Deploying Express with Lifeboat

What is Lifeboat?

Quoted from their Website:

Lifeboat makes it simple to deploy and manage your Xojo Web applications on a Linux server. Connect to a Linux server and Lifeboat will install the components necessary to run and manage Xojo Web apps.

• Designed for Xojo Web 1.0 and 2.0

• Upload and configure web apps without additional tools

• Manage multiple web apps running on one server

• Easily configure load balancing

• Serve regular websites with nginx

• Install free SSL certificates from Let's Encrypt

• Try Lifeboat for free and deploy a web app today!

If you haven't heard of Lifeboat, go ahead and read their excellent Documentation: Lifeboat - Help.

While it is optimized for "Xojo Web Apps", it works perfectly fine with Express.

How to deploy an Express Console App using Lifeboat?

For this example we're going to use a macOS machine to build the Express Console App. Then we'll deploy it using Lifeboat to a Ubuntu 22 Server.

Requirements

- Xojo with a licence allowing you to build a Console App for Target Linux (Intel, 64Bit).
- A Server running Linux (in this example: Ubuntu 22; choose any Linux distribution which is supported by Lifeboat).
- Lifeboat installed on the macOS developer machine.
- Lifeboat connected to your Linux Server.

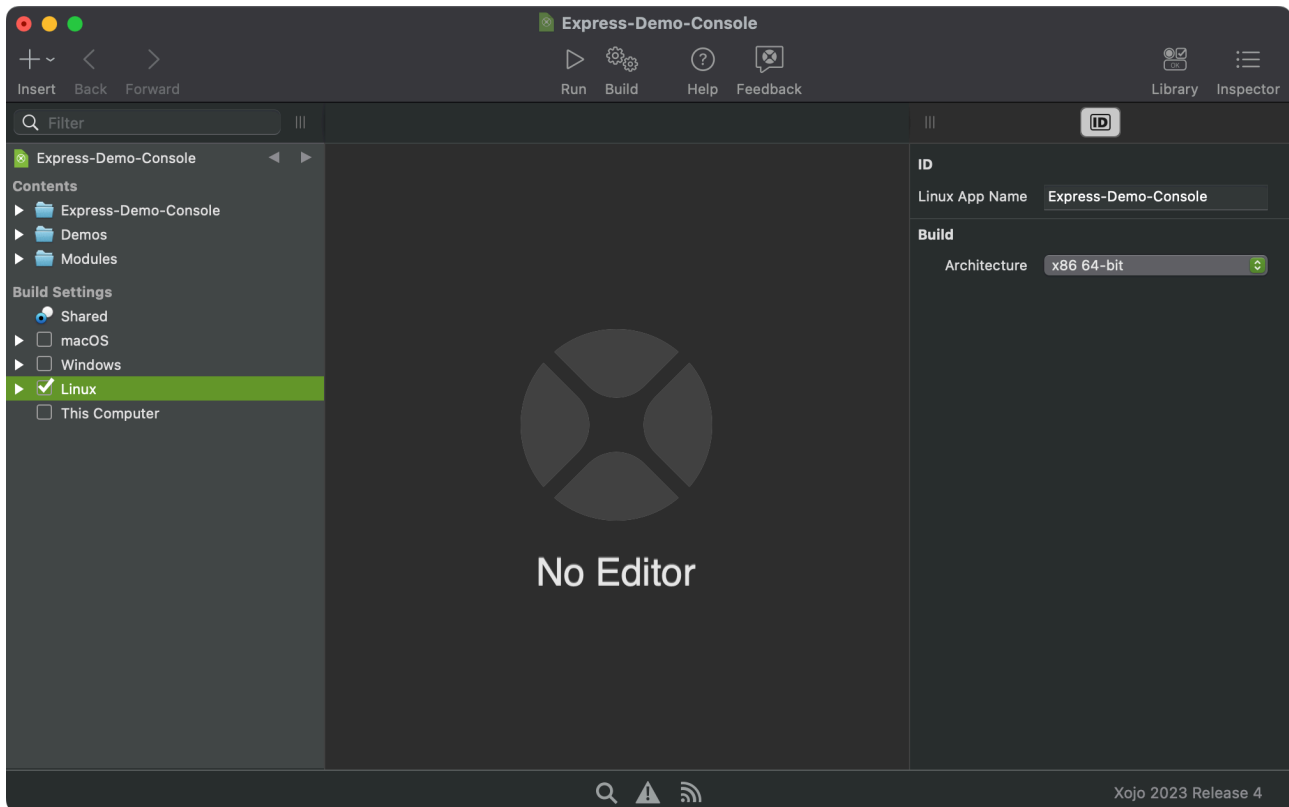Are you ready and set up? So let's go and try this.

Build an Express Console App

I'm going to use the provided example: `Express-Demo-Console.xojo_project`.

First Debug-Run it locally to make sure the Console App works as expected.

Once we're ready to build: Check the following properties in your Xojo Project:

- The most obvious one: Tick the "**Build Target: Linux, x86-64Bit".** Since our Linux Server will run Ubuntu / Linux-64Bit, we need the Express Console App to run in such an environment.
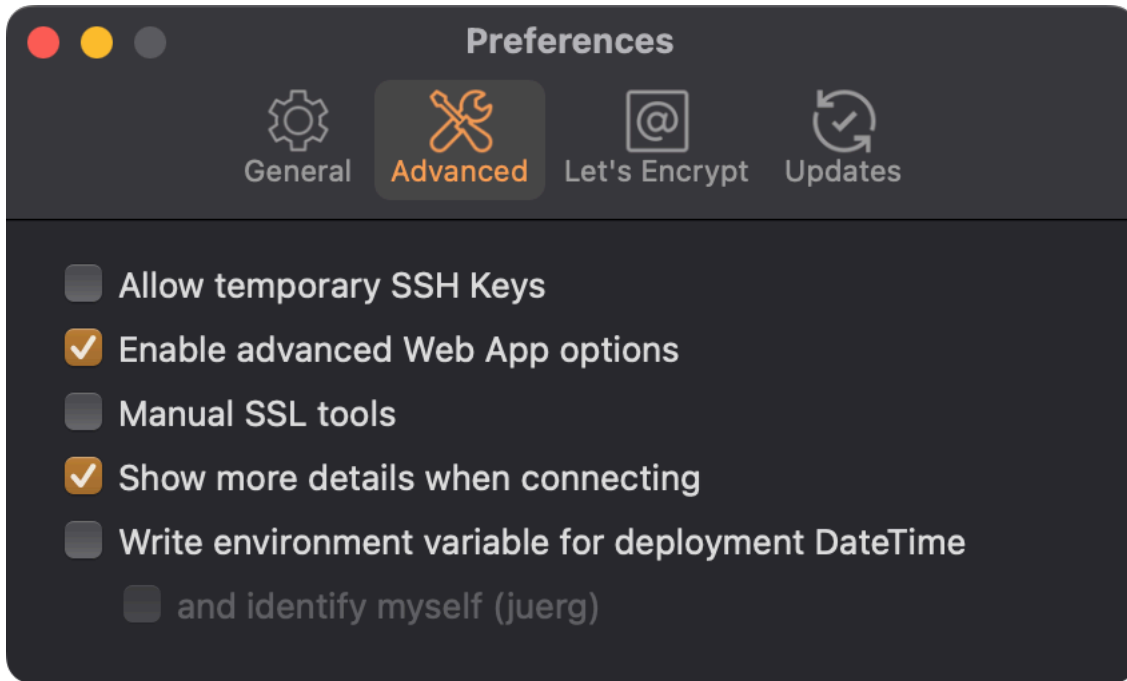

- That's about all we need to do within the Xojo IDE - so let's hit **"Build"**.

We now have the built Express Console App ready. Next step is to deploy it using Lifeboat.
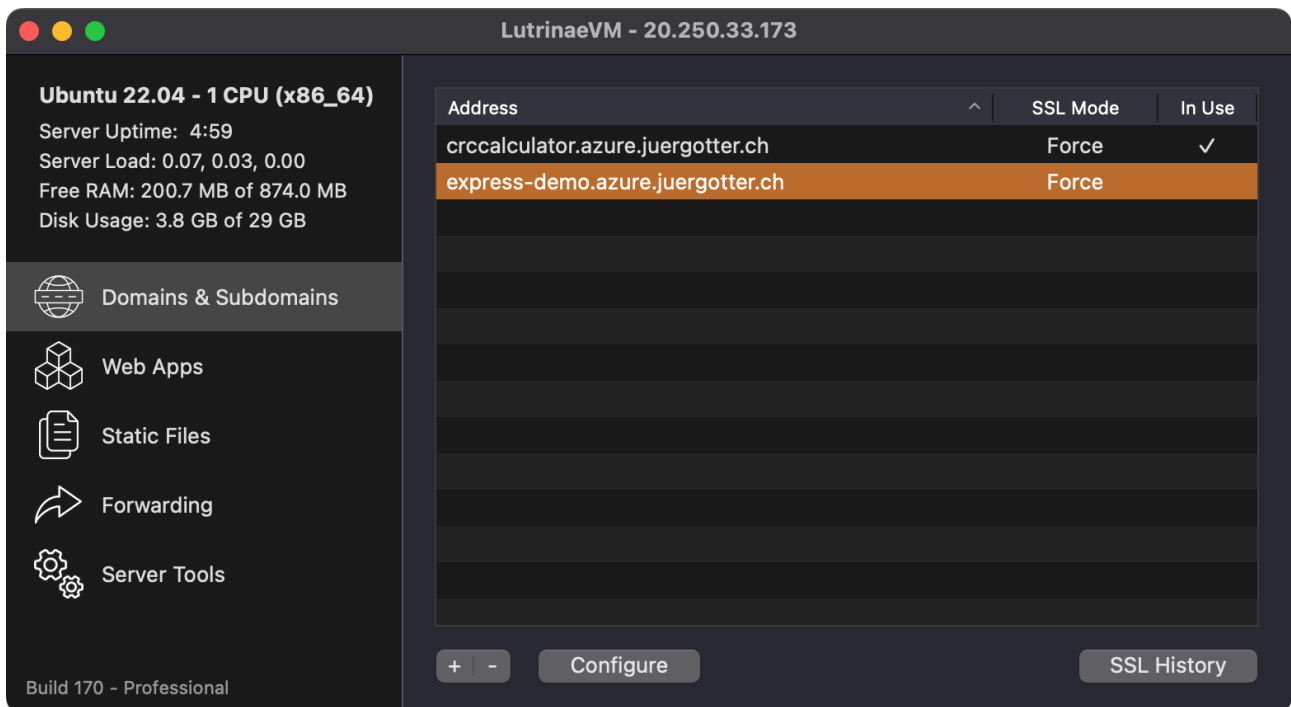
Lifeboat

Open Lifeboat and connect to your server.

First enable the Advanced Options in Lifeboat in "Settings -> Advanced: [x] Enable advanced WebApp options". This will be required to later configure Environment Variables and/or change the Load Balancing method.
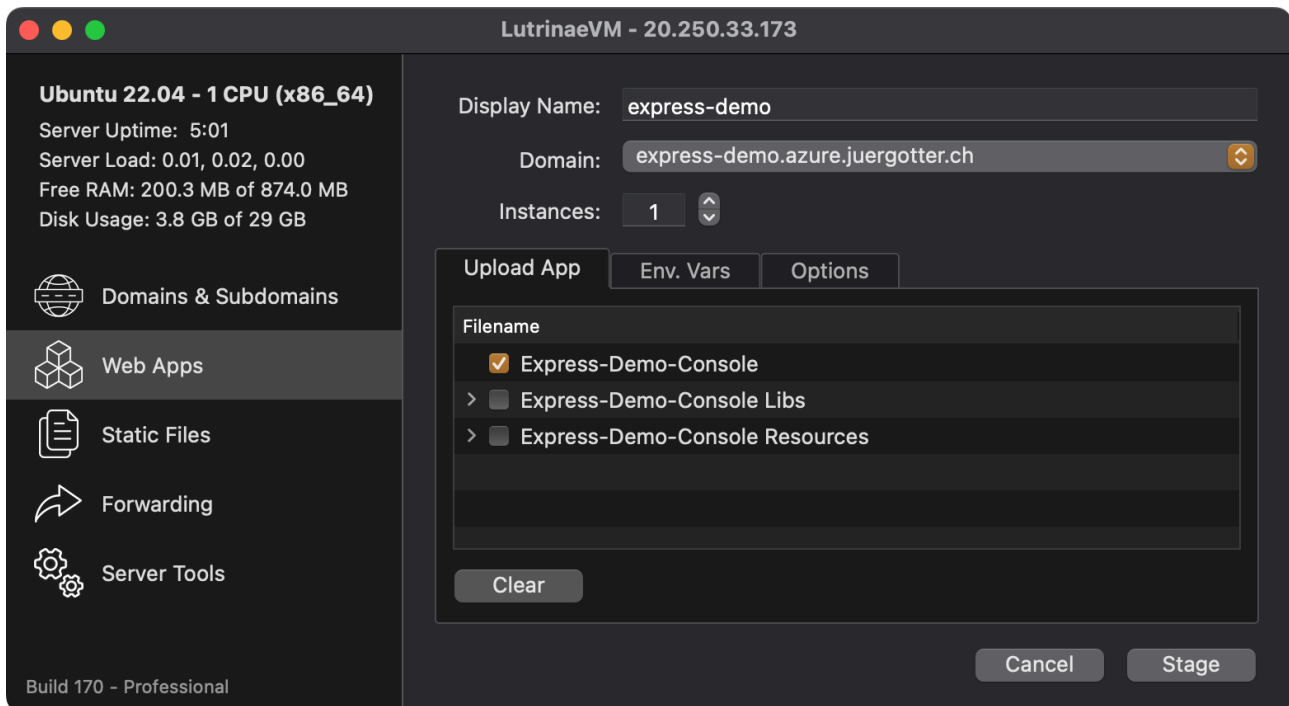
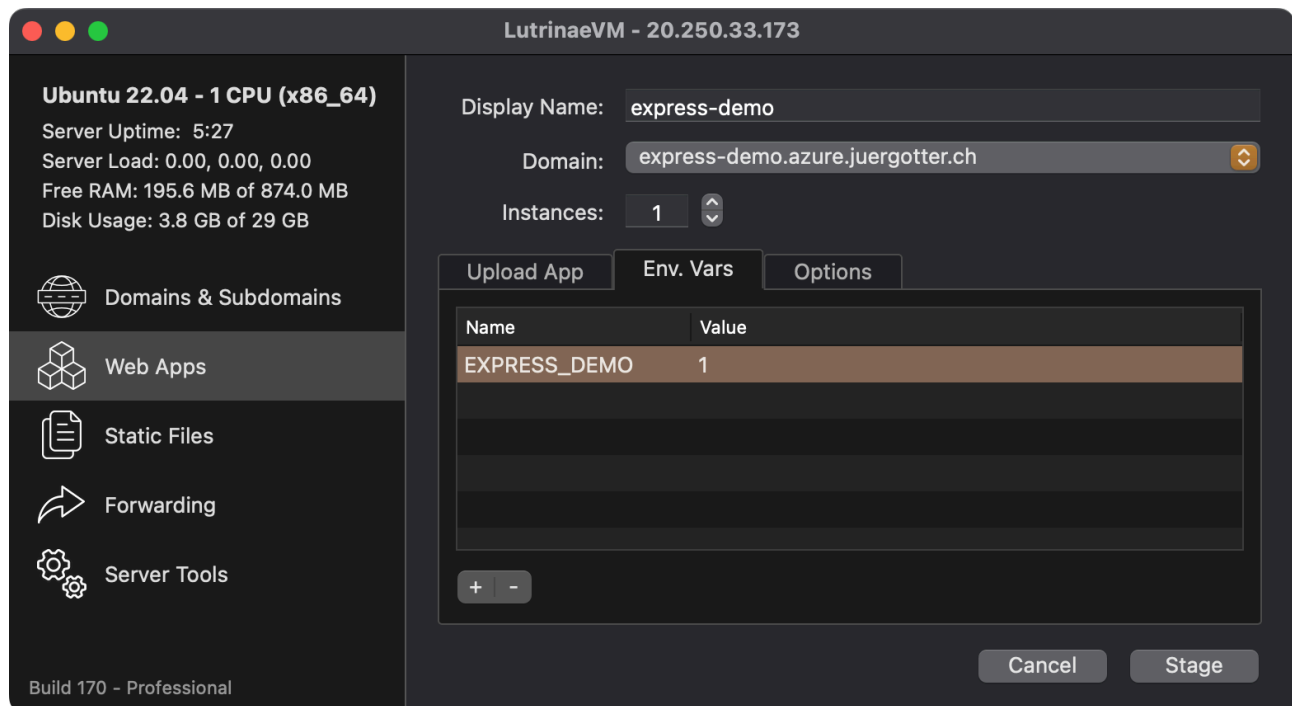

Add the Domain you're going to use:

Then add a new "Web App".

Give it a name, choose the Domain and drop the Build Folder `/path/to/your/xojo-express-project/Builds\ -\ Express-Demo-Console/Linux\ 64\ bit/Express-Demo-Console` into "Upload App":

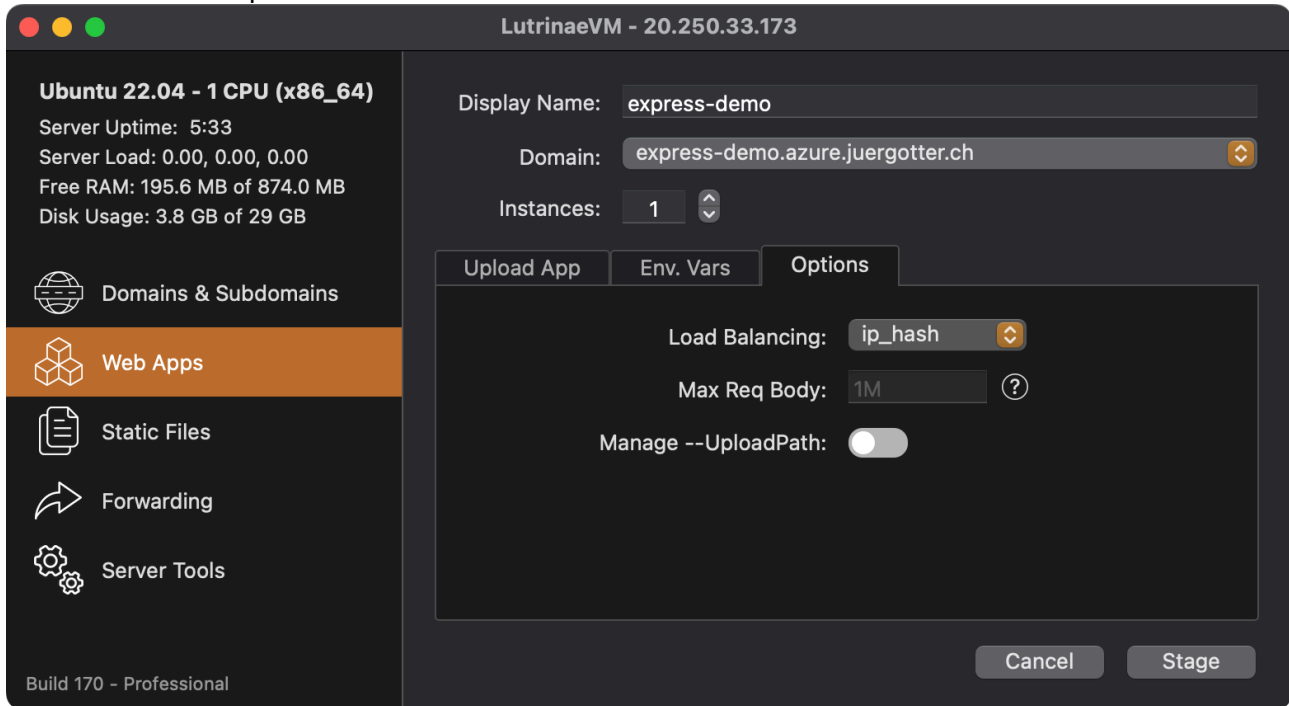Lifeboat should automatically detect the executable and select it.

Switch to the Tab "Env. Vars", and add: `EXPRESS_DEMO=1`

The Express Console Demo is looking for a launch argument or an environment variable to select the demo to be run. This Environment Variable will launch demo 1. If no Launch Argument and no Environment Variable are set, the Express Console Demo would ask for user input - which obviously won't happen without user interaction.

In a real world Express application this would not be necessary, since you wouldn't ask for user input. However, it's good to point out the Environment Variables. You can read them in the Express Console Application and react. This could be useful to set/change Log Verbosity or other behavior depending on the Environment Variables, which you can configure using Lifeboat.
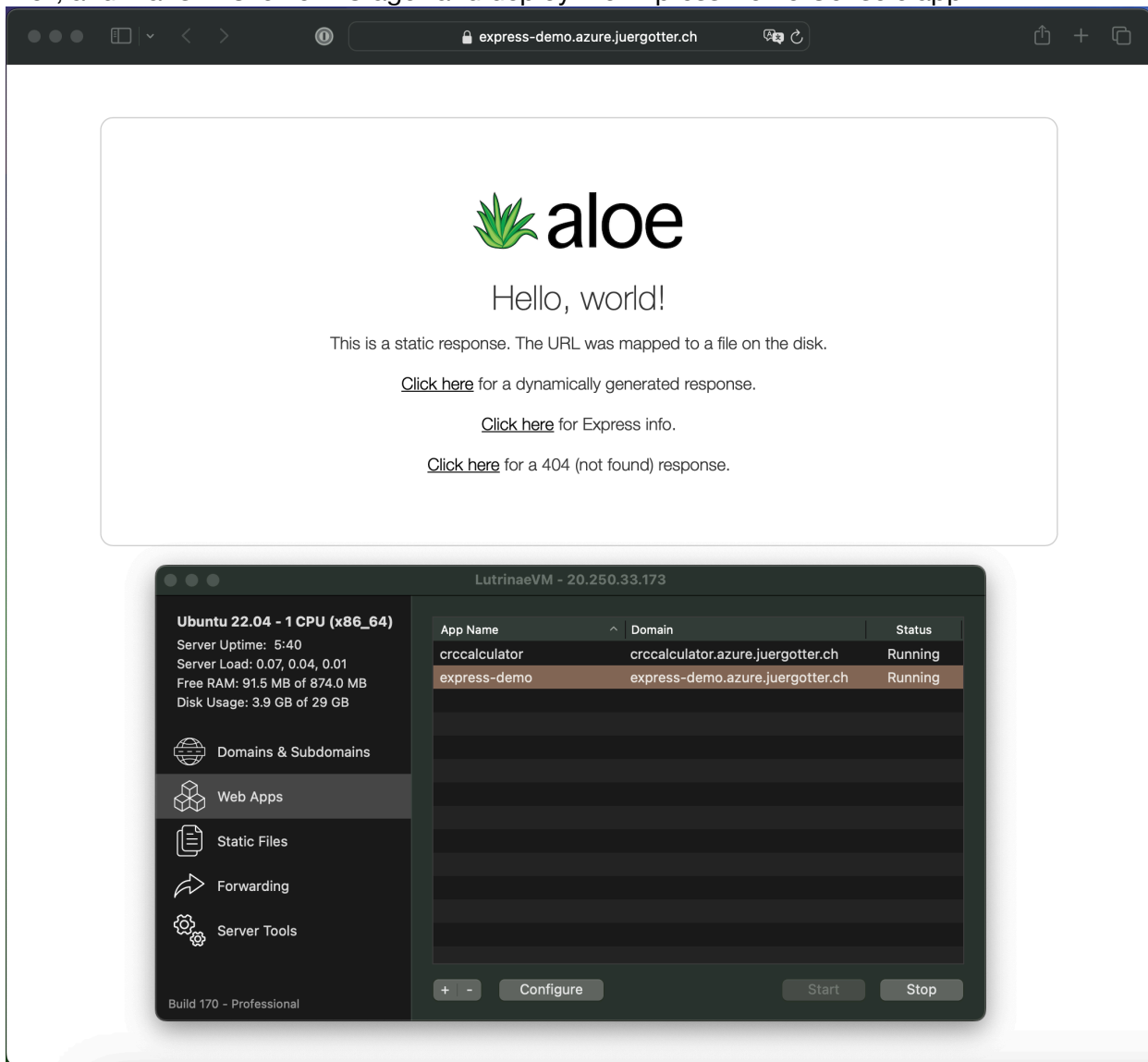
Next to the Tab "Options":

Since we have enabled the advanced options, we could change the Load Balancing Method here. This obviously only has any effect if you're using multiple instances for Load Balancing.

Note: The load balancing method `ip_hash` is the only one that is compatible with a Session based application. If you are running e.g. a REST API and are not using sessions then you can change and use the Load Balancing Method that works best for your use case.

Well, and that's it! Click on "Stage" and deploy the Express Demo Console app.

Note: The cool thing about the Environment Variable `EXPRESS_DEMO=x` is that you can change it at any time in Lifeboat, e.g. set it's value to 4. Then just click "Stage", and you can try Demo Number 4. All without having to re-upload the demo application.