Make Your Own Personal Website

With Amazon Web Services

Based off: https://aws.amazon.com/getting-started/projects/host-static-website/

Cost Breakdown:

Registering a domain for 1 year with Route 53: usually around **$12**
Using a hosted zone on Route 53 per month per domain: **$0.50**
Hosting on S3 per month for a typical lightweight site: around **$0.01** (yes that is one penny)

-
Total annual cost to host a static website: about **$18.12**

*Note that you don't *have* to buy a domain and connect it to your S3 bucket. You could just shorten your S3 URL with something like www.tiny.cc, but having a dedicated domain looks nice!

Tutorial Outline

Part 1: Making a Local Site
Part 2: Registering a Domain
Part 3: Creating S3 Buckets
Part 4: Connecting Your Domain to Your S3 Buckets

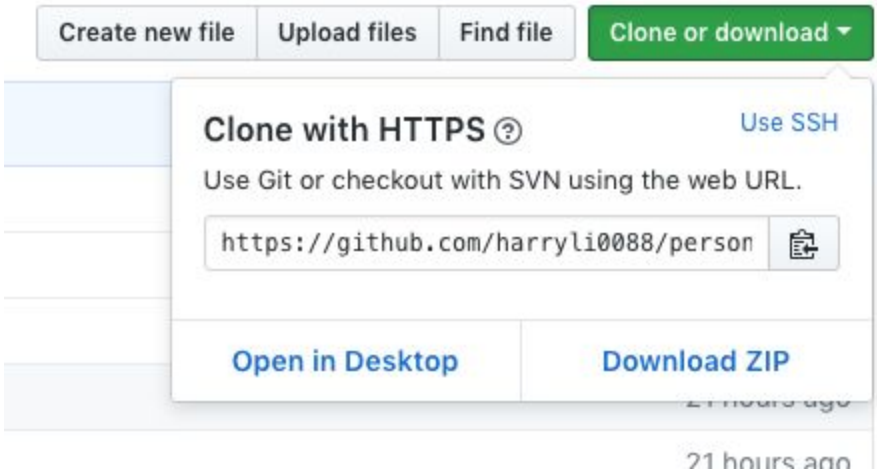
Requirements

- Computer with Internet connection
- Credit card for your AWS account for online payments
- Programming experience helps, but is not necessary
- Willingness to learn :)

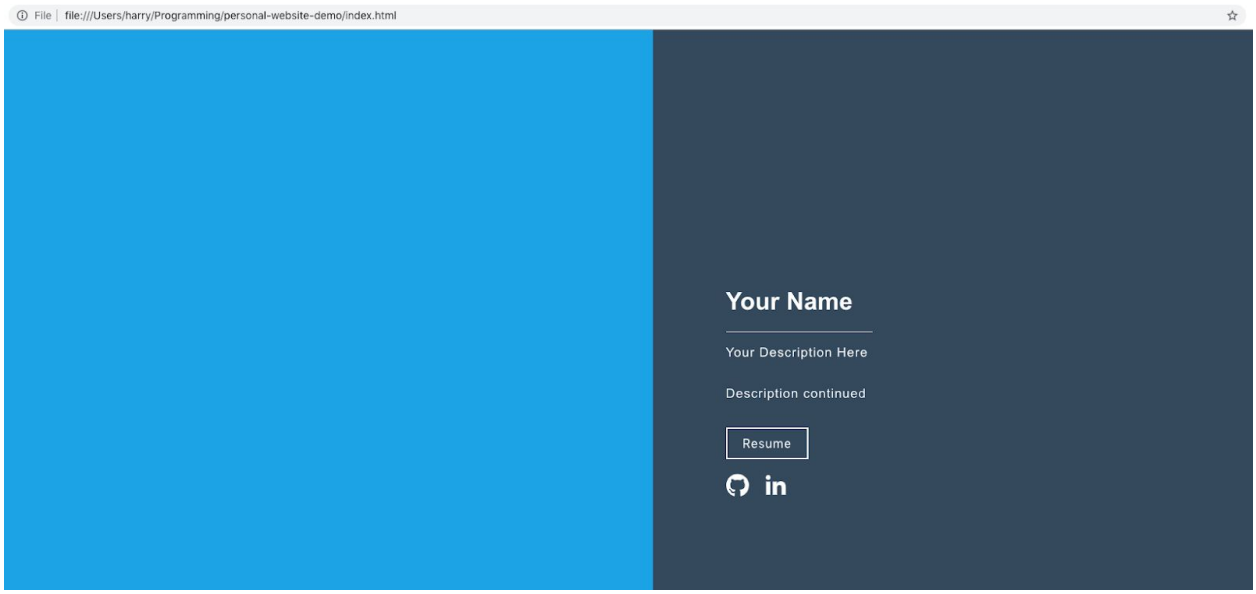
Part 1: Making a Local Site

https://github.com/harryli0088/personal-website-workshop

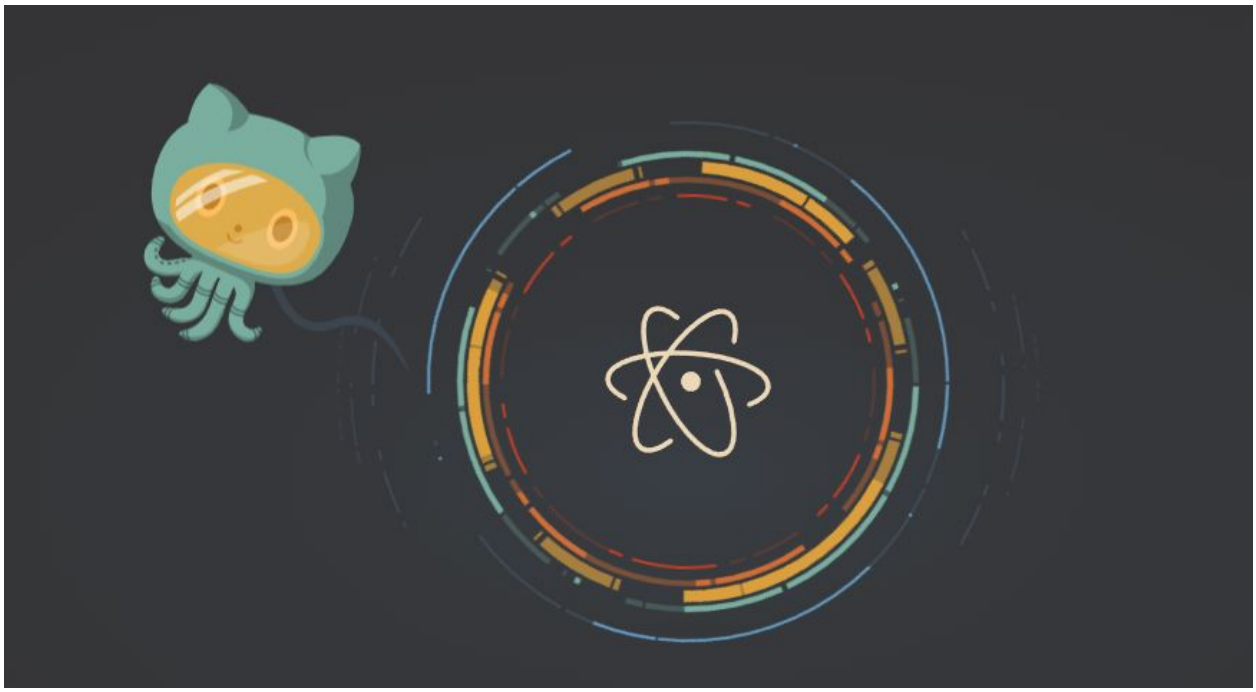
1) Click the link above. Click the green **Clone or download** button and click **Download ZIP** then unzip the project



2) Check out your local website! (Local means that the files are on your own computer, not somewhere else on the internet). Open your the folder you unzipped, then open **index.html** in your browser (Chrome, Firefox, no IE plz). You should see something like this



3) This is a template for you to edit with your own information! The site is laid out in a generic resume format and is even *reactive*, meaning that it is designed to look nice on wide computer monitors and narrow phone screens. You can open up the code in your favorite text editor. I personally like using Atom. You can download it here: https://atom.io/
You could use something like Notepad but you will soon hate yourself.



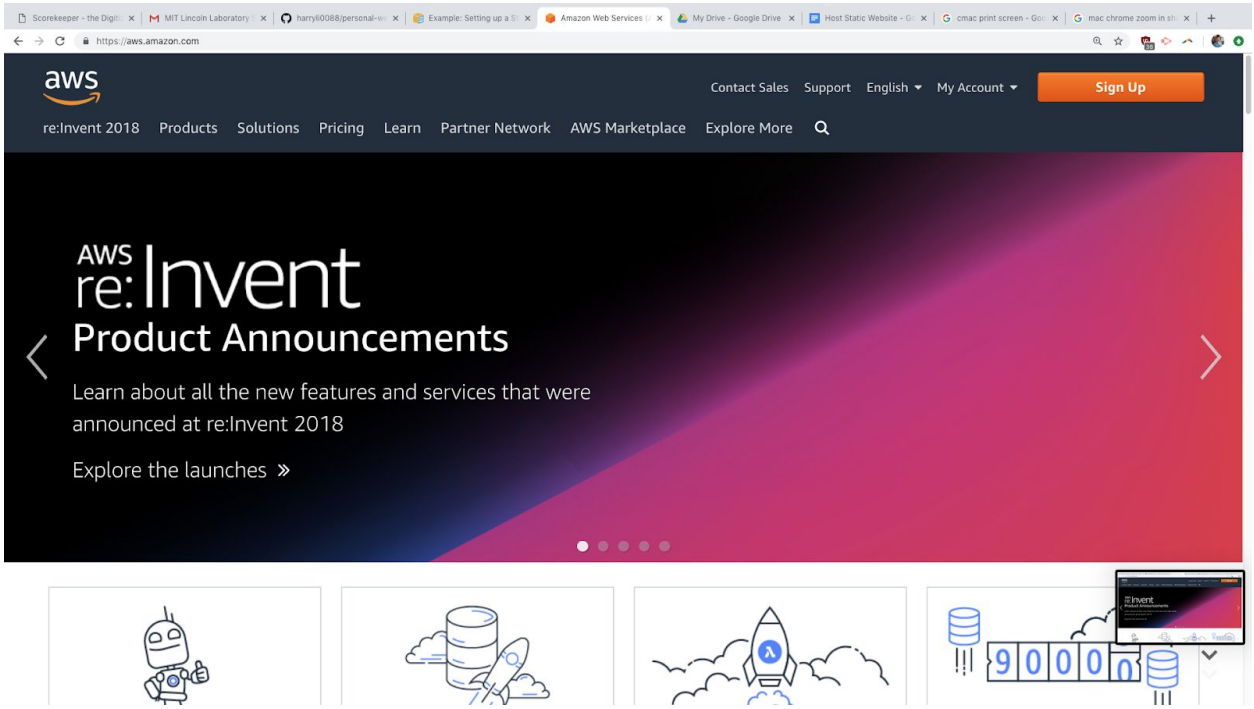
4) Open up **index.html** in your text editor to edit. Everytime you see “<!-- EDIT THIS! -->”, it means there is something in that line that you should replace with your own information. For example, in line 5, replace “Your Name” with, of course, your name.

```
<!-- template made by Harry Li -->
<html>
<head>
  <title>Your Name</title> <!-- EDIT THIS! -->
  <meta name="viewport" content="width=device-width, height=device-height, initial-scale=1">
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
  <meta name="description" content="search engine description here"> <!-- EDIT THIS! -->
  <meta name="keywords" content="search engine keywords here"> <!-- EDIT THIS! -->
  <link rel="stylesheet" href="main.css">
</head>
<body>
```

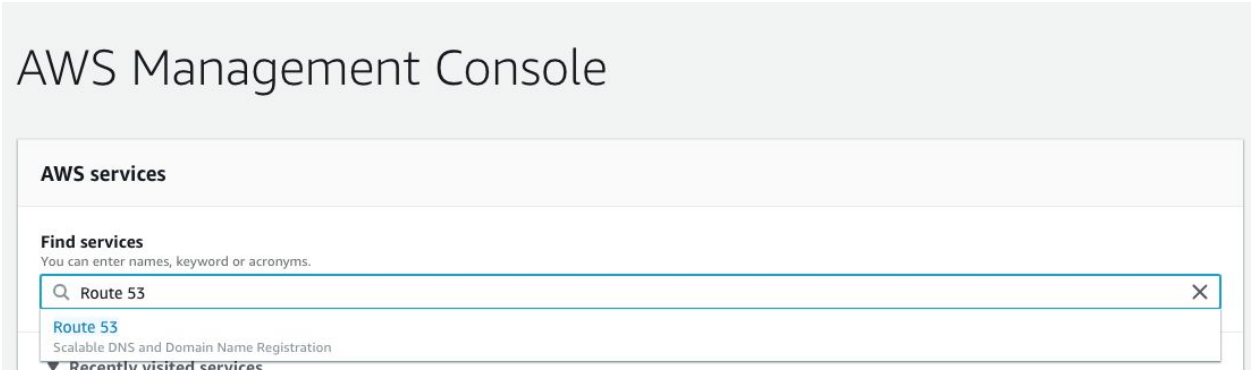
Part 2: Registering a Domain

https://docs.aws.amazon.com/Route53/latest/DeveloperGuide/domain-register.html

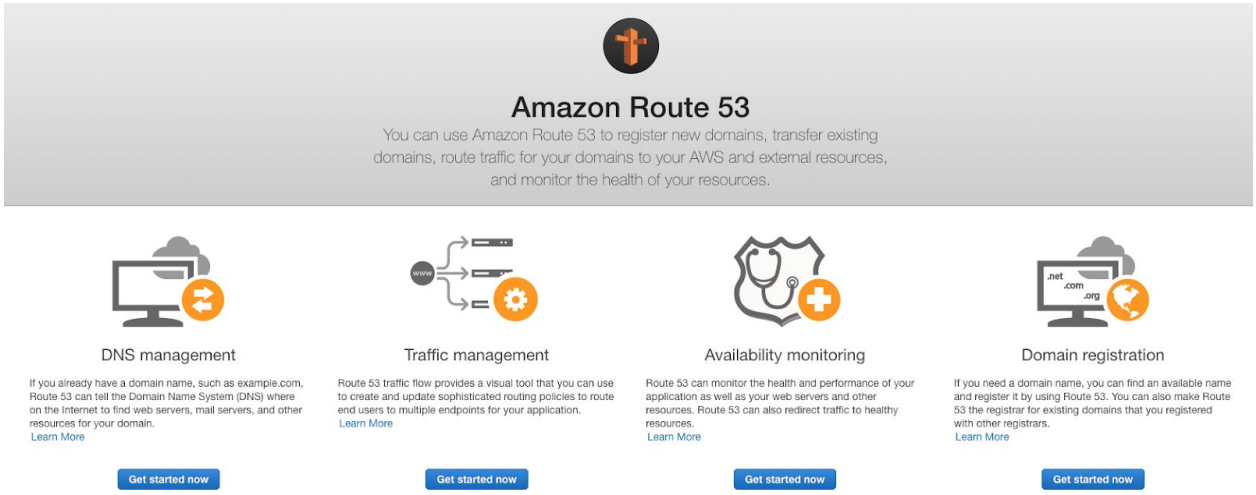
1) It's nice that we have this website locally, but we want anyone on the internet to be able to see it. We will be using **Amazon Web Services** (yes that Amazon) to host your site. Go to https://aws.amazon.com/ and make an account.



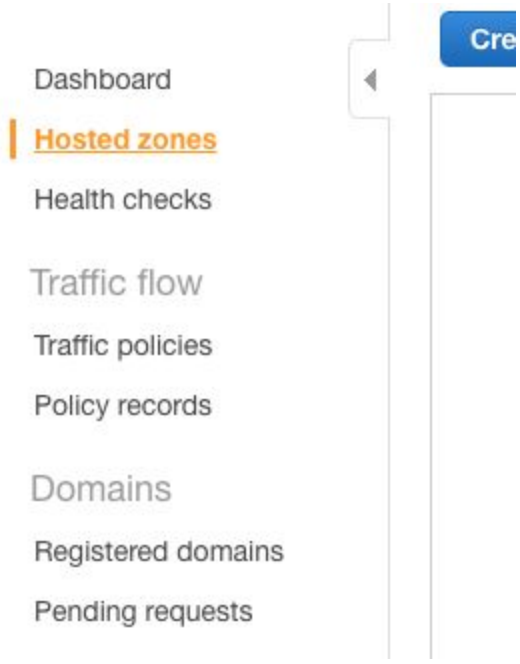
2) In your **AWS Management Console**, search for "Route 53". **Route 53** is Amazon's service to manage domains.



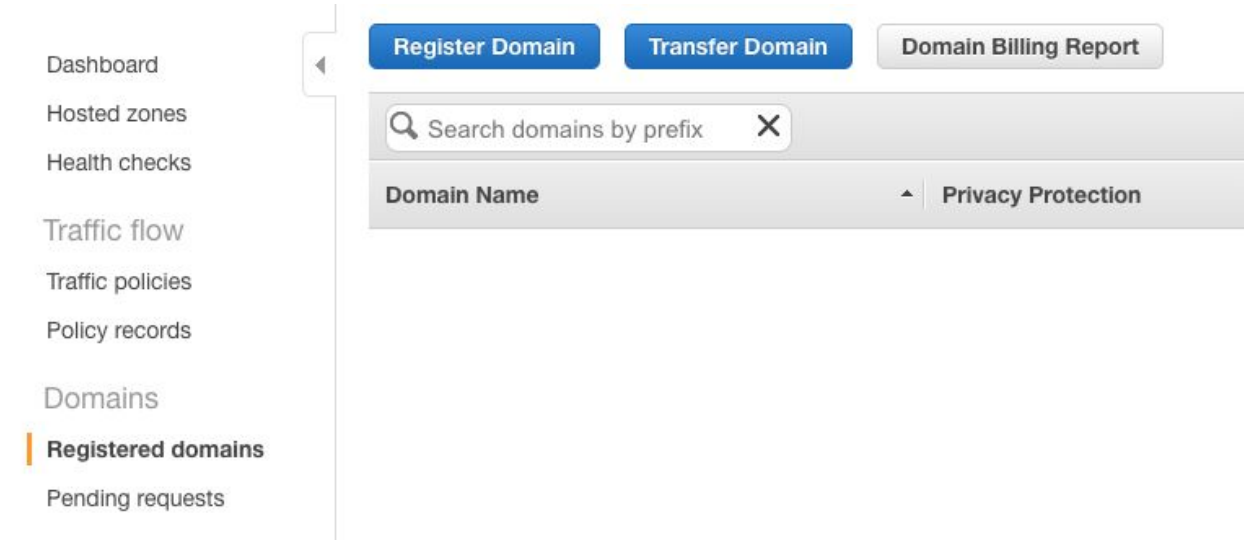
3) Click one of the **Get started now** buttons



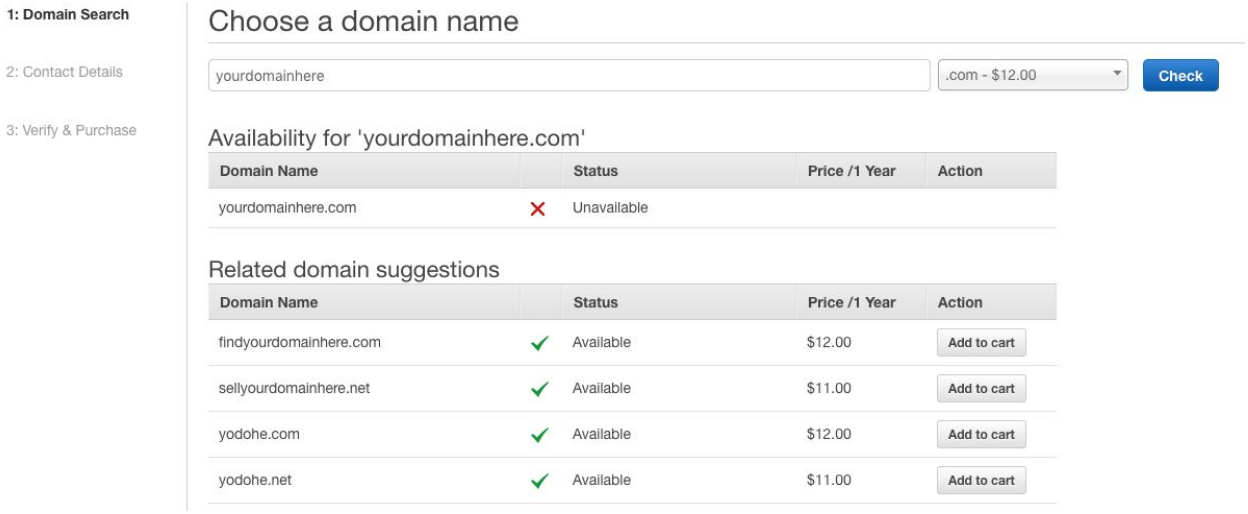
4) On the left-hand side, click the **Registered Domains** option



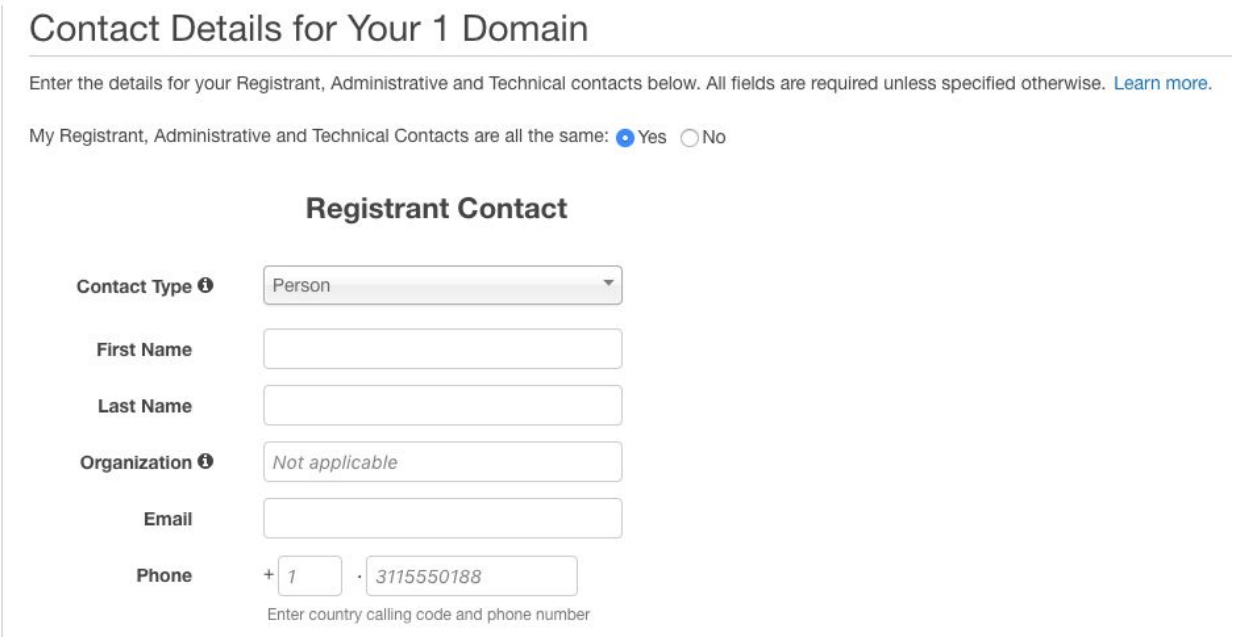
5) Click the **Register Domain** button



6) Search for available domains. You might end up spending a decent amount of time here. If your name is not available, try including your middle name initial. When you find one that you like, add it to your cart and continue.



7) Fill out some personal information to register the domain. Amazon should hide some of your information from outsiders. You can read more about their privacy policies here: https://docs.aws.amazon.com/Route53/latest/DeveloperGuide/domain-privacy-protection.html

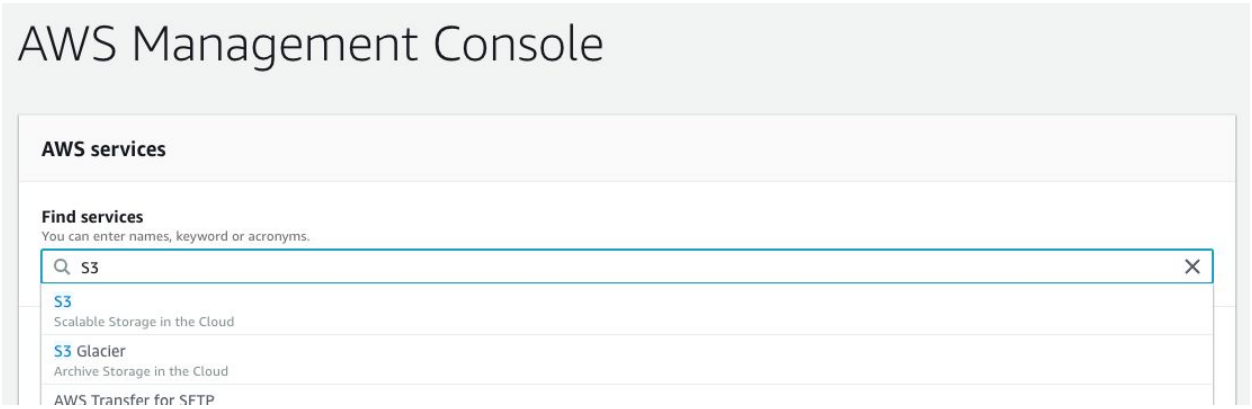


8) The email you enter will get a verification email to prove you have access. Verify that email and check the Terms and Conditions agreement to complete the purchase! You should see your new domain in your dashboard. Congrats! You have your own domain!! We will come back here later.
If you want, you can configure your domain to auto-renew every year. If you forget to renew, your domain will be back up for grabs!

Part 3: Creating S3 Buckets

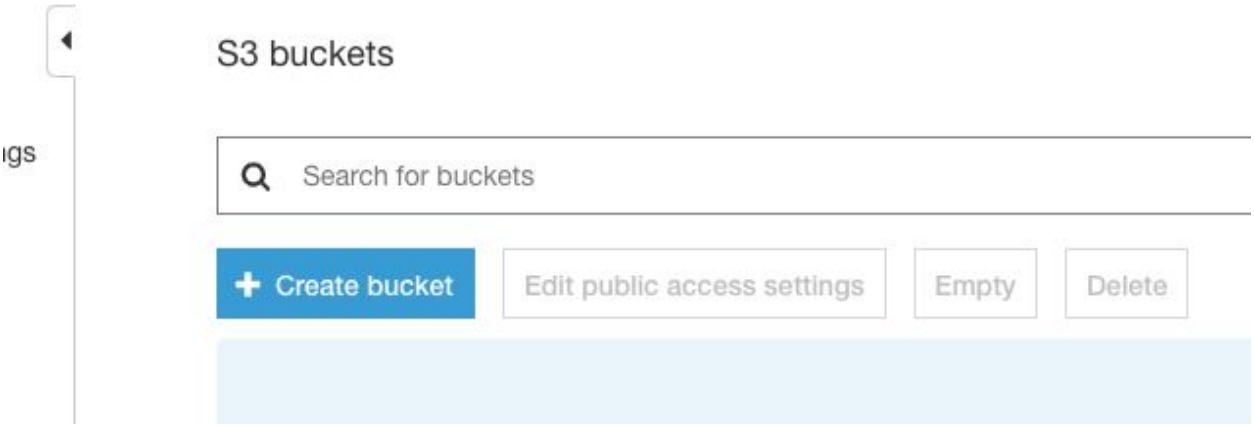
https://docs.aws.amazon.com/AmazonS3/latest/dev/website-hosting-custom-domain-walkthrough.html#root-domain-walkthrough-s3-tasks

1) Go back to your AWS Management Console (you can click the aws logo) and search for “S3”. S3 stands for “Simple Storage Service”. It’s a cheap and fast way to store files and, in our case, to host static websites!

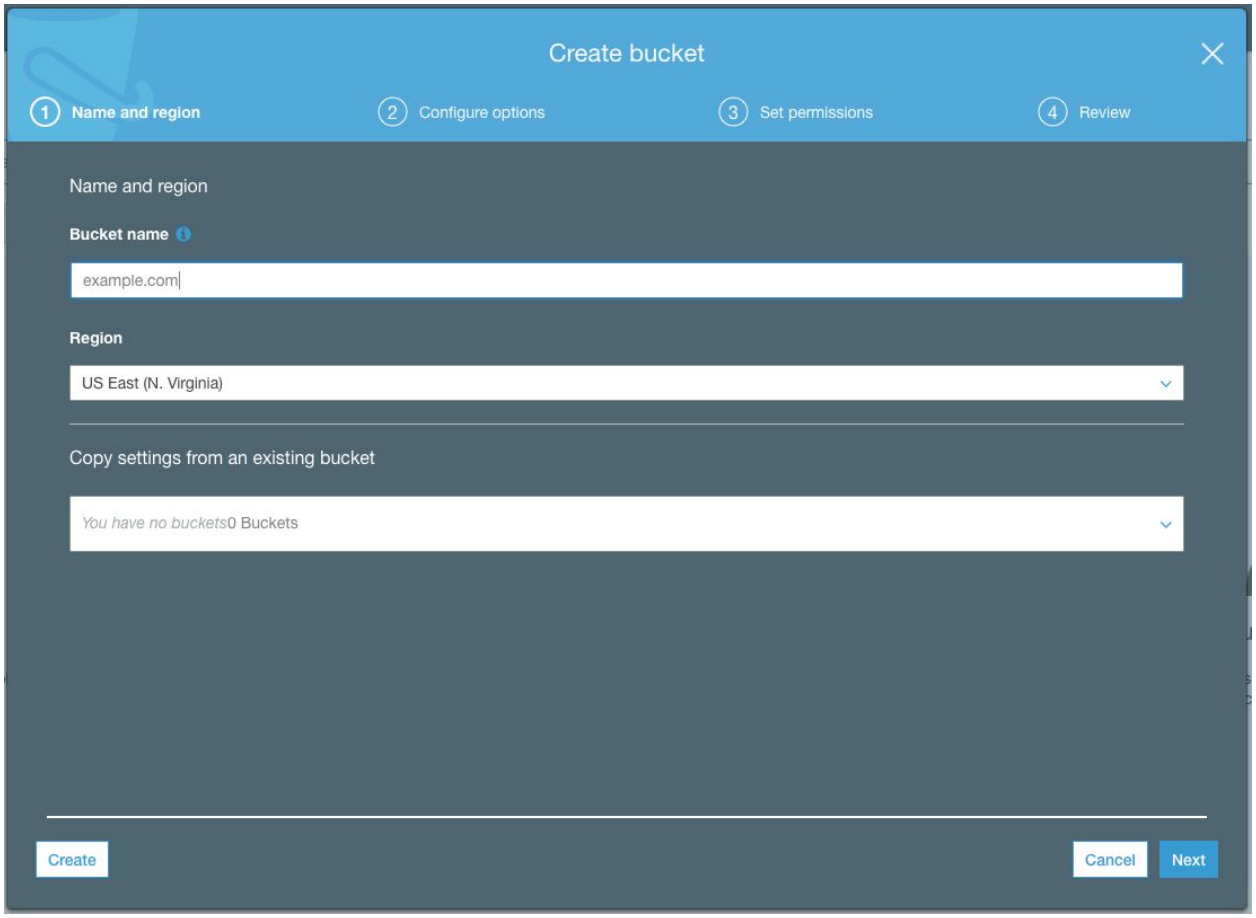


2) We will be creating two S3 buckets with the names:
example.com
www.example.com
IMPORTANT anytime you see this italicized, red text, replace it with the domain you purchased before.

3) To create a new bucket, click the **Create Bucket** button

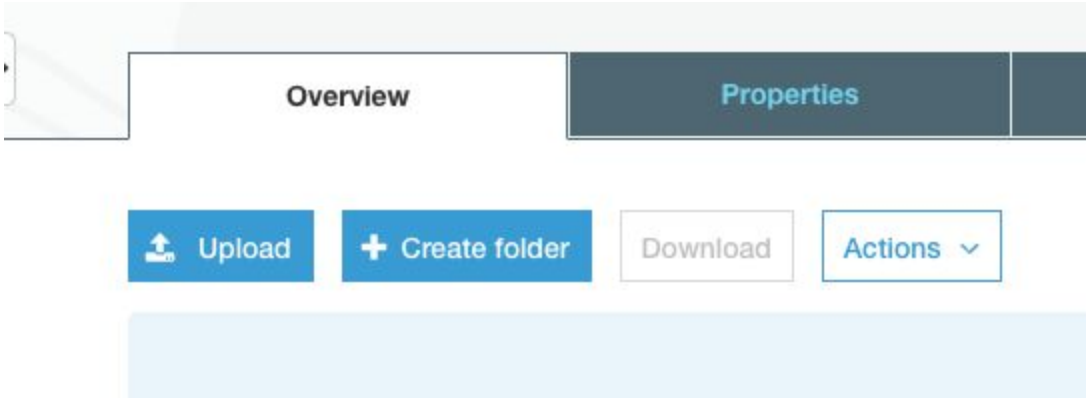


4) Give your new bucket the name ***example.com***
You can select any region you want, US East (N. Virginia) for example. It won't really affect your site as long as you select a region close to you.

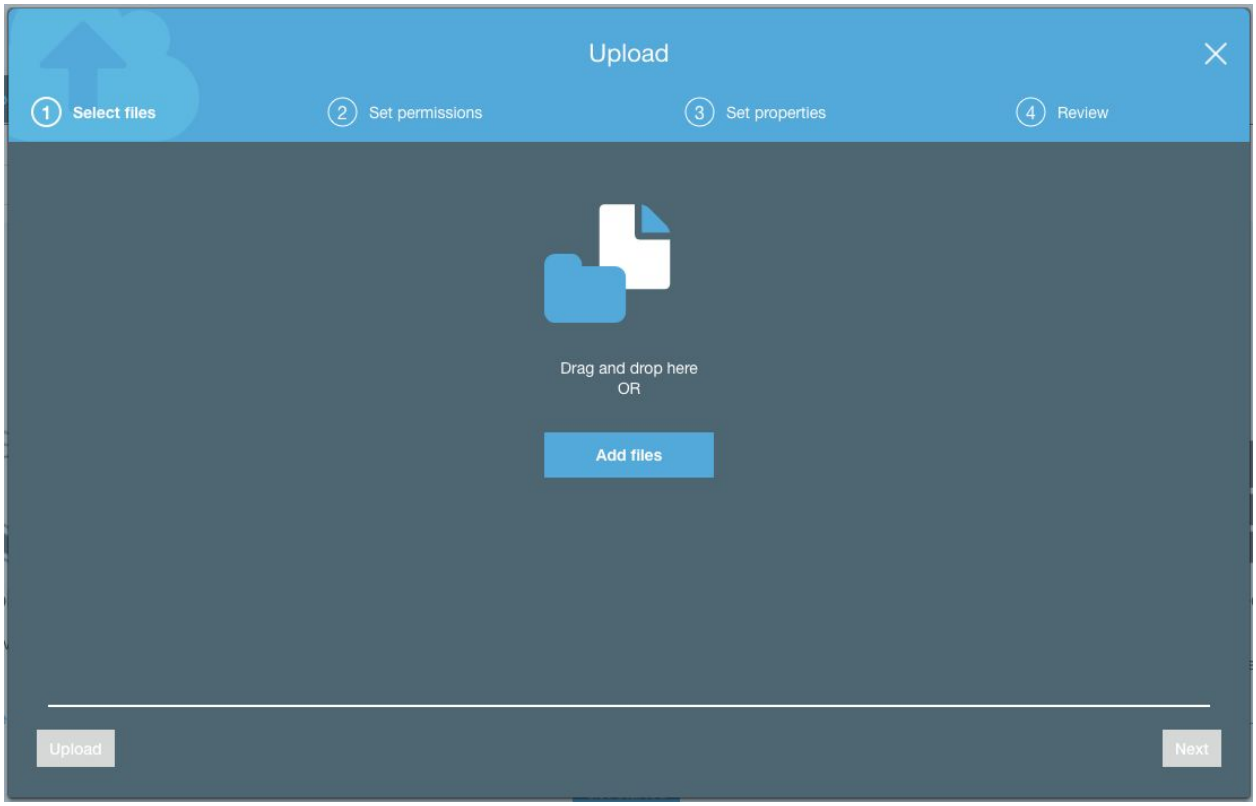


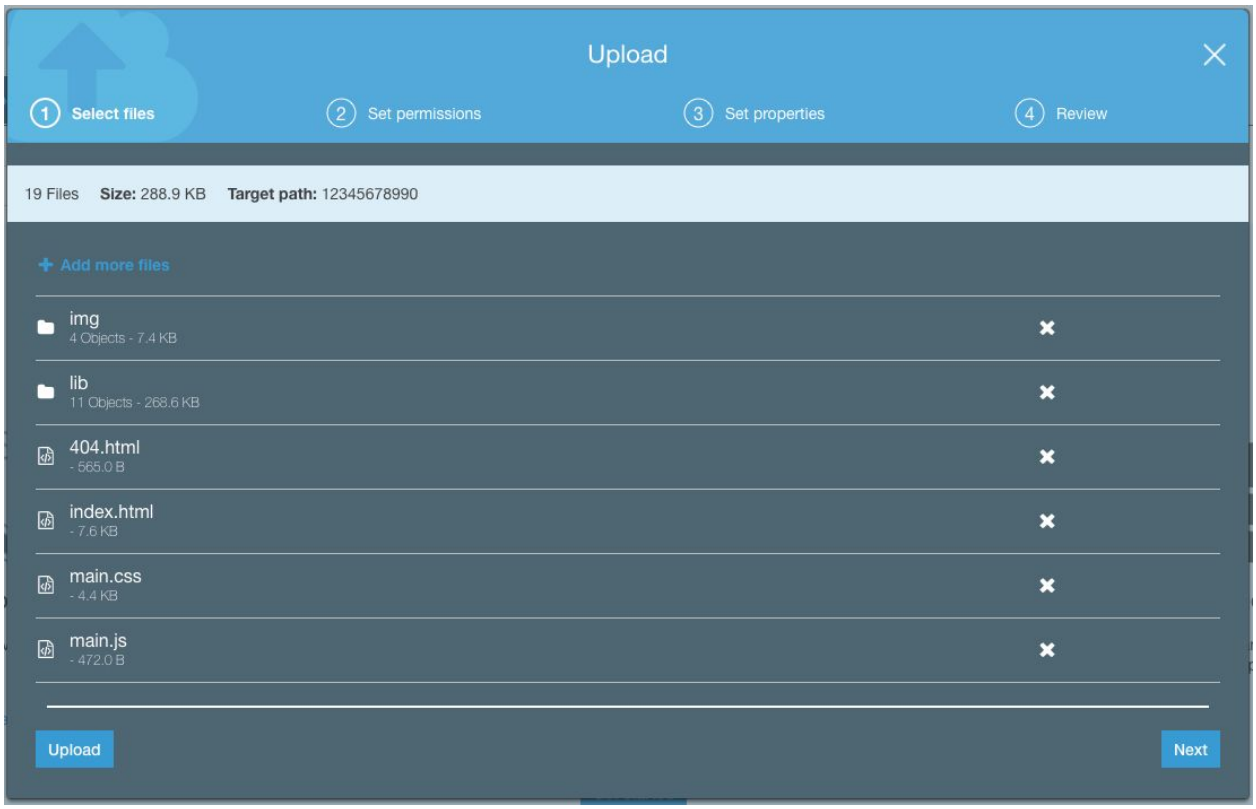
5) Move past the configure options and set permissions, leaving the defaults. Create your new bucket. Congrats! You created your first S3 bucket!! Now repeat steps 3 to 5 but with a new bucket name ***www.example.com***

6) Click your ***example.com*** (no www) bucket and the **Upload** button to begin uploading files.

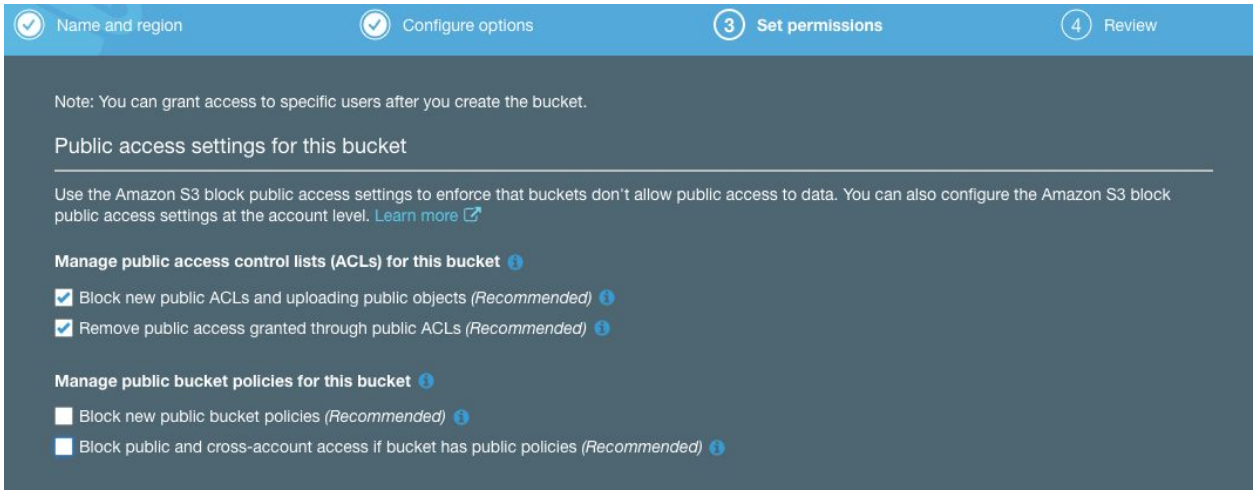


7) Upload everything in the **upload-stuff-in-here** folder. You must drag and drop your files and folders into the UI. Otherwise, S3 doesn't let you upload folders for some odd reason.

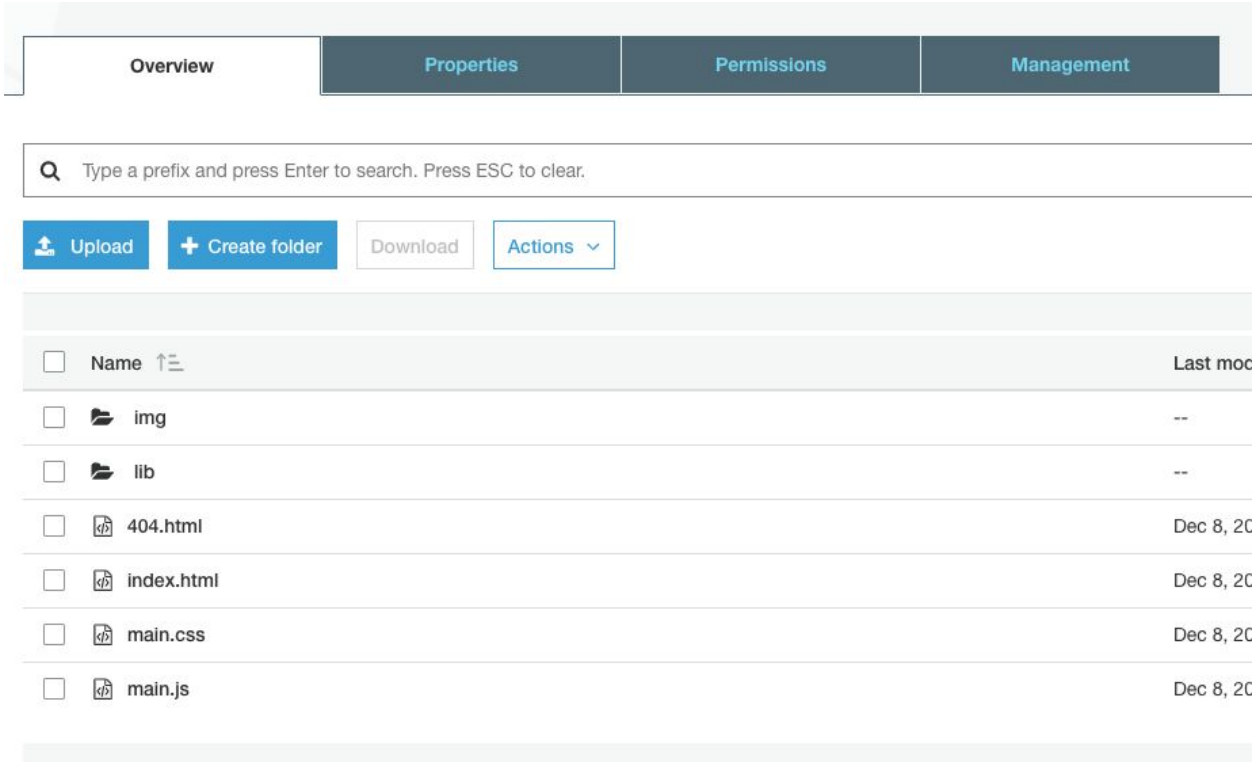




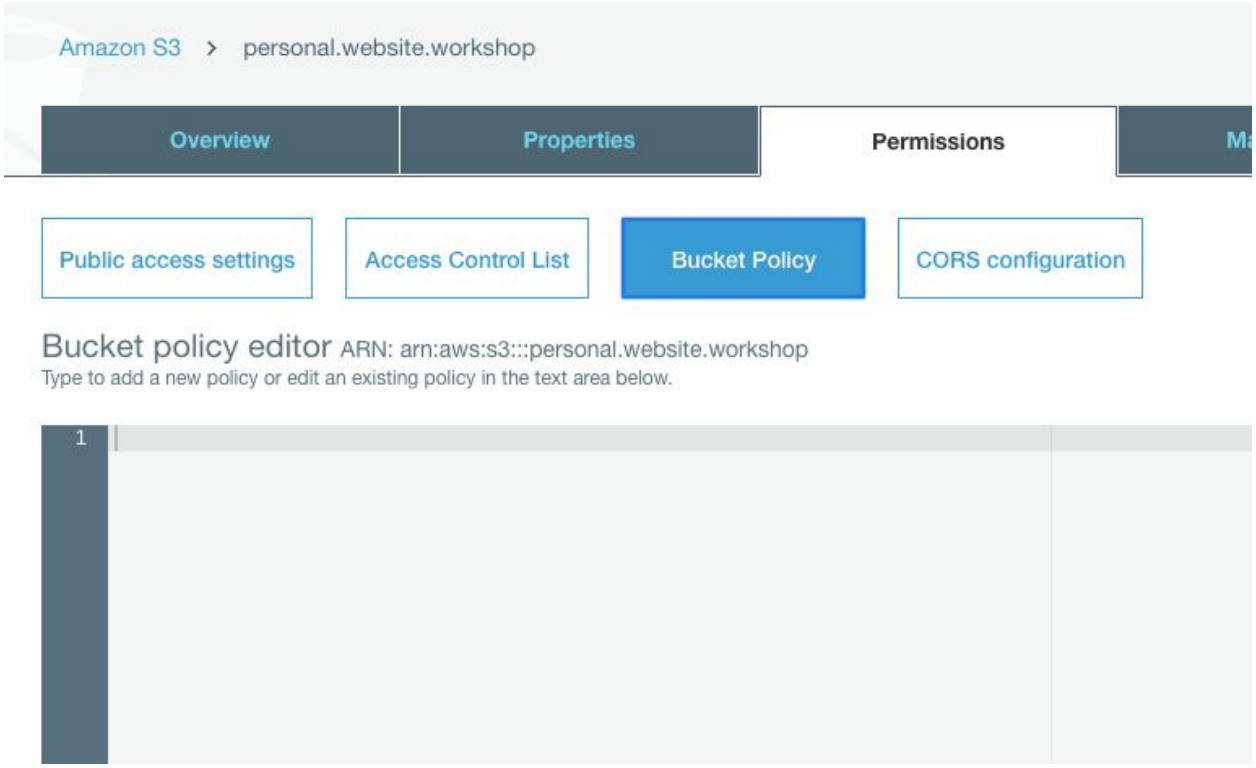
8) Leave the defaults for **Configure options**.
On **Set permissions**, under **Manage public bucket policies for this bucket**,
- uncheck both options
  - Block new public bucket policies (Recommended)
  - Block public and cross-account access if bucket has public policies (Recommended)
This will allow us to configure the public read settings of the bucket later.



9) Click past review and upload your files! Your website files are now uploaded!



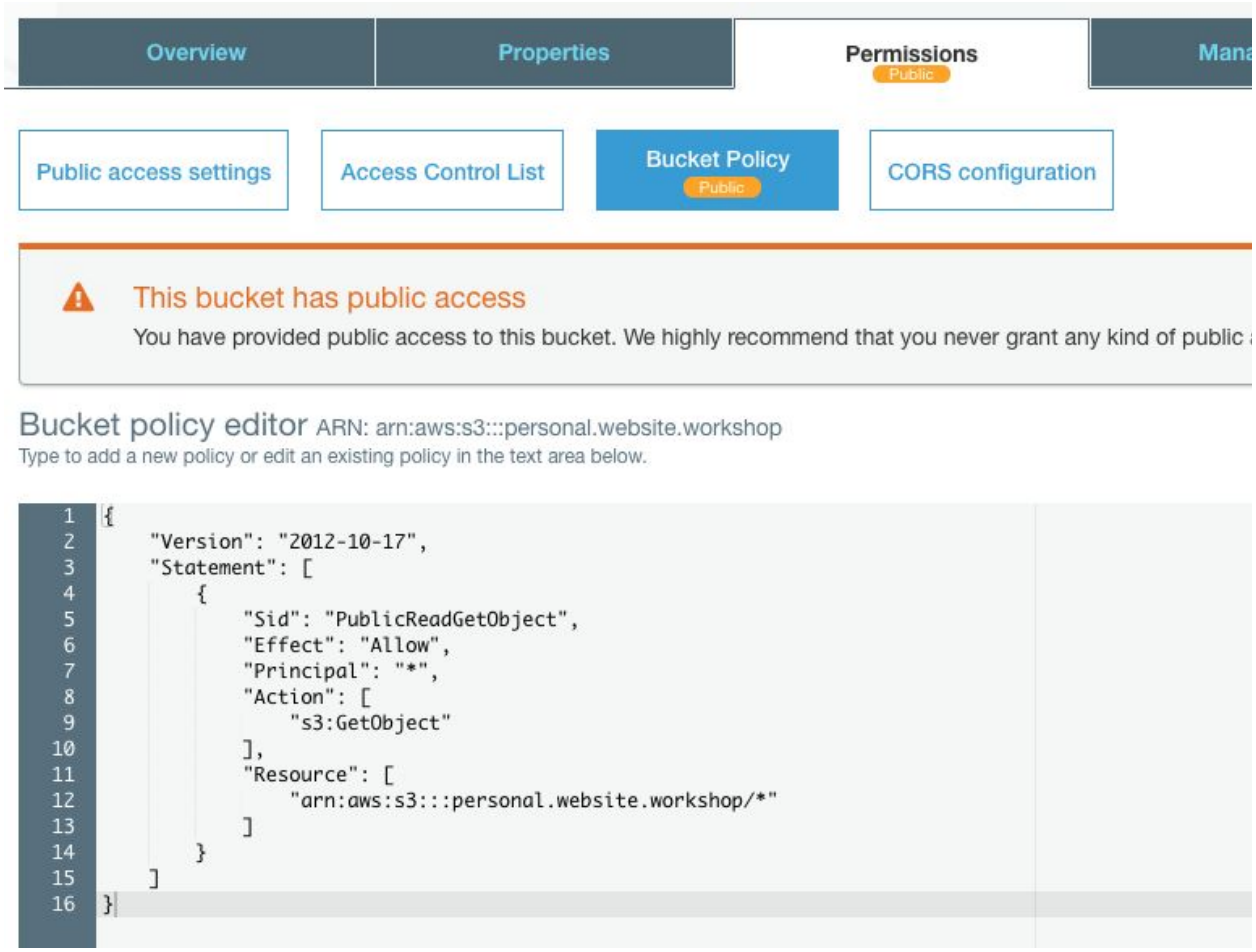
10) Now click **Permission** then **Bucket Policy**



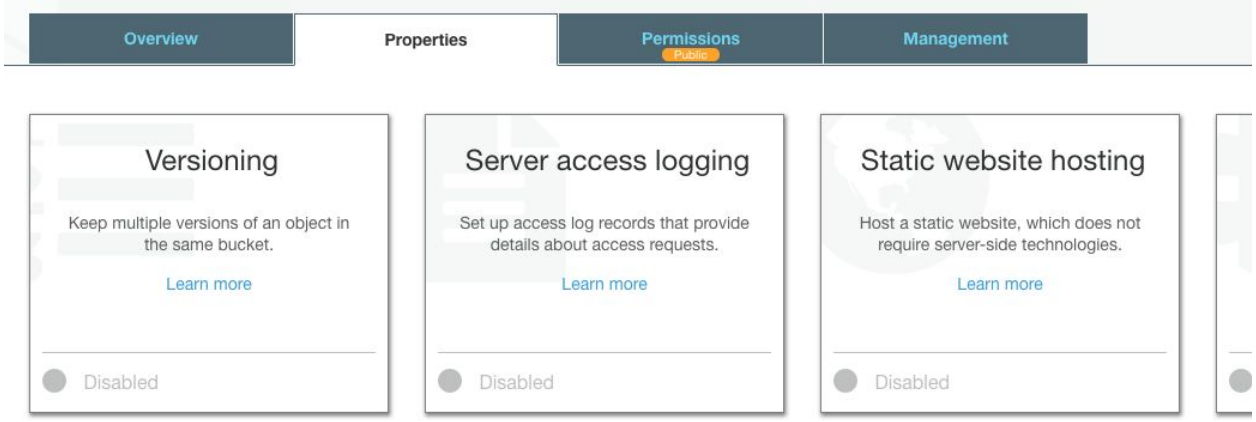
11) Copy and paste this text into your bucket to make it publicly readable. IMPORTANT make sure to replace ***example.com*** with your actual domain.

```
{
  "Version":"2012-10-17",
  "Statement":[{
          "Sid":"PublicReadGetObject",
       "Effect":"Allow",
            "Principal": "*",
     "Action":["s3:GetObject"],
     "Resource":["arn:aws:s3:::example.com/*"
     ]
    }
  ]
}
```

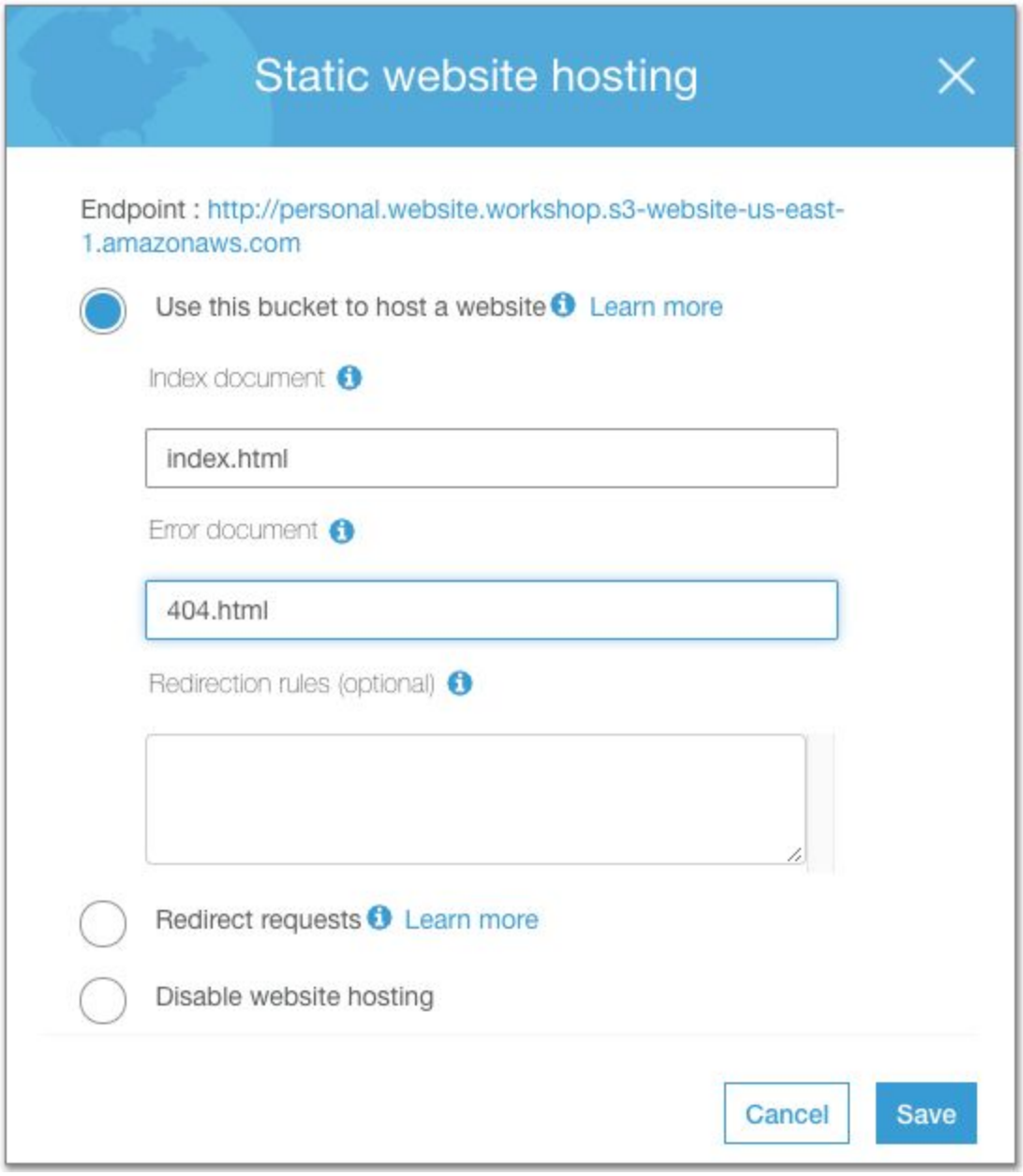
12) Your bucket should now be public.



13) Now we want to configure your buckets for website hosting. Go to “Properties” and select “Static Website Hosting”



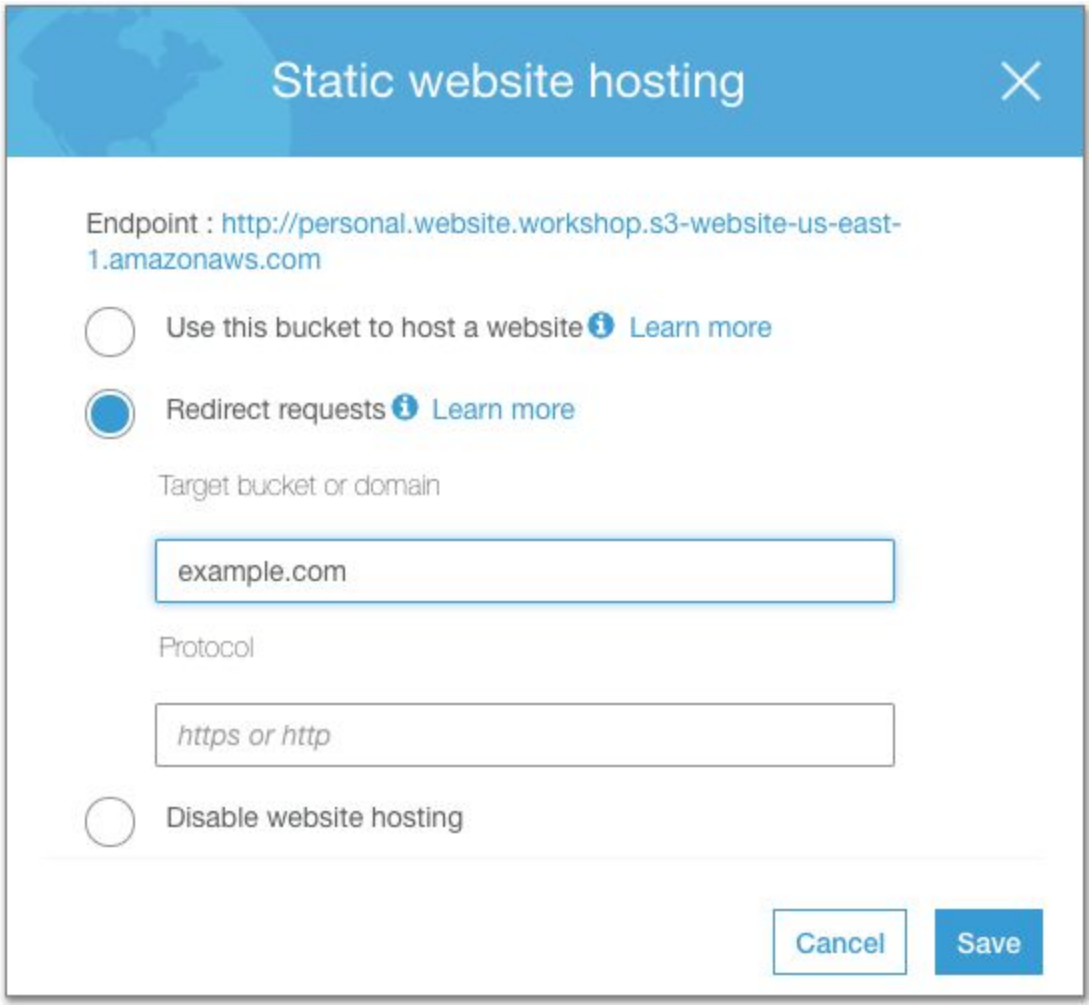
14) Select **Use this bucket to host a website**. In the **Index document** input, type “index.html”. In the **Error document** input, type “404.html”. This way, index.html is the default file to load, and 404.html is the file to load if there is a resource is not found. Save these settings.



15) You should now be able to see your website with the **Endpoint URL** provided! For example, http://personal.website.workshop.s3-website-us-east-1.amazonaws.com/. Hold on to this **Endpoint URL** because we will need it later.
I’ve also shortened it to http://tiny.cc/personal-site-workshop
(You could of course, just shorten your bucket URL and skip the domain name stuff altogether)

16) Now we want to have the ***www.example.com*** bucket redirect to the ***example.com*** bucket. This way, if we want to make changes to our site, we only have to update ***example.com***, not both buckets.

17) Now go back to your S3 dashboard and click on your ***www.example.com*** (with www) bucket. Go to **Properties** and **Static Website Hosting**, but this time click **Redirect Requests**. Type in "***example.com***" as the target bucket and click **Save**.

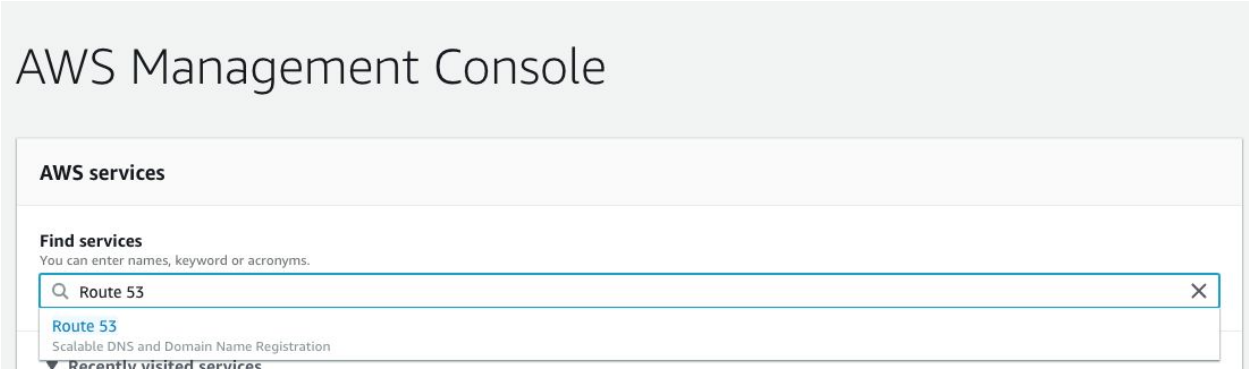


18) Now either bucket's **Endpoint URL** will load your site! Your ***www.example.com*** bucket simple redirects to your ***example.com*** bucket. But typing out these super long **Endpoint URLs** is...pretty lame. So let's connect your easier-to-remember domain to this S3 bucket!
Anytime you want to make an edit, simply upload the a new **index.html** file and you should see your changes in a few minutes!

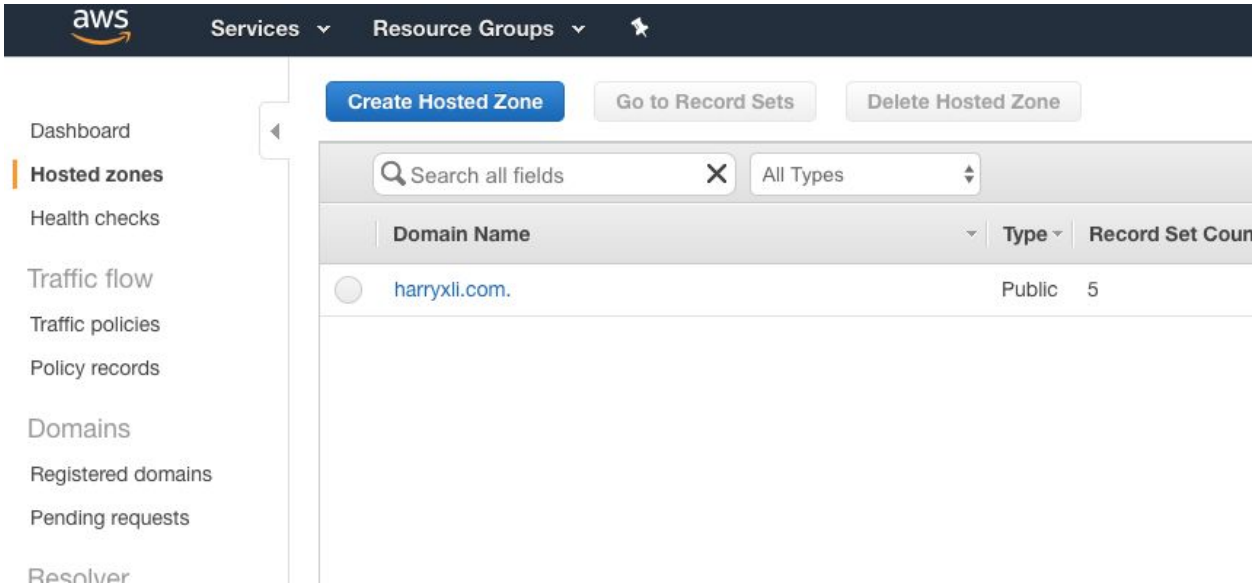
Part 4: Connecting Your Domain to Your S3 Buckets

https://docs.aws.amazon.com/AmazonS3/latest/dev/website-hosting-custom-domain-walkthrough.html#root-domain-walkthrough-add-arecord-to-hostedzone

1) Go back to your **Route 53** console. You can go to your **AWS Management Console** and search for “Route 53” or click this link https://console.aws.amazon.com/route53/



2) Click **Hosted zones** on the left side. Route 53 should have automatically created a hosted zone for you when you registered your domain.



3) Click on your hosted zone, click **Create a Record Set**, and a form should appear on the right.
You will create 2 records for your ***example.com*** and ***www.example.com*** buckets
Name
For ***example.com*** : leave it blank
Do this again for ***www.example.com*** : type “www”

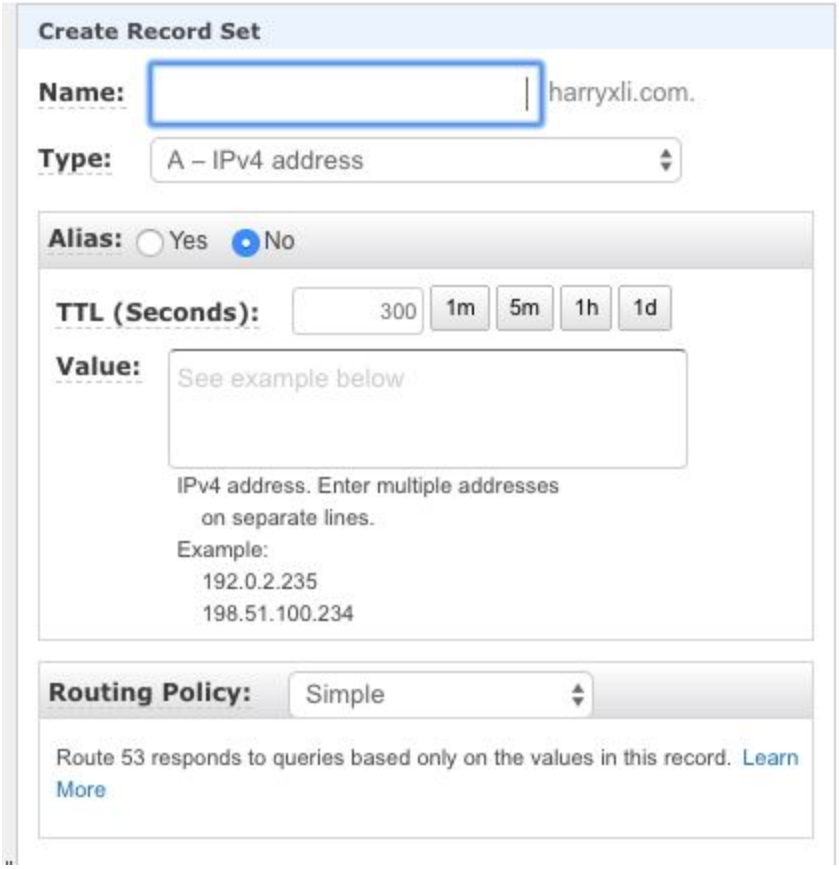
Type: For both: choose **A - IPv4 address** (it should be default)

Alias: For both: select **Yes**

Alias Target: For both: type the bucket’s *RESPECTIVE* **Endpoint URL**

Routing Policy: For both: leave the default value **Simple**

Evaluate Target Health: For both: leave the default value **No**



4) Create your record and make sure to repeat step 3 for your ***www.example.com*** bucket!

5) You should be done!!! Check out the URLs ***example.com*** and ***www.example.com*** and you should see your personal site! Congratulations!

Bonus: Encrypting your site with SSL and Amazon Cloudfront

https://medium.com/@itsmattburgess/hosting-a-https-website-using-aws-s3-and-cloudfront-ee6521df03b9