aws

ROI Training

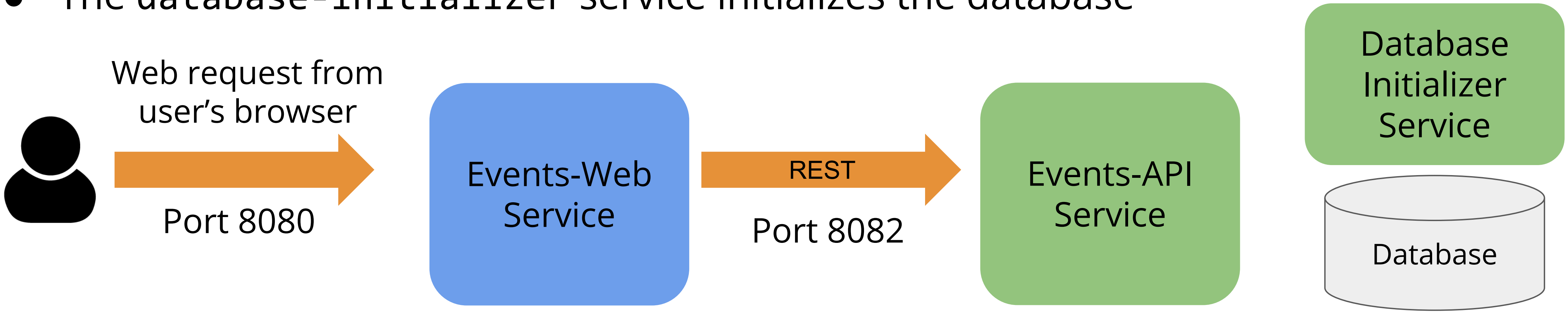
Lab:
Running the Case Study App

The Case Study App

- The case study app is a simple app to store and view events
  - Provides the basis of a simple app a company could use to announce events, news, and important messages
  - Written in Node.js as three microservices
- This course focuses on containering the application and running it on Kubernetes
  - This course does not cover developing/coding the application
  - The code for the running application is provided for you and is in an existing Git repository
    - You will copy the code to your Git repository in this lab

About the Case Study App

- The Events App is comprised of three microservices
- The `events-api` service receives REST requests on port 8082
  - Saves the event data
  - And returns event data
- The `events-website` service is a web frontend
  - Displays JSON data from `events-api` using an HTML template
- The `database-initializer` service initializes the database

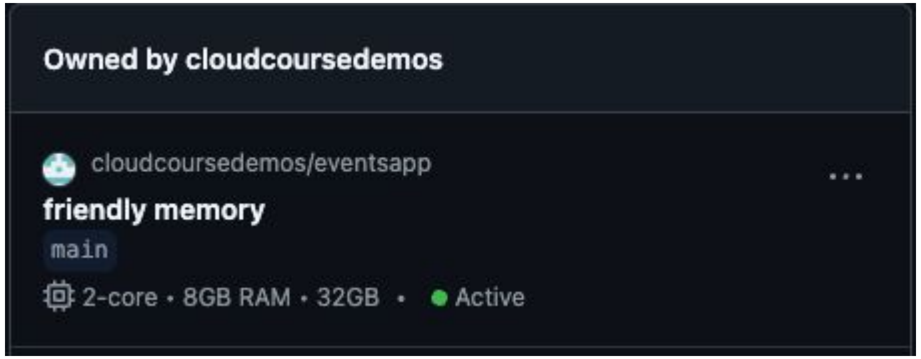


A Database Will Be Added Later

- Currently, the app does not use a database
  - The database will be added later in the course
  - The `database-initializer` service will not be used until then
- For now, the `events-api` service just stores the events data in a local array
  - Which means the data is not persistent
  - You may see a connection refused or fatal error
    - It will be corrected later

Opening the Codespace

- You will perform labs in the GitHub codespace created in the last lab
- If you closed the codespace tab or it has timed out, reopen it with:
  - From your GitHub repo, click the navigation menu and select **Codespaces**
  - Scroll down to the section **Owned by <your-github-account>**
  - Click the codespace name for the **eventsapp** repo to open the codespace
    - It will have a randomly generated name
    - You can rename it by clicking the **...**



Owned by cloudcoursedemos

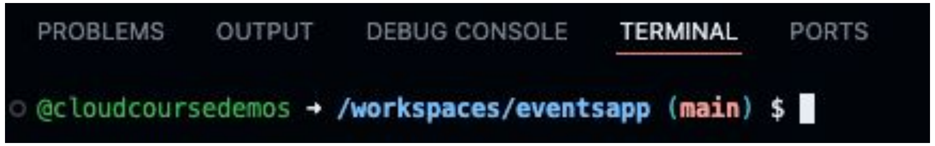
cloudcoursedemos/eventsapp

friendly memory

main

2-core • 8GB RAM • 32GB • ● Active

- Verify the terminal window is in the `/workspaces/eventsapp` folder
  - If not, change to that folder: **`cd /workspaces/eventsapp`**



```
@cloudcoursedemos → /workspaces/eventsapp (main) $
```

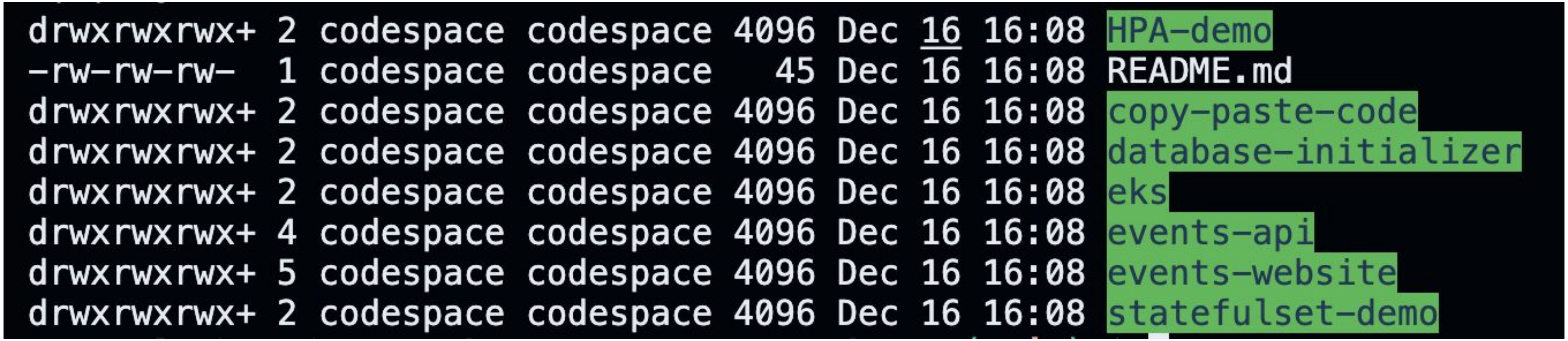
Download the Events App Code

- Run the following commands to copy the events app code from an existing Git repo into your codespace instance

```
git remote add eventsapp-start https://github.com/courserepos/eventsappstart
git fetch eventsapp-start
git checkout eventsapp-start/main -- .
```

- Verify several folders now exist

```
ls -l
```



```
drwxrwxrwx+ 2 codespace codespace 4096 Dec 16 16:08 HPA-demo
-rw-rw-rw-  1 codespace codespace   45 Dec 16 16:08 README.md
drwxrwxrwx+ 2 codespace codespace 4096 Dec 16 16:08 copy-paste-code
drwxrwxrwx+ 2 codespace codespace 4096 Dec 16 16:08 database-initializer
drwxrwxrwx+ 2 codespace codespace 4096 Dec 16 16:08 eks
drwxrwxrwx+ 4 codespace codespace 4096 Dec 16 16:08 events-api
drwxrwxrwx+ 5 codespace codespace 4096 Dec 16 16:08 events-website
drwxrwxrwx+ 2 codespace codespace 4096 Dec 16 16:08 statefulset-demo
```

Running the `events-api` Service

- Change into the `events-api` folder

```
cd events-api
```

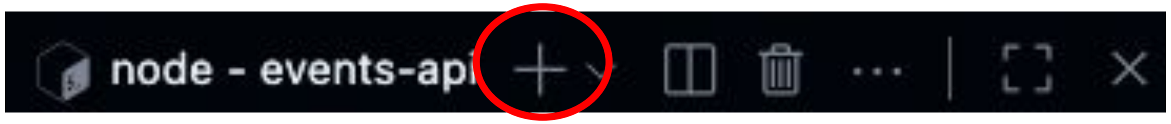
- Run the following commands to install dependencies and run the service (don't worry about any warnings or vulnerabilities at this stage):
  - You may have to click **Allow** to allow copy/paste into codespaces

```
npm install
node server.js
```

- You will see a dialog box saying your app is running on port 8082
  - Click the **X** to close this dialog (or it will close on its own)
- Leave the service running

Running the `events-website` Service

- Open a new terminal session by clicking the **+** near the top right of the terminal



- In the new session:

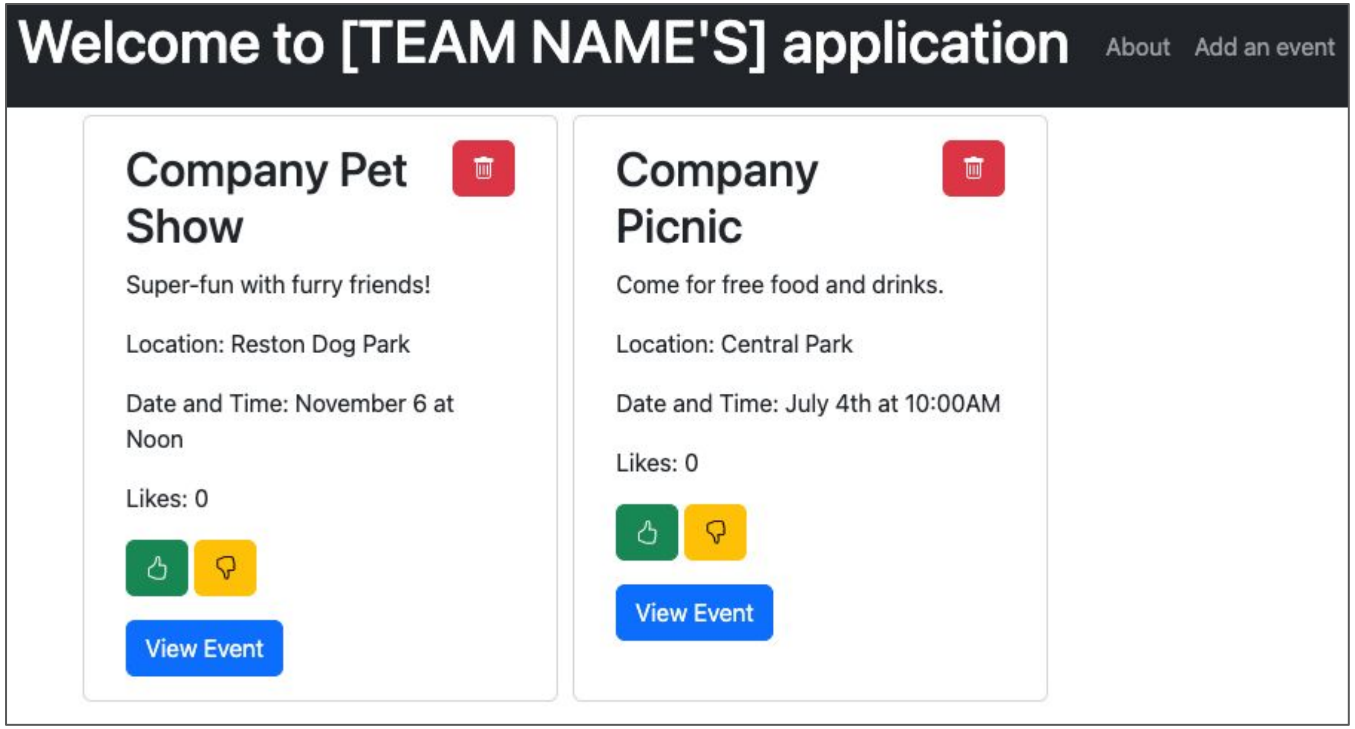
```
cd events-website/
npm install
npm start
```

  (don't worry about any warnings or vulnerabilities at this stage)

- Both services should now be running
- In the dialog box saying your app is running on port 8080, click **Open in Browser**

Testing the Events App

- The app has two sample events hard coded as shown below
- Use the **Add an event** button to add additional events



Welcome to [TEAM NAME'S] application

Company Pet Show

Super-fun with furry friends!

Location: Reston Dog Park

Date and Time: November 6 at Noon

Likes: 0

Company Picnic

Come for free food and drinks.

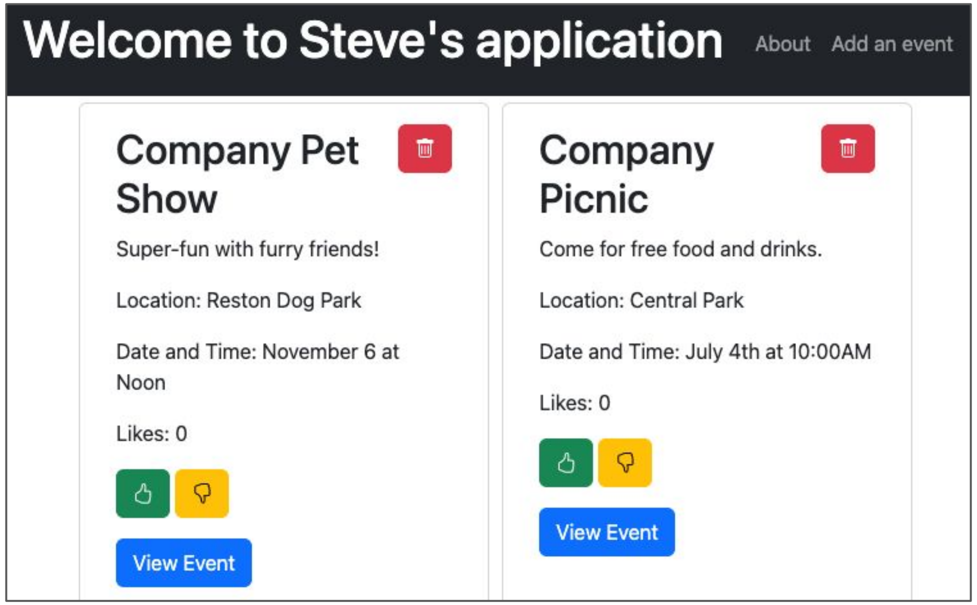
Location: Central Park

Date and Time: July 4th at 10:00AM

Likes: 0

Customizing the `events-website`

- Edit the following file using the codespaces editor and change `[Your Name]` to your name in **two** places:
  `events-website/views/layouts/default.hbs`
- Reload the site to verify the changes



Welcome to Steve's application

Company Pet Show

Super-fun with furry friends!

Location: Reston Dog Park

Date and Time: November 6 at Noon

Likes: 0

Company Picnic

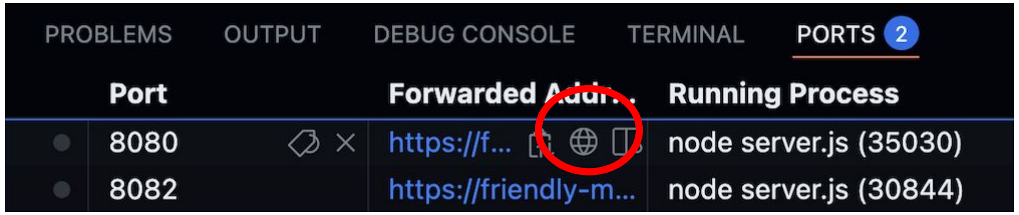
Come for free food and drinks.

Location: Central Park

Date and Time: July 4th at 10:00AM

Likes: 0

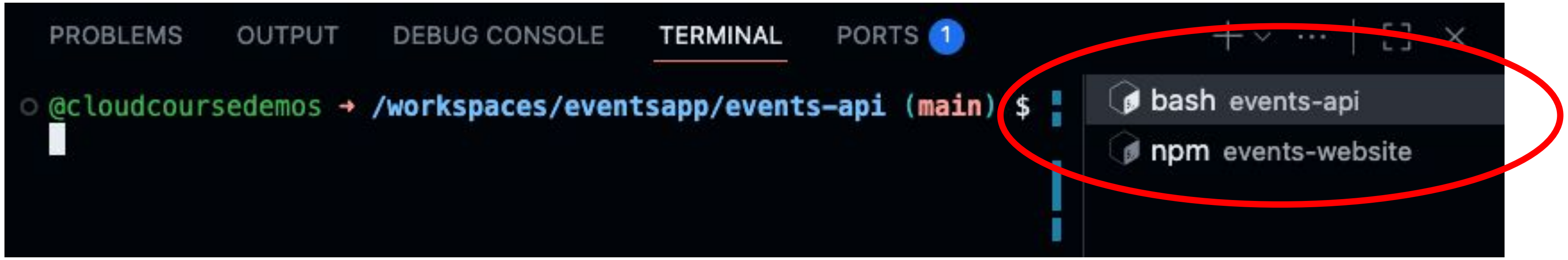
- If you closed the browser displaying the app, it can be opened again by:
  - Click the **Ports** tab, hover over the line for port 8080, and click the globe



Port	Forwarded Addr...	Running Process
8080	https://f...	node server.js (35030)
8082	https://friendly-m...	node server.js (30844)

Stopping the App

- You can switch between the two terminal sessions
  - From the **Terminal** tab
  - Click the terminal sessions in the right pane

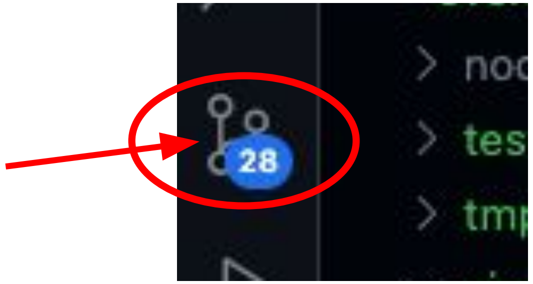


```
@cloudcoursedemos → /workspaces/eventsapp/events-api (main) $
```

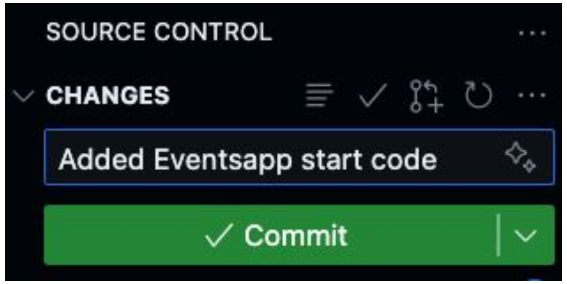
- Stop the services by pressing **CTRL+C** in both sessions
  - Don’t worry if you see an error or a “no connection” message in the events-api terminal
  - That is because we are not using a database yet

Syncing the Changes to Git

- Commit these changes to your Git repository
  - On the left side, click the **Source control** button



  - Most changes should be staged automatically
    - Be sure ALL changes are staged by clicking the **+** button
  - Type a commit message of: **`Added Events app start code`** and click the **Commit** button



Added Eventsapp start code

  - 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 downloaded and run the case study app