Activity: Performing Rolling Updates and Blue/Green Deployments

Kubernetes `config` Folder

- Remember, anytime you are applying Kubernetes YAML files, you must be in the folder containing the YAML files
  - Open Cloud Shell and change into the correct folder:

```
cd ~/eventsapp/kubernetes-config/
```

Rolling Updates

- To make the rolling update more interesting, you will increase the number of replicas of your `events-web` pods:
  - Modify the `web-deployment.yaml` to have `4` replicas
  - Apply the `web-deployment.yaml` file
  - Get the pods and verify `4` are running

```
$ kubectl get pods
NAME                          READY   STATUS    RESTARTS   AGE
events-api-b7c5f7ccf-vg95r    1/1     Running   0          38m
events-web-5d5485cd7d-4ptd2   1/1     Running   0          6s
events-web-5d5485cd7d-4snn2   1/1     Running   0          6s
events-web-5d5485cd7d-68cvl   1/1     Running   0          6s
events-web-5d5485cd7d-72t5c   1/1     Running   0          23m
```

  - Test the application by loading the `events-web-svc EXTERNAL-IP` in a browser to ensure it is still working (keep the browser tab open)

Rolling Updates (continued)

- You will now upgrade to Version 2.0 of the `events-web`
  - Modify the `web-deployment.yaml` and change the version of the container image from `v1.0` to `v2.0`

```
- image: gcr.io/test960/events-website:v2.0
```

  - You don't have to worry about changing any of the labels at this point

Rolling Updates (continued)

- Open a new Cloud Shell tab by clicking the **+**
- In the new tab, run the following command to watch the pods:
  `kubectl get pods -w`
- Switch back to the first Cloud Shell tab
- Run the following command and then quickly switch to the tab that is running the watch command:
  `kubectl apply -f web-deployment.yaml`
- You should see the pods being updated 25% at a time

Testing the Rolling Update

- Test the application again
  - Keep reloading the app until you see Version 2.0
  - If you keep reloading, it may toggle between Version 1.0 and Version 2.0
  - After a few seconds, it should only return Version 2.0

Rollback

- You will now roll back to Version 1.0 of the `events-web`
  - Verify in one tab you are still running the command to watch the pods:
    ```
    kubectl get pods -w
    ```
  - In the other Cloud Shell tab, run the following command:
    ```
    kubectl rollout undo deployments/events-web
    ```
  - Switch to the tab running the watch command
    - You should see the pods being rolled back 25% at a time
- Feel free to perform the update and roll back again if you want to see it in action again

Blue/Green Deployments

- Ensure the `events-web` is back on Version 1.0
  - Modify the `web-deployment.yaml` and change the version of the container image back to `v1.0`
  - Apply the `web-deployment.yaml` file
- You will now make another copy of the `web-deployment.yaml` file
  - In Cloud Shell, execute the following command to copy the file:

    ```
    cp web-deployment.yaml web-deployment-v2.yaml
    ```
  - Edit the new `web-deployment-v2.yaml` file
  - Modify the name from "`name: events-web`" to "`name: events-web-v2.0`"
  - Modify the "`ver: v1.0`" labels to "`ver: v2.0`" (two places)
  - Modify the container image URL from `v1.0` to `v2.0`

The `web-deployment-v2.yaml`

- Your file should look like this (your `projectID` will be different):

```
apiVersion: apps/v1
kind: Deployment
metadata:
 labels:
   app: events-web
 name: events-web-v2.0
spec:
 replicas: 4
 selector:
   matchLabels:
     app: events-web
     ver: v2.0
 template:
   metadata:
     labels:
       app: events-web
       ver: v2.0
```

```
  spec:
    containers:
    - image: gcr.io/ProjectID/events-website:v2.0
      name: events-web
      imagePullPolicy: "Always"
      ports:
      - containerPort: 8080
      env:
      - name: SERVER
        value: "http://events-api-svc:8082"
```

Create the `events-web-2.0` Deployment

- Apply the new `web-deployment-v2.yaml`

  ```
  kubectl apply -f web-deployment-v2.yaml
  ```

- Get the deployments and pods

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

```
$ kubectl get deployments
NAME              READY   UP-TO-DATE   AVAILABLE   AGE
events-api        1/1     1            1           88m
events-web        4/4     4            4           88m
events-web-v2.0   4/4     4            4           9m43s
$
$ kubectl get pods
NAME                               READY   STATUS    RESTARTS   AGE
events-api-b7c5f7ccf-vg95r         1/1     Running   0          89m
events-web-5d5485cd7d-29wjr        1/1     Running   0          37m
events-web-5d5485cd7d-7k728        1/1     Running   0          36m
events-web-5d5485cd7d-82lpl        1/1     Running   0          37m
events-web-5d5485cd7d-w7vvh        1/1     Running   0          37m
events-web-v2.0-7cb5697d84-ggl8c   1/1     Running   0          10m
events-web-v2.0-7cb5697d84-hsqdc   1/1     Running   0          10m
events-web-v2.0-7cb5697d84-qwb4q   1/1     Running   0          10m
events-web-v2.0-7cb5697d84-vgwrg   1/1     Running   0          10m
```

- In the web browser tab running the app, verify you still see Version 1.0

Switch the Load Balancer Labels

- Perform the Blue/Green deployment
  - Edit the `web-service.yaml`
  - In the selector, change "`ver: v1.0`" to "`ver: v2.0`"
  - Apply the `web-service.yaml`
  - Reload the app in the browser until you see Version 2.0
- Rollback
  - Edit the `web-service.yaml`
  - In the selector, change "`ver: v2.0`" to "`ver: v1.0`"
  - Apply the `web-service.yaml`
  - Reload the app in the browser until you see Version 1.0

Canary Release

- Edit the `web-deployment-v2.yaml` file
  - Modify the replicas from `4` to `1`
  - Apply the `web-deployment-v2.yaml` file
- Edit the `web-service.yaml`
  - In the selector, remove the entire "`ver: v1.0`" line so the entire selector is:

```
selector:
  app: events-web
```

  - This will cause the load balancer to select all pods not matter the version
  - Apply the `web-service.yaml` file

Testing the Canary Release

- Reload the app in the browser
  - It is now load balancing between four copies of Version 1.0 and one copy of Version 2.0
  - If you are having trouble getting both versions, you can try running this script in Cloud Shell:

```
while true; do curl http://EXTERNAL-IP/ | grep "version 2.0" && sleep 1; done;
```

  - Replace the `EXTERNAL-IP` with your `EXTERNAL-IP`
  - Whenever “version 2.0” is found, it will be displayed

Clean Up

- Edit the `web-service.yaml`
  - In the selector, put the "`ver: v1.0`" line back in so the entire selector is as shown here:
  - Apply the `web-service.yaml` file
- Edit the `web-deployment.yaml`
  - Set the replicas to `2`
  - Apply the `web-deployment.yaml` file
- Delete the Version 2.0 deployment with:
  `kubectl delete deployment events-web-v2.0`
- Verify the app still works in the browser

```
selector:
  app: events-web
  ver: v1.0
```

Success

- **Congratulations**! You have successfully updated Kubernetes workloads
  - Performed a rolling update of your pods
  - Implemented a Blue/Green deployment
  - Created a simple canary release