ROI TRAINING
MAXIMIZE YOUR TRAINING INVESTMENT

Activity:
Building Docker Images

ROI TRAINING
MAXIMIZE YOUR TRAINING INVESTMENT

Dockerfile

- Open Cloud Shell if it has closed
- The following `Dockerfile` will work for both the api and website services
  - Create a file called **`Dockerfile`** in both the `events-api` and `events-website` folders
- You can use the Cloud Shell visual editor to create these files

```
# Use Google base image for NodeJS
FROM launcher.gcr.io/google/nodejs

# Copy application code.
COPY . /app/

# Change the working directory
WORKDIR /app

# Install dependencies.
RUN npm install

# Start the Express app
CMD ["node", "server.js"]
```

.dockerignore

- Create a file called **.dockerignore** in both the `events-api` and `events-website` folders:

```
node_modules
npm-debug.log
```

  - You can create this file using the Cloud Shell editor
  - Be sure the name starts with a "." and is all lowercase
- The Cloud Shell editor does not display files starting with a "." by default
  - Use **View | Toggle hidden files** to show them

Build

- To build `events-api`, from a cloud shell terminal in the **`events-api`** folder:
  ```
  docker build . -t events-api:v1.0
  ```
- To build `events-website`, from a cloud shell terminal in the **`events-website`** folder:
  ```
  docker build . -t events-website:v1.0
  ```
- To view the Docker images just built:
  ```
  docker images
  ```

Run Locally

- To run `events-api`:

  ```
  docker run -d -p 8082:8082 events-api:v1.0
  ```

- To run `events-website`:

  ```
  docker run -d -p 8080:8080 -e SERVER='http://localhost:8082' --network="host" events-website:v1.0
  ```

- Test your app by previewing on port 8080
- Other commands to try:
  - `docker images`
  - `docker ps -a`
  - `docker stop <ContainerID>`
  - `docker rm <ContainerID>`

Success!

- **Congratulations**! You have successfully containerized the Events app