ASSIGN A CUSTOM DOMAIN NAME TO YOUR WEBSITE

What is a domain name?

In the last unit, you deployed your site and GitHub Pages assigned your site a default URL, or domain name.

In this unit, you'll purchase your own custom domain name and assign it to your GitHub Pages website. At the end of the unit, you'll be able to access your site using both your new domain name and your default GitHub Pages domain name.

Before you choose a custom domain name, it's important to first understand what domain names actually are.

Domain names are human-friendly names that identify servers on the Internet. A global system known as the Domain Name System (DNS) is used for storing which domain names correspond to which servers.

For example, Codecademy's domain name is `www.codecademy.com`. When you type the domain name into your browser, your computer asks the DNS to identify which servers should receive the request in order to load our website.

Note: This unit is optional. We'll walk you through the steps required to purchase a custom domain name and assign it to your GitHub Pages site. If you do not want to purchase a domain name, you won't have to. However, feel free to follow along to see what steps are required to do so (for future reference).

Instructions

Observe the diagram to the right.

In the top portion of the diagram:

1. Bob wants to call Sue, but does not have Sue's phone number memorized.
2. Bob types "Sue" into his phone and his phone identifies the phone number associated with Sue.
3. The phone call is then made. Sue's phone receives a request (it rings). If Sue answers, Bob and Sue are connected and can now communicate.

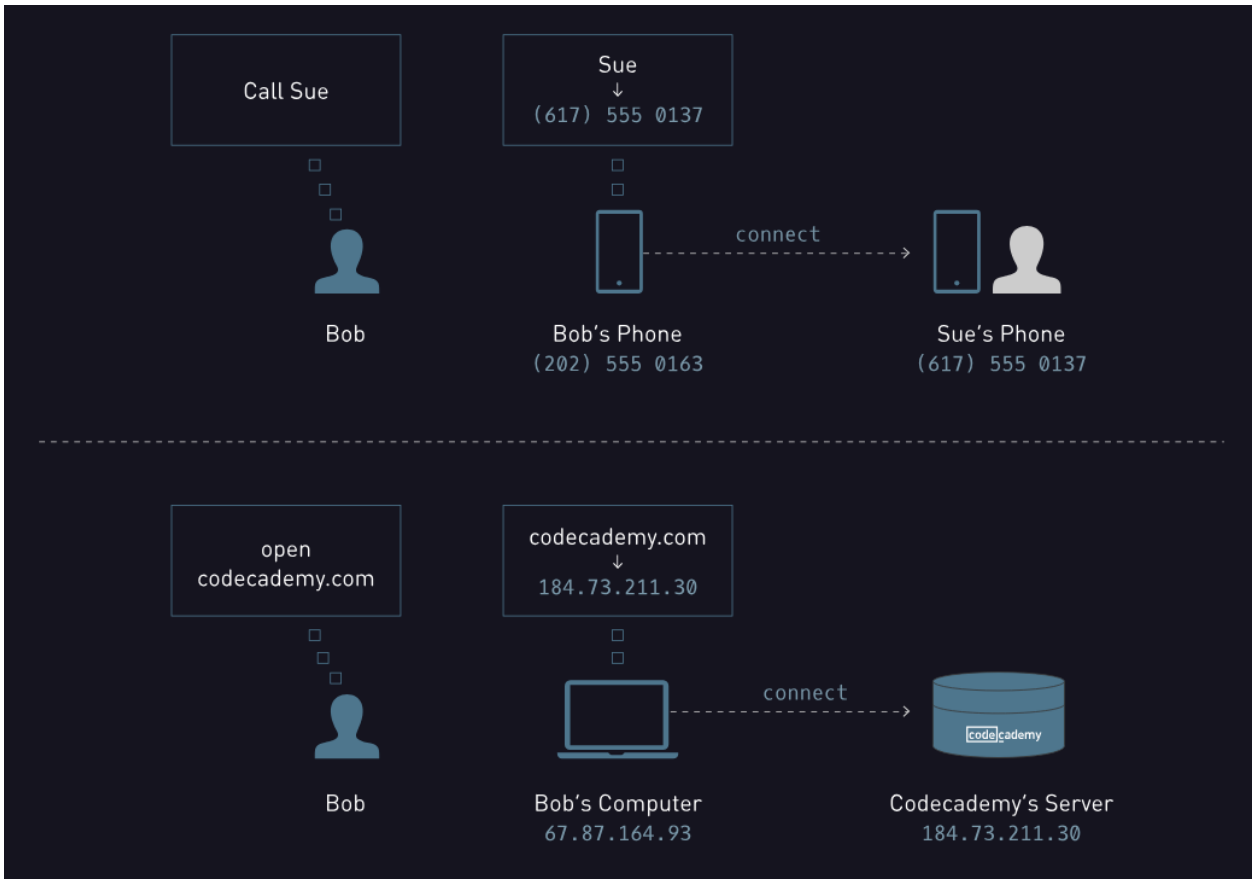
Your computer communicates with websites in a simliar fashion.

In the bottom portion of the diagram:

1. Bob wants to load the Codecademy website, but does not have Codecademy's IP address memorized.
2. Bob types `www.codecademy.com` into his browser and the DNS then identifies the IP address associated with `www.codecademy.com`.
3. The request to load Codecademy's website is then made. If the server successfully responds, Bob can now load and communicate with Codecademy's website.

For more details, read the following Codecademy resource:

1. HTTP Requests



Custom Domain I

Often, the most time consuming part of buying a domain name is actually deciding what you'd like it to be. Be aware that not all domain names are available; many have already been claimed by others.

We're going to use Amazon Web Services (AWS) to purchase your custom domain.

AWS is an industry standard suite of web infrastructure services used frequently by developers. The specific service we're going to use to purchase your domain name is called Route 53.

Instructions

If you don't already have an account, create one at https://aws.amazon.com.

AWS uses a thorough verification process. You'll have to confirm the following:

1. Contact information
2. Payment information (*no* purchase is necessary to open an account, but a debit or credit card is)
3. Identity verification (a PIN entered via a phone call)
4. Support Plan (there is a free plan)

Remember, creating an AWS account is free — there are no required purchases. The only purchase that you'll (optionally) make will be the purchase of your custom domain name.

Custom Domain II

AWS offers many services used for web development like servers, databases, and networking configuration. If you're new to using AWS it's easy to feel overwhelmed at first by all that it can offer, but we'll walk through the exact steps needed to create a custom domain name.

After logging into your account, you'll land on the AWS console. The console displays the many different services we mentioned earlier.

In this lesson we're going to focus on a service called "Route 53," under the "Networking" category. Route 53 can be used to purchase domain names and create DNS records.

Instructions

1. Log into your AWS account.

2. In the AWS console, click on “Route 53” located under the “Networking” title.
3. On the next page, click on the “Get Started Now” button under the “Domain Registration” section.
4. On the next page, locate the two buttons at the top of the page. Click on the button titled “Register Domain.”

Custom Domain III

Now it’s time to select a domain name and make sure it’s available.

Route 53 allows you to search the availability of a domain name you have in mind. It also offers many suffixes, like `.com`, `.io`, `.me`, and `.pizza`. If the domain name you want is unavailable as a `.com` for example, you can try using a different suffix.

The suffixes of domain names are known as top-level domains (TLDs). Different TLDs cost different annual prices.

Note: `.com` domains are the most popular and are therefore generally unavailable (or expensive).

Instructions

1. In the text field, type in a domain name you have in mind. Do not type in `www.` before the domain name.

For example, if you wanted to try purchasing our domain name, you would type in `codecademy` and select the `.com` TLD in the dropdown menu.

2. Once you’ve entered a domain name you like, click the button on the right labeled “Check” to see if it’s available.

3. If you’d like to purchase the domain name, select it and continue through the rest of the checkout process.

Setting Up Your Custom Domain

Congrats — you're the proud owner of a custom domain name!

You might notice, however, that your new domain name doesn't work yet — you can't visit it in your web browser. We have to connect it to your GitHub Pages website first.

There are two steps required:

1. Inform GitHub of the new domain name we'll be using (the one you purchased)
2. Set up DNS records in Route 53 that direct to GitHub

Let's start with the first step.

Instructions

1. Open GitHub and access the repo you created earlier titled `your-user-name.github.io`.
2. Click the "New file" button.
3. Name the new file `CNAME`. Do not add a file name extension.
4. In the file, on line 1, type the custom domain name you just purchased in the following format:

```
yourcustomdomain.com
```

You may have purchased a domain name with a TLD other than `.com`. In that case, make sure to use that TLD when creating the `CNAME` file.

5. Commit the new file.

6. Under the title of the repo, click on "Settings." Scroll down to the section titled "GitHub Pages" and confirm that there is a message *similar* to the following:

```
Your site is published at http://yourcustomdomain.com.
```

7. Try navigating to your website in your browser using your new domain name. It still doesn't work! Let's finish setting it up in the next few exercises.

Access Your Hosted Zone

The new `CNAME` file in your repo informs GitHub that you're assigning a new custom domain name to your GitHub Pages site.

Next, we have to let the rest of the Internet know that we want to associate the custom domain name with your GitHub Pages site.

We can do this by creating DNS records, which are globally accessible records that map domain names to servers.

The DNS records are created inside of a Hosted Zone in Route 53. A Hosted Zone is essentially a group of DNS records for a single domain.

Instructions

1. Access Route 53 once again. On the left side of the page, click on the title that says "Hosted Zones." Notice that you have a Hosted Zone for your new domain name. Click on it to open it.

Confirm NS Record

Domain names are associated with the correct DNS records by setting the domain name's *name servers*.

After a domain name is typed into a browser, the computer first retrieves the name servers that correspond to that domain name. The name servers are important because they're responsible with providing the computer with other important information (in the form of DNS records) associated with the domain name.

Setting your domain's name servers is important. The DNS is a global system, which means that anyone can create DNS records. We must verify that the DNS records we create were actually created by the owner of the domain name (in this case, you).

By doing this, the owner of a domain name ensures that only they have exclusive control over their domain's DNS records.

Instructions

1. Notice that the Hosted Zone for your domain name already has an `NS` (`Name server`) record. This record contains four values. These are the Hosted Zone's unique name servers. Take note of these values and copy them down somewhere.
2. On the left hand side, under "Domains," click on "Registered domains." Then, click on your domain name.
3. On the right hand side of the page, locate the section titled "Name Servers." Notice that these are the same name servers that your Hosted Zone's `NS` record contained. Route 53 did this for you automatically.

Create an A record

Now that your domain name is associated with the correct name servers, it's time to create some additional DNS records within the Hosted Zone.

The records that we'll create will be used by the name servers to help locate your site when a computer wants to load it. Specifically, the name servers will be responsible for providing that computer with important information stored in the records.

There are several different types of DNS records.

We're going to start by creating an `A` record, which stands for `Address` record.

An `A` record directs a domain name to an IP address. This record will associate our new custom domain name with Github's servers.

Instructions

1. Inside of your Hosted Zone, click on the button at the top labeled "Create Record Set." A form will appear to the right. Leave the "Name:" field blank. Set the "Type:" field to `A - IPv4 address`.
2. Leave the "TTL (Seconds)" value at the default of `300`.
3. In the "Value" text box, enter the following IP addresses (keep them on separate lines):

```
192.30.252.153
192.30.252.154
```

These IP addresses belong to GitHub. We are specifying that when your custom domain name is requested, the DNS should direct the request to GitHub. Read more information about this here.

4. Click the “Save Record Set” button at the bottom of the form.

Create a CNAME record

When setting up a website, it’s also conventional to also set up a `www` subdomain. `www` stands for `world wide web`.

Subdomains are part of a main (or root) domain. For example, `www.yourcustomdomain.com` is a subdomain of the `yourcustomdomain.com` root domain.

We can set up a subdomain using a `CNAME` record, which stands for `Canonical Name`.

A `CNAME` record specifies that a domain name will be used as an alias, or substitute, for the true (canonical) domain name.

Instructions

1. Inside of your Hosted Zone, click on the button at the top labeled “Create Record Set”.

A form will appear to the right. In the “Name:” field, enter *only* `www`. Set the “Type: “ field to `CNAME - Canonical name`. This step sets up the subdomain.

2. In the `Value` text box, enter the domain name that GitHub assigned to you earlier (the canonical domain name:

```
your-user-name.github.io
```

3. Click the “Save Record Set” button at the bottom of the form.

DNS Overview

Let’s review your DNS setup so far.

In Route 53, your domain name’s Hosted Zone contains the following:

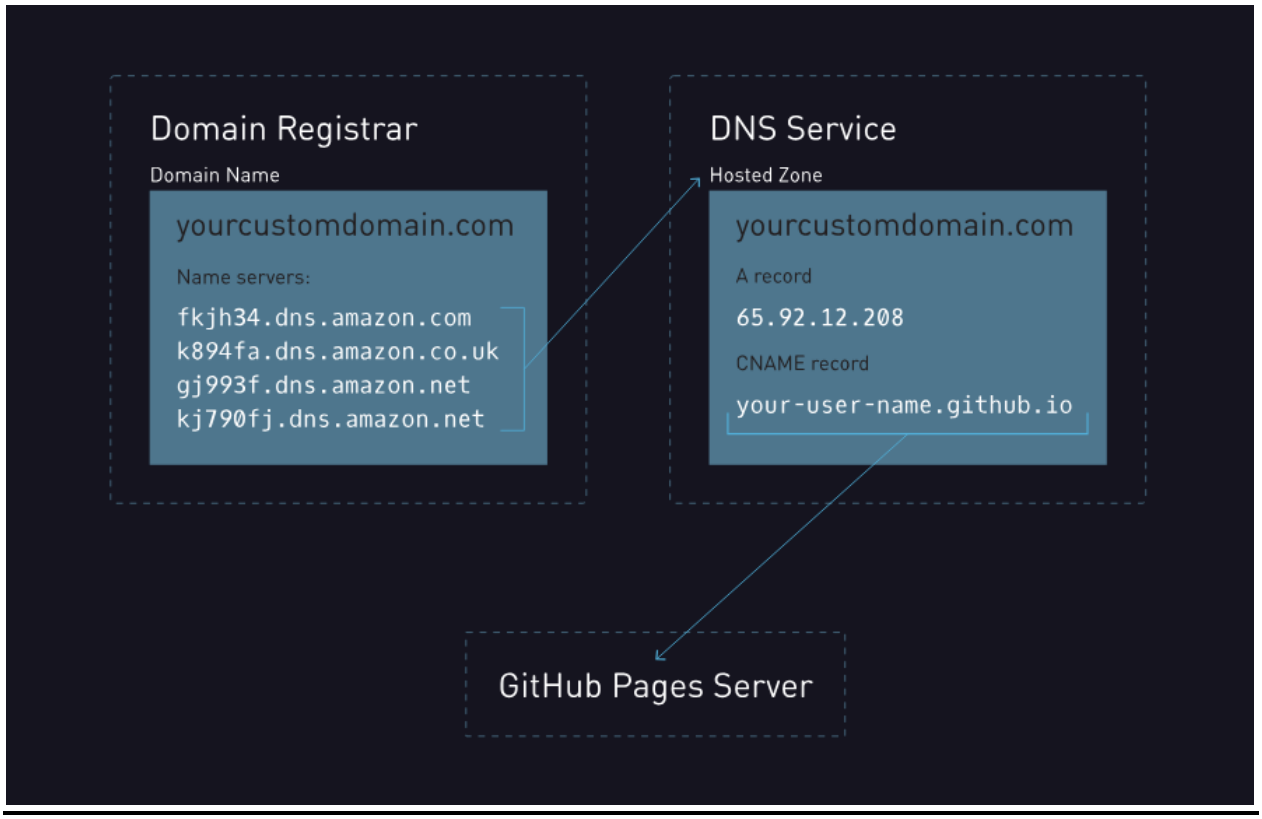
1. The `NS` (`Name Server`) record for your domain name. When a domain name is typed into a browser, the DNS looks to these name servers to help direct the request.

2. The `A` (or `Alias`) record. This record is used to direct requests of your domain name to GitHub's servers using their IP addresses.
3. The `CNAME` (or `Canonical name`) record. This record specifies what custom domain will point to your true (canonical) domain.

Instructions

Observe the diagram to the right. It outlines the overall setup of your DNS records within Route 53.

1. You purchased a custom domain name through a Domain Registrar, which in this case, is Route 53.
2. Four unique name servers were assigned to your custom domain name after your purchase.
3. To assign your custom domain name to your web site, you had to set up a Hosted Zone with multiple DNS records for your custom domain name. The Hosted Zone was set up within Route 53.
4. Inside of the Hosted Zone, the `NS` record was created automatically for you by Route 53. However, you created the `A` record and the `CNAME` record.
5. This setup allows you to visit your personal website with your new custom domain name, even though it's hosted on GitHub!



Testing your CNAME record

You've now created two DNS records: an `A` record for `yourcustomdomain.com` and a `CNAME` record for `www.yourcustomdomain.com`. Let's make sure they both work.

Instructions

1. Try opening your website using your root domain in the web browser. You should see your new GitHub Pages site.
2. Try opening your website using your `www` subdomain in the web browser. You should see your new GitHub Pages site.

Note: It may take a few minutes for the DNS record to take effect.

Review

Congratulations! You now have a static site with a custom domain name published on the Internet.

Let's review the process we followed in this lesson:

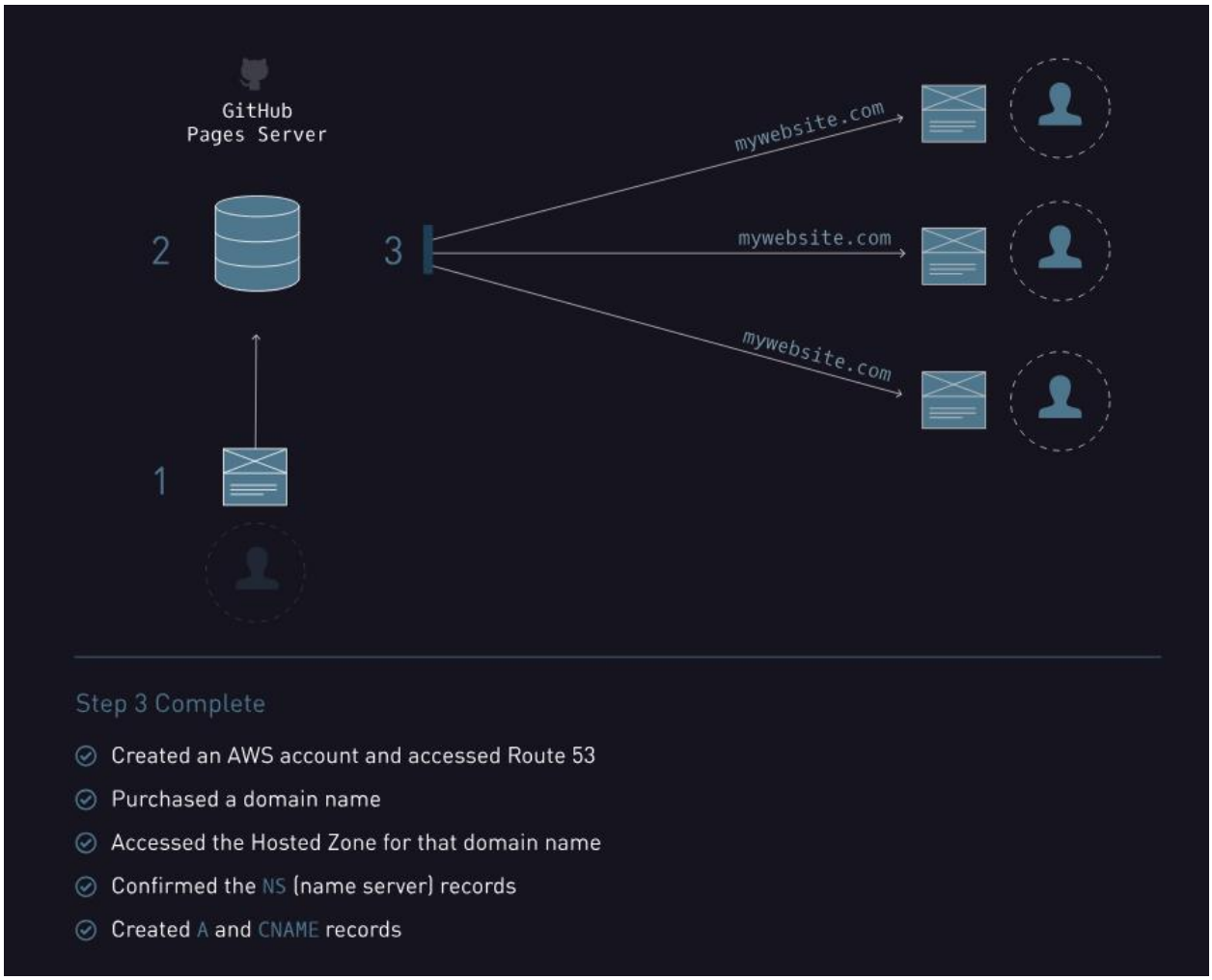
1. Created an AWS account and accessed Route 53
2. Purchased a domain name
3. Accessed the Hosted Zone for that domain name
4. Confirmed the `NS` (name server) records
5. Created `A` and `CNAME` records

Instructions

Take a look at the diagram to the right. In this unit, you successfully accomplished the third and final step: assigning a custom domain name to your GitHub Pages site.

Remember, there are many different ways to deploy a site to the Internet. In this course, you learned how to use Jekyll to generate site content and deploy that content to GitHub Pages, a free service provided by GitHub.

You're now set up to maintain and update your own personal site on the Internet. Happy coding!



Step 3 Complete

- Created an AWS account and accessed Route 53
- Purchased a domain name
- Accessed the Hosted Zone for that domain name
- Confirmed the NS (name server) records
- Created A and CNAME records