

# **Project Milestone**

# **IaaS: Virtualization and Containerization**

**Name: Mihir Patel**

**Student Number: 100702168**

**Date: January 31, 2022**

**Video Link:** [**https://drive.google.com/drive/folders/17EnSZpAIZPjl6oymphOvR-a6cpggRB8S?usp=sharing**](https://drive.google.com/drive/folders/17EnSZpAIZPjl6oymphOvR-a6cpggRB8S?usp=sharing)

**Procedure:**

1. **What are docker images, container, and registry?**

* A container is a standard component that allows the user to package the application and its dependencies. Containers are portable, isolated and lightweight. Docker images are readable templates that include the instruction for creating a container. They are convenient because it provides a convenient way to package applications. A docker registry is a place that stores the docker images and the version that it's associated with.

1. **List the Docker commands used in the video with a brief description for each command and option.**

* In the video following are the commands that are being used

1. **Docker builds -t hello-world:1.0** - This command executes the docker file that contains the instruction to create a docker image.
2. **Docker images** - list all the docker images available on the system
3. **Docker run hello-world:1.0** - This command uses the docker image to make a docker container and execute the content specified in the docker file.
4. **Docker ps** - This command lists all the running containers in the system. It also shows the detailed version of the container.
5. **docker ps -a** - This command lists all the containers in the system. It also shows the detailed version of the container.
6. **Docker run -d hello-world-** 1.0- This command will execute the container in the background without displaying the output on the screen.
7. **docker logs [container id]-** show the out of container that is being executed in the background.
8. **At the end of the video, there are two running containers, what commands can be used to stop and delete those two containers?**

* The command used for stopping and deleting the containers are as follows
  + **Stop:** docker container stop [container id]
  + **Delete:** docker container rm [container id]

**4. What’s a multi-container Docker application?**

* A multi-container docker application is an application that is able to link multiple docker containers with the help of the docker network protocol.

**5. How these containers are communicated together?**

* Docker network allows containers to communicate with one another as this is done by creating a network bridge between the two containers.

**6. What command can be used to stop the Docker application and delete its images?**

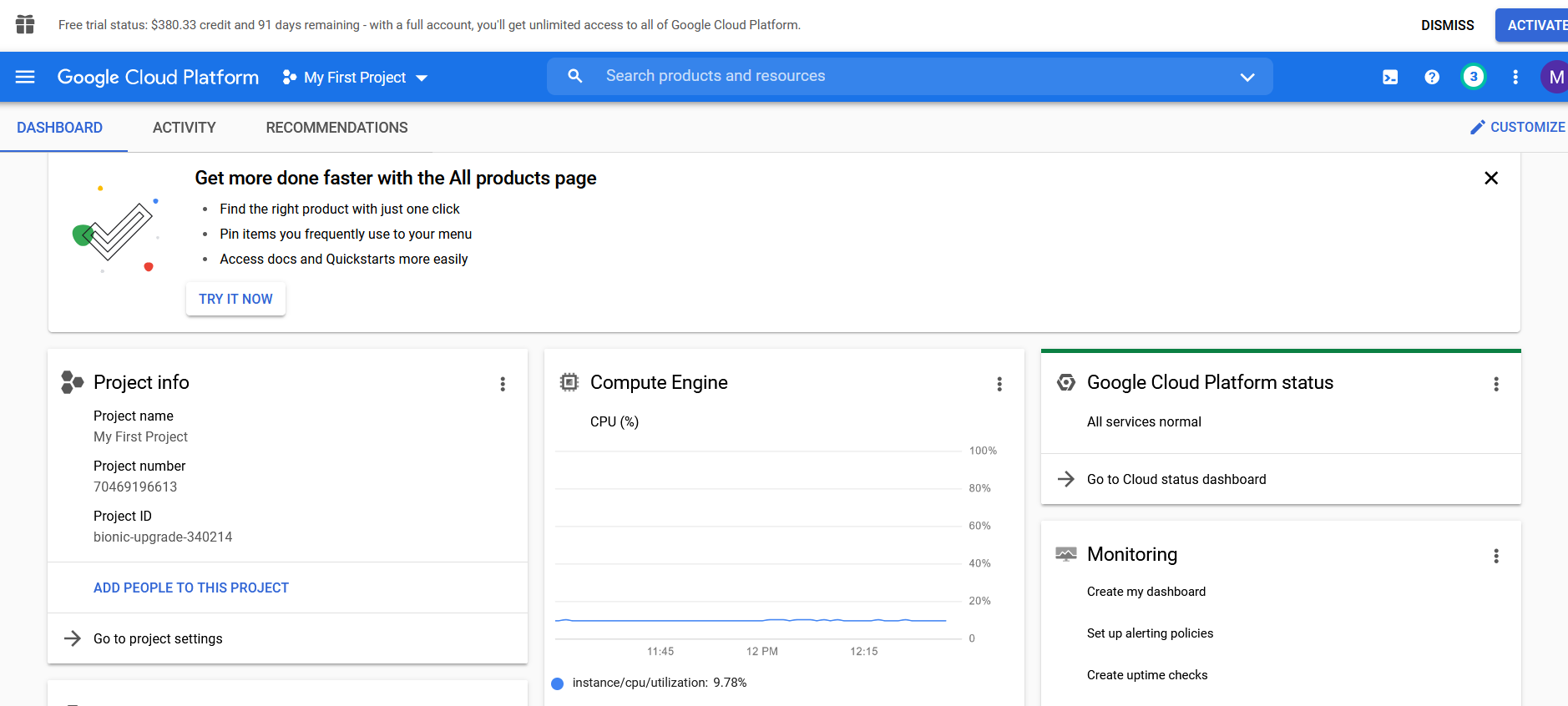
-The command used for stopping and deleting the containers are as follows

* + **Stop:** docker container stop [container id]
  + **Delete:** docker container rm [container id]

**7. List the new docker commands used in the video with a brief description for each command and option.**

* **Docker pull mysql:** This command pull the latest version of mysql from docker hub
* **Docker run –name app–db -d:** This command will create a container called app-db in the detached mode.
* **Docker run -name app-db -d -e MYSQL\_ROOT\_PASSWORD=password -e MYSQL\_DATABASE=myDB mysql:** This command will create the root password that will be used to initialize the database with the image mysql.
* **Docker rm -f [container id]:** This command will delete the container without stopping it.
* **Docker network create [network name]:** create an additional network bridge that works similar to the default bridge.
* **Docker network ls:** It displays all the available networks.
* **Docker run –name app -d -p 8080:8080 my-web-app:1.0:** This command will connect the docker port number with the host machine port number so that they can communicate and execute the web application. The first port number is the host machine port number and the second port number is the container port number.
* **Docker-compose up:** This command combines all the commands mentioned above and executes in an easy to run format.

**8. Create a free Google Cloud Account. The first two videos in the following playlist may be helpful**

* ****

**9. List all used GCP shell commands and their description in your report.**

* Create a sample website using docker container
  + **docker run -p 8080:80 nginx:latest :**create a docker image with latest nginx image
  + **docker cp index.html [container-id]:/usr/share/nginx/html/:** copy the index.html file and replace it with the container default html file.
  + **docker commit [container-id] cad/web:version1:** creates a new version of the docker image
  + **docker tag cad/web:version1 us.gcr.io/milestone-1/cad-site:version1:** creates the tag for the google container repository.
  + **docker push us.gcr.io/milestone-1/cad-site:version1:** push the container that was created with the specific tag in the previous command.
* Deploying container in GKE cluster
  + gcloud config set project bionic-upgrade-340214: switches the project to the given name
  + gcloud config set compute/zone us-central1-a
  + Creating a GKE cluster
    - gcloud container clusters create gk-cluster --num-nodes=1
    - gcloud container clusters get-credentials gk-cluster
* Deploying an application to the cluster
  + kubectl create deployment web-server --image=us.gcr.io/youtube-demo-255723/cad-site:version1
  + Exposing the Deployment

kubectl expose deployment web-server --type LoadBalancer --port 80 --target-port 80

**10. Prepare a Kubernetes YML (or YAML) file to load the webApp used in steps 6:8 and deploy it using the Kubernetes engine on GCP. The file is a little different than that used by docker-compose.**

* The hostname of all containers is the same and can be accessed by localhost, the address of the MySQL should be changed to localhost and recompiled. (Note: if you have a problem generating the war file, you can download it from the repository from the path “/KGS/target/MyWebApp.war”).
* Create a new image using the new war file and push it to Google Container Registry.
* Follow the comments and fill the missing lines in the “/webApp.yml” file.

Apply the YML file into Kubernetes and run the server (what is the appropriate Cloud shell command?).

**11. What is Kubernetes’ pod, service, node, and deployment?**

* **Node:** A node is a virtual machine that runs the kubernetes workloads.
* **Pod:**  Pods are the smallest deployment of a container that shares storage/resources.
* **Node:** Node provides services that are used to run pods
* **Deployment:** It's used to inform kumbernets to create the instance of the pods that further can be used to run containerized applications

**12. What’s meant by replicas?**

* As the name suggests it is replicas of the post that are currently running on the machine. It is necessary to ensure that a specific number of replicas are running at any given time such that the user does not lose access to the application.

**13. What are the types of Kubernetes’ services? What is the purpose of each?**

* **ClusterIP.** Exposes a service which is only accessible from within the cluster.
* **NodePort.** Exposes a service via a static port on each node’s IP.
* **LoadBalancer.** Exposes the service via the cloud provider’s load balancer.
* **ExternalName.**