ROI Training

Lab:
Using a Container Registry

Using Docker Hub Container Registry

- Go to Docker Hub (https://hub.docker.com)
  - Log in with your account (or create an account if you did not already)
- Open your codespace if it has closed
- Rebuild your Docker images using your Docker ID in the tag
  - From a terminal in the `events-api` folder:

    ```
    cd /workspaces/eventsapp/events-api/
    docker build -t your-docker-hub-id/events-api:v1.0 .
    ```
  - From a terminal in the `events-website` folder:

    ```
    cd /workspaces/eventsapp/events-website/
    docker build -t your-docker-hub-id/events-website:v1.0 .
    ```

Using Docker Hub Container Registry (continued)

- From a terminal, authenticate to Docker Hub with:

  ```
  docker login -u your-docker-hub-id
  ```

  - Provide your Docker Hub password when prompted
- Push the containers to Docker Hub with:

  ```
  docker push your-docker-hub-id/events-api:v1.0
  docker push your-docker-hub-id/events-website:v1.0
  ```
- Reload the browser logged into Docker Hub and verify the images were pushed

Remove All Local Containers

- If you have 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>
```

Running the Containers from the Registry

- Run the case study directly from the container registry
  - The Docker run commands are the same as the last lab except the image name contains your docker hub ID

```
docker run -d -p 8082:8082 your-docker-hub-id/events-api:v1.0
```

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

Creating Another Version

- Using the codespaces editor, edit the following file:
  `/workspaces/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>
```

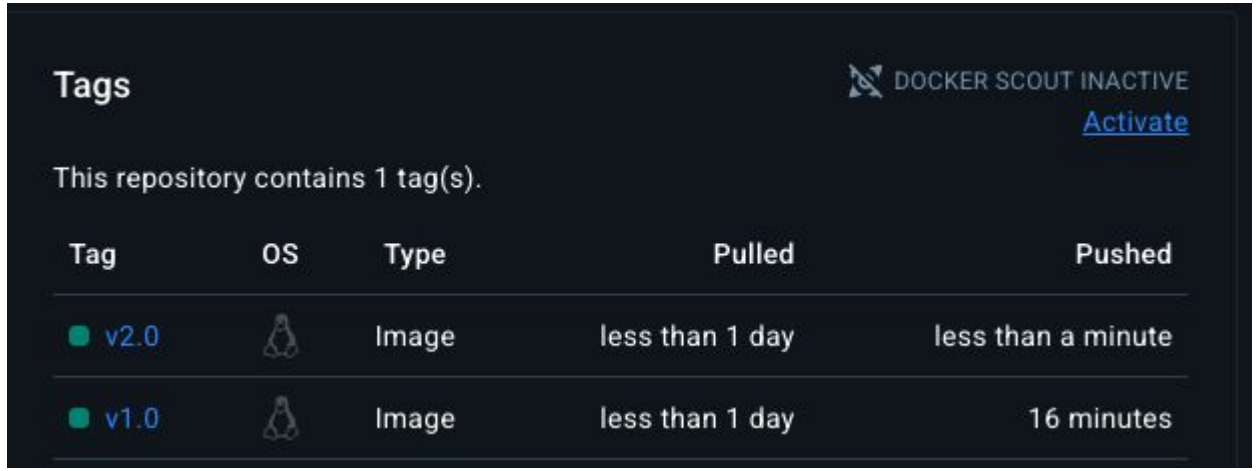
- You can edit the `<title>` tag as well, but **be sure to edit the `<h1>`**

Creating Another Version (continued)

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

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

- Reload the browser logged into Docker Hub
  - The `events-website` container should now have two versions
- From the terminal:
  - Try stopping Version 1.0 and running Version 2.0



Success

- **Congratulations**! You have successfully used a container registry

The Following Slides are for Reference Only

- Do **not** perform the steps on these slides now
- They contain the commands needed to use the AWS Elastic Container Registry instead of Docker Hub
  - These commands would require an AWS account and the AWS CLI configured on your system

Reference Only: Using a Container Registry

- In the AWS console, click the Navigation menu and select **Elastic Container Registry**
- Click the **Create Repository** button
  - Name the repo **events-api**
  - Set **Mutable**
  - Leave all other settings at the defaults
  - Click the **Create** button
- Do the same for the **events-website**

Reference Only: Using a Container Registry

Run the following command to authenticate the Docker CLI to container registry and push the image to the registry

NOTE: Items marked in red will need to be changed based on your account number and region. Do this from the events-api directory

```
docker login -u AWS -p $(aws ecr get-login-password --region us-east-1)
account#.dkr.ecr.us-east-1.amazonaws.com/events-api
```

```
docker build -t events-api .
```

```
docker tag events-api:latest
account#.dkr.ecr.us-east-1.amazonaws.com/events-api
```

```
docker push account#.dkr.ecr.region.amazonaws.com/events-api:latest
```

Reference Only: Using a Container Registry

Run the following command to authenticate the Docker CLI to container registry and push the image to the registry

Do this from the events-website directory

```
docker login -u AWS -p $(aws ecr get-login-password --region us-east-1)
account#.us-east-1.amazonaws.com/events-website

docker build -t events-website .


docker tag events-website:latest
account#.dkr.ecr.us-east-1.amazonaws.com/events-website

docker push account#.dkr.ecr.region.amazonaws.com/events-website:latest
```

Reference Only: Running Containers from the Registry

- In the deployment server, run the case study directly from the container registry
  - For example, your run commands will look similar to:

```
docker run -d -p 8082:8082 account#.dkr.ecr.us-east-1.amazonaws.com/events-api:latest
```

```
docker run -d -p 8080:8080 -e SERVER='http://localhost:8082' --network="host"
account#.dkr.ecr.us-east-1.amazonaws.com/events-website:latest
```