

Professional Documentation: Introduction to Docker Lab

(Simplified for Technical and Non-Technical Readers)

What is Docker?

Think of Docker as a **digital shipping container** for software. It packages an app with all its parts (code, settings, libraries) into a standardized "container" that runs *the same way anywhere* – like a physical shipping container works on ships, trains, or trucks.

Key Terms Simplified:

- **Container:** A lightweight, ready-to-run software package.
 - **Image:** A blueprint to build containers (like a recipe).
 - **Docker Hub:** Public library of pre-built images (e.g., `hello-world`).
 - **Artifact Registry:** Google Cloud's private storage for your custom images.
-

Lab Walkthrough

Task 1: Run Your First Container

Goal: Test if Docker works.

Steps:

1. In Cloud Shell, run:
2. `docker run hello-world`
 - **What Happened?:**
 - Docker downloaded the `hello-world` image from Docker Hub.
 - It created a container from the image.
 - The container printed "Hello from Docker!" → Success!

Non-Technical Insight:

Like downloading an app from an app store and opening it.

The screenshot shows the Google Cloud Platform dashboard for the project 'qwiklabs-gcp-00-6bfa9df86695'. The left sidebar includes sections for Cloud Hub, Cloud overview, and Solutions, along with a PINNED PRODUCTS section for 'Pin your top products here' and a PRODUCTS section listing Billing, IAM & Admin, Marketplace, APIs & Services, Vertex AI, Compute Engine, Kubernetes Engine, and Cloud Storage. The main content area features a 'Project info' card with details like Project name (qwiklabs-gcp-00-6bfa9df86695), Project number (262529971764), and Project ID (qwiklabs-gcp-00-6bfa9df86695). It also includes an 'ADD PEOPLE TO THIS PROJECT' section and a 'Go to project settings' link. To the right, there are several cards: 'Compute Engine' showing CPU utilization (0-100%) with a current value of 13.49% at 12:30 PM; 'Google Cloud Platform status' showing 'All services normal' with a 'Go to Cloud status dashboard' link; 'Billing' showing estimated charges for USD \$0.00 for the period Jul 1 – 22, 2025, and a 'Take a tour of billing' link; 'Monitoring' showing options to 'Create my dashboard', 'Set up alerting policies', and 'Create uptime checks'; and an 'APIs' card showing requests per second (0.060/s). The bottom of the dashboard has a 'VIEW ALL PRODUCTS' link and a 'View all dashboards' link.

The screenshot shows the Google Cloud Platform dashboard for the project `qwiklabs-gcp-00-6bfa9df86695`. The left sidebar lists various services: Cloud Hub, Cloud overview, Solutions, Recently visited (with a NEW badge), PINNED PRODUCTS (empty), and PRODUCTS (empty). The main content area is divided into several sections:

- Project info**: Displays the project name (`qwiklabs-gcp-00-6bfa9df86695`), project number (`529971764`), project ID (`qwiklabs-gcp-00-6bfa9df86695`), and settings.
- Compute Engine**: Shows CPU utilization over time, with a current value of 16.78%.
- Google Cloud Platform status**: Shows all services as normal and provides a link to the Cloud status dashboard.
- Billing**: Shows estimated charges of USD \$0.00 for the period July 1 - 22, 2025, and a link to take a tour of billing.
- Cloud Shell**: A terminal window titled `(qwiklabs-gcp-00-6bfa9df86695) x` showing the command line interface for the project.

Google Cloud

Cloud Hub

Cloud overview

Solutions

Recently visited

PINNED PRODUCTS

Billing

IAM & Admin

Marketplace

APIs & Services

Vertex AI

Compute Engine

Kubernetes Engine

VIEW ALL PRODUCTS

Google Cloud

qwיקlabs-gcp-00-6bfa9df86695

Search (/) for resource

Search

DASHBOARD

ACTIVITY

RECOMMENDATIONS

CUSTOMIZE

Project info

Project name: qwík labs-gcp-00-6bfa9df86695

Project number: 262529971764

Compute Engine

CPU (%): 100%

Google Cloud Platform status

All services normal

Go to Cloud status dashboard

Welcome to Cloud Shell! Type "help" to get started.

Your Cloud Platform project in this session is set to **qwík labs-gcp-00-6bfa9df86695**.

Use "gcloud config set project [PROJECT_ID]" to change to a different project.

```
student_01_8579d88be1c2@cloudshell:~ (qwík labs-gcp-00-6bfa9df86695)$ gcloud auth list
```

Credentialed Accounts

```
ACTIVE: *
```

```
ACCOUNT: student-01-8579d88be1c2@qwík labs.net
```

To set the active account, run:

```
$ gcloud config set account 'ACCOUNT'
```

```
student_01_8579d88be1c2@cloudshell:~ (qwík labs-gcp-00-6bfa9df86695)$ gcloud config list project
```

```
[core]
```

```
project = qwík labs-gcp-00-6bfa9df86695
```

Your active configuration is: [cloudshell-29428]

```
student_01_8579d88be1c2@cloudshell:~ (qwík labs-gcp-00-6bfa9df86695)$
```

<http://docs.docker.com/get-started/>

```
student_01_8579d88be1c2@cloudshell:~ (qwík labs-gcp-00-6bfa9df86695)$ docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
hello-world	latest	74cc54e27dc4	6 months ago	10.1kB

```
student_01_8579d88be1c2@cloudshell:~ (qwík labs-gcp-00-6bfa9df86695)$
```

<https://cloud.google.com/compute/docs/quickstart-linux>

```
student_01_8579d88be1c2@cloudshell:~ (qwík labs-gcp-00-6bfa9df86695)$ docker run hello-world
```

```
student_01_8579d88be1c2@cloudshell:~ (qwík labs-gcp-00-6bfa9df86695)$ docker run hello-world
```

Hello from Docker!

This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:

1. The Docker client contacted the Docker daemon.
2. The Docker daemon pulled the "hello-world" image from the Docker Hub. (amd64)
3. The Docker daemon created a new container from that image which runs the executable that produces the output you are currently reading.
4. The Docker daemon streamed that output to the Docker client, which sent it to your terminal.

To try something more ambitious, you can run an Ubuntu container with:

```
$ docker run -it ubuntu bash
```

Share images, automate workflows, and more with a free Docker ID:

<https://hub.docker.com/>

For more examples and ideas, visit:

<https://docs.docker.com/get-started/>

```
student_01_8579d88be1c2@cloudshell:~ (qwík labs-gcp-00-6bfa9df86695)$
```

<https://cloud.google.com/compute/docs/quickstart-linux>

```
student_01_8579d88be1c2@cloudshell:~ (qwík labs-gcp-00-6bfa9df86695)$ docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
student_01_8579d88be1c2@cloudshell:~ (qwík labs-gcp-00-6bfa9df86695)\$						

```

student_01 8579d88be1c2@cloudshell:~ (qwiklabs-gcp-00-6bfa9df86695)$ docker ps -a
CONTAINER ID   IMAGE      COMMAND   CREATED      STATUS      PORTS      NAMES
bf5elb84bb83   hello-world   "/hello"  5 minutes ago  Exited (0) 5 minutes ago
2d4257853447   hello-world   "/hello"  12 minutes ago  Exited (0) 12 minutes ago
student_01_8579d88be1c2@cloudshell:~ (qwiklabs-gcp-00-6bfa9df86695)$

CLOUD SHELL
Terminal (qwiklabs-gcp-00-6bfa9df86695) x +  Open Editor
student_01 8579d88be1c2@cloudshell:~ (qwiklabs-gcp-00-6bfa9df86695)$ mkdir test && cd test
student_01_8579d88be1c2@cloudshell:~/test (qwiklabs-gcp-00-6bfa9df86695)$ cat > Dockerfile <<EOF
# Use an official Node runtime as the parent image
FROM node:18

# Set the working directory in the container to /app
WORKDIR /app

# Copy the current directory contents into the container at /app
ADD . /app

# Make the container's port 80 available to the outside world
EXPOSE 80

# Run app.js using node when the container launches
CMD ["node", "app.js"]
EOF

CLOUD SHELL
Terminal (qwiklabs-gcp-00-6bfa9df86695) x +  Open Editor
student_01 8579d88be1c2@cloudshell:~ (qwiklabs-gcp-00-6bfa9df86695)$ mkdir test && cd test
student_01_8579d88be1c2@cloudshell:~/test (qwiklabs-gcp-00-6bfa9df86695)$ cat > Dockerfile <<EOF
# Use an official Node runtime as the parent image
FROM node:18

# Set the working directory in the container to /app
WORKDIR /app

# Set the working directory in the container to /app
WORKDIR /app

# Copy the current directory contents into the container at /app
ADD . /app

# Copy the current directory contents into the container at /app
ADD . /app container's port 80 available to the outside world
EXPOSE 80

# Make the container's port 80 available to the outside world
EXPOSE 80

# Run app.js using node when the container launches
CMD ["node", "app.js"]
# Run app.js using node when the container launches
CMD ["node", "app.js"]
EOF

student_01 8579d88be1c2@cloudshell:~/test (qwiklabs-gcp-00-6bfa9df86695)$ cat > app.js << EOF;
const http = require("http");

const hostname = "0.0.0.0";
const port = 80;

const server = http.createServer((req, res) => {
  res.statusCode = 200;
  res.setHeader("Content-Type", "text/plain");
  res.end("Hello World\n");
});

server.listen(port, hostname, () => {
  console.log(`Server running at http://${hostname}:${port}`);
});

process.on("SIGINT", function () {
  console.log("Caught interrupt signal and will exit");
  process.exit();
});
EOF
cat > app.js << EOF;
const http = require("http");

process.exit(); // caught interrupt signal and will exit
EOF
student_01_8579d88be1c2@cloudshell:~/test (qwiklabs-gcp-00-6bfa9df86695)$

```

Task 2: Build Your Own Container

Goal: Create a custom Node.js app container.
Steps:

1. Create files:
 - o Dockerfile (instructions to build the image).
 - o app.js (a simple web server).
2. Build the image:

```
3. docker build -t node-app:0.1 .
```

- o **What Happened?**

- Docker read the `Dockerfile` and built a reusable image named `node-app` (version 0.1).

Key Term:

- **Dockerfile:** A text file with step-by-step commands to assemble the image.

```
EOF      process.exit(); // Exit if interrupt signal and will exit();, port);  
student_01_8579d88be1c2@cloudshell:~/test (qwiklabs-gcp-00-6bfa9df86695)$ docker build -t node-app:0.1 .  
[+] Building 34.2s (8/8) FINISHED  
   docker: default  
=> [internal] load build definition from Dockerfile  
      0.0s  
=> [internal] transfer dockerfile: 39B  
      0.0s  
=> [internal] load metadata for docker.io/library/node:lts  
      1.0s  
=> [internal] load .dockerignore  
      0.0s  
=> [internal] transfer context: 2B  
      0.0s  
=> [1/3] FROM docker.io/library/node:1ts@sha256:079b6a683dc47a87673a615 30.6s  
=> resolve docker.io/library/node:1ts@sha256:079b6a683dc47a87673a6159 0.0s  
=> sha256:37e838b71c6b82c581b7543a13235bb8c99c23cc9 64.40MB 1.8s  
=> sha256:079b6a683dc47a87673a6159c9e9b522b0e87d045330 6.41kB 0.0s  
=> sha256:4:aab86785c23bd679b291b038f18605f46f7f6ab26 2.49kB 0.0s  
=> sha256:bed137c7c18cb1906fb8314abc10611dd4f9a28 48.49MB 1.1s  
=> sha256:c2e75a9483f2d17a3e370639403df2c53a3da148 24.02MB 1.1s  
=> sha256:424bd0c472e5c9cdd0bdfa3c4b66110ed60695883 6.63kB 0.0s  
=> sha256:63a15a4eb345d2dc9cf010dfbd9dc9c5e356ee7c309a66 3.32kB 1.3s  
=> sha256:873aa4c80287477653c01b20948fc34bb1baacf0f 211.36MB 5.1s  
=> sha256:873aa4c80287477653c01b20948fc34bb1baacf0f9a281f8c1b5 4.6s  
=> extracting sha256:bed137c7c18cb1906fb8314abc10611dd4f9a28 56.45MB 2.5s  
=> sha256:7f09a0105c217b304567d5677ef597831ba74318195 1.25MB 2.1s  
=> sha256:352d22a9ba9f0f8127b1c4f5566d095d2b3536614d726c54 446B 2.5s  
=> extracting sha256:c2e76a9483f2d17a3e370639403df2c53a3da1480d53311 1.2s  
=> extracting sha256:873aa4c80287477653c01b20948fc34bb1baacf0f826bcc2d 12.1s  
=> extracting sha256:3a1ba4eb345d2dc9cf010dfbd9dc9c5e356ee7c309a66a1c0f 0.0s  
=> extracting sha256:fd0fe17b5f09bf536bd020a3bb4lae09152ebe3b47a74 4.3s  
=> extracting sha256:7f09a0105c217b304567d5677ef597831ba743181951a223 0.1s  
=> extracting sha256:352d22a9ba9f0f8127b1c4f5566d095d2b3536614d726c54a 0.0s  
=> [internal] load build context:  
=> transferring context: 1.35kB 0.0s  
[2/3] WORKDIR /app 2.3s  
[3/3] ADD . /app 0.0s  
=> exporting to image 0.1s  
=> exporting layers 0.0s  
=> writing image sha256:72aa2528da606c1cf440db937910a3d16efd2c4845e42 0.0s  
=> naming to docker.io/library/node-app:0.1 0.0s  
student_01_8579d88be1c2@cloudshell:~/test (qwiklabs-gcp-00-6bfa9df86695)$
```

```
student_01_8579d88be1c2@cloudshell:~/test (qwiklabs-gcp-00-6bfa9df86695)$ docker images  
REPOSITORY      TAG          IMAGE ID      CREATED        SIZE  
node-app        0.1          72aa2528da60  6 minutes ago  1.12GB  
hello-world     latest       74cc54e27dc4   6 months ago  10.1kB  
student_01_8579d88be1c2@cloudshell:~/test (qwiklabs-gcp-00-6bfa9df86695)$
```

Task 3: Run Your Container

Goal: Start your app and access it.

Steps:

1. Run the container:
2. docker run -p 4000:80 --name my-app node-app:0.1
 - o **-p 4000:80:** Maps your computer's port 4000 to the container's port 80.
 - o **Access the app:** curl `http://localhost:4000` → **Output:** Hello World.

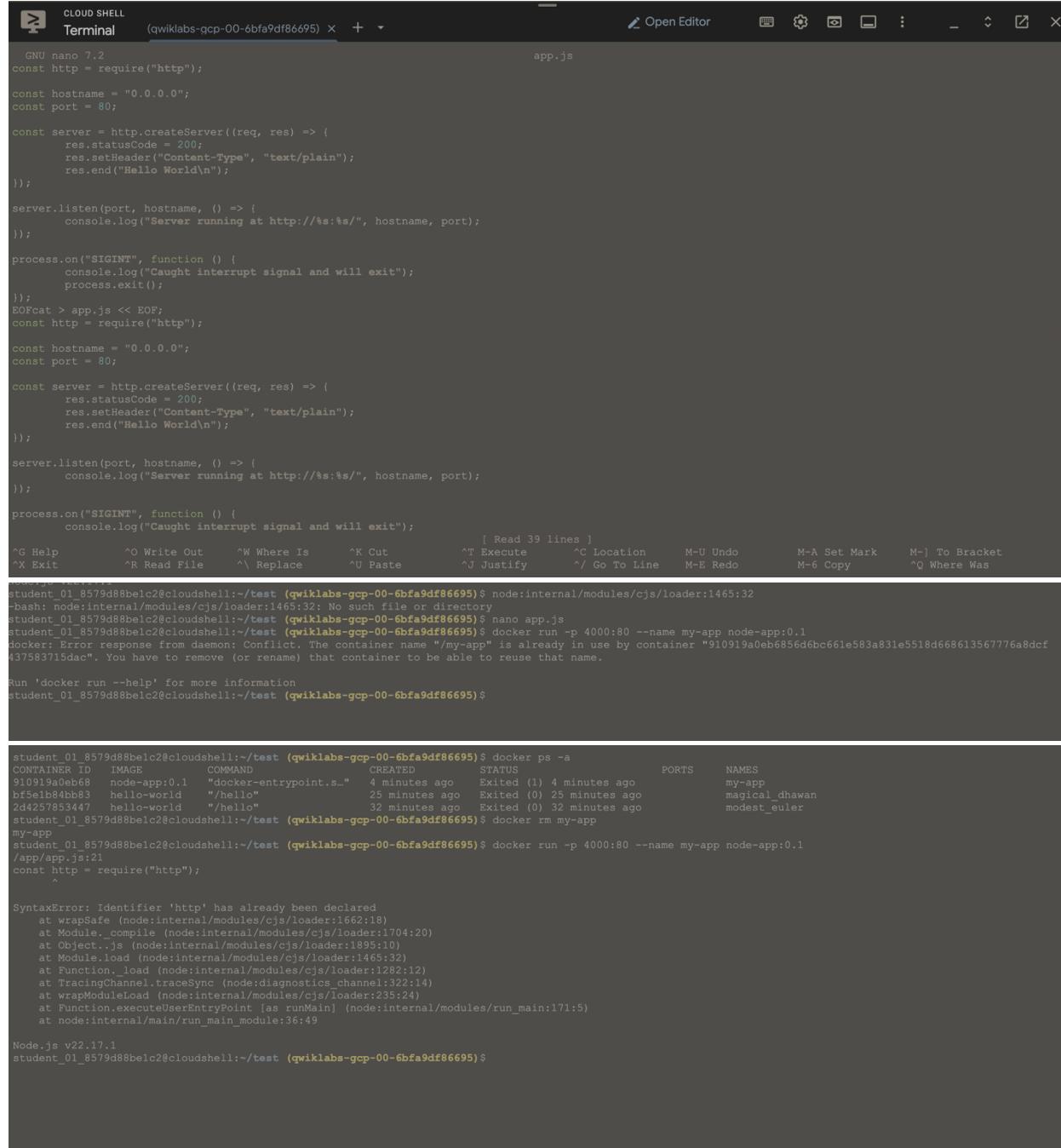
Non-Technical Insight:

Like opening a website by typing `localhost:4000` in your browser.

```
student_01_8579d88belc2@cloudshell:~/test (qwiklabs-gcp-00-6bfa9df86695)$ docker run -p 4000:80 --name my-app node-app:0.1
/app/app.js:21
const http = require("http");
^

SyntaxError: Identifier 'http' has already been declared
  at wrapSafe (node:internal/modules/cjs/loader:1662:18)
  at Module._compile (node:internal/modules/cjs/loader:1704:20)
  at Object..js (node:internal/modules/cjs/loader:1895:10)
  at Module.load (node:internal/modules/cjs/loader:1465:32)
  at Function.load (node:internal/modules/cjs/loader:1282:12)
  at TracingChannel.traceSync (node:diagnostics_channel:322:14)
  at wrapModuleLoad (node:internal/modules/cjs/loader:235:24)
  at Function.executeUserEntryPoint [as runMain] (node:internal/modules/run_main:171:5)
  at node:internal/main/run_main_module:36:49

Node.js v22.17.1
student_01_8579d88belc2@cloudshell:~/test (qwiklabs-gcp-00-6bfa9df86695)$
```



The screenshot shows a terminal window with the following content:

```
student_01_8579d88belc2@cloudshell:~/test (qwiklabs-gcp-00-6bfa9df86695) x + - Open Editor
```

The terminal window displays the following code in `app.js`:

```
GNU nano 7.2
const http = require("http");

const hostname = "0.0.0.0";
const port = 80;

const server = http.createServer((req, res) => {
  res.statusCode = 200;
  res.setHeader("Content-Type", "text/plain");
  res.end("Hello World\n");
});

server.listen(port, hostname, () => {
  console.log("Server running at http://%s:%s/", hostname, port);
});

process.on("SIGINT", function () {
  console.log("Caught interrupt signal and will exit");
  process.exit();
});

EOFcat > app.js << EOF;
const http = require("http");

const hostname = "0.0.0.0";
const port = 80;

const server = http.createServer((req, res) => {
  res.statusCode = 200;
  res.setHeader("Content-Type", "text/plain");
  res.end("Hello World\n");
});

server.listen(port, hostname, () => {
  console.log("Server running at http://%s:%s/", hostname, port);
});

process.on("SIGINT", function () {
  console.log("caught interrupt signal and will exit");
  [ Read 39 lines ]
^G Help      ^O Write Out    ^W Where Is    ^K Cut          ^T Execute      ^C Location    M-U Undo      M-A Set Mark    M-] To Bracket
^X Exit      ^R Read File    ^\ Replace      ^U Paste        ^J Justify      ^/ Go To Line  M-E Redo      M-6 Copy       ^Q Where Was

student_01_8579d88belc2@cloudshell:~/test (qwiklabs-gcp-00-6bfa9df86695)$ node:internal/modules/cjs/loader:1465:32
bash: node:internal/modules/cjs/loader:1465:32: No such file or directory
student_01_8579d88belc2@cloudshell:~/test (qwiklabs-gcp-00-6bfa9df86695)$ nano app.js
student_01_8579d88belc2@cloudshell:~/test (qwiklabs-gcp-00-6bfa9df86695)$ docker run -p 4000:80 --name my-app node-app:0.1
docker: Error response from daemon: Conflict. The container name "/my-app" is already in use by container "910919a0eb6856d6bc661e583a831e5518d668613567776a8dcf437583715dac". You have to remove (or rename) that container to be able to reuse that name.

Run 'docker run --help' for more information
student_01_8579d88belc2@cloudshell:~/test (qwiklabs-gcp-00-6bfa9df86695)$
```

Below the terminal, the command `docker ps -a` is run, showing the following table:

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
910919a0eb68	node-app:0.1	"docker-entrypoint.s..."	4 minutes ago	Exited (1) 4 minutes ago		my-app
bf5e1b84bb83	hello-world	"hello"	25 minutes ago	Exited (0) 25 minutes ago		magical_dhawan
2d4257853447	hello-world	"/hello"	32 minutes ago	Exited (0) 32 minutes ago		modest_euler
student_01_8579d88belc2@cloudshell:~/test (qwiklabs-gcp-00-6bfa9df86695)\$						

The terminal then runs `docker rm my-app`, followed by the Node.js code again, and finally the command `node:internal/modules/cjs/loader:1465:32`.

```

GNU nano 7.2
const http = require("http");

const hostname = "0.0.0.0";
const port = 80;

const server = http.createServer((req, res) => {
  res.statusCode = 200;
  res.setHeader("Content-Type", "text/plain");
  res.end("Hello World\n");
});

server.listen(port, hostname, () => {
  console.log(`Server running at http://${hostname}:${port}`);
});

process.on("SIGINT", function () {
  console.log("Caught interrupt signal and will exit");
  process.exit();
});

EOFcat > app.js << EOF;
const http = require("http");

const hostname = "0.0.0.0";
const port = 80;

const server = http.createServer((req, res) => {
  res.statusCode = 200;
  res.setHeader("Content-Type", "text/plain");
  res.end("Hello World\n");
});

server.listen(port, hostname, () => {
  console.log(`Server running at http://${hostname}:${port}`);
});

process.on("SIGINT", function () {
  console.log("Caught interrupt signal and will exit");
  [ Wrote 39 lines ]
PG Help      ^O Write Out      ^W Where Is      ^K Cut          ^T Execute      ^C Location      M-U Undo      M-A Set Mark      M-] To Bracket
student_01 8579d88be1c2@cloudshell:~/test (qwiklabs-gcp-00-6bfa9df86695)$ nano app.js
student_01 8579d88be1c2@cloudshell:~/test (qwiklabs-gcp-00-6bfa9df86695)$ docker build -t node-app:0.1 .
[+] Building 0.7s (8/8) FINISHED
--> [internal] load build definition from Dockerfile
--> => transferring dockerfile: 397B
--> [internal] load metadata for docker.io/library/node:lts
--> [internal] load .dockerignore
--> => transferring context: 2B
--> [1/3] FROM docker.io/library/node:lts@sha256:079b6a683dc47a87673a6159c9e9b22b0687d04533087cf144c96fac8c26ecd3
--> [internal] load build context
--> => transferring context: 985B
--> CACHED [2/3] WORKDIR /app
--> CACHED [3/3] ADD . /app
--> exporting to image
--> => exporting layers
--> => writing image sha256:72aa2528da606c1cf440db937910a3d16efd2c4845e42bd60f1acd6d4b9ea379
--> => naming to docker.io/library/node-app:0.1
student_01 8579d88be1c2@cloudshell:~/test (qwiklabs-gcp-00-6bfa9df86695)$

student_01 8579d88be1c2@cloudshell:~/test (qwiklabs-gcp-00-6bfa9df86695)$ docker rm my-app      # cleanup if needed
my-app
/app/app.js:21
const http = require("http");
^

SyntaxError: Identifier 'http' has already been declared
  at wrapSafe (node:internal/modules/cjs/loader:1662:18)
  at Module._compile (node:internal/modules/cjs/loader:1704:20)
  at Object..js (node:internal/modules/cjs/loader:1895:10)
  at Module._load (node:internal/modules/cjs/loader:1465:32)
  at Function._load (node:internal/modules/cjs/loader:1282:12)
  at TracingChannel.traceSync (node:diagnostics_channel:322:14)
  at wrapModuleLoad (node:internal/modules/cjs/loader:235:24)
  at Function.executeUserEntryPoint [as runMain] (node:internal/modules/run_main:171:5)
  at node:internal/main/run_main_module:36:49

Node.js v22.17.1
student_01 8579d88be1c2@cloudshell:~/test (qwiklabs-gcp-00-6bfa9df86695)$

```

```
student_01_8579d88belc2@cloudshell:~/test (qwiklabs-gcp-00-6bfa9df86695)$ cat > app.js << 'EOF'
const http = require("http");

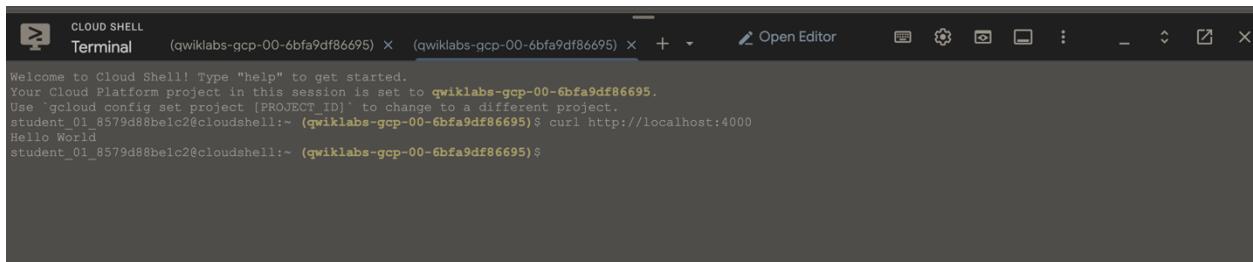
const hostname = "0.0.0.0";
const port = 80;

const server = http.createServer((req, res) => {
  res.statusCode = 200;
  res.setHeader("Content-Type", "text/plain");
  res.end("Hello World\n");
});

server.listen(port, hostname, () => {
  console.log(`Server running at http://$s:$s/, ${hostname}, ${port}`);
});

process.on("SIGINT", function () {
  console.log("Caught interrupt signal and will exit");
  process.exit();
});
EOF
student_01_8579d88belc2@cloudshell:~/test (qwiklabs-gcp-00-6bfa9df86695)$ docker build -t node-app:0.1 .
[+] Building 0.5s (8/8) FINISHED
  => [internal] load build definition from Dockerfile
  => transferring dockerfile: 397B
  => [internal] load metadata for docker.io/library/node:lts
  => [internal] load .dockerignore
  => transferring context: 2B
  => [1/3] FROM docker.io/library/node:lts@sha256:079b6a683dc47a87673a6159c9e9b22b0687d04533087cf144c96fac8c26ecd3
  => [internal] load build context
  => transferring context: 521B
  => CACHED [2/3] WORKDIR /app
  => [3/3] ADD . /app
  => exporting to image
  => => exporting layers
  => => writing image sha256:af52a49ddb06df1211869324706d4d56b02862436c555df66ce06f558af81397
  => => naming to docker.io/library/node-app:0.1
student_01_8579d88belc2@cloudshell:~/test (qwiklabs-gcp-00-6bfa9df86695)$
```

```
);
EOF
student_01_8579d88belc2@cloudshell:~/test (qwiklabs-gcp-00-6bfa9df86695)$ docker build -t node-app:0.1 .
[+] Building 0.5s (8/8) FINISHED
  => [internal] load build definition from Dockerfile
  => transferring dockerfile: 397B
  => [internal] load metadata for docker.io/library/node:lts
  => [internal] load .dockerignore
  => => transferring context: 2B
  => [1/3] FROM docker.io/library/node:lts@sha256:079b6a683dc47a87673a6159c9e9b22b0687d04533087cf144c96fac8c26ecd3
  => [internal] load build context
  => transferring context: 521B
  => CACHED [2/3] WORKDIR /app
  => [3/3] ADD . /app
  => exporting to image
  => => exporting layers
  => => writing image sha256:af52a49ddb06df1211869324706d4d56b02862436c555df66ce06f558af81397
  => => naming to docker.io/library/node-app:0.1
student_01_8579d88belc2@cloudshell:~/test (qwiklabs-gcp-00-6bfa9df86695)$ docker rm my-app
my-app
student_01_8579d88belc2@cloudshell:~/test (qwiklabs-gcp-00-6bfa9df86695)$ docker run -p 4000:80 --name my-app node-app:0.1
Server running at http://0.0.0.0:80/
```



Task 4: Debug Containers

Goal: Inspect running containers.

Tools:

- **Logs:**
- `docker logs <container_id>` # View app logs
- **Shell Access:**
- `docker exec -it <container_id> bash` # Open a terminal inside the container
- **Inspect Details:**

- docker inspect <container_id> # View container configuration (e.g., IP address)

Non-Technical Insight:

Like checking an app's settings or logs on your phone.

```
student_01_8579d88be1c2@cloudshell:~ (qwiklabs-gcp-00-6bfa9df86695)$ docker logs -f [container_id]
Error response from daemon: No such container: [container_id]
student_01_8579d88be1c2@cloudshell:~ (qwiklabs-gcp-00-6bfa9df86695)$

(qwiklabs-gcp-02-9f4bf1607b78) × (qwiklabs-gcp-02-9f4bf1607b78) + Editor ☰ ː _ ⌂ X
Welcome to Cloud Shell! Type "help" to get started.
Your Cloud Platform project in this session is set to qwiklabs-gcp-02-9f4bf1607b78.
Use `gcloud config set project [PROJECT_ID]` to change to a different project.
student_00_fe2cfb9a9c0c@cloudshell:~ (qwiklabs-gcp-02-9f4bf1607b78)$ curl http://localhost:4000
Hello World
student_00_fe2cfb9a9c0c@cloudshell:~ (qwiklabs-gcp-02-9f4bf1607b78)$

Hello World
student_00_fe2cfb9a9c0c@cloudshell:~ (qwiklabs-gcp-02-9f4bf1607b78)$ docker stop my-app && docker rm my-app
my-app
my-app
student_00_fe2cfb9a9c0c@cloudshell:~ (qwiklabs-gcp-02-9f4bf1607b78)$
student_00_fe2cfb9a9c0c@cloudshell:~ (qwiklabs-gcp-02-9f4bf1607b78)$ docker run -p 4000:80 --name my-app -d node:app:0.1
docker ps
05e9a8642bb2f4039836a80e9e9f0e9e358ebcb6c0cc646b89b1614925bfb57f
CONTAINER ID   IMAGE          COMMAND           CREATED          STATUS
  PORTS          NAMES
05e9a8642bb2   node:app:0.1   "docker-entrypoint.s..."   Less than a second ago   Up Less than a second
  0.0.0.0:4000->80/tcp   my-app
student_00_fe2cfb9a9c0c@cloudshell:~ (qwiklabs-gcp-02-9f4bf1607b78)$

  0.0.0.0:4000->80/tcp   my-app
student_00_fe2cfb9a9c0c@cloudshell:~ (qwiklabs-gcp-02-9f4bf1607b78)$ docker logs 05e9a8642bb2
Server running at http://0.0.0.0:80/
student_00_fe2cfb9a9c0c@cloudshell:~ (qwiklabs-gcp-02-9f4bf1607b78)$
```

```
GNU nano 7.2                                     app.js
const http = require("http");

const hostname = "0.0.0.0";
const port = 80;

const server = http.createServer((req, res) => {
    res.statusCode = 200;
    res.setHeader("Content-Type", "text/plain");
    res.end("Hello World\n");
});

server.listen(port, hostname, () => {
    console.log("Server running at http://%s:%s/", hostname, port);
});

process.on("SIGINT", function () {
    console.log("Caught interrupt signal and will exit");
    process.exit();
});
```

```
(qwiklabs-gcp-02-9f4bf1607b78) x + ▾ Editor ⚙️ 🌐 🌐 🌐 : _ ▾ ▾
```

```
GNU nano 7.2                                app.js *
```

```
const http = require("http");
```

```
const hostname = "0.0.0.0";
const port = 80;
```

```
const server = http.createServer((req, res) => {
    res.statusCode = 200;
    res.setHeader("Content-Type", "text/plain");
    res.end("Welcome to the Cloud\n");
});
```

```
server.listen(port, hostname, () => {
    console.log("Server running at http://%s:%s/", hostname, port);
});
```

```
process.on("SIGINT", function () {
    console.log("Caught interrupt signal and will exit");
    process.exit();
});
```

```
student_00_fe2cfb9a9c0c@cloudshell:~/test (qwiklabs-gcp-02-9f4bf1607b78)$ docker build -t node-app:0.2
[+] Building 0.5s (8/8) FINISHED
  => [internal] load build definition from Dockerfile
  => => transferring dockerfile: 397B
  => [internal] load metadata for docker.io/library/node:lts
  => [internal] load .dockerignore
  => => transferring context: 2B
  => [1/3] FROM docker.io/library/node@sha256:ec09419096c9cb8ff4c6fcf9c7b63332bbecab56d4cb6d
  => [internal] load build context
  => => transferring context: 524B
  => CACHED [2/3] WORKDIR /app
  => [3/3] ADD . /app
  => exporting to image
  => => exporting layers
  => => writing image sha256:c71a7522997ab92709820bcb7f5035d64ce62ebe87cf9572b40509ca64bbb4b
  => => naming to docker.io/library/node-app:0.2
student_00_fe2cfb9a9c0c@cloudshell:~/test (qwiklabs-gcp-02-9f4bf1607b78)$
```

```

  Starting to dockerize my app...
student_00_fe2cfb9a9c0c@cloudshell:~/test (qwiklabs-gcp-02-9f4bf1607b78)$ docker run -p 8080:80 --name my-app-2 -d node-app:0.2
docker ps
cd6830d49b78484d64e9da8a9da06d81100a16a4b2329ac06316f182f2675fc5
CONTAINER ID        IMAGE               COMMAND             CREATED            STATUS              PORTS
cd6830d49b78        node-app:0.2      "docker-entrypoint.s..."   Less than a second ago   Up Less than a second
0.0.0.0:8080->80/tcp      my-app-2
05e9a8642bb2        node-app:0.1      "docker-entrypoint.s..."   6 minutes ago       Up 6 minutes
0.0.0.0:4000->80/tcp      my-app
student_00_fe2cfb9a9c0c@cloudshell:~/test (qwiklabs-gcp-02-9f4bf1607b78)$

  0.0.0.0:8080->80/tcp      my-app
student_00_fe2cfb9a9c0c@cloudshell:~/test (qwiklabs-gcp-02-9f4bf1607b78)$ curl http://localhost:8080
Welcome to the Cloud
student_00_fe2cfb9a9c0c@cloudshell:~/test (qwiklabs-gcp-02-9f4bf1607b78)$

student_00_fe2cfb9a9c0c@cloudshell:~/test (qwiklabs-gcp-02-9f4bf1607b78)$ curl http://localhost:4000
Hello World
student_00_fe2cfb9a9c0c@cloudshell:~/test (qwiklabs-gcp-02-9f4bf1607b78)$

student_00_fe2cfb9a9c0c@cloudshell:~/test (qwiklabs-gcp-02-9f4bf1607b78)$ docker logs -f cd6830d49b78
[...]
Server running at http://0.0.0.0:80/
[...]
student_00_fe2cfb9a9c0c@cloudshell:~/test (qwiklabs-gcp-02-9f4bf1607b78)$ docker exec -it cd6830d49b78 bash
root@cd6830d49b78:/app# ls
Dockerfile  app.js
root@cd6830d49b78:/app#


student_00_fe2cfb9a9c0c@cloudshell:~/test (qwiklabs-gcp-02-9f4bf1607b78)$ docker inspect cd6830d49b78
[...]
{
  "Id": "cd6830d49b78484d64e9da8a9da06d81100a16a4b2329ac06316f182f2675fc5",
  "Created": "2025-07-22T22:15:16.597760652Z",
  "Path": "docker-entrypoint.sh",
  "Args": [
    "node",
    "app.js"
  ],
  "State": {
    "Status": "running",
    "Running": true,
    "Paused": false,
    "Restarting": false,
    "OOMKilled": false,
    "Dead": false,
    "Pid": 1929,
    "ExitCode": 0,
    "Error": "",
    "StartedAt": "2025-07-22T22:15:16.644307677Z",
    "FinishedAt": "0001-01-01T00:00:00Z"
  },
  "Image": "sha256:c71a7522997ab92709820bcb7f5035d64ce62ebe87cfcc9572b40509ca64bbb4b",
  "ResolvConfPath": "/var/lib/docker/containers/cd6830d49b78484d64e9da8a9da06d81100a16a4b2329ac06316f182f2675fc5/resolv.conf",
  "HostnamePath": "/var/lib/docker/containers/cd6830d49b78484d64e9da8a9da06d81100a16a4b2329ac06316f182f2675fc5/hostname",
  "HostsPath": "/var/lib/docker/containers/cd6830d49b78484d64e9da8a9da06d81100a16a4b2329ac06316f182f2675fc5/hosts",
  "LogPath": "/var/lib/docker/containers/cd6830d49b78484d64e9da8a9da06d81100a16a4b2329ac06316f182f2675fc5/cd6830d49b78484d64e9da8a9da06d81100a16a4b2329ac06316f182f2675fc5-journal.log",
  "Name": "/my-app-2",
  "RestartCount": 0,
  "Driver": "overlay2",
  "Platform": "linux",
  "MountLabel": "",
  "ProcessLabel": "",
  "AppArmorProfile": "docker-default",
  "ExecIDs": null,
  [...]
}
student_00_fe2cfb9a9c0c@cloudshell:~/test (qwiklabs-gcp-02-9f4bf1607b78)$ docker inspect --format='{{range .NetworkSettings.Addresses}}{{end}}' cd6830d49b78
172.17.0.3
student_00_fe2cfb9a9c0c@cloudshell:~/test (qwiklabs-gcp-02-9f4bf1607b78)$

[...]
student_00_fe2cfb9a9c0c@cloudshell:~/test (qwiklabs-gcp-02-9f4bf1607b78)$ docker inspect --format='{{range .NetworkSettings.Networks}}{{.IP Address}}{{end}}' cd6830d49b78
172.17.0.3
student_00_fe2cfb9a9c0c@cloudshell:~/test (qwiklabs-gcp-02-9f4bf1607b78)$

```

Task 5: Publish to Google Artifact Registry

Goal: Save your image in Google Cloud and reuse it.

Steps:

1. Create a private repository (storage space) in Artifact Registry.
2. Tag your image:
3. `docker build -t us-east1-docker.pkg.dev/PROJECT_ID/my-repo/node-app:0.2 .`
4. Push to Artifact Registry:
5. `docker push us-east1-docker.pkg.dev/PROJECT_ID/my-repo/node-app:0.2`
6. Pull and run it anywhere:
7. `docker run -p 4000:80 us-east1-docker.pkg.dev/PROJECT_ID/my-repo/node-app:0.2`

Non-Technical Insight:

Like saving a document to Google Drive and downloading it on another device.

ories

js

← Create repository

Learn

Cleanup policies

Define policies to automatically clean up artifacts.

Delete artifacts
Artifacts will be deleted according to cleanup policy criteria.

Dry run
Artifacts will not be deleted. Cleanup policies will be evaluated, and test delete events sent to Cloud Audit Logging. [Learn more](#)

[Add a cleanup policy](#)

Artifact Analysis

Vulnerability scanning

Automatically scan images that are pushed to this repository for vulnerabilities.

Note: Scanning is currently only supported for Docker standard and remote repositories. [Learn more](#)

Enabled

Disabled

Failed to create repository: Location "us-central1" violates organization policy on the project qwiklabs-gcp-02-9f4bf1607b78

student_00_fe2cfb9a9c0c@cloudshell:~/test (qwiklabs-gcp-02-9f4bf1607b78)\$ gcloud beta resource-manager org-policies describe constraints/gcp.resourceLocations --effective --project=qwiklabs-gcp-02-9f4bf1607b78

```
constraint: constraints/gcp.resourceLocations
listPolicy:
  allowedValues:
  - asia
  - europe-west
  - eu
  - europe-west4
  - aws-ap-southeast-2
  - europe-west4-locations
  - aws-eu-central-1
  - aws-locations
  - azure-eastus2
  - global
  - US
  - azure-locations
  - europe
  - us
  - europe-west4-a
  - nam5
  - aws-us-east-1
  - EU
  - aws-ap-northeast-2
  - us-central
  - europe-west4-b
  - europe-west4-c
  - eur3
```

--location=us-central1 \
--description="My Docker repo for Node app"

ERROR: (gcloud.artifacts.repositories.create) FAILED_PRECONDITION: Location "us-central1" violates organization policy on the project qwiklabs-gcp-02-9f4bf1607b78

```
student_00_fe2cfb9a9c0c@cloudshell:~/test (qwiklabs-gcp-02-9f4bf1607b78)$ gcloud artifacts repositories create my-repository \
--repository-format=docker \
--location=europe-west4 \
--description="Node app repo"
Create request issued for: [my-repository]
Waiting for operation [projects/qwiklabs-gcp-02-9f4bf1607b78/locations/europe-west4/operations/d6596faa-2d70-4473-871c-8632a8fd1804] to complete...done.
Created repository [my-repository].
student_00_fe2cfb9a9c0c@cloudshell:~/test (qwiklabs-gcp-02-9f4bf1607b78)$
```

```
student_00_fe2cfb9a9c0c@cloudshell:~/test (qwiklabs-gcp-02-9f4bf1607b78)$ gcloud auth configure-docker europe-west4-docker.pkg.dev
WARNING: Your config file at [/home/student_00_fe2cfb9a9c0c/.docker/config.json] contains these credential helper entries:
```

```
{
  "credHelpers": {
    "gcr.io": "gcloud",
    "us.gcr.io": "gcloud",
    "eu.gcr.io": "gcloud",
    "asia.gcr.io": "gcloud",
    "staging-k8s.gcr.io": "gcloud",
    "marketplace.gcr.io": "gcloud",
    "africa-south1-docker.pkg.dev": "gcloud",
    "docker.africa-south1.rep.pkg.dev": "gcloud",
    "asia-docker.pkg.dev": "gcloud",
    "asia-east1-docker.pkg.dev": "gcloud",
    "asia-east1-docker.rep.pkg.dev": "gcloud",
    "asia-east2-docker.pkg.dev": "gcloud",
    "docker.asia-east2.rep.pkg.dev": "gcloud",
    "asia-northeast1-docker.pkg.dev": "gcloud",
    "docker.asia-northeast1.rep.pkg.dev": "gcloud",
    "asia-northeast2-docker.pkg.dev": "gcloud",
    "docker.asia-northeast2.rep.pkg.dev": "gcloud",
    "asia-northeast3-docker.pkg.dev": "gcloud",
    "docker.asia-northeast3.rep.pkg.dev": "gcloud",
    "asia-south1-docker.pkg.dev": "gcloud",
    "docker.asia-south1.rep.pkg.dev": "gcloud",
    "asia-south2-docker.pkg.dev": "gcloud",
    "docker.asia-south2.rep.pkg.dev": "gcloud",
    "asia-southeast1-docker.pkg.dev": "gcloud",
    "docker.asia-southeast1.rep.pkg.dev": "gcloud",
    "asia-southeast2-docker.pkg.dev": "gcloud",
    "docker.asia-southeast2.rep.pkg.dev": "gcloud",
    "us-central1-docker.pkg.dev": "gcloud",
    "docker.us-central1.rep.pkg.dev": "gcloud",
    "us-east1-docker.pkg.dev": "gcloud",
    "docker.us-east1.rep.pkg.dev": "gcloud",
    "us-east4-docker.pkg.dev": "gcloud",
    "docker.us-east4.rep.pkg.dev": "gcloud",
    "us-east5-docker.pkg.dev": "gcloud",
    "docker.us-east5.rep.pkg.dev": "gcloud",
    "us-east7-docker.pkg.dev": "gcloud",
    "docker.us-east7.rep.pkg.dev": "gcloud",
    "us-south1-docker.pkg.dev": "gcloud",
    "docker.us-south1.rep.pkg.dev": "gcloud",
    "us-west1-docker.pkg.dev": "gcloud",
    "docker.us-west1.rep.pkg.dev": "gcloud",
    "us-west2-docker.pkg.dev": "gcloud",
    "docker.us-west2.rep.pkg.dev": "gcloud",
    "us-west3-docker.pkg.dev": "gcloud",
    "docker.us-west3.rep.pkg.dev": "gcloud",
    "us-west4-docker.pkg.dev": "gcloud",
    "docker.us-west4.rep.pkg.dev": "gcloud",
    "us-west8-docker.pkg.dev": "gcloud"
  }
}
}
Adding credentials for: europe-west4-docker.pkg.dev
gcloud credential helpers already registered correctly.
student_00_fe2cfb9a9c0c@cloudshell:~/test (qwiklabs-gcp-02-9f4bf1607b78)$
```

```
"docker.southamerica-west1.rep.pkg.dev": "gcloud",
student_00_fe2cfb9a9c0c@cloudshell:~/test (qwiklabs-gcp-02-9f4bf1607b78)$ gcloud artifacts repositories create my-repository --repository-format=docker --location=europe-west4 --description="Docker repository"
ERROR: (gcloud.artifacts.repositories.create) ALREADY EXISTS: the repository already exists
student_00_fe2cfb9a9c0c@cloudshell:~/test (qwiklabs-gcp-02-9f4bf1607b78)$
```

```
student_00_fe2cfb9a9c0c@cloudshell:~/test (qwiklabs-gcp-02-9f4bf1607b78)$ docker build -t europe-west4-docker.pkg.dev/qwiklabs-gcp-02-9f4bf1607b78/my-repository/node-app:0.2 .
[+] Building 0.4s (8/8) FINISHED
--> [internal] load build definition from Dockerfile
--> => transferring dockerfile: 397B
--> [internal] load metadata for docker.io/library/node:lts
--> [internal] load .dockerrigore
--> => transferring context: 2B
--> [1/3] FROM docker.io/library/node@sha256:ec09419096c9cb8ff4c6fcf9c7b63332bbecab56d4cb6dcc83d98c180a7cddd
--> [internal] load build context
--> => transferring context: 58B
--> CACHED [2/3] WORKDIR /app
--> CACHED [3/3] ADD . /app
--> exporting to image
--> => exporting layers
--> => writing image sha256:c71a7522997ab92709820bcb7f5035d64ce62ebe87cfc9572b40509ca64bbbb4b
--> => naming to europe-west4-docker.pkg.dev/qwiklabs-gcp-02-9f4bf1607b78/my-repository/node-app:0.2
student_00_fe2cfb9a9c0c@cloudshell:~/test (qwiklabs-gcp-02-9f4bf1607b78)$
```

```
student_00_fe2cfb9a9c0c@cloudshell:~/test (qwiklabs-gcp-02-9f4bf1607b78)$ docker images
REPOSITORY          TAG      IMAGE ID      CREATED     SIZE
node-app            0.2      c71a7522997a   30 minutes ago  1.12GB
europe-west4-docker.pkg.dev/qwiklabs-gcp-02-9f4bf1607b78/my-repository/node-app  0.2      c71a7522997a   30 minutes ago  1.12GB
node-app            0.1      7ad163c14cd5   39 minutes ago  1.12GB
hello-world         latest   74cc54e27dc4   6 months ago   10.1kB
student_00_fe2cfb9a9c0c@cloudshell:~/test (qwiklabs-gcp-02-9f4bf1607b78)$
```

```
student_00_fe2cfb9a9c0c@cloudshell:~/test (qwiklabs-gcp-02-9f4bf1607b78)$ docker push europe-west4-docker.pkg.dev/qwiklabs-gcp-02-9f4bf1607b78/my-repository/node-app:0.2
The push refers to repository [europe-west4-docker.pkg.dev/qwiklabs-gcp-02-9f4bf1607b78/my-repository/node-app]
97a7cae6a970: Pushed
fbf3ca3c2545: Pushed
e5b77f05d0e5: Pushed
bd8201ae8f5d: Pushed
16a5fc601df: Pushed
2f10455f8757: Pushed
1b90a4eeb596: Pushed
a9b908787288: Pushed
20c8aca21338: Pushed
175a19836175: Pushed
0.2: digest: sha256:718f2ab8857c0df42c33a02120394ccd215928c87c2862a956575f4223025elb size: 2417
student_00_fe2cfb9a9c0c@cloudshell:~/test (qwiklabs-gcp-02-9f4bf1607b78)$
```

The screenshot shows a cloud-based repository interface. At the top, there are buttons for 'Edit repository', 'Delete', 'Refresh', and 'Show info panel'. Below this is a search bar with the placeholder 'Enter property name or value'. A table lists a single repository entry:

Name	Format	Type	Location	Scanning	Description	Labels
<u>my-repository</u>	Docker	Standard	europe-west4 (Netherlands)	Active	Node	▼

```
student_00_fe2cfb9a9c0c@cloudshell:~/test (qwiklabs-gcp-02-9f4bf1607b78)$ docker stop $(docker ps -q)
docker rm $(docker ps -q)
```

```
cd6830d49b78
05e9a8642bb2
cd6830d49b78
05e9a8642bb2
9d391adc3c36
8da32c0ab69b
student_00_fe2cfb9a9c0c@cloudshell:~/test (qwiklabs-gcp-02-9f4bf1607b78)$
student_00_fe2cfb9a9c0c@cloudshell:~/test (qwiklabs-gcp-02-9f4bf1607b78)$
```

```
student_00_fe2cfb9a9c0c@cloudshell:~/test (qwiklabs-gcp-02-9f4bf1607b78)$ docker rmi europe-west4-docker.pkg.dev/qwiklabs-gcp-02-9f4bf1607b78/my-repository/node-app:0.2
docker rmi node:its
docker rmi -f $(docker images -aq) # remove remaining images
docker images
Untagged: europe-west4-docker.pkg.dev/qwiklabs-gcp-02-9f4bf1607b78/my-repository/node-app:0.2
Untagged: europe-west4-docker.pkg.dev/qwiklabs-gcp-02-9f4bf1607b78/my-repository/node-app@sha256:718f2ab8857c0df42c33a02120394ccd215928c87c2862a956575f4223025elb
Error response from daemon: No such image: node:its
Untagged: node-app:0.2
Deleted: sha256:c71a7522997ab92709820bcb7f5035d64ce62ebe87cf9572b40509ca64bbb4b
Untagged: node-app:0.1
Deleted: sha256:7ad163c14cd5fc80397812849d473de88a902de6ff81ee04d7e53ea2f9c9a6c2
Untagged: hello-world:latest
Untagged: hello-world@sha256:ec153840d1e635ac434fab5e377081f17e0e15afab27beb3f726c3265039cff
Deleted: sha256:74cc54e27dc41bb10dc4b2226072d469509f2f22fla3ce74fa59661a1d44602
Deleted: sha256:63a41026379f4391a306242eb0b9f26dc3550d863b7fdb97d899f6eb89efe72
REPOSITORY TAG IMAGE ID CREATED SIZE
student_00_fe2cfb9a9c0c@cloudshell:~/test (qwiklabs-gcp-02-9f4bf1607b78)$
```

```
student_00_fe2cfb9a9c0c@cloudshell:~/test (qwiklabs-gcp-02-9f4bf1607b78)$ docker run -p 4000:80 -d europe-west4-docker.pkg.dev/qwiklabs-gcp-02-9f4bf1607b78/my-repository/node-app:0.2
Unable to find image 'europe-west4-docker.pkg.dev/qwiklabs-gcp-02-9f4bf1607b78/my-repository/node-app:0.2' locally
0.2: Pulling from qwiklabs-gcp-02-9f4bf1607b78/my-repository/node-app
0873b8cf9e3: Already exists
c2e16af9483f: Already exists
37f838b71c6b: Already exists
873a4c802874: Already exists
63a15a4eb345: Already exists
fd0fe17b5f09: Already exists
7f09a0105c21: Already exists
352d22a9ba9f: Already exists
67177375ccf1: Already exists
5e6d392bee41: Already exists
Digest: sha256:718f2ab8857c0df42c33a02120394ccd215928c87c2862a956575f4223025e1b
Status: Downloaded newer image for europe-west4-docker.pkg.dev/qwiklabs-gcp-02-9f4bf1607b78/my-repository/node-app:0.2
f3d28290a9be8254cb21d0bf4f9eb1c3888475876dc55ae68830a9cafef233be
student_00_fe2cfb9a9c0c@cloudshell:~/test (qwiklabs-gcp-02-9f4bf1607b78)$
```

```
student_00_fe2cfb9a9c0c@cloudshell:~/test (qwiklabs-gcp-02-9f4bf1607b78)$ curl http://localhost:4000
Welcome to the Cloud
student_00_fe2cfb9a9c0c@cloudshell:~/test (qwiklabs-gcp-02-9f4bf1607b78)$
```

Why This Matters

1. **Consistency:** Your app runs identically on any machine.
2. **Speed:** Build once, run anywhere (no "it works on my machine" issues).
3. **Scalability:** Easily deploy containers to cloud services like Google Kubernetes Engine.

For Non-Technical Readers:

Docker simplifies software deployment – like shipping a self-contained "app box" that works everywhere without setup hassles.

Conclusion

This hands-on Docker lab gave me a deeper understanding of how containerization works in real-world DevOps and cloud environments. I went beyond just running public containers — I built my own Node.js application image, pushed it to Google Artifact Registry, and verified its portability by running it in a fresh environment.

This project sharpened my skills in:

- Container lifecycle management (build → run → debug → publish)
- Working with Docker CLI in Google Cloud Shell
- Private container image hosting using Artifact Registry
- Port mapping and image versioning for custom apps

Whether deploying microservices in production or testing isolated environments for security, Docker is now a practical tool in my toolkit — not just a buzzword.