**LAPORAN PRAKTIKUM**

**TEKNOLOGI CLOUD**

**PERTEMUAN KE – 12**

**Disusun Oleh :**

**NAMA : TARISA DWI SEPTIA**

**NIM : 205410126**

**JURUSAN : TEKNIK INFORMATIKA**

**JENJANG : S1**

**UNIVERSITAS TEKNOLOGI DIGITAL INDONSIA**

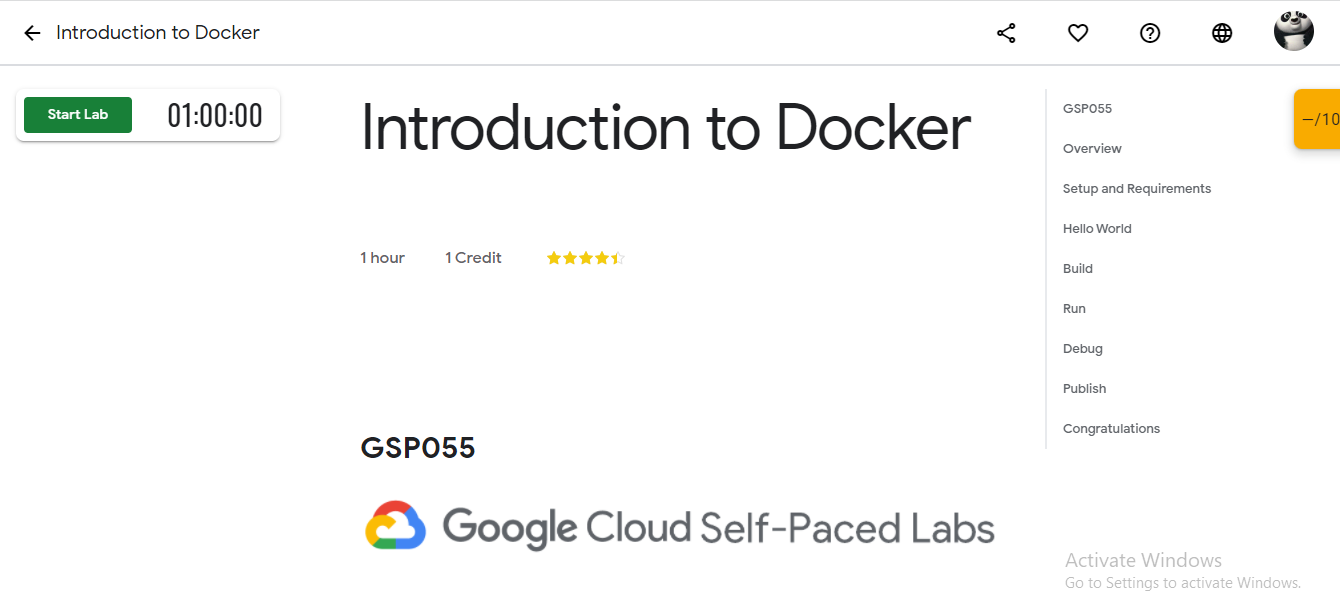
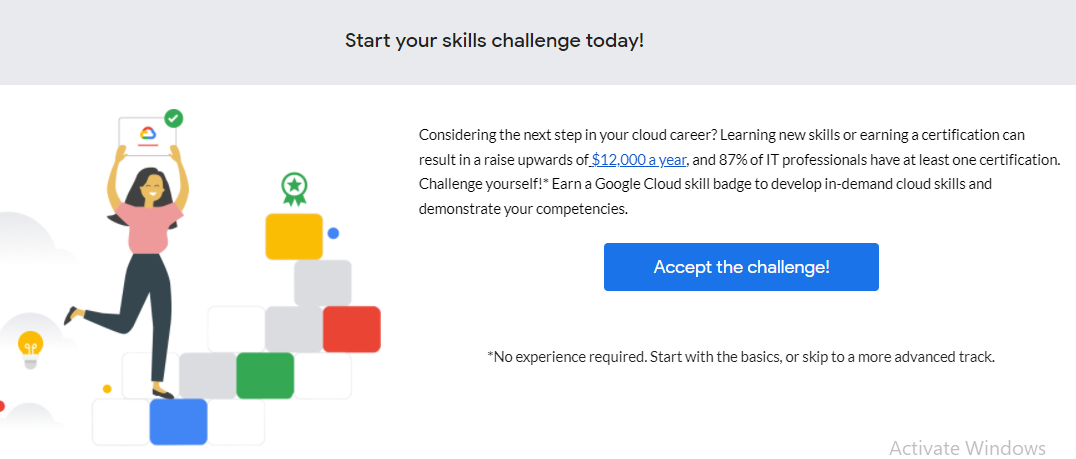
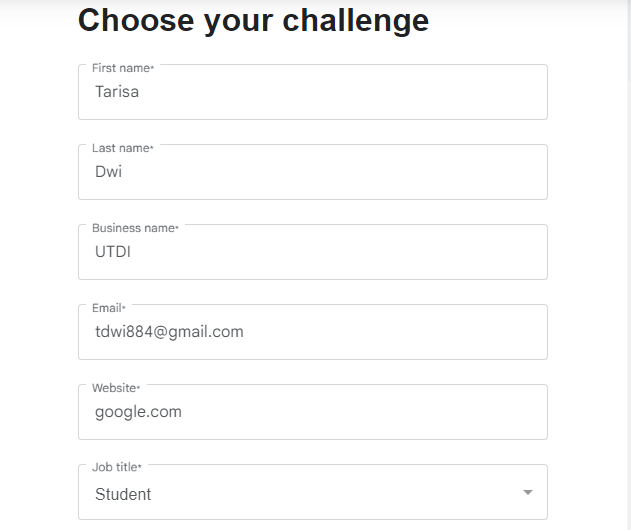
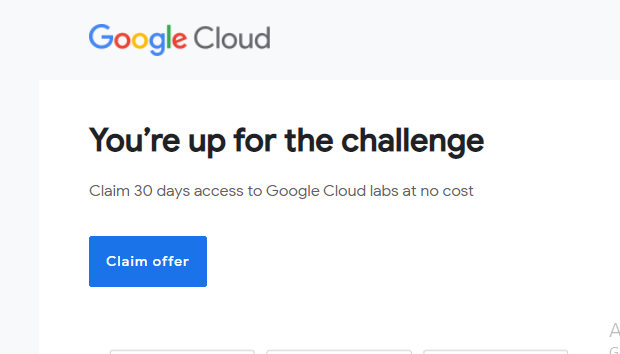
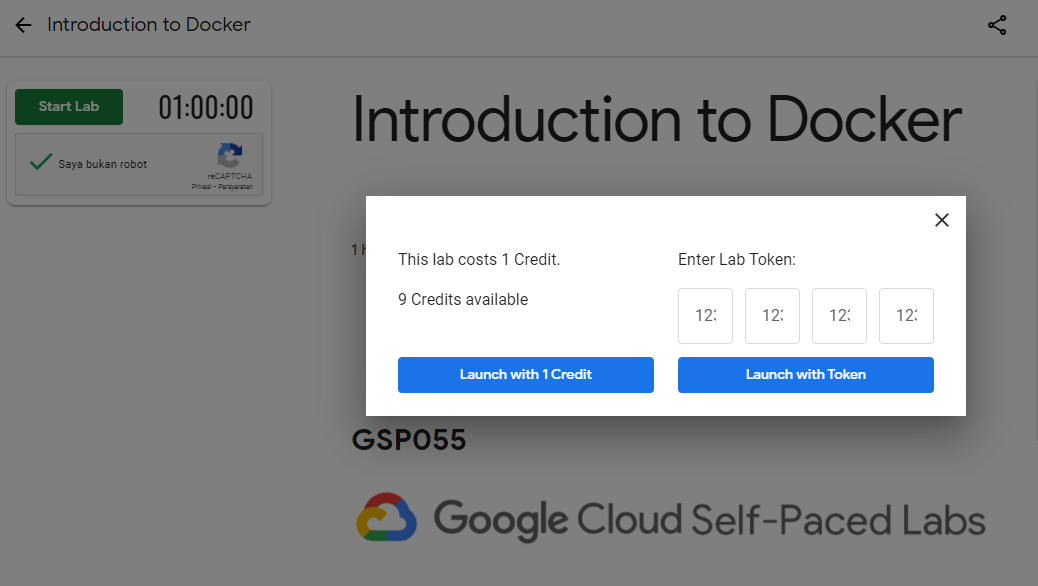
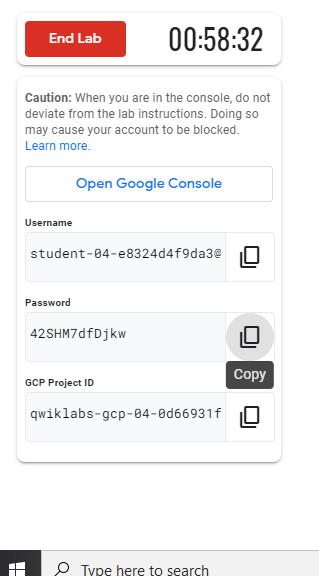
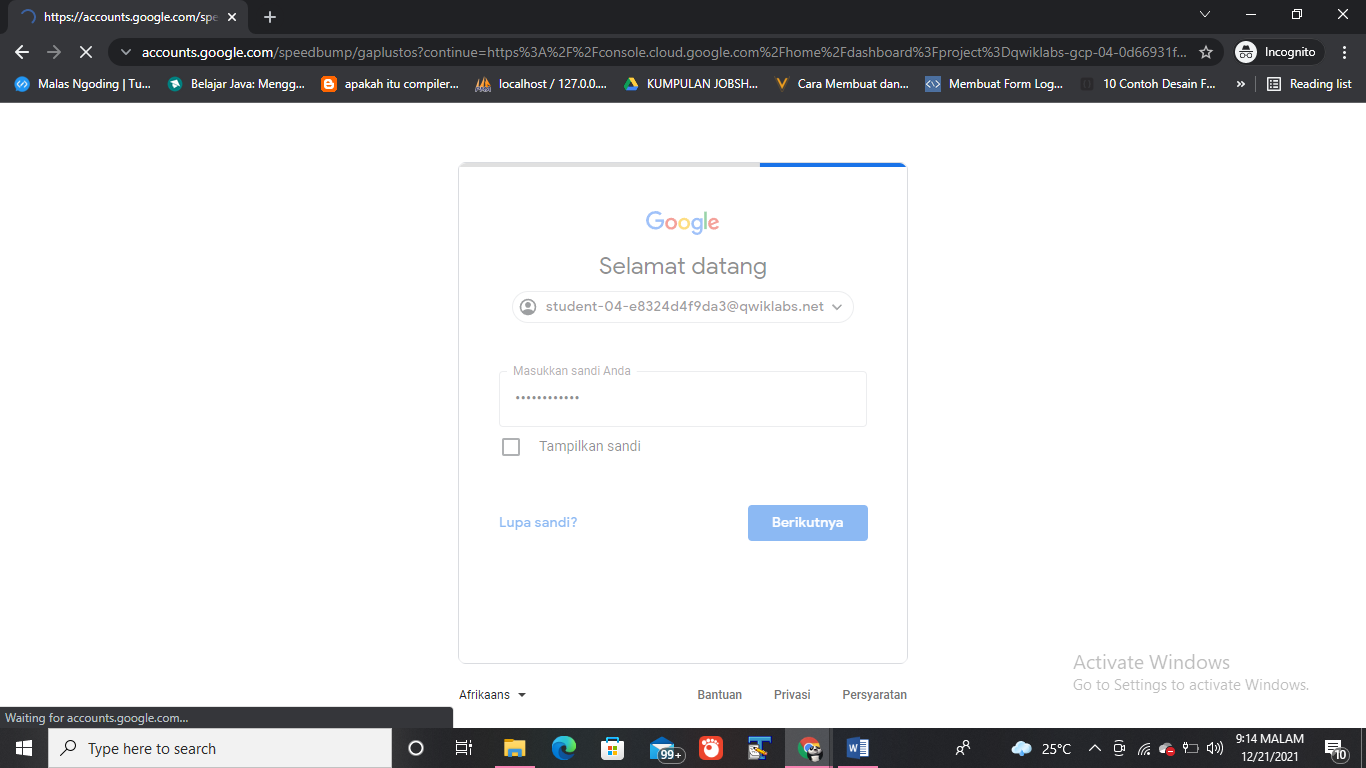
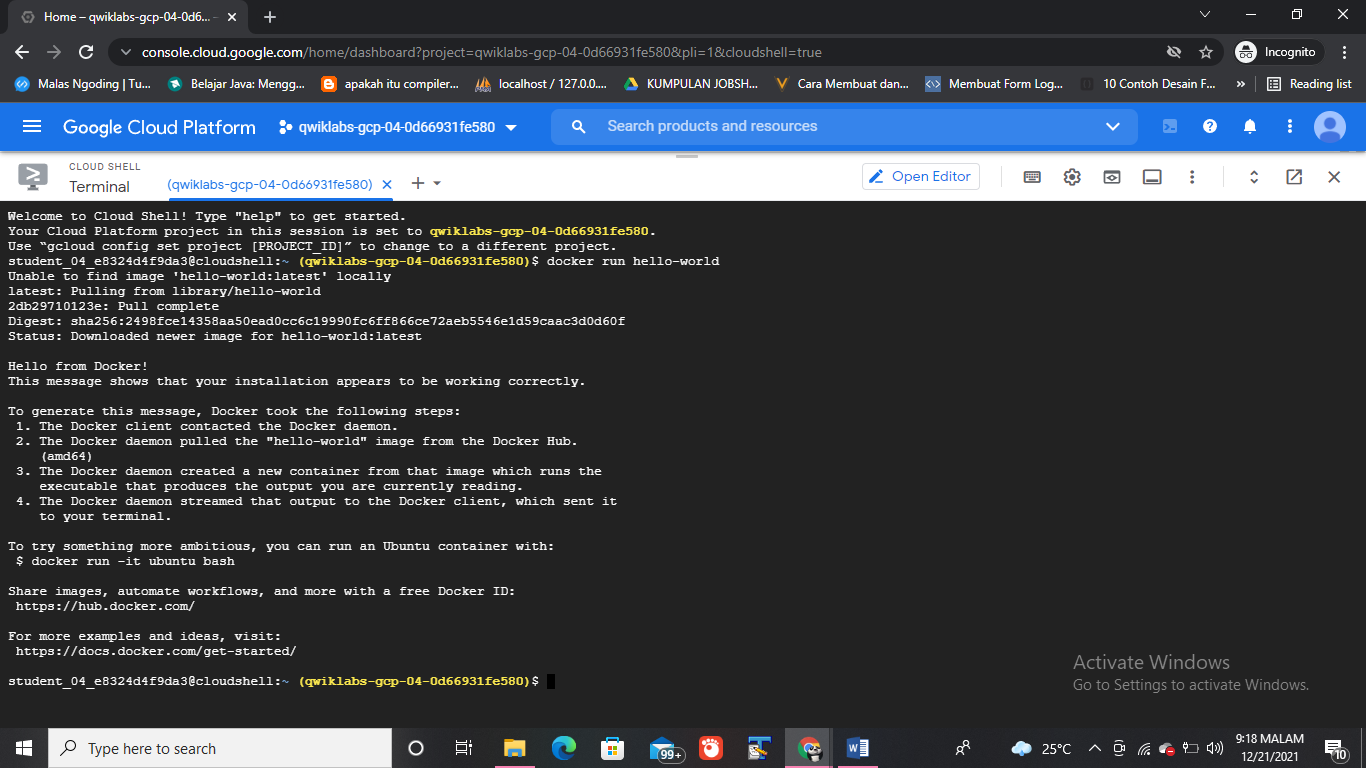
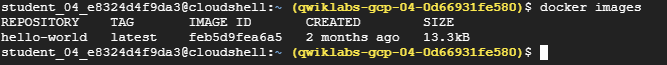
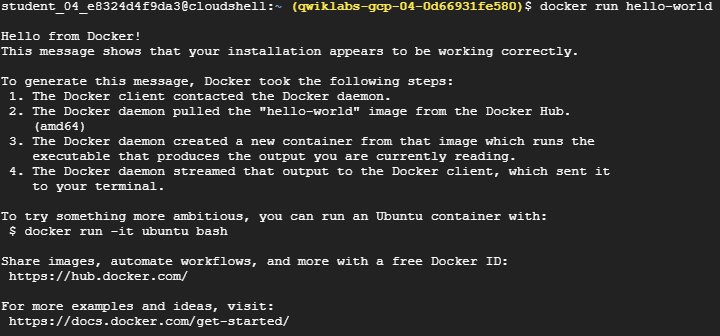
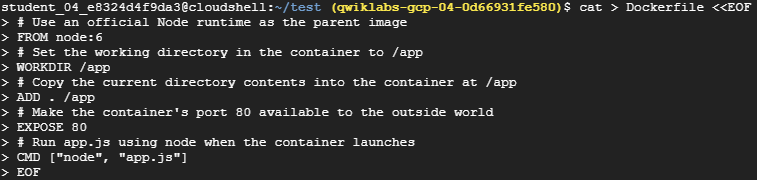
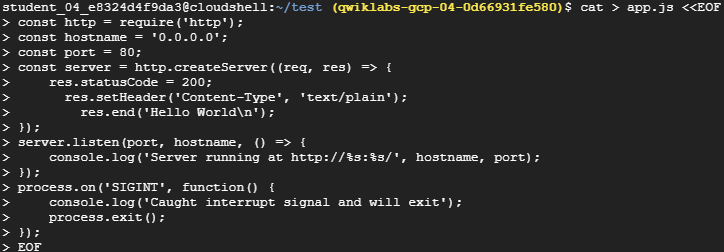
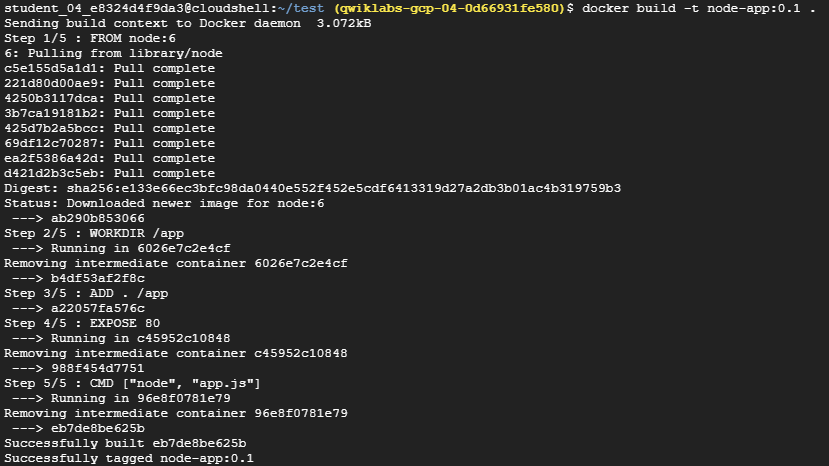
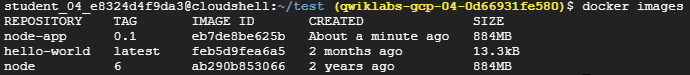
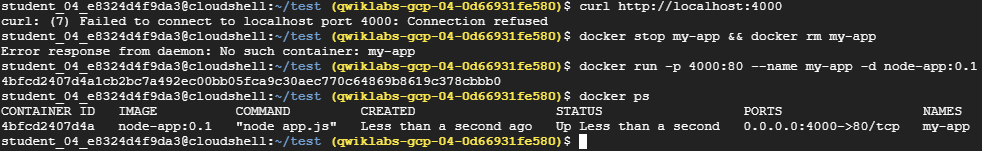
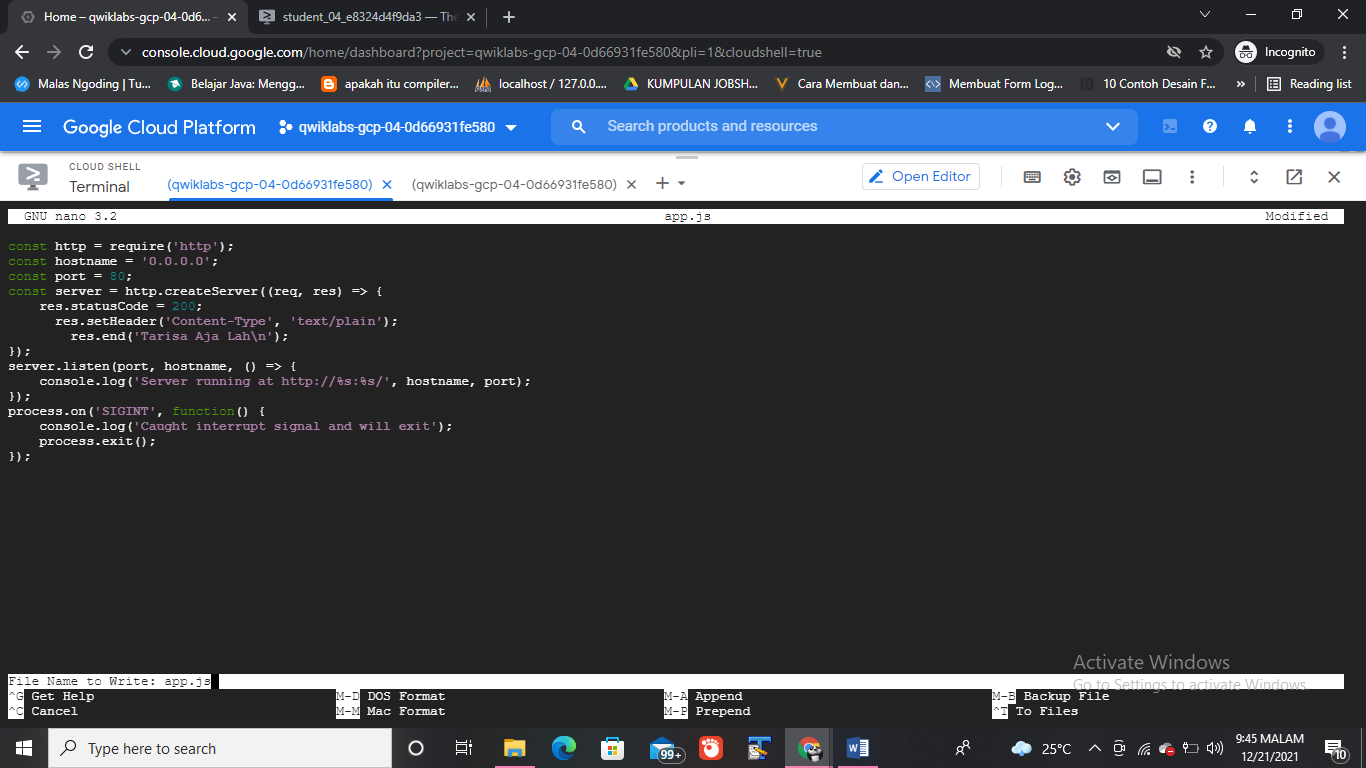
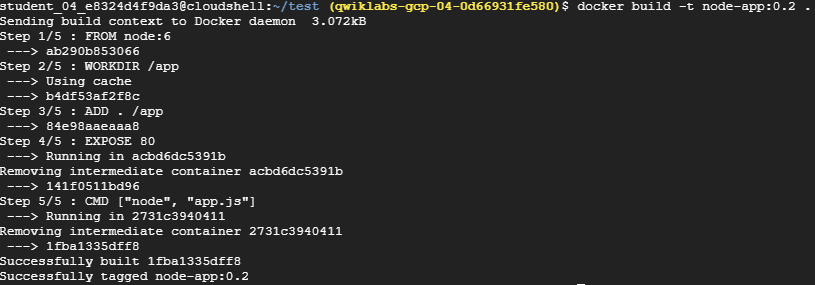
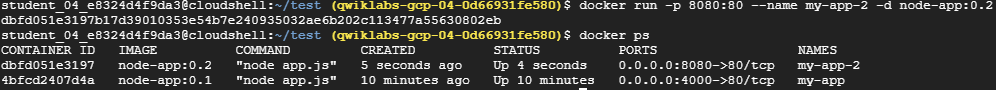
**YOGYAKARTA**

**2020**

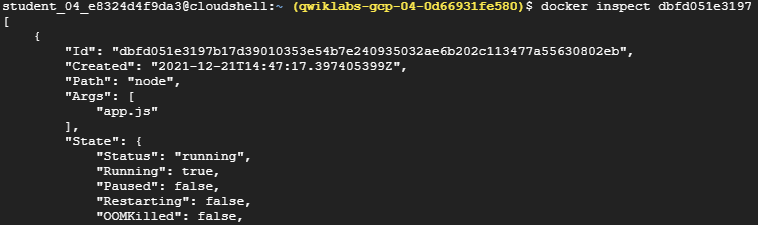
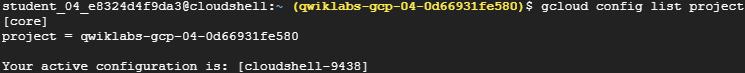
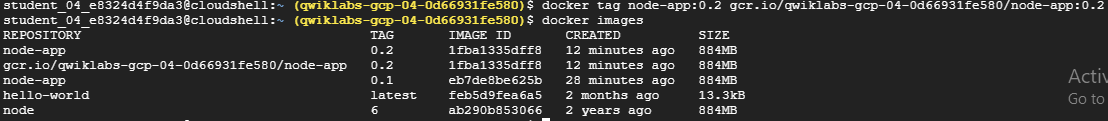
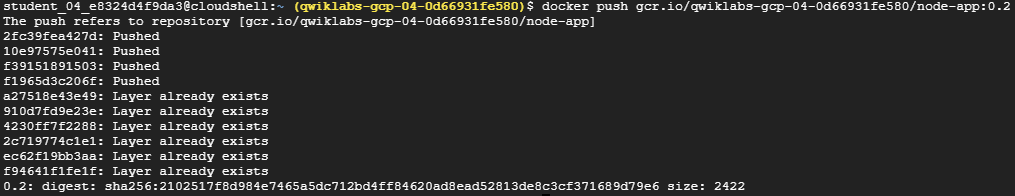
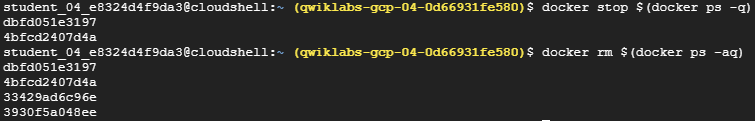
**Docker Compose**

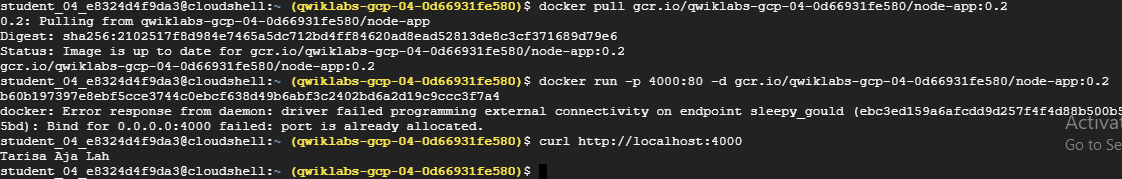
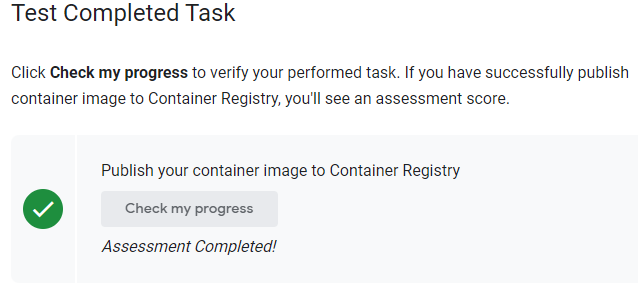
1. **Tujuan**

* Mahasiswa dapat mem-build, menjalankan, dan melakukan debug container Docker.
* Mahasiswa dapat mengambil image Docker dari Docker Hub dan Google Container Registry.
* Mahasiswa dapat menerapkan image Docker ke Google Container Registry.

1. **Praktikum**
2. **Persiapan**
3. Masuk ke lab
4. Mengambil token untuk mengakses lab
5. Mengisi biodata untuk mendapatkan token
6. Cek email dari qwicklab, setelah itu klik claim offer dan login akun google melalui link yang diberikan
7. Setelah itu mencari lab Introcution to Docker, kemudian start lab. Setelah di klik start lab akan muncul popup seperti di atas, klik saja lunch with 1 credit
8. Lab sudah mulai
9. Copy link ke tab samara agar akunya tidak tertindih dengan akun qwiklabs
10. **Hello Word**
11. Docker run hello-world
12. Docker – images
13. Docker run hello-world
14. Docker ps
15. Docker ps –a
16. **Build**
17. mkdir test && cd test
18. Create a Dockerfile
19. Create the node application
20. Docker build -t node-app:0.1 .
21. Run the following command to look at the images you built
22. **Run**
23. use this code to run containers based on the image you built
24. Open another terminal (in Cloud Shell, click the + icon), and test the server
25. Close the initial terminal and then run the following command to stop and remove the container:
26. You can look at the logs by executing docker logs [container\_id].
27. Edit app.js with a text editor of your choice (for example nano or vim) and replace "Hello World" with another string:
28. Build this new image and tag it with 0.2:
29. Run another container with the new image version. Notice how we map the host's port 8080 instead of 80. We can't use host port 4000 because it's already in use.
30. Test the containers:
31. And now test the first container you made:

1. **Debug**
2. You can look at the logs of a container using docker logs [container\_id]. If you want to follow the log's output as the container is running, use the -f option.
3. Open another terminal (in Cloud Shell, click the + icon) and enter the following command:
4. Look at the directory.

1. Exit the Bash session:
2. You can examine a container's metadata in Docker by using Docker inspect:
3. Use --format to inspect specific fields from the returned JSON. For example:
4. **Publish**
5. You can find your project ID by running:
6. Tag node-app:0.2. Replace [project-id] with your configuration..
7. Push this image to gcr. Remember to replace [project-id].
8. Stop and remove all containers:
9. You have to remove the child images (of node:6) before you remove the node image. Replace [project-id].

1. Pull the image and run it. Remember to replace the [project-id].
2. **Test Completed Task**
3. **Kesimpulan**

Setelah melakukan praktik seperti diatas, dapat disimpulkan bahwa mahasiswa dapat mem-build, menjalankan, dan melakukan debug container Docker. Mahasiwa juga dapat mengambil image Docker dari Docker Hub dan Google Container Registry dan juga dapat menerapkan image Docker ke Google Container Registry.