ROITRAINING
MAXIMIZE YOUR TRAINING INVESTMENT

Activity:
Using a Container Registry

Using a Container Registry

- In the Google Cloud Console, click the Navigation menu, select **Container Registry**
  - Enable the API if needed
  - Click **Settings** on the left
  - Enable Vulnerability Scanning

Note: `$GOOGLE_CLOUD_PROJECT` is an environment variable containing your project ID that Cloud Shell sets automatically

- In Cloud Shell:
  - From the `events-api` folder, run the following commands:

```
docker build -t gcr.io/$GOOGLE_CLOUD_PROJECT/events-api:v1.0 .
docker push gcr.io/$GOOGLE_CLOUD_PROJECT/events-api:v1.0
```

  - From the `events-website` folder, run the following commands:

```
docker build -t gcr.io/$GOOGLE_CLOUD_PROJECT/events-website:v1.0 .
docker push gcr.io/$GOOGLE_CLOUD_PROJECT/events-website:v1.0
```

Investigate Vulnerability Scanning

- In the Google Cloud Console, go to **Container Registry**
  - Locate your containers
  - Click a container image and investigate the vulnerability scanner findings
  - If the scanner is not done, check back in a few minutes

Remove All Local Containers and Images

- If you have previous containers running locally, 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 image in the registry for the image names
  - For example, your run commands will look similar to:

```
docker run -d -p 8082:8082 gcr.io/$GOOGLE_CLOUD_PROJECT/events-api:v1.0

docker run -d -p 8080:8080 -e SERVER='http://localhost:8082' --network="host"
gcr.io/$GOOGLE_CLOUD_PROJECT/events-website:v1.0
```

Creating Another Version

- Using the Cloud Shell editor, edit the following file:
  `/eventsapp/events-website/views/layouts/default.hbs`
- Locate the `<h1>...</h1>` tag (approximately line 16) that looks similar to below:
  - Your name will be different

    ```
    <h1>Doug's Events App</h1>
    ```
- Add the text “Version 2.0” to the tag, for example:

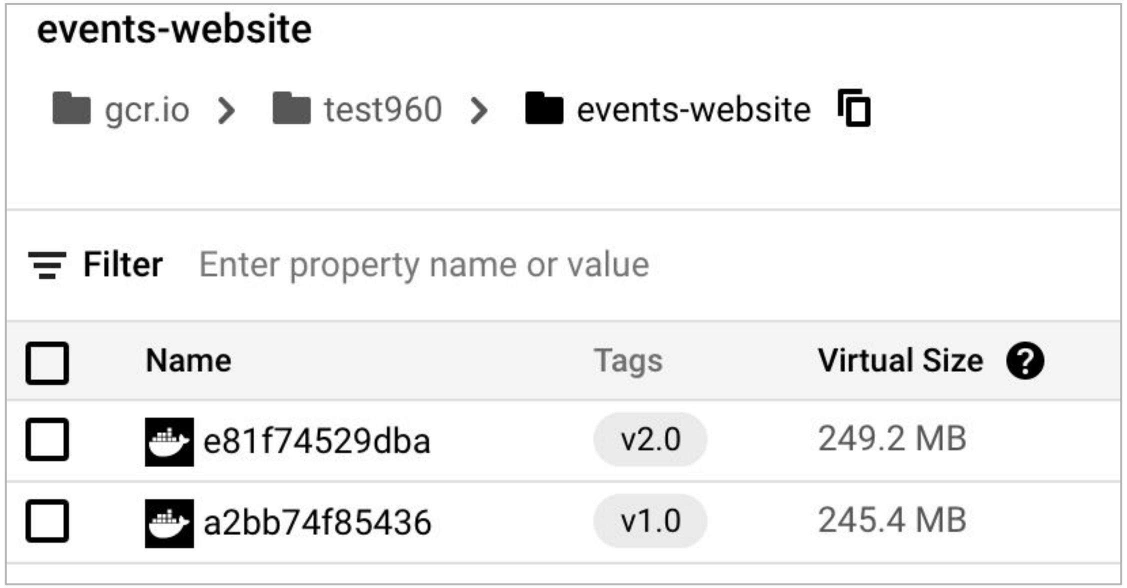
  ```
  <h1>Doug's Events App **Version 2.0**</h1>
  ```
- You can edit the `<title>` tag as well, but **be sure to edit the `<h1>`**

Creating Another Version (continued)

- In Cloud Shell:
  - From the events-website folder, run the following commands:

```
docker build -t gcr.io/$GOOGLE_CLOUD_PROJECT/events-website:v2.0 .
docker push gcr.io/$GOOGLE_CLOUD_PROJECT/events-website:v2.0
```

- In the Google Cloud Console, go to **Container Registry**
  - The events-website container should now have two versions
- Try stopping Version 1.0 and running Version 2.0



events-website

gcr.io > test960 > events-website

Filter Enter property name or value

□	Name	Tags	Virtual Size
□	e81f74529dba	v2.0	249.2 MB
□	a2bb74f85436	v1.0	245.4 MB

Optional: Using Docker Hub Container Registry

- You can optionally try this if you want to use Docker Hub
- Go to Docker Hub (https://hub.docker.com)
  - Create an account
- Rebuild your Docker images using your Docker ID
  - From the `events-api` folder:

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

  - From the `events-website` folder:

    ```
    docker build -t your-docker-hub-id/events-website:v2.0 .
    ```

- In Cloud Shell:

  ```
  docker login
  docker push your-docker-hub-id/events-api:v1.0
  docker push your-docker-hub-id/events-website:v2.0
  ```

Success

- **Congratulations**! You have successfully containerized the Events app