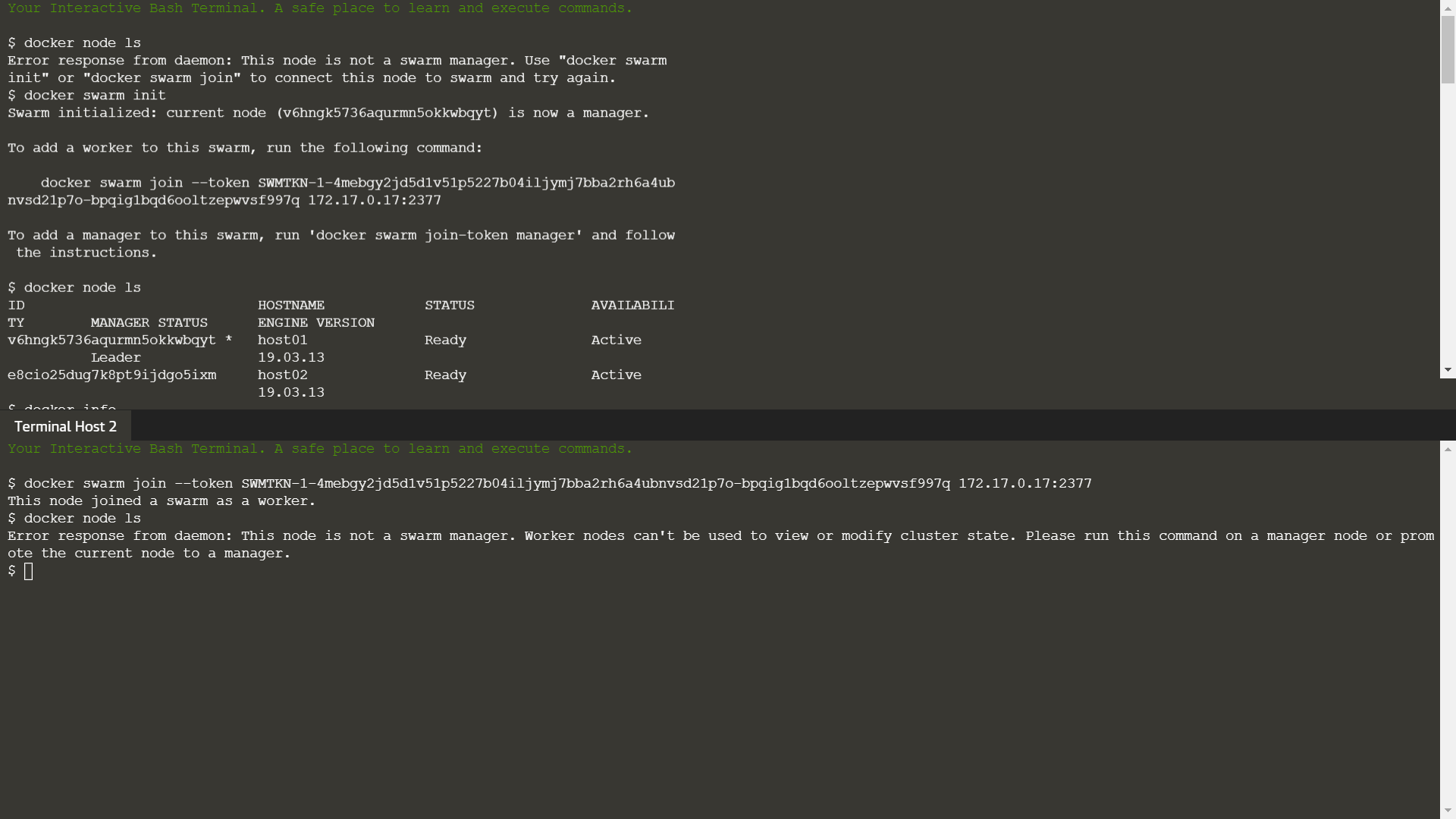
Application Containerization Lab

EXPERIMENT - 7 (Docker Swarm)

**Name: Harivansh Mathur, Roll No: R171218043**

Open two terminals one for being the master and the other for being the slave (node), check if the swarm is active or not by the command ‘docker node ls’. If the swarm is active the command will list the master and the slaves but if the swarm isn’t active the command will throw an error.

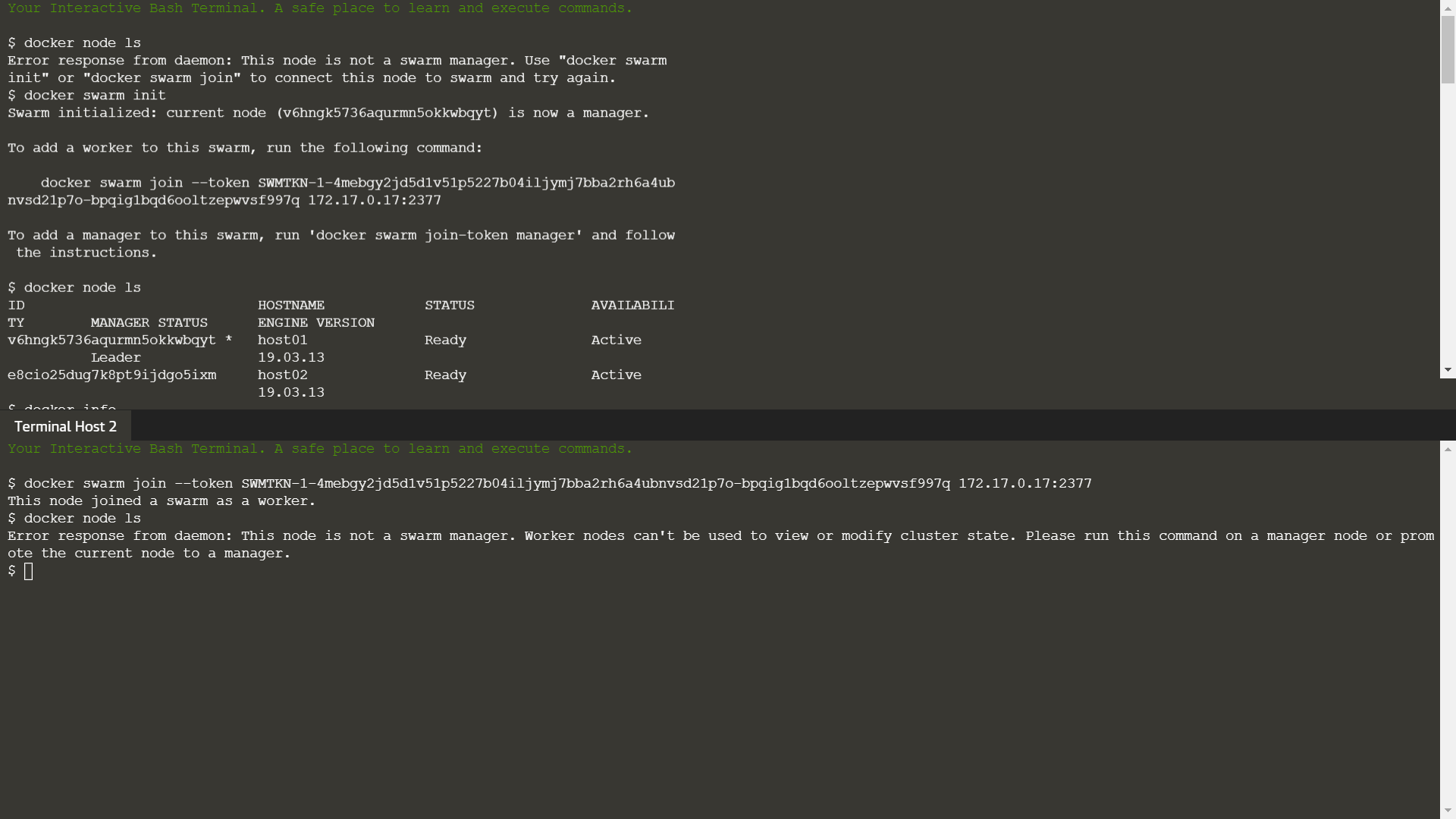
*Terminal 1 -*



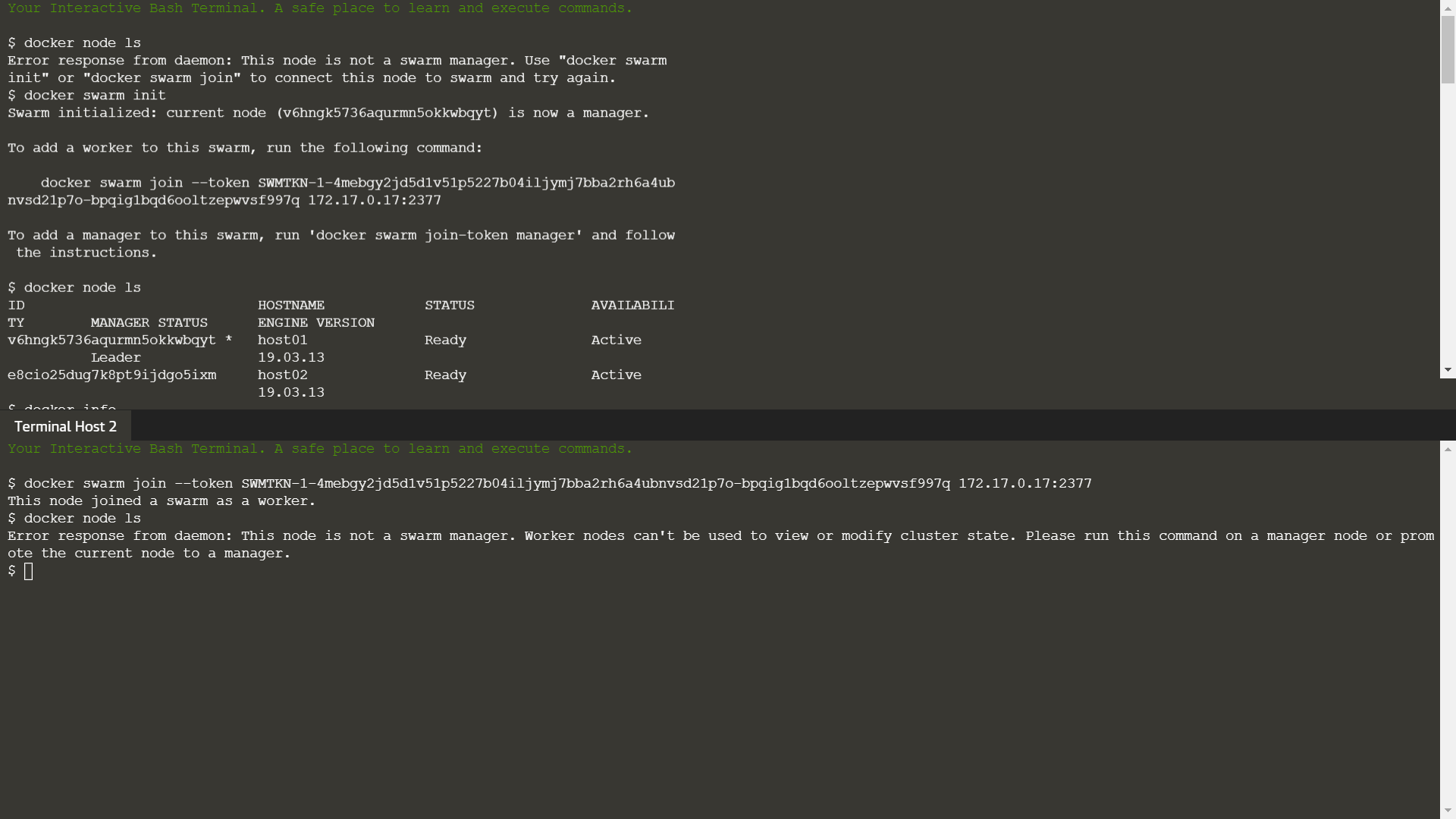
Activate the swarm using the command ‘docker swarm init’, the output of the command will have a command having a unique token and the IP address. Through this command, we can connect the other terminal as a worker to the master by running the command in the terminal.

The basic syntax of the command from the output is ‘docker swarm join –token <token> <ip and port no.>’

*Terminal 1 -*

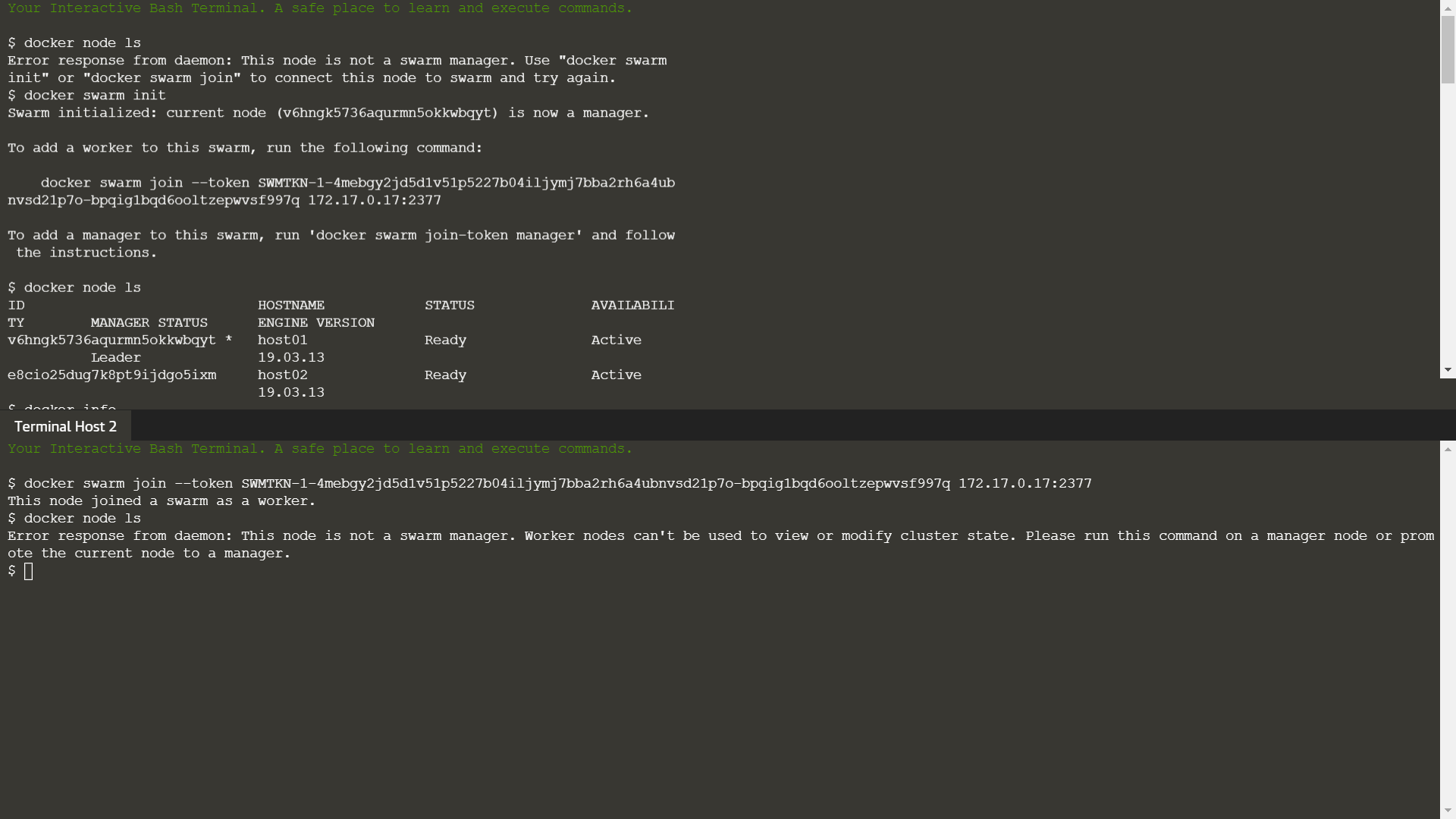


*Terminal 2 -*



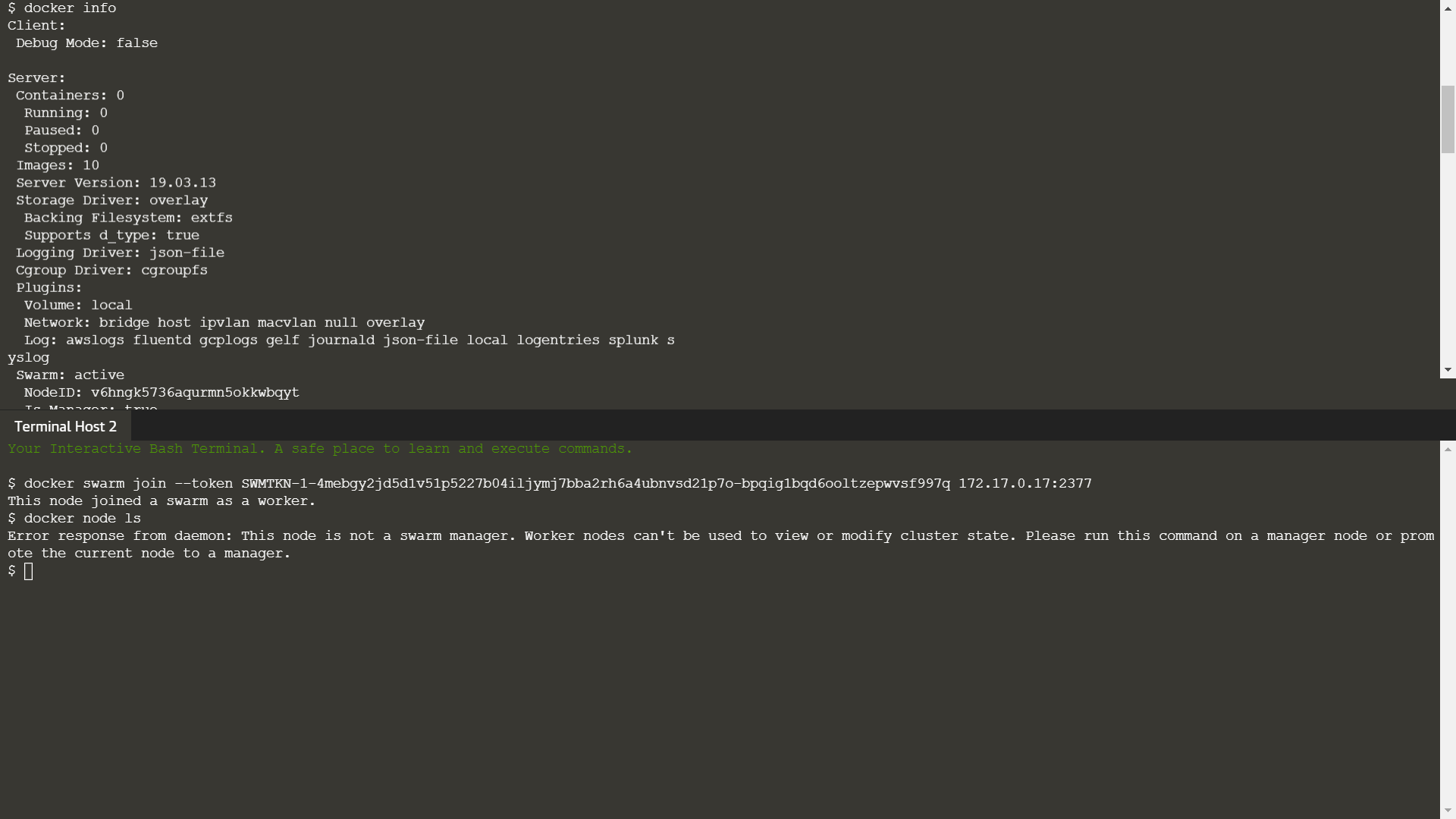
Again run the command ‘docker node ls’ to view all the nodes in the swarm we just activated, the nodes which are acting as a worker will be written directly but the node which is acting as a manager will have ‘Leader’ written with it and a ‘\*’ mark on it.

*Terminal 1 -*



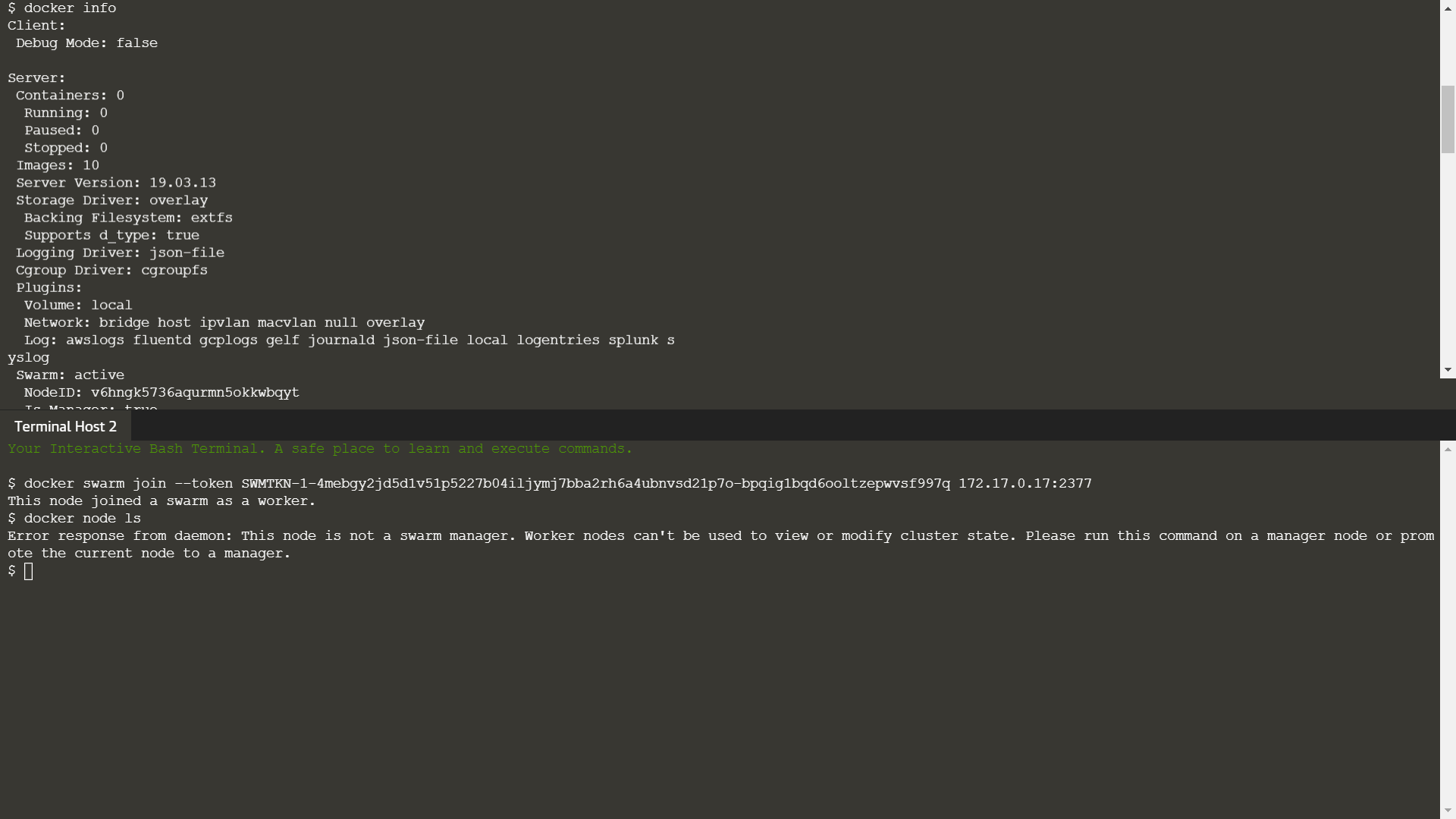
Only the manager can access the information regarding the number of rows and nodes in the swarm so if we try to run the command in a worker node terminal, it will fail.

*Terminal 2 -*



We can check all the details of the swarm and the nodes by running the command ‘docker info’.

*Terminal 1 -*



When we want to remove a node from the swarm we use the command ‘docker swarm leave --force’ if the node is a manager and ‘docker swarm leave’ if it’s a worker node. If a manager leaves the swarm, the manager properties are automatically inherited by a worker node and it becomes the new manager

*Terminal 1 -*

