💻 Activity D4: Setting Up DNS

Re-submit Assignment

Due Jan 11 by 11:59pm **Points** 5 **Submitting** a website url

In this activity, you will setup a DNS configuration for your server that is hosted with Digital Ocean. The Domain Name Service (DNS) allows you to translate the *name* of the server you want to connect to (e.g. emmasmith.org) into the *IP address* of this server (e.g. 176.32.98.166).

We will show you how to use DNS and Nginx to serve your website on a domain of your choosing, for example emmasmith.org.

Doing this involves the following steps:

Buy a Domain Name

Visit NameCheap to purchase a domain name. You will need to create an account there. A domain name may cost you about $10 for a year. For example, you may be able to purchase a domainname that uses your real name, such as emmasmith.org. You can pick whatever you like.

Configure Your Domain Name

After you purchase your domain name, you need to set it up so that anyone who uses DNS to look up your domain name will retrieve the IP address of your Digital Ocean server.

To do this, the first step is to add a new DNS record. In Name Cheap, navigate to your domain and click "Advanced DNS".

You need to add two DNS records and delete the two default records that are there. (The defaults are a CNAME and a URL redirect that shows a parked web page).

1. Click "Add new record". Choose "A record" and enter "@" for the host, the IP address of your Digital Ocean machine, and "20 minutes" for the TTL. Save this.
2. Click "Add new record". Choose "A record" and enter "*" for the host, the IP address of your Digital Ocean machine, and "20 minutes" for the TTL. Save this.

Delete the other two records.

It may take some time for these changes to be updated throughout the Internet (15 to 30 minutes), but after some time you should be able to browse to your website using the domain name you purchased and any subdomains. For example, if you try the following domains, they should go to the same site:

- emmasmith.org

- lab1.emmasmith.org
- cp1.emmasmith.org

Understanding NGINX configuration

Let's take a look at the configuration for your Nginx web server so that we can understand how it works. This is located in /etc/nginx/sites-enabled/default. You should be able to navigate and view this file using the command line:

```
cd /etc/nginx/sites-available
less default
```

Here is the default configuration:

```
server {
    listen 80 default_server;
    listen [::]:80 default_server;
    root /var/www/html;

    index index.html index.htm index.nginx-debian.html;

    server_name _;

    location / {
        # First attempt to serve request as file, then
        # as directory, then fall back to displaying a 404.
        try_files $uri $uri/ =404;
    }
}
```

Note that there are many other lines that begin with the "#" character. These are all comments. Starting lines with a "#" is the typical way for Linux configuration files to include comments.

In nginx, the web server configuration is given as a "server block", which is why the first line starts with "server". You can setup multiple websites using the same web server, and each website gets its own server block. Later, we will setup a special server block for "lab1.emmasmith.org" and "lab2.emmasmith.org". This file contains the default server block and will be used for any website that hasn't otherwise been defined.

The first "listen" line says to run a web server on port 80, which is the standard port for HTTP. This will be the "default" server, meaning any website (emmasmith.org, www.emmasmith.org, lab1.emmasmith.org, etc.) that you have not otherwise configured will use this configuration.

The second "listen" line allows web browsers to connect to your website using IPv6 (the newest version of the IP protocol) in addition to IPv4 (the version we have been running for nearly 50 years).

The "root" line tells the web server where to find the files it will serve. These are located in /var/www/html.

The "index" line tells the web server which files will be used when browsing a directory. For example, when you visit http://emmasmith.org, you are technically visiting the directory named "/" and the web server will look for a file called "index.html" or "index.htm" or any of the other names listed in the /var/www/html directory. If you visit http://emmasmith.org/papers" then it will likewise look for one of these files in the /var/www/html/papers directory.

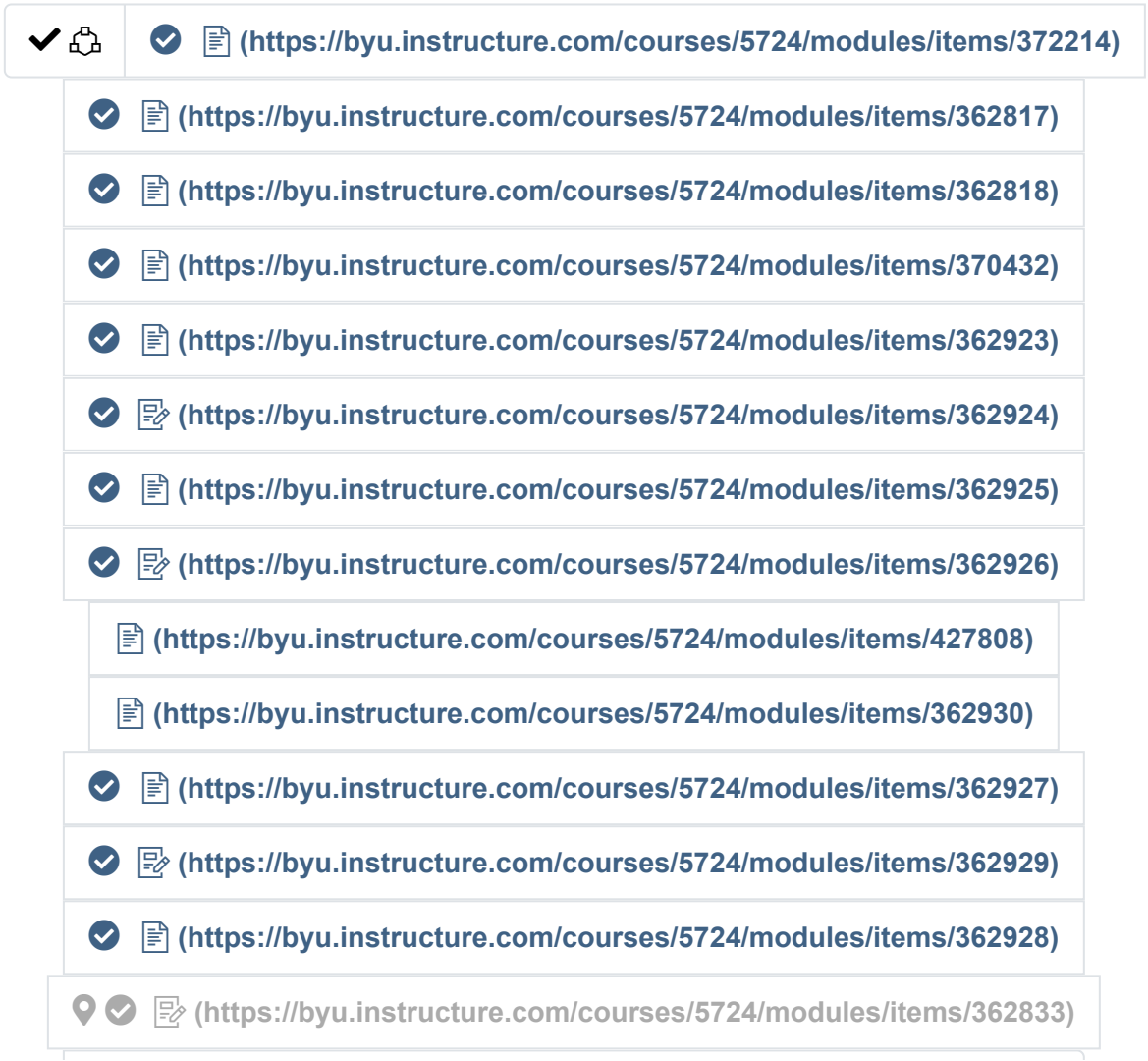
The "server_name" line lists the names your server can use. Since this line uses an invalid hostname of "_", it will not conflict with any other server names we set up later for other server blocks.

The "location" line tells the web server how to handle requests for your website. Since it lists "/", known as "root", this location block will handle all of the requests for your website.

The "try_files" line tells the web server to first use the URL being visited (e.g., emmasmith.org), then try adding a slash to the end of it (e.g., emmasmith.org/) and if both of these fail, then return a 404 Not Found error.

Submission

Turn in a URL for your equivalent of emmasmith.org. This should let us see the code you setup from the previous activity.



(https://byu.instructure.com/courses/5724/modules/items/415269)