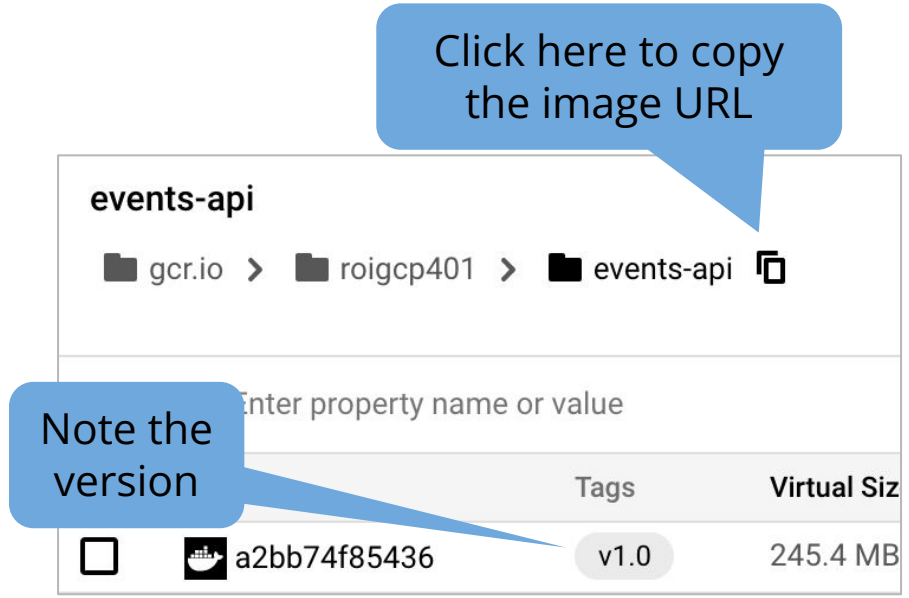
ROI TRAINING
MAXIMIZE YOUR TRAINING INVESTMENT

Activity:
Creating Kubernetes Deployments

Case Study Container Images

- In this lab, you will be deploying the container images you created earlier
  - `events-api` and `events-website`
- You need to know the location of these images in the container registry
- In the Google Cloud Console, click the Navigation menu, select **Container Registry**
  - Click your **`events-api`** image
  - Click the button to copy the image URL
  - Save this URL in a text file somewhere
  - To the end of the URL, add `:` and the version
- Your URL should look similar to this:
  `gcr.io/projectid/events-api:v1.0`



Case Study Container Images (continued)

- Perform the same steps you just did for the `events-website` image
  - You should have two versions of this image, use `v1.0`
  - Save this URL as well—you need the URLs later
- Your URL should look similar to this:
  `gcr.io/projectid/events-website:v1.0`

Creating the `api-deployment.yaml`

- In Cloud Shell, change into your `kubernetes-config` folder:
  `cd ~/eventsapp/kubernetes-config`
- In the `kubernetes-config` folder, create a new file named `api-deployment.yaml`
  - You can create the file with the Cloud Shell editor
  - Copy the contents of the file from the next slide
    - Replace the container image with the URL saved earlier

api-deployment.yaml

- Copy/paste the contents of this file
  - Replace the container image with the URL saved earlier

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
 labels:
   app: events-api
 name: events-api
spec:
 replicas: 1
 selector:
   matchLabels:
     app: events-api
     ver: v1.0
 template:
   metadata:
     labels:
       app: events-api
       ver: v1.0
   spec:
     containers:
     - image: PUT-EVENTS-API-IMAGE-URL-HERE
       name: events-api
       ports:
       - containerPort: 8082
```

Deploying the `api-deployment.yaml`

- Deploy the `events-api` deployment, in cloud shell from the `kubernetes-config` folder, run:

  ```
  kubectl apply -f api-deployment.yaml
  ```

- Verify it was deployed:

  ```
  kubectl get deployments
  kubectl get pods
  ```

Creating the `web-deployment.yaml`

- In Cloud Shell, in your `kubernetes-config` folder, create a new file named `web-deployment.yaml`
  - For the contents, copy the contents of your `api-deployment.yaml`
  - You can create this file in the editor or just copy the file at the prompt
- Edit the new `web-deployment.yaml` file
  - Change all occurrences of "`events-api`" to "`events-web`"
  - Change the image URL to your `events-website` image URL copied earlier
  - Change the container port to `8080`

Deploying the `web-deployment.yaml`

- Deploy the `events-web` deployment:
  ```
  kubectl apply -f web-deployment.yaml
  ```
- Verify it was deployed:
  ```
  kubectl get deployments
  kubectl get pods
  ```

- If your `web-deployment.yaml` is not working correctly, there is an example of how this file should look at the end of this document

Investigating ReplicaSets

- Kubernetes deployments use ReplicaSets to manage pods
  - Even if there is only one pod
- From the list of running pods, copy one of the pod names
- Try deleting that pod:
  `kubectl delete pod <paste pod name here>`
- View the pods again:
  `kubectl get pods`
- You should see a new pod was started to replace the deleted one
- Feel free to experiment with deleting another pod

Investigating ReplicaSets (continued)

- Edit the `web-deployment.yaml`
  - Change the replicas to `3` and re apply the file:

    ```
    kubectl apply -f web-deployment.yaml
    ```

- View the pods again:

  ```
  kubectl get pods
  ```

- You should see two new pods were started to make a total of three
- Edit the `web-deployment.yaml` and change the replicas back to `1`
  - Apply the file:

    ```
    kubectl apply -f web-deployment.yaml
    ```

- View the pods again:

  ```
  kubectl get pods
  ```

Clean Up

- There is no need to clean up the deployments or pods
- Leave them running on your cluster

Success

- **Congratulations**! You have successfully created Kubernetes deployments
  - Created a deployment yaml for both containers of the case study
  - Deployed the deployments to the cluster
  - Experimented with deleting pods in a deployment

web-deployment.yaml

- Here is an example file if you need it

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
 labels:
   app: events-web
 name: events-web
spec:
 replicas: 1
 selector:
   matchLabels:
     app: events-web
     ver: v1.0
 template:
   metadata:
     labels:
       app: events-web
       ver: v1.0
   spec:
     containers:
     - image: PUT-URL-HERE
       name: events-web
       ports:
       - containerPort: 8080
```