**PRACTICAL: 7**

**AIM:**

Kubernetes in Google Cloud: Kubernetes is the most popular container orchestration system and the Google Kubernetes Engine was designed specifically to support managed Kubernetes deployments in the Google Cloud. In this advanced-level quest, you will get hands-on practice configuring Docker images and containers, and deploying fully-fledged Kubernetes Engine applications. This quest lab will teach you the practical skills needed for integrating container orchestration into your own workflow.

**THEORY:**

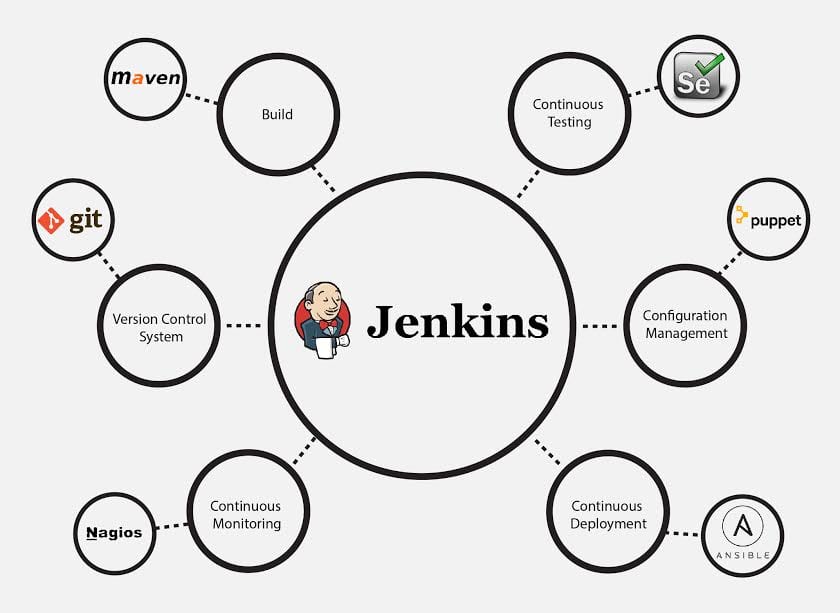
Docker is an open platform for developing, shipping, and running applications. With Docker, you can separate your applications from your infrastructure and treat your infrastructure like a managed application. Docker helps you ship code faster, test faster, deploy faster, and shorten the cycle between writing code and running code.

Docker does this by combining kernel containerization features with workflows and tooling that helps you manage and deploy your applications. Docker containers can be directly used in Kubernetes, which allows them to be run in the Kubernetes Engine with ease. After learning the essentials of Docker, you will have the skillset to start developing Kubernetes and containerized applications.

Kubernetes is an open source project (available on kubernetes.io) which can run on many different environments, from laptops to high-availability multi-node clusters, from public clouds to on-premise deployments, from virtual machines to bare metal.

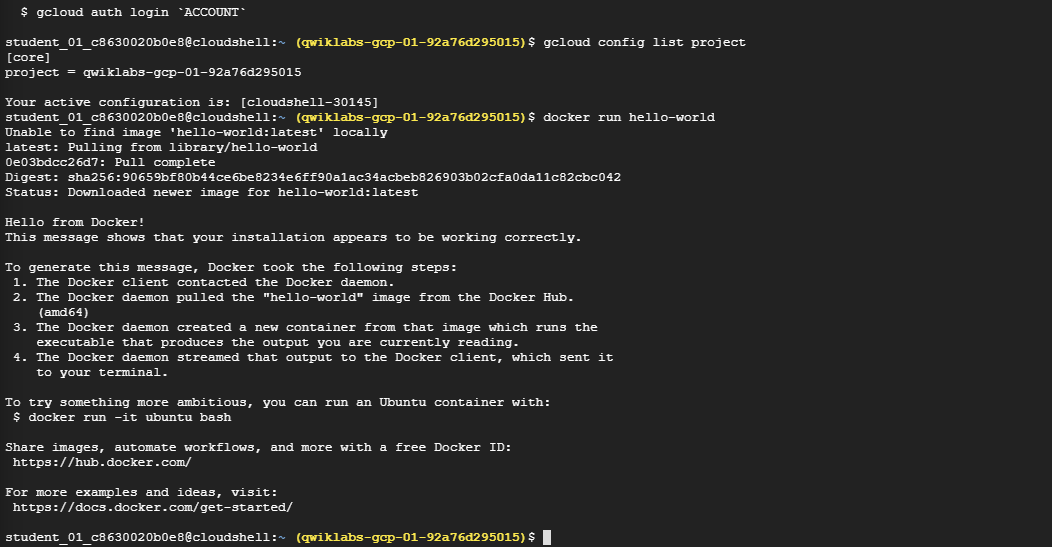
Dev Ops practices will regularly make use of multiple deployments to manage application deployment scenarios such as "Continuous Deployment", "Blue-Green Deployments", "Canary Deployments" and more.

Jenkins is a free and open source automation server. It helps automate the parts of software development related to building, testing, and deploying, facilitating continuous integration and continuous delivery. It is a server-based system that runs in servlet containers such as Apache Tomcat.

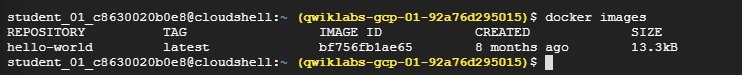


**OUTPUT:**

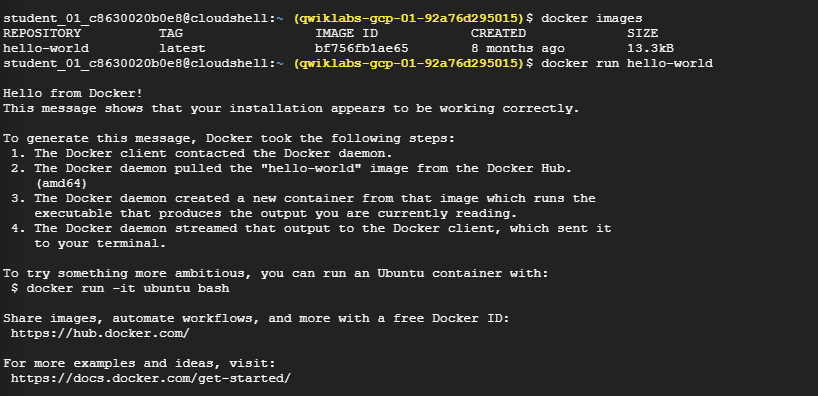
**7.1 - Introduction to Docker**



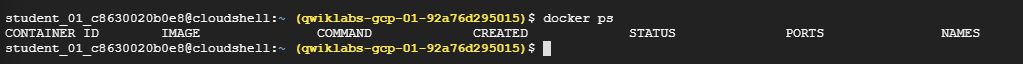
**Running docker hello-world**



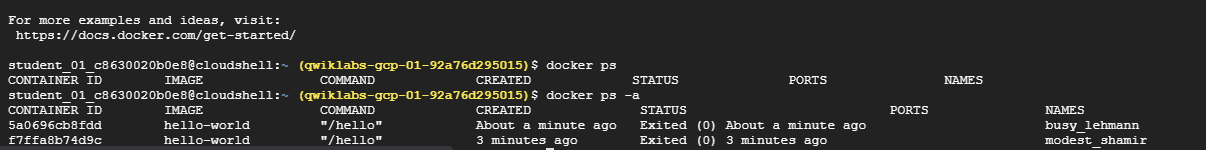
**Docker images**



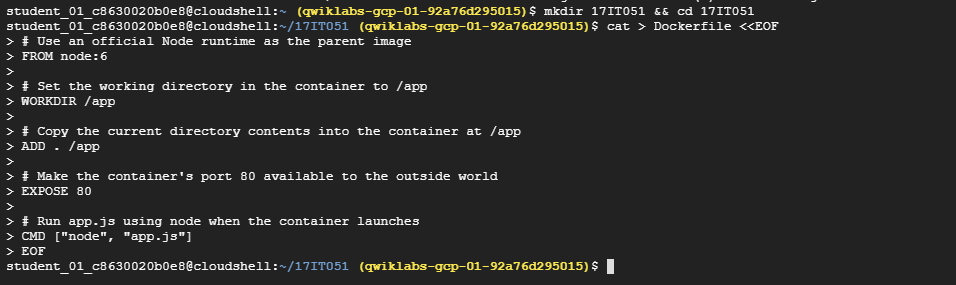
**Docker run hello-world again**



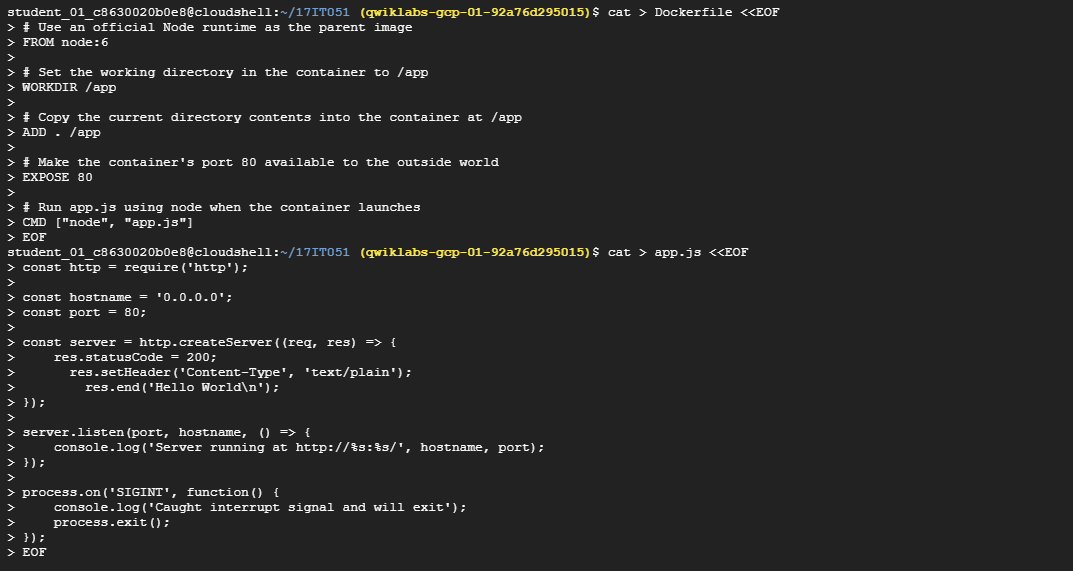
**Listing docker process**



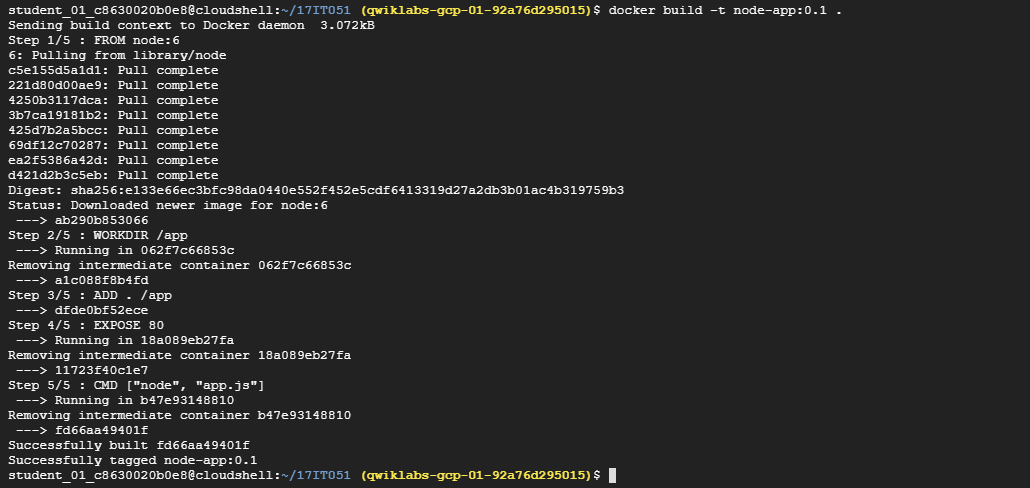
**Listing docker process with all flag**



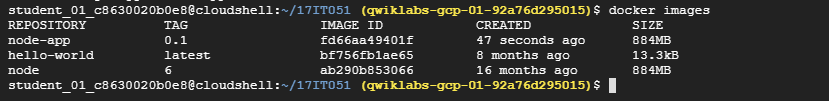
**Creating directory and navigation into it**



**Viewing docker file**



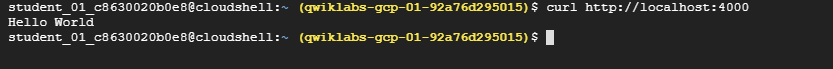
**Building docker**



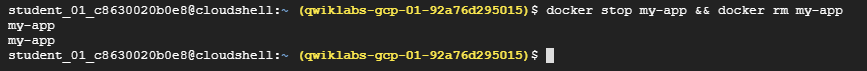
**Visualize docker image**



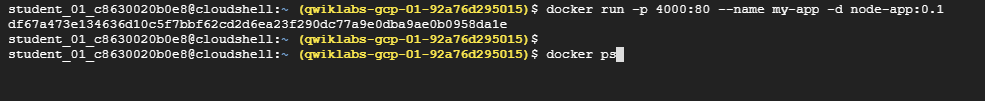
**Running docker**



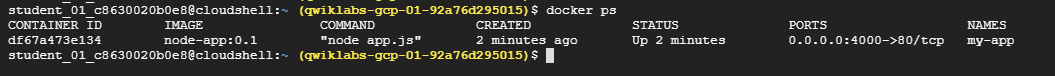
**Curl to IP on which docker activated**



**Stopping and removing my-app**



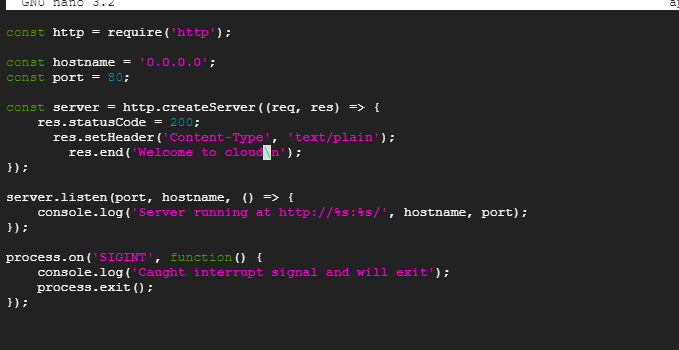
**Running docker again**



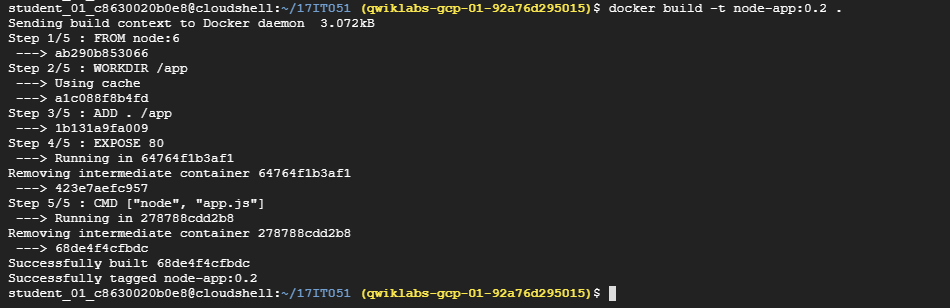
**Visualize docker process**



**Docker logs**



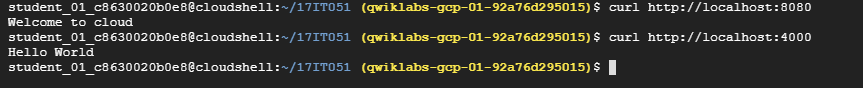
**Node server**



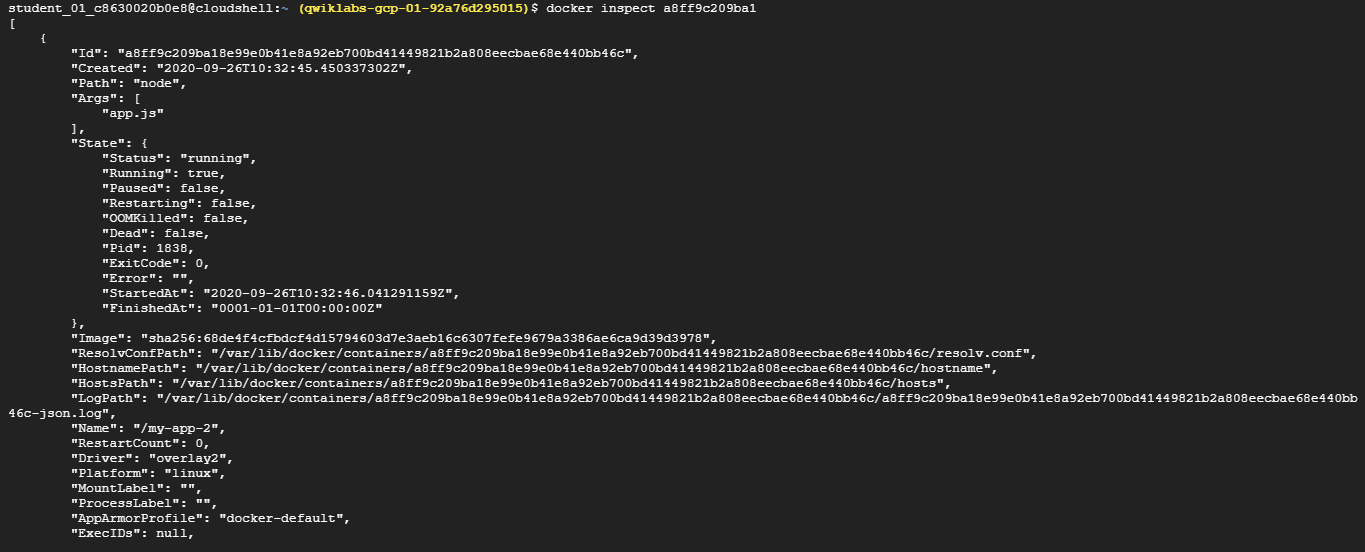
**Building docker**



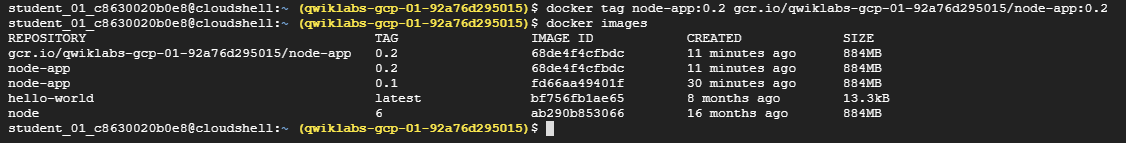
**Running docker again**



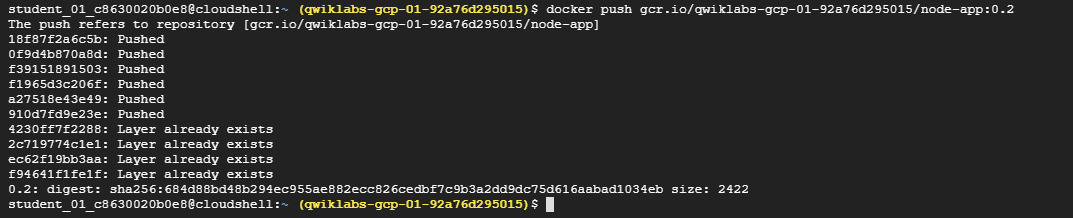
**Curl both ports 8080 and 4000**



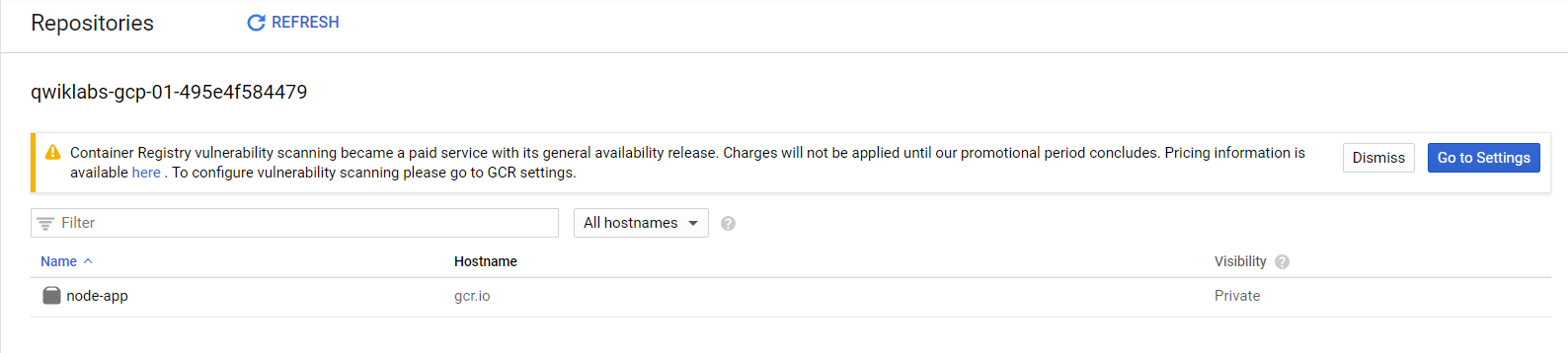
**Inspecting the docker**



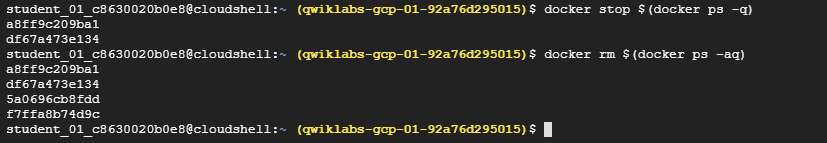
**Viewing all docker images again**



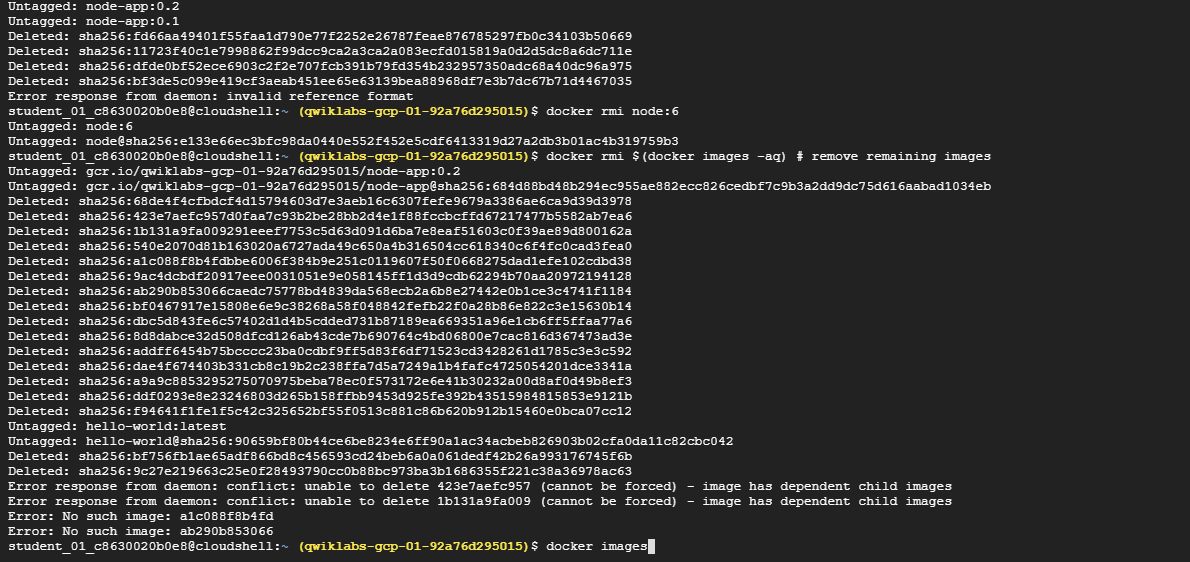
**Pushing into docker image**



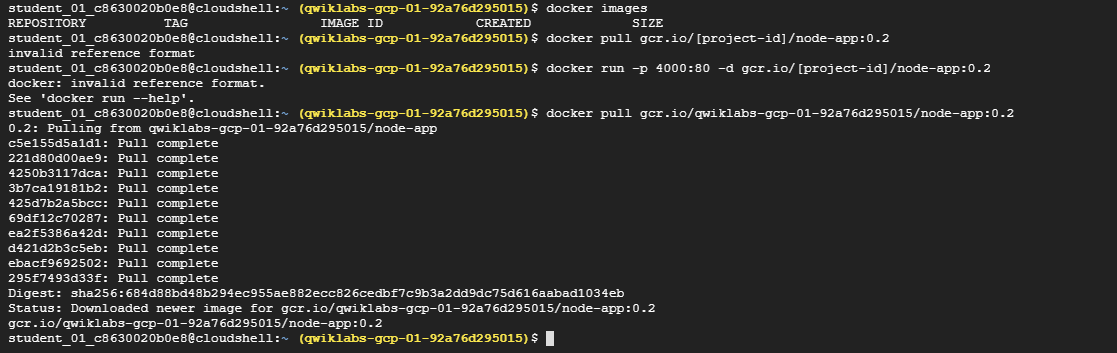
**Repositories dashboard contain node-app docker**



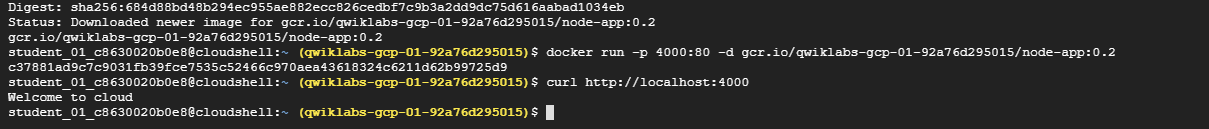
**Stopping docker and then removing using nesting commands**



**Removing remaining images**



**Docker pulling**



**Running docker again on port 4000**

**LATEST APPLICATIONS:**

**LEARNING OUTCOME:**

**REFERENCE:**

1. <https://google.qwiklabs.com/quests/29>