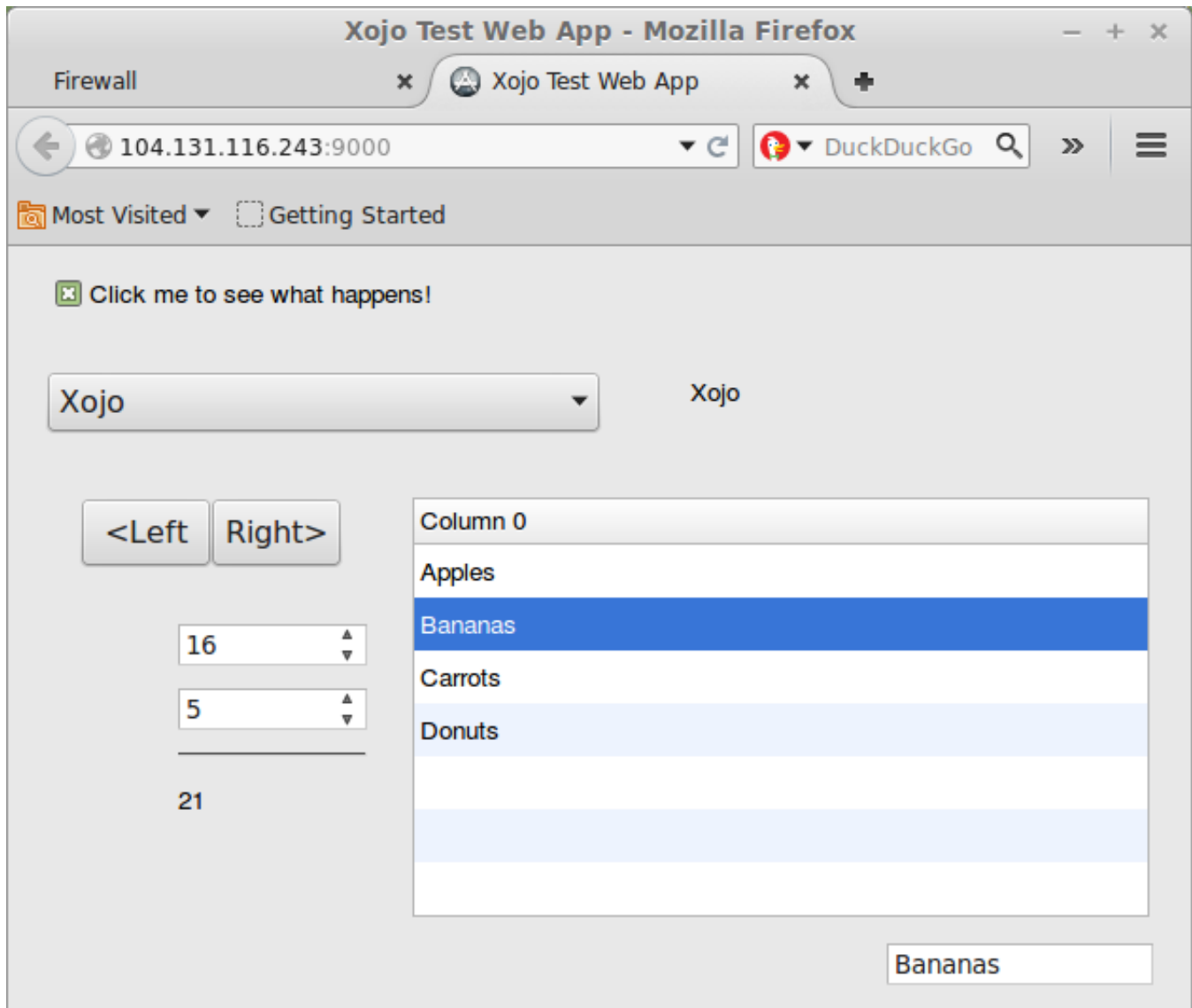


Xojo Web Application Deployment on the Cheap

When you want to publish your Xojo Web Application to the web, you have several great choices. The most obvious is to use the [Xojo Cloud Service](#). Their servers are RackSpace based servers, optimized for Xojo Web Applications with security and (some) maintenance built in. Prices begin at \$49 per month. Another option is from [1701 Software](#). They have a wealth of experience in getting Xojo based Web Applications deployed. CubeSQL database support comes standard. Prices start at \$15 per month with discounts for paying a year at a time.



What if you want to just dip your toes in the water however? Or what if you need rock bottom pricing but still at a cloud based solution? Today we'll be looking at one such provider: [DigitalOcean](#). Prices start at \$5 per month and up. That's \$0.007 per hour for the base level hosting.

How to build your Web App for deployment

We're going to build a "Console" application, so in your project make sure to switch the build type from "CGI" to

“Stand Alone” from the Xojo IDE / Build Settings / Shared / Deployment Type setting. You also want to make sure that you have the Xojo IDE / Build Settings / Linux checkbox checked, and the “Xojo Cloud” checkbox un-checked. Now you can build your application and have it deployable to our Digital Ocean droplet.

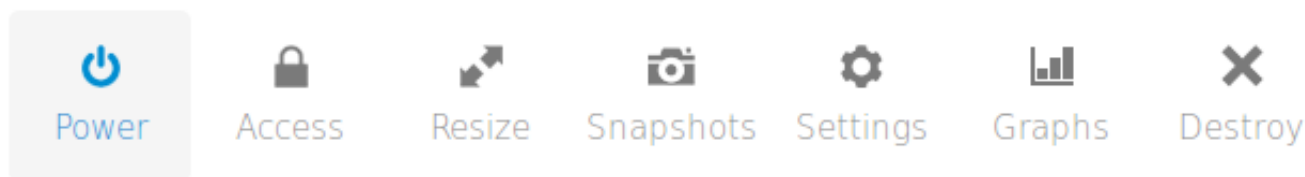
Server Setup and Creation

Here’s the basic steps to get up and running. This won’t be comprehensive on how to fully secure a website from attack, but it will get you started. You’ve been warned. First step is to register with Digital Ocean. This includes supplying your credit card information. With that taken care of, head over to the “Create” Droplet. A “Droplet” is the term that they use for their virtual server. (A "droplet" in the "Digital Ocean". Get it?)

Here are the options and choices needed to create a droplet:

1. Droplet Name. I made my TestXojoApp1.
2. Select Size. Yes, the \$5/mo or \$0.007/hour works just fine for simple Xojo Web Applications. Start here. They make it easy to upsize your droplets.
3. Select Region. I’m on the Eastern side of the USA so I chose the “New York” region. Pick the one closest geographically to you, or your customers.
4. Available Settings. I left this blank by default.
5. Select Image. Here’s where it gets interesting. You can choose various preselected server setups by “Application” such as LAMP, Drupal, etc. As we’re just starting up our server, and Xojo builds 32 bit applications, we’ll choose “Ubuntu 14.04 x32”. No extra packages are needed to be installed with this option so that makes it a little easier!
6. Click the "Create Droplet" button. In under a minute, you will have a server instantiated! Mine took only 39 seconds the first time I created it. My second droplet took 46 seconds. It is amazing the speed of these SSD drives.

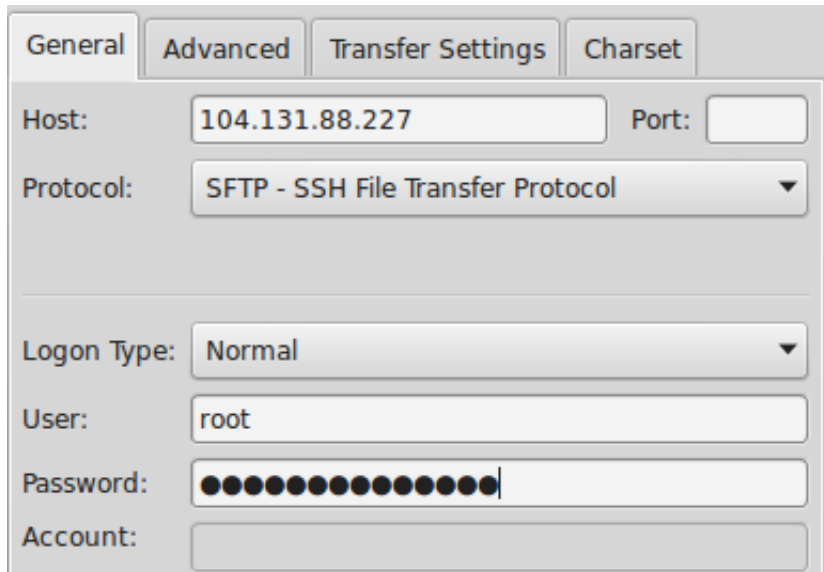
Your droplet will be started automatically, and you get a console control that looks like this:



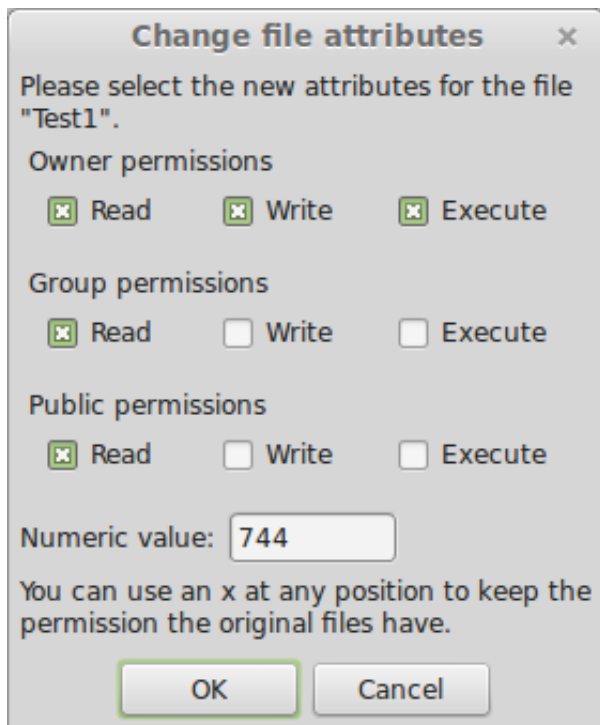
Digital Ocean will email you the server information including the server IP address, and the root user password. Don’t worry, you’ll be required to change the password the first time you log in. While you’re waiting on the email, let us go ahead and click on the “Access” button and then the “Console Access” button which will give us SSH terminal access to the server from within the browser. When the droplet email arrives, then log in as “root” and use the password you received. It will tell you that you need to change the password. *You need to retype the root password again!* I’m not a very careful reader evidently and this tripped me up for a minute. Next, type in your new password ... and then again to verify. We’re in! Type (that’s a lower case “LL”) and press ENTER at the prompt and you’ll see that you’re in the root directory of your server!

Xojo Application Setup

I'm assuming that you have your test web application already built and tested. We'll transfer the application and supporting libraries to the server using SFTP. I'll use FileZilla to do the SFTP transfer for me. I'm going to use the FileZilla Site Manager to help make the connections easier to maintain.



Transfer the files and all subdirectories of your application to the server. For the main application, make sure that you set the execution privileges.



We need that. In my case, my application is built to "Test1" so not only do I need that file on the server, I also need the "Test1 Libs" folder as well as the "Resources" folder.

| Filename ^ | Filesize | Filetype | Last modified | Permissions |
|---------------|----------|-----------|------------------------|-------------|
| .. | | | | |
| .cache | | Directory | 04/17/2014 12:51:00 PM | drwx----- |
| .ssh | | Directory | 10/10/2014 11:34:00 AM | drwx----- |
| Resources | | Directory | 10/10/2014 12:01:00 PM | drwxr-xr-x |
| Test1 Libs | | Directory | 10/10/2014 12:02:00 PM | drwxr-xr-x |
| .bash_history | 0 B | File | 10/02/2014 03:08:00 PM | -rw-r--r-- |
| .bashrc | 819 B | File | 02/19/2014 | -rw-r--r-- |
| .profile | 140 B | File | 02/19/2014 | -rw-r--r-- |
| Test1 | 4.1 MB | File | 10/10/2014 12:02:00 PM | -rwxr--r-- |

Note the 'X' for Execute!

Let's go back to the SSH console in the browser. You can do another "ll" to see the files and folders that you uploaded. From the console, I type ". /Test1" and press ENTER which gets my application running on the server. I built my application to monitor port 9000, so in another browser tab, I type in my server IP address and then a colon and then "9000". If all is configured properly, you should see your application start up in the browser.

Notice that we didn't need to install any 32bit Linux libraries as they came default with the server! Easy! If you need a 64bit Linux server for some reason and you want to run a Xojo Web Application, follow the guidelines to install the 32bit libraries from the [Xojo Documentation](#).

Security

A note about Internet security: Security is a full time job. Security is a process, not a destination! Please remember that. However, in our case we'll at least do a tiny bit with security. Ubuntu comes with a very simple firewall that may help protect your app. The firewall is called "ufw" for "Uncomplicated Firewall".

On the server, it is enabled by default. You can check this by issuing a `sudo ufw status`. If it is not running, you can issue a `sudo ufw enable`. Because I'm not using a standard port I may need to open up that port by issuing a `sudo ufw allow 9000`. If you want more detail on the status of the firewall try `sudo ufw status verbose`.

For a more secure firewall, research [IPTables](#) and the numerous tutorials that can be found on the Internet. However, for getting started, UFW does the trick, and Digital Ocean make it easy to get started

By now, you may be wondering how much expense we've racked up? Unfortunately Digital Ocean updates their billing tab once a day, it seems. We may be up to \$0.02 depending on how long this has taken for you.

An additional hosting option: Another competitor to Digital Ocean is [Vultr](#). Their pricing is competitive to Digital Ocean and has additional hosting options such as Windows and BSD servers.

Shutting down your server

You will be charged for your droplet, even if it is powered off. The way around this, is to take a *snapshot* of your

server. The steps are [1] shut down the applications then [2] power off the server then [3] take a snapshot and finally [4] destroy the server. You can restore the droplet from the snapshot instead of creating a new server from scratch if you want the server to be restored.

First let's stop our application from running, by doing a "Ctrl+C". From the console we can ask the server to shutdown by doing a `sudo shutdown now`. We will then power off the droplet. Do this by going to the "Power" option, and then select the "Power Off" button. When you go to the list of droplets, you will see the status as "off". From the menu, then choose "Snapshots" and then "Take a Snapshot". Taking a snapshot may take some time so be patient. Once the snapshot has been created, then choose the "Destroy" option for your droplet server and from this point on, you will not be charged for the droplet.

I hope you found this guide to be useful. Now get out there and build some tremendous Xojo web applications! Even if you want your server to run for an hour.