

<WA/>

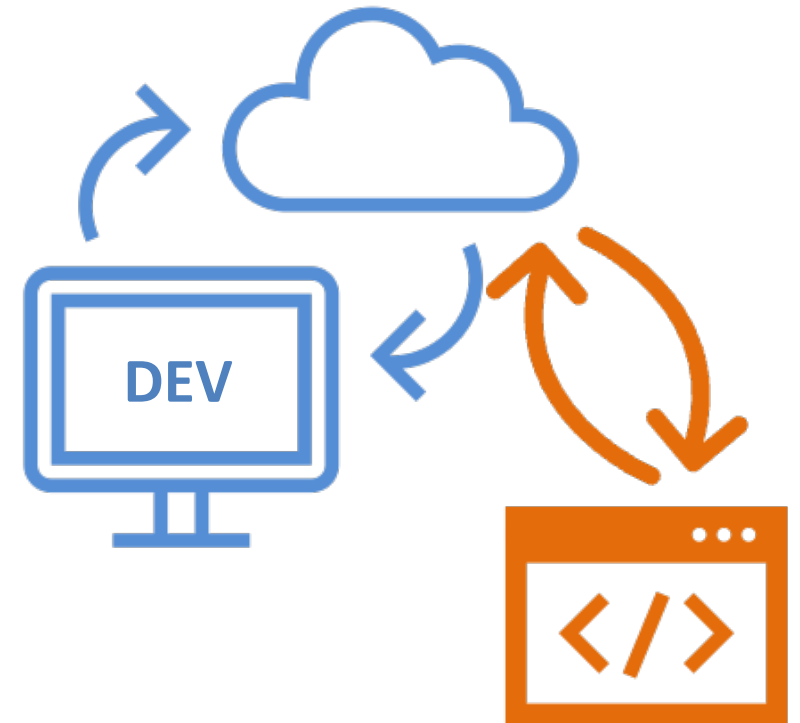
2024

Deployment

How/where to put the app in the “cloud”

Enrico Masala

Antonio Servetti



Deployment to Cloud Servers

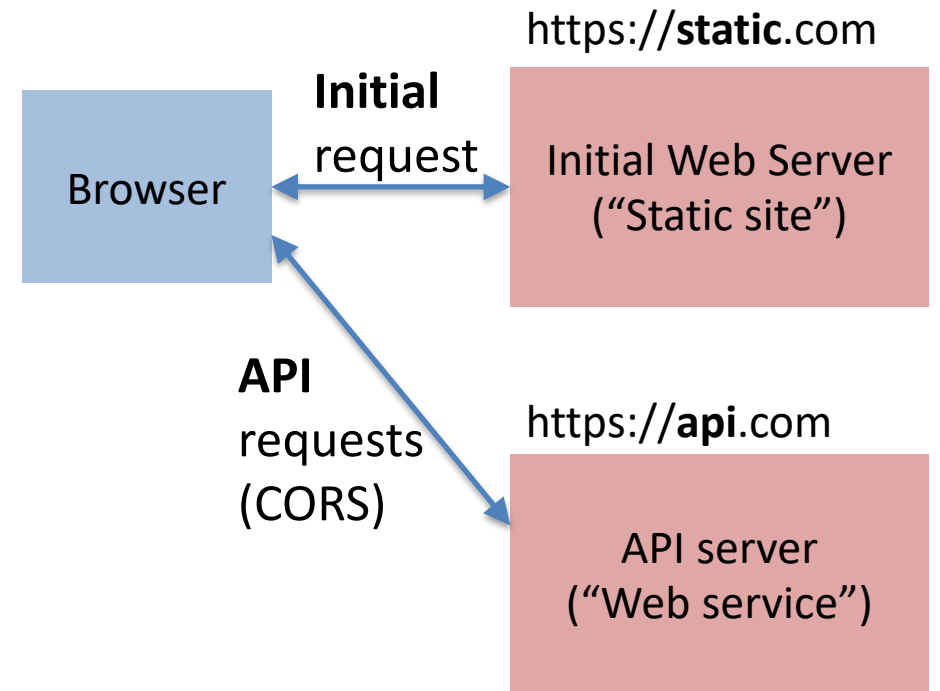
- Real applications are accessible by anybody connected to the Internet
- Need to set up a web server to
 - Serve the static files of the React client application (“landing page”)
 - Provide the API services used by the client
- Configuring servers from scratch, starting from bare virtual machines, is difficult, error prone, and requires expertise in system administration
 - Selecting the best web server software (Apache, nginx, ...)
 - Configuring it
 - Configure the security part (HTTPS, certificates, private keys, algorithms, etc.)

Deployment to Cloud Servers

- Most of the activity is mechanical and the same for many cases
- Some online services offer the possibility to simplify deployment
 - **Pre-configured static or API servers**
 - Possibility to run node.js as backend server
 - Possibility to use a database instance
- Platform-as-a-Service approach (PaaS)
 - Many (paid) services exists
 - Very few free ones, especially for the API server (require computational resources)
 - Mostly oriented to become familiar with the system by deploying a toy project, then upgrade to a paid profile

Deployment Architecture

- Use two servers, configured with CORS
- The server runs as-is
- Client: create a “build” with vite
 - `npm run build`
- The build will be put in the `/dist` folder
 - These are all the files for the static web server
 - Note: keep the original files!
From the build you cannot go back to the original sources



Vite Build

- `npm run build` produces all the files needed for a static web server
 - A static `index.html` file that loads:
 - JS file (including all libraries, Bootstrap, ...), CSS file,
 - other resources (icons, fonts, etc.)

```
> react-qa@0.0.0 build
> vite build

vite v5.2.9 building for production...
✓ 350 modules transformed.
dist/index.html                0.46 kB | gzip: 0.30 kB
dist/assets/bootstrap-icons-BtvjY1KL.woff2 130.40 kB
dist/assets/bootstrap-icons-B0rJxbIo.woff  176.03 kB
dist/assets/index-CFLmWFf5.css             307.59 kB | gzip: 44.37 kB
dist/assets/index-BHo4gWFZ.js              220.24 kB | gzip: 72.33 kB
✓ built in 692ms
```

Static Build

- Any needed resource is put into static files, served by the static server
 - Libraries (e.g., Bootstrap etc.) are minified to save bandwidth
- No leak of information towards third-party servers (CDNs) due to css, fonts, etc.)
 - Might be important for legal reasons (privacy law disclosures, etc.)

```
<!doctype html>
<html lang="en">
  <head> <meta charset="UTF-8" />
    <link rel="icon" type="image/svg+xml" href="/vite.svg" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Vite + React</title>
    <script type="module" crossorigin src="/assets/index-BHo4gWFZ.js"></script>
    <link rel="stylesheet" crossorigin href="/assets/index-CFLmWFf5.css">
  </head>
  <body><div id="root"></div></body>
</html>
```

index.html

index-BHo4gWFZ.js

```
function gp(e,t){for(var
n=0;n<t.length;n++){const
r=t[n];if(typeof
r!="string"&&!Array.isArr
ay(r)){for(const l in
r)if(l!="default"&&!(l
in e)){...
```

Suggested Deployment Approach

- Identify a free service that can support an API server running node.js
 - Example: <https://render.com>
- Create an account (free for basic, simple services)
- Make your projects (client & server) accessible to the service
 - Typically, via a git repository from which the service load the files
 - Git approach is easy to maintain and update: automatic deploy after commit/push
 - With render.com: create 1 GitHub repository (either private or public) with your code, in two separate folders (client and server)
 - If private, later you will have to grant permission to render.com to access it

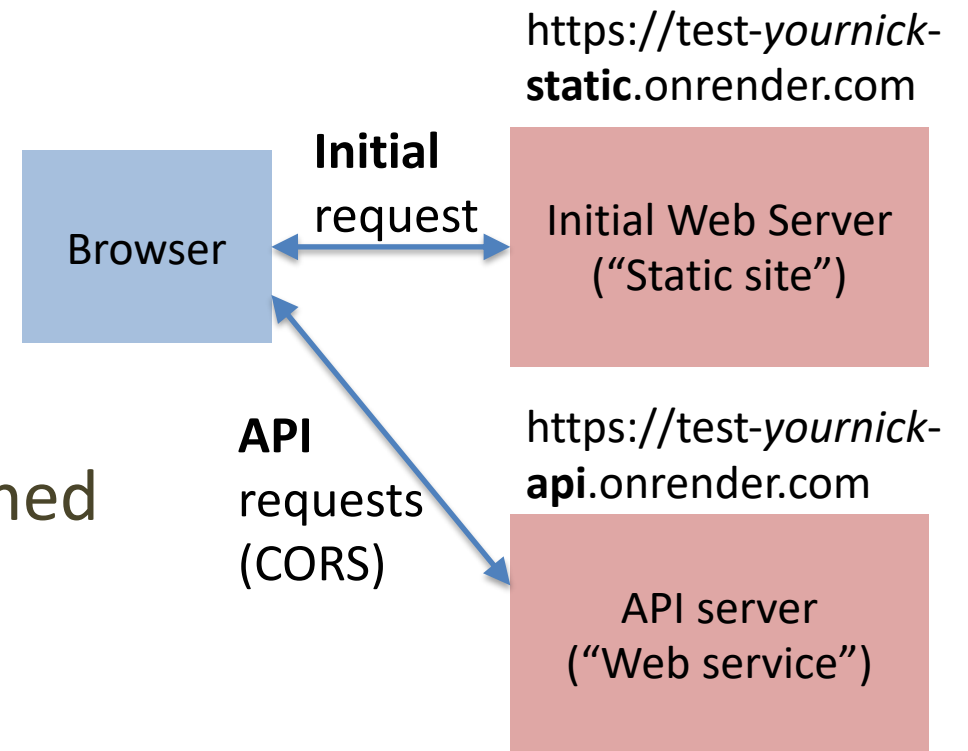
Implement the Architecture

- Go to the <https://render.com> dashboard, create the following two services with names of your choice

- **Static site**, e.g. *test-yournick-static*
- **Web service**, e.g., *test-yournick-api*

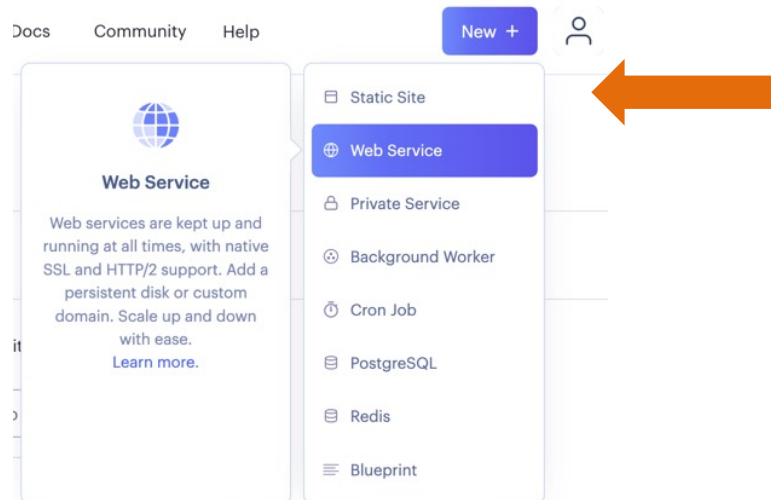
- Note that the names will determine part of the URL at which the server can be reached

- <https://test-yournick-static.onrender.com>
- <https://test-yournick-api.onrender.com>



Static Site

- Root Directory: `client`
- Build command:
`npm ci; npm run build`
- Publish directory: `dist`



You are deploying a static site for `polito-WA-2024/deployment`.

Name

A unique name for your static site.

wa2024-em-static

Branch

The repository branch used for your static site.

main

Root Directory Optional

Defaults to repository root. When you specify a [root directory](#) that is different from your repository root, Render runs all your commands in the [specified directory](#) and ignores changes outside the directory.

client

Build Command

This command runs in the root directory of your repository when a new version of your code is pushed, or when you deploy manually. It is typically a script that installs libraries, runs migrations, or compiles resources needed by your app.

client/ \$ npm ci; npm run build

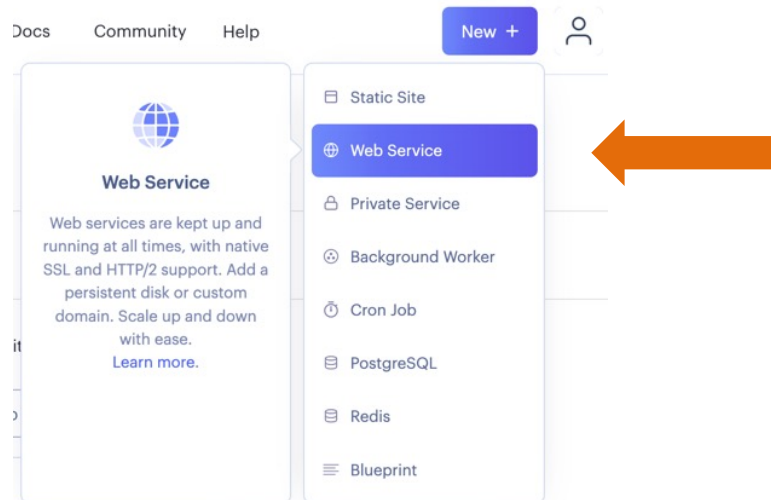
Publish directory

The [relative](#) path of the directory containing built assets to publish. Examples: `./`, `./build`, `dist` and `frontend/build`.

client/ dist

Web Service

- Root Directory: server
- Build command: `npm ci`
- Start command: `node index.js`



You are deploying a web service for **polito-WA-2024/deployment**.

Name
A unique name for your web service.
wa2024-em-api

Region
The **region** where your web service runs.
Frankfurt (EU Central)

Branch
The repository branch used for your web service.
main

Root Directory Optional
Defaults to repository root. When you specify a **root directory** that is different from your repository root, Render runs all your commands in the **specified directory** and ignores changes outside the directory.
server

Runtime
The runtime for your web service.
Node

Build Command
This command runs in the root directory of your repository when a new version of your code is pushed, or when you deploy manually. It is typically a script that installs libraries, runs migrations, or compiles resources needed by your app.
server/ \$ npm ci

Start Command
This command runs in the root directory of your app and is responsible for starting its processes. It is typically used to start a webserver for your app. It can access environment variables defined by you in Render.
server/ \$ node index.js

Instance Type

For hobby projects	Free \$0 / month 512 MB (RAM) 0.1 CPU	Upgrade to enable more features Free instances spin down after periods of inactivity. They do not support SSH access, scaling, one-off jobs, or persistent disks. Select any paid instance type to enable these features.
For professional use For more power and to get the most out of Render, we recommend using one of our paid instance types. All paid	Starter \$7 / month 512 MB (RAM) 0.5 CPU	Standard \$25 / month 2 GB (RAM) 1 CPU

Preparing the Project for Deployment

- Configure CORS and HTTP requests correctly
 - The server must allow requests from `https://test-yournick-static.onrender.com`
 - The client must send API requests to `https://test-yournick-api.onrender.com`
- Make sure that the serve runs on the correct port
 - The render.com system requires the server to listen on localhost, port 10000
 - The port will be exposed to the public internet via a reverse-proxy server mechanism, that will also provide the TLS certificate for HTTPS
- Configure the use of the secure option in cookies
 - Modify the express-session configuration

Environment Variables

- In production environments, many configuration parameters are made available as environment variables
 - Accessible by the “`process.env`” Javascript special variable
 - **Very convenient to change configuration without changing files!**
 - Also, without having to commit/push such files to the git repository!
 - In render.com such values can be set/changed externally, using the dashboard
- Note: a manual re-deployment may be needed for changes to environment variables to take effect

Client Config

- For the client build with Vite, use
 - “import.meta.env” JS variable
 - Every variable name **must start** with **VITE_...**
 - This is to avoid accidentally exposing other environment variables

```
// API.js file
//const URL = 'http://localhost:3001/api';
const URL = import.meta.env.VITE_API_SERVER_URL;
```

Value to be set
after starting the
“Web Service”
from the
Dashboard

Environment Variables Optional

Set environment-specific config and secrets
(such as API keys), then read those values from
your code. [Learn more.](#)

VITE_API_SERVER_URL

https://wa2024-em-api.onrender.com/api



+ Add Environment Variable

Add from .env

Server Config

- Use two environment runtime variables, PORT and ORIGIN

```
const app = express();  
//const port = 3001;  
const port = process.env.PORT || 3001;  
  
const corsOptions = {  
  //origin: 'http://localhost:5173',  
  origin: process.env.ORIGIN || 'http://localhost:5173',  
  credentials: true,  
};
```

Server Config

- Set the environment variables depending on the public URL of “Static Site”

WEB SERVICE

wa2024-em-api Node Free Upgrade your instance →

polito-WA-2024 / deployment ↗ main

<https://wa2024-em-api.onrender.com>

Connect Manual Deploy

Events

Logs

Disks

Environment

Shell

Previews

Jobs

Metrics

Scaling

Settings

Environment Variables

Set environment-specific config and secrets (such as API keys), then read those values from your code. [Learn more.](#)

Key	Value
ORIGIN	https://wa2024-em-static.onrender.com
PORT	10000

Create Environment Group

+ Add Environment Variable Add from .env Save Changes

Secret Files

Store plaintext files containing secret data (such as a .env file or a private key).

STATIC SITE

wa2024-em-static

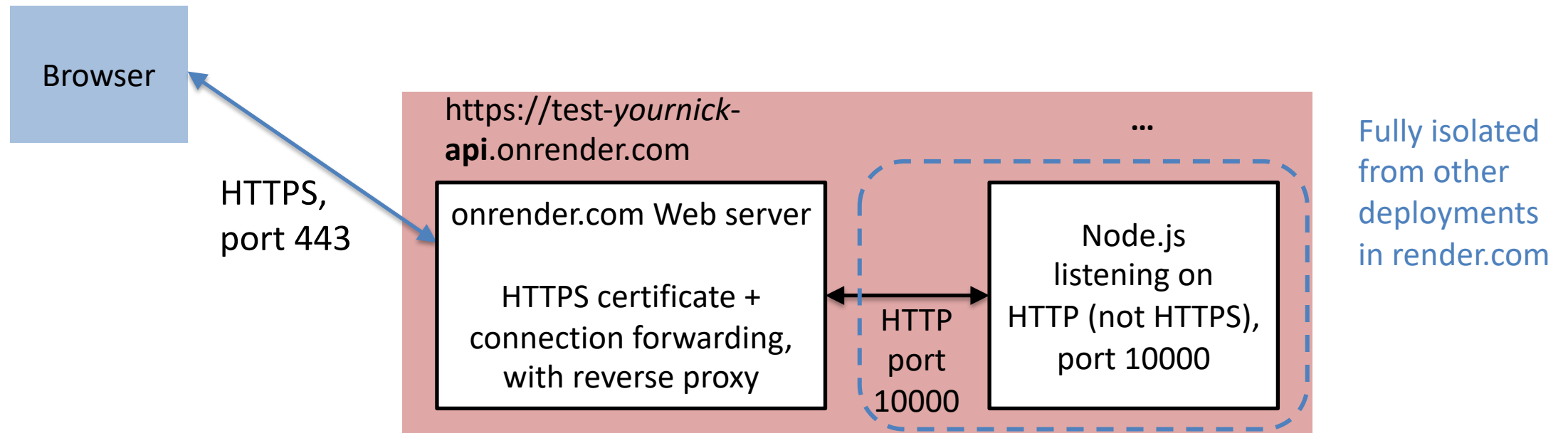
polito-WA-2024 / deployment ↗ main

<https://wa2024-em-static.onrender.com>

Value to be set
by looking at the
“Static Site” URL
from the
Dashboard

Server Config: HTTPS management

- Node.js must be informed that it is sending a secure cookie on an insecure connection because it is behind a reverse proxy that will handle the HTTPS
 - Otherwise, it will silently refuse to do it



Server Config: HTTPS & Proxy Configuration

- Check if node is running in production mode
- Use the built-in `NODE_ENV` variable (already set by render.com)

```
if (process.env.NODE_ENV === 'production')  
  app.set('trust proxy', 1); // To work behind reverse-proxy  
  // NB: Without this setting, “secure” cookies will not work  
...  
app.use(session({  
  secret: '.....', // this could also be an env variable if desired  
  resave: false, saveUninitialized: false,  
  cookie: { httpOnly: true,  
    secure: (process.env.NODE_ENV === 'production'? true : false) },  
}));
```

Actual Deployment

- From the dashboard, **deploy** both the “Static Site” and the “Web Service”
 - The first time you configure the service, deployment is automatically started
- You can check the result and errors by looking at the **console** which is integrated into the page

The screenshot displays the Render dashboard for a web service named 'wa2024-em-api'. The top navigation bar includes a 'Manual Deploy' button, which is highlighted by a blue arrow from the text 'deploy' in the slide. Below the service name, there are tabs for 'Node', 'Free', and 'Upgrade your instance'. The 'Events' section shows a deployment event from May 30, 2024, at 11:51 AM, with a 'Live' status. The 'Logs' section is expanded, showing a live tail of the deployment process. The logs indicate that the service is using Node version 20.12.2 and Bun version 1.1.0. It shows the execution of 'node index.js' and a warning about the MemoryStore not being designed for a production environment. The logs also show the service listening on http://localhost:10000 and receiving a HEAD request with a 404 status and a GET request with a 404 status.

WEB SERVICE
wa2024-em-api Node Free Upgrade your instance → Connect Manual Deploy

polito-WA-2024 / deployment main
<https://wa2024-em-api.onrender.com>

Events
Logs
Disks
Environment
Shell
Previews
Jobs
Metrics
Scaling
Settings

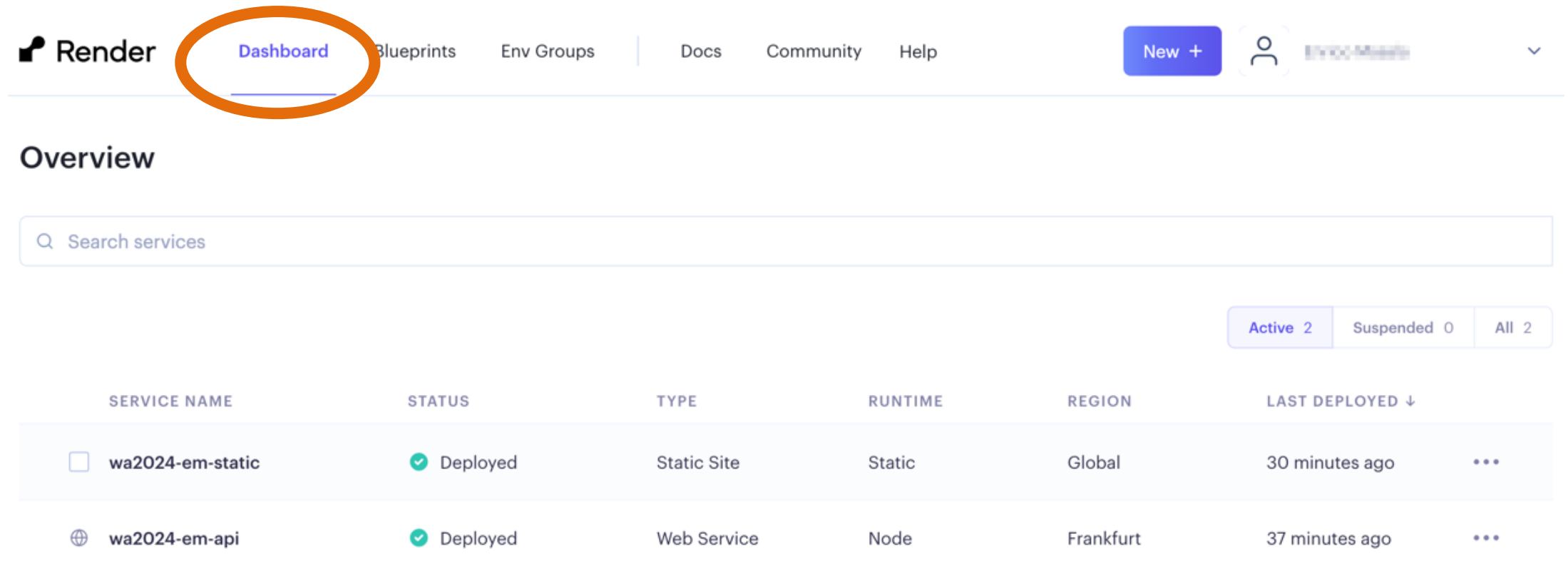
May 30, 2024 at 11:51 AM Live
f0b5f84 fix bu

All logs Search Live tail GMT+2

May 30 11:51:51 AM ==> Using Node version 20.12.2 (default)
May 30 11:51:51 AM ==> Docs on specifying a Node version: <https://render.com/docs/node-version>
May 30 11:51:51 AM ==> Using Bun version 1.1.0 (default)
May 30 11:51:51 AM ==> Docs on specifying a bun version: <https://render.com/docs/bun-version>
May 30 11:51:54 AM ==> Running 'node index.js'
May 30 11:51:55 AM Warning: connect.session() MemoryStore is not
May 30 11:51:55 AM designed for a production environment, as it will leak
May 30 11:51:55 AM memory, and will not scale past a single process.
May 30 11:51:55 AM qa-server listening at http://localhost:10000
May 30 11:51:56 AM HEAD / 404 3.705 ms - 140
May 30 11:52:02 AM ==> Your service is live 🎉
May 30 11:52:03 AM GET / 404 1.284 ms - 139

Active Services

- Also look at the dashboard to check the status of your services

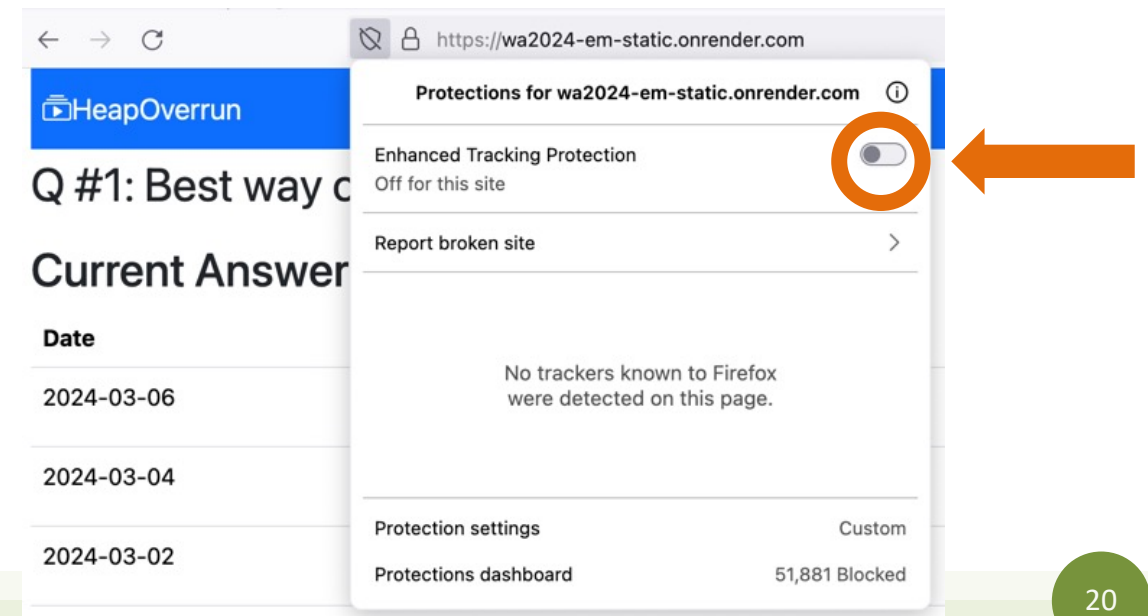
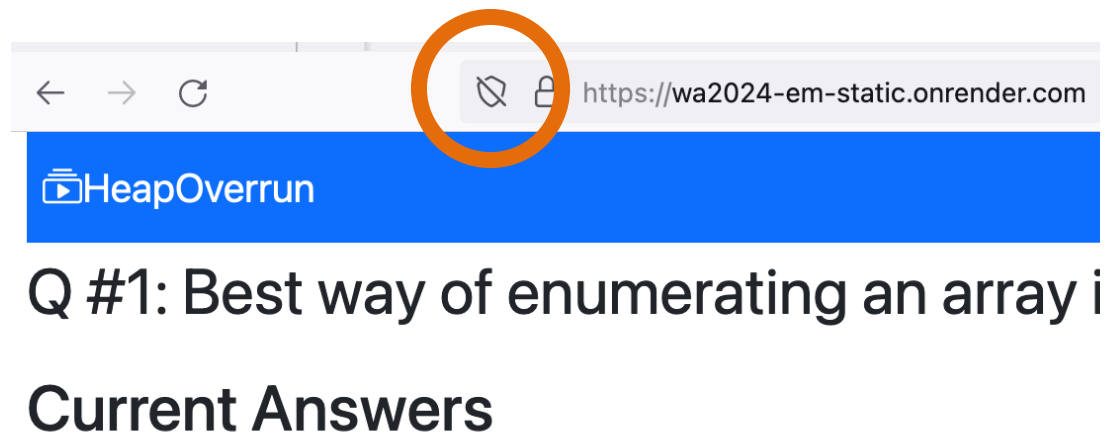


The screenshot shows the Render Dashboard interface. The top navigation bar includes the Render logo, a 'Dashboard' link (highlighted with an orange circle), and other links like Blueprints, Env Groups, Docs, Community, and Help. On the right, there is a 'New +' button, a user profile icon, and a dropdown arrow. Below the navigation bar, the 'Overview' section features a search bar labeled 'Search services'. To the right of the search bar, there are filters for 'Active 2', 'Suspended 0', and 'All 2'. The main content area displays a table of services with columns for Service Name, Status, Type, Runtime, Region, and Last Deployed. Two services are listed: 'wa2024-em-static' (Static Site, Static runtime, Global region, deployed 30 minutes ago) and 'wa2024-em-api' (Web Service, Node runtime, Frankfurt region, deployed 37 minutes ago). Each service row has a checkbox and a three-dot menu icon.

SERVICE NAME	STATUS	TYPE	RUNTIME	REGION	LAST DEPLOYED ↓
<input type="checkbox"/> wa2024-em-static	✓ Deployed	Static Site	Static	Global	30 minutes ago
<input checked="" type="checkbox"/> wa2024-em-api	✓ Deployed	Web Service	Node	Frankfurt	37 minutes ago

Test the App!

- Open a browser, go to <https://test-yournick-static.onrender.com> and test!
- **IMPORTANT: DISABLE browser cross-site “Tracking Protection”,** e.g. in Firefox, click on the Shield Icon on the left side of the URL box
 - This is needed because you are using a public, shared domain name (onrender.com)



Beware: Notes

- In the free “Web Service”, files will be often deleted, after few minutes of inactivity (in particular, the sqlite3 file)
 - This limitation can only be removed by subscribing to non-free tiers
 - In any case, useful to test “toy” examples and become familiar with the system before paying for one
- Browsers are extremely careful when sharing information on domains where anybody can register subdomains for free (e.g., onrender.com)
 - Registering your own domain name is possible but not free
 - Remember: in this case you need to handle/provide your own HTTPS certificate

License

- These slides are distributed under a Creative Commons license “**Attribution-NonCommercial-ShareAlike 4.0 International (CC BY-NC-SA 4.0)**”
- **You are free to:**
 - **Share** — copy and redistribute the material in any medium or format
 - **Adapt** — remix, transform, and build upon the material
 - The licensor cannot revoke these freedoms as long as you follow the license terms.
- **Under the following terms:**
 - **Attribution** — You must give [appropriate credit](#), provide a link to the license, and [indicate if changes were made](#). You may do so in any reasonable manner, but not in any way that suggests the licensor endorses you or your use.
 - **NonCommercial** — You may not use the material for [commercial purposes](#).
 - **ShareAlike** — If you remix, transform, or build upon the material, you must distribute your contributions under the [same license](#) as the original.
 - **No additional restrictions** — You may not apply legal terms or [technological measures](#) that legally restrict others from doing anything the license permits.
- <https://creativecommons.org/licenses/by-nc-sa/4.0/>

