**Overview**

Use Hugo template to publish your website in minutes.

**Steps**

|  |
| --- |
| 1. **Install Hugo**   1.a. Install brew  /bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh)"  1.b. Install hugo  port install hugo   1. **Download hugo template**   First, start a new hugo project (replace site name with the name of the folder you want to create on your computer)  hugo new site SITENAME  Now you have an empty project. You can add any of the hugo themes (templates) from https://themes.gohugo.io by running the following command  git submodule add LINK\_TO\_GIT\_REPO.git themes/THEME\_NAME  Afterwards, copy and paste all the contents in the folder themes/THEME\_NAME to the root directory of the hugo project   1. **Edit hugo**   In order to edit the contents of the website, you go to the following folders:   * content: edit text and images on the website * static/images: replace media files * layout: edit HTML structure of the pages * config.toml: edit page names, order and other website settings  1. **Github**   Before we move to AWS Amplify, create a new git repository and push all the code you have written at this point to a public or private repository.   1. **AWS Amplify**   Next, go to Amazon Web Services (AWS) and create an account if you don’t have one. Go to AWS Amplify and choose “Host your web app”. Next connect the git repository that contains your code and the branch that you will be hosting the final code on. In the next page, under the build and test settings click edit. Add the following code under commands  - git submodule update --init --recursive --depth 1  - wget https:*//github.com/gohugoio/hugo/releases/download/v0.70.0/hugo\_extended\_0.70.0\_Linux-64bit.tar.gz*  - tar -xf hugo\_extended\_0.70.0\_Linux-64bit**.tar.gz** hugo  - mv hugo /usr/bin/hugo  - rm -rf hugo\_extended\_0.70.0\_Linux-64bit**.tar.gz**  - hugo |
| Hit save and deploy and wait for a few minutes to make sure the build was executed without errors.    Using AWS amplify will not cost money as long as you spend less than 1000 build min (average build takes less than 5 min, so you can make approximately 200 pushes to the git repository without paying). |

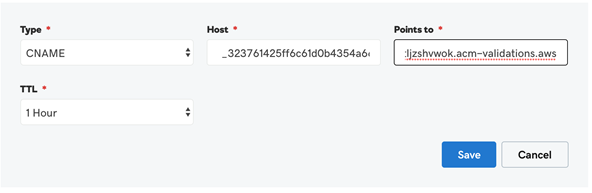
1. **Buy domain (godaddy)**

In order to direct the website to your domain, first make sure that you own a domain on GoDaddy. A basic package will suffice, you don’t need to pay for email service, storage, or website builder. You can use a different domain provider if you already have one. In the rest of this article, I will go over the steps to set up the domain on GoDaddy.

1. **Link your domain to the website**

In the navigation pane, choose App Settings, Domain management. On the Domain management page, choose Add domain. For Domain, enter the name of your root domain, and then choose Configure domain.

Go to GoDaddy and log in to your account. In your list of domains, find the domain to add and choose DNS. GoDaddy displays a list of records for your domain. You need to add two new CNAME records.Create the first CNAME record to point your subdomains to the Amplify domain. Then, create another record to point to AWS Certificate Manager (ACM) validation server. Bellow is a sample of how ACM validation record looks like



Finally Scroll down to the bottom of the DNS Management page to find the Forwarding box For Forward to, choose http://, and then enter the name of your subdomain to forward to (for example, www.example.com). For Forward Type, choose Temporary (302). For Settings, choose Forward only.

1. **Update git repository to update website**

Update content or layout and push to the same branch that AWS amplify is connected to. Wait 5 min for updates to go live.