Cheat Sheet: Container Registry

Docker Hub

- Prerequisites:
  - Create a Docker Hub account if you don't already have one
  - Go to Docker Hub (https://hub.docker.com)
  - Click the **Register** button and create an account

Using a Container Registry

- In Cloud Shell:
  - From the **`external`** folder, run the following command:

    ```
    docker build -t your-docker-hub-id/external:v1.0 .
    ```

    Replace *your-docker-hub-id* with your docker hub account
  - From the **`internal`** folder, run the following command:

    ```
    docker build -t your-docker-hub-id/internal:v1.0 .
    ```

- Next, run the following commands to push your images to dockerhub:

```
docker login
docker push your-docker-hub-id/internal:v1.0
docker push your-docker-hub-id/external:v1.0
```

Remove All Local Containers and Images

- If you have any previous containers running, you will need to stop them
  - Or you will get a port number already in use error
- Below are a few commands to help you stop any containers:
  - List all Docker processes with: **`docker ps -a`**
  - Stop and remove all Docker processes:

    `docker stop <container_id>`

    `docker rm <container_id>`
  - List and delete all the local Docker images:

    `docker images`

    `docker rmi <image-id>`

Running the Containers from the Registry

- In Cloud Shell, run the case study directly from the container registry
  - Refer back to the last activity for the Docker run commands
  - Use the URL to the images in Docker Hub for the image names
  - For example, your run commands will look similar to:

```
docker run -d -p 8082:8082 -e SERVICE_PORT=8082 your-docker-hub-id/internal:v1.0

docker run -d -p 8080:8080 -e SERVER='http://localhost:8082' --network="host"
your-docker-hub-id/external:v1.0
```

- Test your app by previewing on port 8080

Success!

- **Congratulations**! You have successfully stored images in a container registry

Optional: Using Google Container Registry (gcr)

- In the Google Cloud Console, go to **Container Registry**
  - Enable the API if needed
- In Cloud Shell:
  - From `external` folder, run the following command:

```
docker build --tag gcr.io/$GOOGLE_CLOUD_PROJECT/external-image:v1.0 .
```

```
docker push gcr.io/$GOOGLE_CLOUD_PROJECT/external-image:v1.0
```

    - Type `Y` if asked to enable API
  - From `internal` folder, run the following command:

```
docker build --tag gcr.io/$GOOGLE_CLOUD_PROJECT/internal-image:v1.0 .
```

```
docker push gcr.io/$GOOGLE_CLOUD_PROJECT/internal-image:v1.0
```