https://github.com/NancyFx/Nancy/wiki/Hosting-Nancy-with-Nginx-on-Ubuntu

# Hosting Nancy with Nginx on Ubuntu

This tutorial describes how to install and run a NancyFx powered website on Ubuntu 14.04, but newer versions will probably work fine.

### Install mono on your ubuntu machine

> Note: Due to how Ubuntu updates its packages, the version of Mono that comes bundled on Ubuntu will almost always be outdated, and as of April 2015 is not suitable for our purposes.

In order to install Mono, it is highly recommended that you:

1. Follow these instructions to get the latest version from the official mono repositories; or

2. Compile it yourself (for advanced users familiar with Linux)

To compile it yourself, go to the mono download page to retrieve the the latest mono version. In our case, we build the latest source from github:

```
$ sudo apt-get install git autoconf automake libtool g++ gettext
$ mkdir ~/src
$ cd ~/src
$ git clone git://github.com/mono/mono.git
$ cd mono
$ ./autogen.sh --prefix=/usr/local
$ make get-monolite-latest
$ make
$ sudo make install
```

Now we are ready to run .NET applications under linux.

### Create a Nancy Website

Open Visual Studio (this tutorial was written using VS 2015 Community Edition), and create a new Console Application.

Install the following nuget packages:

```
Install-Package Nancy.Hosting.Self
Install-Package Mono.Posix
```

Edit the Program.cs file and add the following code:

```
using Mono.Unix;
using Mono.Unix.Native;
using Nancy.Hosting.Self;
using System;

namespace NancyDemo
{
    class Program
    {
        static void Main(string[] args)
        {
            var uri = "http://localhost:8888";
            Console.WriteLine("Starting Nancy on " + uri);

            // initialize an instance of NancyHost
            var host = new NancyHost(new Uri(uri));
            host.Start();  // start hosting

            // check if we're running on mono
            if (Type.GetType("Mono.Runtime") != null)
            {
                // on mono, processes will usually run as daemons - this allows you to listen
                // for termination signals (ctrl+c, shutdown, etc) and finalize correctly
                UnixSignal.WaitAny(new[] {
                    new UnixSignal(Signum.SIGINT),
                    new UnixSignal(Signum.SIGTERM),
                    new UnixSignal(Signum.SIGQUIT),
                    new UnixSignal(Signum.SIGHUP)
                });
            }
            else
            {
                Console.ReadLine();
            }

            Console.WriteLine("Stopping Nancy");
            host.Stop();  // stop hosting
        }
    }
}
```

Create a new file HelloModule.cs with the following code:

```
using Nancy;

namespace NancyDemo
{
    public class HelloModule : NancyModule
    {
        public HelloModule()
        {
            Get["/"] = parameters => "Hello World!";
        }
    }
}
```

Make sure it works locally first! Go ahead and start the debugger. You should be able to go to `http://localhost:8888` in your web browser and see the "Hello World!" message.

### Install nginx

nginx is the webserver we're using. We configure it to forward all requests to the nancy self hosted application. The content folder with static files will be handled by nginx.

```
$ sudo apt-get install nginx
```

Create the website configuration file in /etc/nginx/sites-available/nancydemo with the following content. The server_name is the domain on which the request will be handled. Change this to your own value.

```
server {
    listen       80;
    server_name  yourdomainname.com;
    root /var/www/nancydemo;

    location /Content/ {
        alias /var/www/nancydemo/Content/;
        location ~*  \.(jpg|jpeg|png|gif|ico|css|js|ttf)$ {
            expires 365d;
        }
    }

    location / {
            proxy_pass http://127.0.0.1:8888;
    }
}
```

When testing, instead of `yourdomainname.com`, you can substitute the local IP address, eg `192.168.1.200`.

To enable the website, create a symbolic link from the sites-available to the sites-enabled folder. This will make it easy to temporary disable sites in the future.

```
$ sudo ln -s /etc/nginx/sites-available/nancydemo
/etc/nginx/sites-enabled/nancydemo
```

The configuration is completed, reload Nginx to apply.

```
$ sudo /etc/init.d/nginx reload
```

### Install supervisor

To make sure our nancy self hosted website never stops, we use supervisor. This program makes sure that NancyDemo keeps running.

```
$ apt-get install supervisor
```

Configure supervisor by creating a new file /etc/supervisor/conf.d/nancydemo.conf

```
[program:nancydemo]
command=mono NancyDemo.exe -d
user=www-data
stderr_logfile = /var/log/supervisor/nancydemo-err.log
stdout_logfile = /var/log/supervisor/nancydemo-stdout.log
directory=/var/www/nancydemo/
```

**NOTE** If you compiled Mono yourself in the first step, change the second line to `command=/usr/bin/mono NancyDemo.exe -d`

Start the control manager of supervisor

```
$ sudo supervisorctl
```

And update the configuration. You should see that there is a new process added. Now start nancydemo.

```
$ supervisor>update
$ supervisor>start nancydemo
```

### Put your hands in the air

Go to http://yourdomain.com and see our baby saying "Hello World".

### Notes

Linux is very critical about uppercase and lowercase letters, so if a view can't be found, make sure you've used the exact name with the exact uppercase and lowercase letters.