ROI TRAINING
MAXIMIZE YOUR TRAINING INVESTMENT

Activity:
Modifying the Events-API App to use the Database

Events-API Environment Variables

- The Events-API app was written to look for several environment variables
  - `DBHOST, DBUSER, DBPASSWORD, DBDATABASE`
  - These variables tell it how to connect to a database
  - If these variables do not exist, the event data is stored in a local array
    - That is what has been happening so far in the class
- Now that we have a MariaDB running, we just need to set the environment variables for the database
  - The app will then start storing the event data in the database

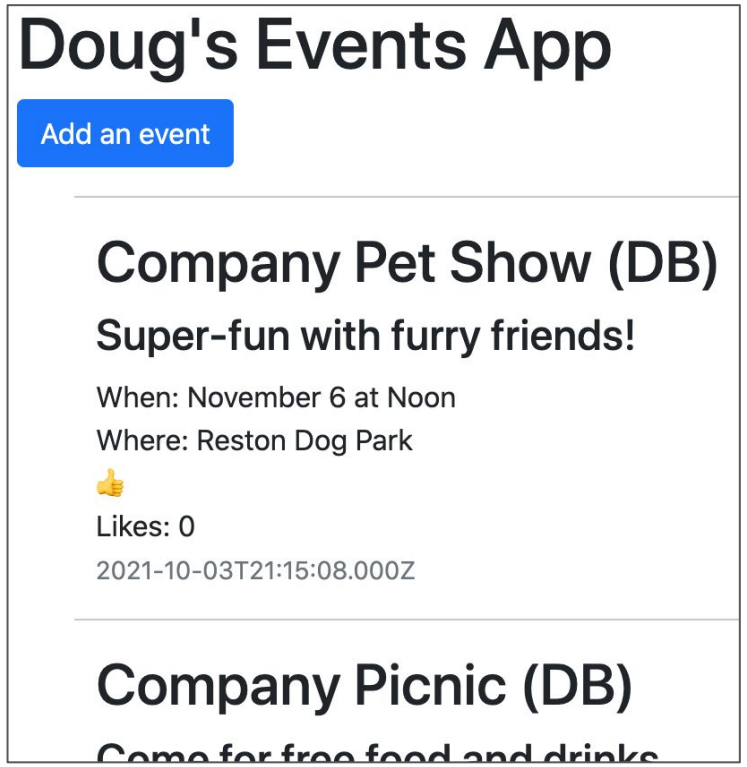
Modify the `api-deployment.yaml`

- Edit your `api-deployment.yaml` file
  - At the end of the file, add the highlighted lines shown here
    - That sets 4 environment variables
  - Be sure to note the indentation
- Then reapply the file:
  `kubectl apply -f api-deployment.yaml`
- This should cause the pods for this deployment to be replaced

```
.....
  spec:
    containers:
    - image: EVENTS-API-IMAGE-URL-HERE
      name: events-api
      ports:
      - containerPort: 8082
        env:
        - name: DBHOST
          value: "database-server-mariadb.default"
        - name: DBUSER
          value: "root"
        - name: DBPASSWORD
          valueFrom:
            secretKeyRef:
              name: database-server-mariadb
              key: mariadb-root-password
        - name: DBDATABASE
          value: "events_db"
```

Testing the Database

- List the pods and verify the events-api pod(s) were replaced:
  `kubectl get pods`
- Test the application by viewing the External IP in a browser
  - If you need the IP again: `kubectl get service`
- The DB initialization job added 2 rows to the database that have “(DB)” in their title
  - This is just so you can verify the DB is working
- Try adding some new events



Experiment with Replicas

- Scale the `events-web` service:
  - Modify the `web-deployment.yaml` to have `3` replicas
  - Apply the file and test the application
  - Everything should still work fine
- Scale the `events-api` service:
  - Modify the `api-deployment.yaml` to have `3` replicas
  - Apply the file and test the application
  - Everything should still work fine
  - This is because the `events-api` service is now storing state in a database

Clean Up

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

Success

- **Congratulations**! You have successfully added a database to the Kubernetes services