Using GitHub to Manage or Host an MIT Website

Using GitHub to Manage or Host an MIT Website

There are two common use cases:

1. You can use the public GitHub Pages (github.io) as a hosting service, and have DNS records created on MIT's side to point to it, e.g. `alpha.mit.edu` will display the content hosted on `bravo.github.io`.
2. You can use the internal MIT instance of GitHub (github.mit.edu) to store your static site files, make & track changes to them, then check out these files in an AFS locker, which will then be hosted from `web.mit.edu` (e.g. `web.mit.edu/alpha`)

Example URLs and Usernames

In this article, many example URLs and usernames are needed to explain functionality. For the purposes of clarity and internal consistency, the following are used:

Code	Usage	Examples
Alpha	Name of Project/Website/Locker	Alpha Group, Alpha Project, `alpha.mit.edu`, `web.mit.edu/alpha`, `/mit/alpha`
Bravo	Username on Public GitHub	`github.com/bravo`, `bravo.github.io`
Charlie	MIT Username and Associated AFS Homedir Locker	`charlie@mit.edu`, `/mit/charlie`
Delta	A Subdirectory	`bravo.github.io/delta`, `web.mit.edu/charlie/delta`

Option One: `github.io`

Having DNS records created for your site requires a **Cost Object**. While there is no charge for this process, it is intended for official DLC purposes, and the Cost Object helps assign long-term ownership for the record.

Create a GitHub Pages Repo

Make a GitHub Pages repository, see the instructions here.

Point an MIT URL to the New Site

Once the site is ready, IS&T can make DNS records to point to it. From Connecting an MIT URL to an Outsourced Website:

> *You should establish a bravo.github.io hostname for the entire site; we will not be able to work with a requested path that is bravo.github.io/delta.*
>
> *Note also that we do not provide Github Pages functionality on our internal github (github.mit.edu) - you must use the externally hosted github if you wish to host web content there.*
>
> *We will provide a single CNAME record:*
>
> ```
> CNAME <alpha.mit.edu> bravo.github.io
> ```

Option Two: `github.mit.edu`, AFS, `web.mit.edu`

Initial Set Up

Github Repository

Create a repo on github.mit.edu to manage your site files. If you are a staff, faculty member or student and haven't already signed up, just go to github.mit.edu to get started. See this page for more details.

AFS

Request or Use Existing Locker

Pick a location to store your site files in AFS, either:

1. Request an organizational locker
2. Use a subdirectory of your own homedir or other existing locker, e.g. `/mit/charlie/delta`. Note that this means the URL for your site will be: `web.mit.edu/charlie/delta/`

Clone the Repository

Clone your repo into your **EMPTY** locker or subdirectory via HTTPS:

```
git clone https://github.mit.edu/charlie/website-repo-name.git .
```

The trailing period ("`.`") is critical. Without it, `git` will make a new directory to store these files.

Add the Directory to "Safe Directories," if Working in a Shared Environment

As of v. 2.35.3, Git will complain if you are not the owner of the working directory.

This a security measure intended to prevent compromises of your code, but if you are planning to have multiple contributors working in this AFS locker, you will need to add an exception.

You can do so by running the following in the root directory of your local repo:

```
git config --global --add safe.directory '*'
```

Make Directory World-Viewable

Set the permissions in AFS

```
fs sa /path/to/site/directory system:anyuser rl
```

If you have an organizational locker, the path would look something like: `/mit/alpha`, or explicitly `/afs/athena.mit.edu/org/a/alpha`

Wait 15 minutes, and check your site, e.g.: `web.mit.edu/alpha`

To set permissions for this directory *and all subdirectories*, use `fsr` instead of `fs`. You will need to have access to `/mit/consult` do so, either by running `add consult` or by manually appending it to your `PATH` variable.

For a more detailed look at how this works, see: How do permissions work in AFS?

For details on a more sophisticated permissions configuration, e.g. restricting access by certificate, see: Web Publishing - Access privileges on web.mit.edu

Maintaining the Site

Ideally this repo would be cloned on a personal machine(s) to make changes, which would then be tested, commited and then pushed back into `github.mit.edu`.

Once they are ready for public consumption, they can be pulled down into AFS.

1. Navigate to the directory in AFS
2. Pull the latest data from github.mit.edu, in this case from the `master` branch:

```
git pull origin master
```

3. Clear your cache and refresh your browser, and you should now see the latest copy of the page(s).