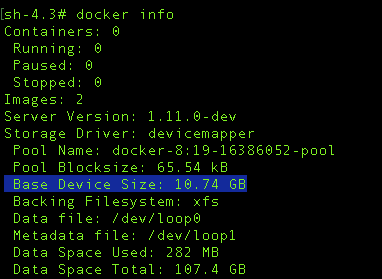
**Docker Assignment 5**

1. How big can Docker containers get? Are there any prerequisites for this?

* the default base size is ~10Gb but theoretically it can grow approx 100Gb but it is not advisable since container will take more time to load we can check that with **docker info**
* Large web deployments like Google and Twitter and platform providers such as Heroku and dotCloud, all run on container technology. Containers can be scaled to hundreds of thousands or even millions of them running in parallel. Talking about requirements, containers require the memory and the OS at all the times and a way to use this memory efficiently when scaled.



1. What operating systems does Docker support?

* The Docker platform runs natively on Linux (on x86-64, ARM and many other CPU architectures) and on Windows (x86-64).
* Docker is supported and can run on several platforms such as:
* Fedora
* Ubuntu
* Gentoo
* RHEL 6.5+
* ArchLinux
* CentOS 6+
* CRUX 3.0+
* openSUSE 12.3+
* Windows
* Docker can be used also in production through Cloud platforms with the given services:
* Amazon ECS
* Amazon EC2
* Microsoft Azure
* Google Compute Engine
* Rackspace

1. How can I figure out what a Docker container's state is?

* To check only for active container : **docker ps**
* To check for container whether running/stopped.. : **docker ps -a**

1. How do you unpause a Docker container?

* command to unpause : **docker unpause container\_name**

1. Is it possible for a container to restart on its own?

* No By default the flag is set to false, but yes if we configure then yes restart is possible, docker provides restart policies after they exit. Docker policy will make sure that it started in correct order. Docker recommends that you use restart policies, and avoid using process managers to start containers.

1. Is it better to use the RM command to delete the container or to stop it by removing it?

* It's always better to first stop the container and then remove it using the remove command.
* **docker stop <container\_id>**
* **docker rm -f <container\_id>**
* Stopping the container and then removing it will allow sending SIG\_HUP signal to recipients. This will ensure that all the containers have enough time to clean up their tasks. This method is considered a good practice, avoiding unwanted errors.

1. Is it a smart idea to use Docker to run stateful applications?

* For satefull applications their data gets stored on the local file system. So, when you move the application to another device, it will become difficult for you to retrieve data. So, we do not recommend running stateful applications here.
* It is preferable to create a Stateless application for Docker Container. We can create a container out of our application and take out the configurable state parameters from the application. Now we can run the same container in Production as well as QA environments with different parameters. This helps in reusing the same Image in different scenarios.
* Also, a stateless application is much easier to scale with Docker Containers than a stateful application.

1. Will cloud overtake the concept of containerization?

* Docker containers are gaining popularity but at the same time, Cloud services are giving a good fight.
* In my personal opinion, Docker will never be replaced by Cloud.
* Using cloud services with containerization will definitely hype the game.
* Organizations need to take their requirements and dependencies into consideration into the picture and decide what’s best for them.
* Most of the companies have integrated Docker with the cloud. This way they can make the best out of both the technologies.

1. How do you load a pre-exported Docker image onto another Docker host using a command?

* Once the target machine has the .tar file, you can load the image into the local registry using command docker load :
* Load an image or repository from a tar archive (even if compressed with gzip, bzip2, or xz) from a file or STDIN. It restores both images and tags. **docker load < file\_name.tar**
* Load an image from a tar archive or STDIN - **docker image load [OPTIONS]**