Lab:
Creating Kubernetes Deployments

Verify Minikube Is Running

- From a codespaces terminal, check if Minikube is still running with:

  ```
  minikube status
  ```

- If it is stopped, start it again with:

  ```
  minikube start
  ```

  - It will take 1-2 minutes to start the cluster

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 Docker Hub
  - They should be in this form:
  - `your-docker-hub-id/events-api:v1.0`
  - `your-docker-hub-id/events-website:v1.0`

Creating the `api-deployment.yaml`

- From a codespaces terminal, change into your `kubernetes-config` folder:
  ```
  cd /workspaces/eventsapp/kubernetes-config
  ```
- In the `kubernetes-config` folder, create a new file named `api-deployment.yaml`
  - You can use the visual editor, or vi, or nano
  - Copy the contents of the file from the next slide
    - Correct the container image value and use your Docker Hub ID

api-deployment.yaml

- Copy/paste the contents of this file
  - Correct the container image value and use your ID

```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: your-docker-hub-id/events-api:v1.0
       name: events-api
       ports:
       - containerPort: 8082
```

Deploying the `api-deployment.yaml`

- Deploy the `events-api` deployment, run:

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

- Verify it was deployed:

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

- You can retrieve more information about a pod (i.e. what node it is on)

```
kubectl describe pod <pod-name>
```

Creating the `web-deployment.yaml`

- 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

Syncing the Changes to Git

- Commit these changes to your Git repository
  - On the left side, click the **Source Control** button
  - Be sure ALL changes are staged by clicking in the + button
  - Type a commit message of: **`Added Events app start code`** and click the **Commit** button
  - Press the **Sync Changes** button and press **OK** to push the changes
- The code has now been saved to your **eventsapp** Git repository created earlier

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
```