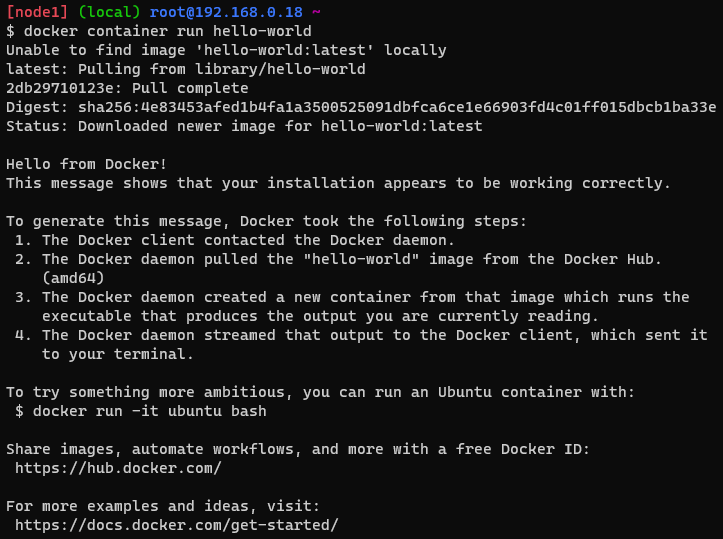
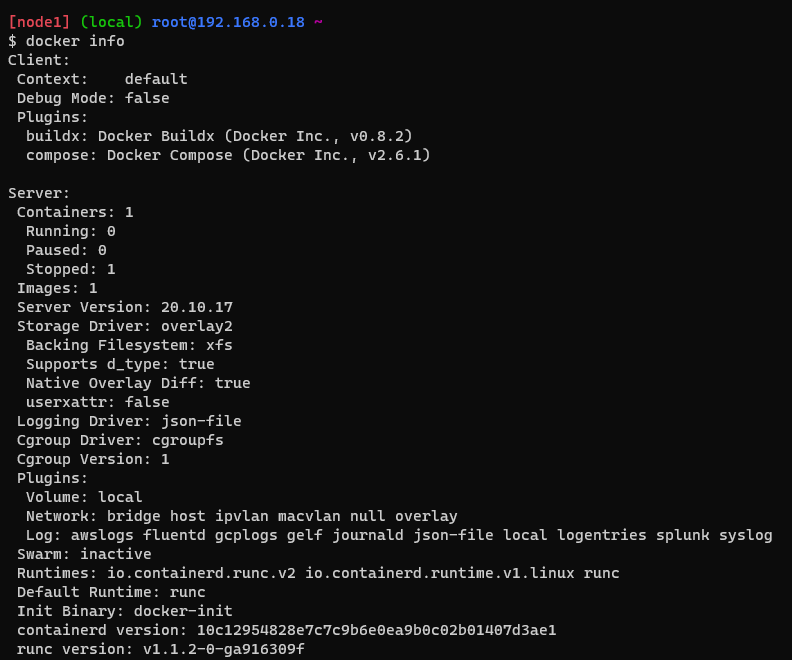
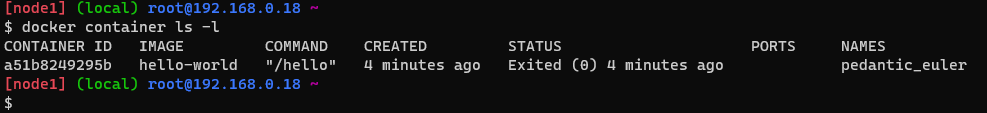
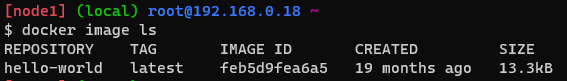
1. Run hello-world docker container and observe the container status



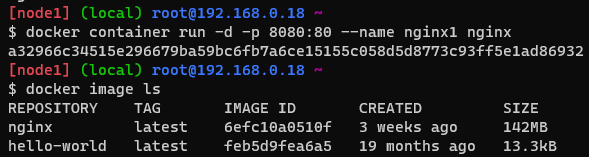


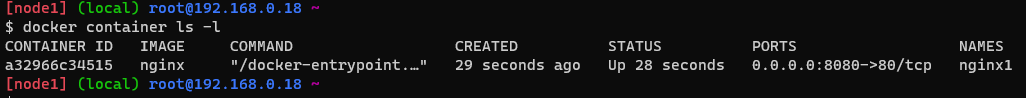


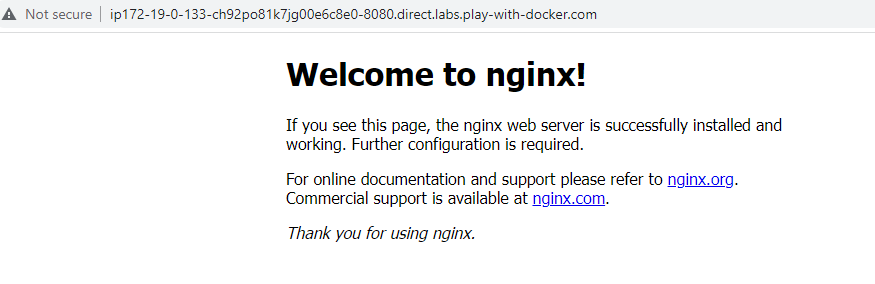
2. Check the docker images and also write down the size of hello-world image



3. Run the nginx container with name as nginx1 and expose it on 8080 port on docker host



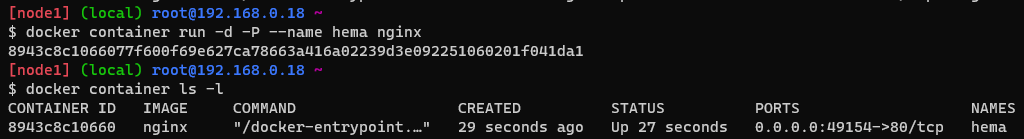




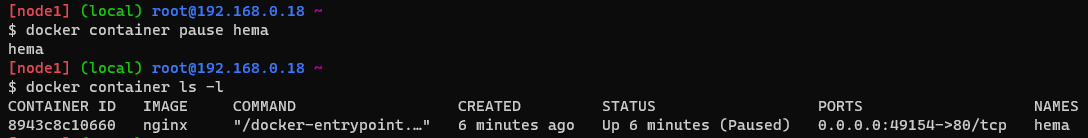
4. Explain docker container lifecycle

Docker container lifecycle has 5 stages.They are

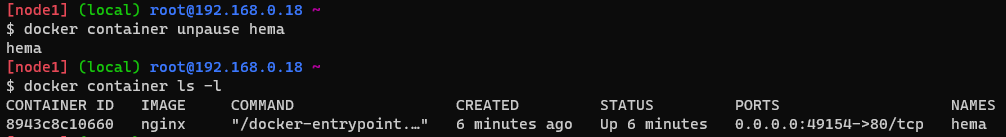
* Create
* Run
* Pause
* Stop
* Delete
* DOCKER CONTAINER CREATE



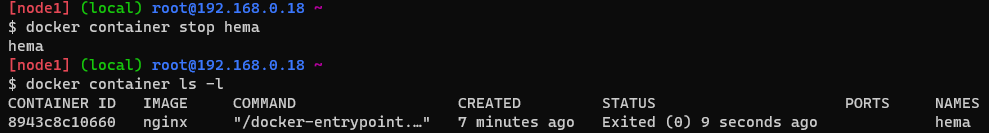
* DOCKER CONTAINER PAUSE



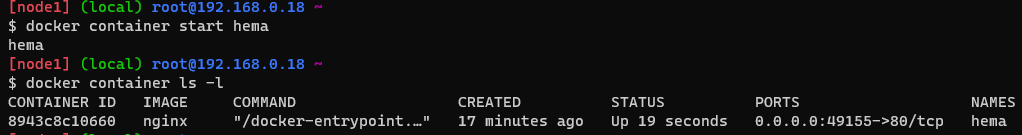
* DOCKER CONTAINER UNPAUSE



* DOCKER CONTAINER STOP



* DOCKER CONTAINER START



* DOCKER CONTAINER DELETE



5. Explain what happens when you run the docker container

The docker run command creates and starts the docker container. Docker daemon checks if the image is available locally or not. If the image exists locally it creates the container. If the image doesn’t exists it pulls the image from the local repository. Once the image is pulled the container gets created.

6. Explain the Docker Architecture

* **First generation**:

Docker client (command line) will send request to docker daemon, it forwards the request to lxc (linux container) internally uses namespaces, c groups to create containers.

* **Second generation:**

Due to frequent updates to linux kernel,there is a chance of container breakdown .As docker is relying on lxc to create containers it has created its own component called as lib container (libc)

* **Third generation:**

In this generation docker is revamped from monolith to micro services i.e., multi component system. Here docker daemon exposes its api to interact with docker client then it sends the request to container-d and then passes to run-c which creates the container and manages the docker lifecycle.