login into amazonlinux in docker.



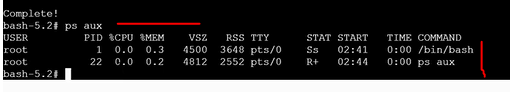
Now I am inside amazon linux container. Than what is the difference. In EC2 Linux by default so many dependency and process are running in back ground but in docker linux image we don’t have any dependency so we have full control. If we will check how many process are running in docker amazon linux than it will throw error means **ps** command dependency not present. Mean no dependency and no process.



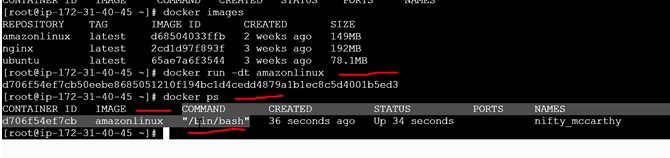
1. Now we will install the **ps** dependency for to check how many process are running.



1. Once its installed then we will execute **ps** command than it will display.

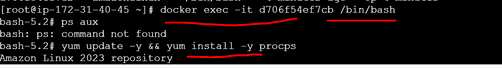


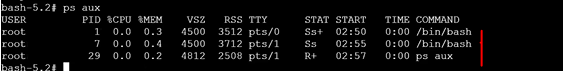
1. If process is not running than no container also not running. If no container running than no server running.
2. Again We pull the images and started running linux image as **dt** mode with out giving /bin/bash. Than when we show which container with which process is running than we see as in linux image /bin/bash is running internally even we are in given the process to start. Means in Dockerfile have CMD that can run the process automatically for DT mode.



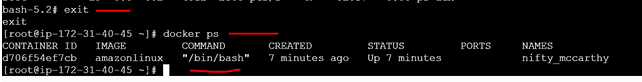
1. Running Container how we will login.

When we give below command than we will see 3 process are running because we are starting a process by giving /bin/bash in login command and another /bin/bash process started when we run a image in container.

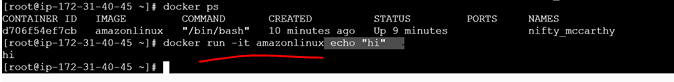




When we will exist from this login container than still it running because we existed from current **it** mode login /bin/process.



1. In **Dt** mode we can’t give argument explicitly by default CMD will run a **/bin/bash/** process if we want to login container and interact than we will execute **it** mode with another **/bin/bash/** process. In it mode if we are giving argument than it will override the existing CMD command and only we have one command in CMD to execute.
2. If I am running It run command with different argument than it will override and run it than terminate.

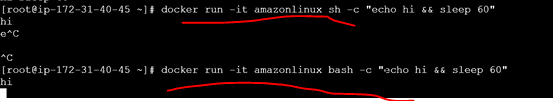




But /bin/bash/ is a continuous process that’s why we are able to login to container. Untill kill the process you are inside container.



1. If you will run below 2 command than it will execute first than sleep for 60 sec than kill the process because its not a /bin/bash/ command for continuous login.



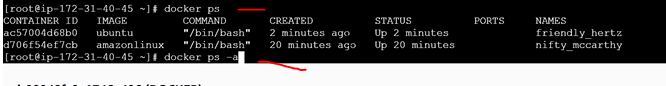
After 60 sec its automatically killed.





Here main understanding is in container at least one container should run continuously so that container is active.

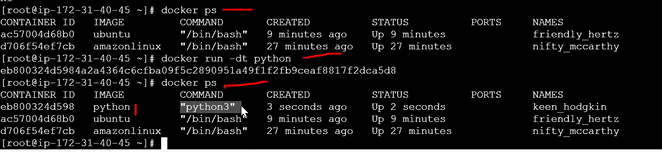
1. To know the how many containers are running and all the containers what ever also killed.



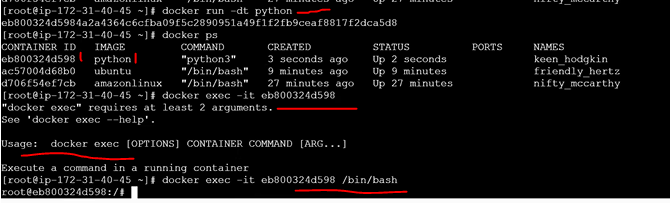
1. If I want to check only existed containers.



1. If its **dt** mode or its **it** mode not matter. Its depend on what command given in CMD based on that it will run as continuous or run and exist.



1. If a application is running and developer try to login docker container image using default argument than it will not allow in **dt** mode becaue if user give exist than container will delete.



1. To kill the process using job id.
2. In Python docker image cmd value is python3 which is continuous run. If we will run in **dt** mode than it will continuous run if same python image we will run in **it** mode by giving /bin/bash/ process than it will override the python3 cmd command it run the **/bin/bash/** process.

