GIT101

Website Building

Building the gitMechanics Website from Source

Arizona State University

Brian Chevalier

Updated: April 20, 2020

Contents

1 Introduction

In this set of notes we'll be looking at how the website is automatically built from the source code, and all the steps that must happen to make a change to the site.

There are quite a few pieces that go into building and hosting the website. The goal of this setup is to completely automate and whatever is uploaded automatically so quite changes can be easily made and the site is responsive and dynamic enough to understand what must be rebuilt.

So let's look at a quick overview of the overall process and then dive into each step.

1. A push/pull request is made to the master branch of repo
2. Travis-CI detects a change in the branch
    - Travis-CI starts a new instance of a Linux distribution specified in a configuration file
    - Travis-CI runs code against the files
    - If all code is run without error Travis-CI pushes changes to the gh-pages branch of the repo
3. Github sees a change to the gh-pages branch and uses Jekyll to generate the static HTML files and publishes them at gitMechanics.com

2 Travis-CI

Travis-CI is a *continuous integration* service. This type of service is intended to automatically test and deploy code. This means that code pushed into the repo will always be tested by the same series of tests each time and if it passes it will deploy the code. For our use we're mostly interested in using Travis-CI to automatically build all the LaTeX files from the repo to make sure they work without error, build all of the associated pdf files. If it fails the website will not be updated thus preventing bad code from being deployed.

3 Github Pages

Github will host a website for any user using a platform called Jekyll (discussed in detail later). This is completely free and can even have a custom domain name. I bought a custom domain from hover.com and configured the website to work with the custom domain in the github repository settings page. In the Github repo settings you can choose enter the custom domain name you buy, choose a theme, and the branch that Github will look at to build and publish the website. I have the branch set to gh-pages. This is a branch of the repo made specifically to host the production website files.

4 Snakemake

Snakemake is a workflow management system to create reproducible and scalable data analyses. Snakemake was inspired by Make, which essentially automates the building of output files from input files. Snakemake does the same thing but using a nicer syntax and uses Python (or R) for scripting.

What this means is Snakemake can be used to track input files (.tex and .py files) and automatically produce output files (.pdfs). This avoids the need to manually run LaTeX on each file. And if a dependency changes (style files) we can ensure all .tex files will still compile. Furthermore, using Python we can automate finding the files to build so adding additional Main.tex files does not require adding the file path to some static list. Snakemake will just automatically build all necessary files.

4.1 Snakefile

The Snakefile is the configuration file for Snakemake, found in the root directory of the gitMechanics repository. You can see the most up to date Snakefile here. The Snakefile has three key parts: The initilization code, the first pseudo-rule, and the actual rules. The initilization code is python code that is run to find all of the output files that must be generated. The first rule also known as a pseudo-rule simply takes all of the output file paths as input which triggers the other rules to be run. The actual rules specify file paths as inputs and outputs and also specifies the shell commands necessary to produce the output from the input files.

5 Miniconda

Miniconda is a very lightweight Anaconda distribution, which downloads and manages Python, R packages. Anaconda/Miniconda can both provide a package manager, called *conda*, that will download Tectonic. Conda will also download any other needed Python dependencies, for instance, matplotlib for generating plots to include in LaTeX documents.

6 Tectonic

Tectonic is a modernized, complete, self-contained TeX/LaTeX engine. When you run Tectonic on a .tex file, it will automatically download any necessary dependencies and support files to process your TeX. When Travis-CI runs Tectonic, it will automatically download all required files, and nothing more, which speeds up compile time. Furthermore, Tectonic stores these files in a cache directory. Travis can also cache this directory for usage between builds (cacheing these files reduces compile time from 12-15 minutes down to 4). Tectonic will also run pdfLaTeX and BibTeX as many times as necessary automatically.

You can use Tectonic and Snakemake when contributing to gitMechanics, but it is not necessary. As long as your .tex files compile Tectonic will happily compile your files serverside and deploy your files to gitMechanics.

7 Jekyll

Jekyll is a simple, blog-aware, static site generator.

7.1 The _config.yml file

This is the special file that Jekyll will look for to configure the website.

7.2 The _layouts folder

The layouts folder is a special folder used by Jekyll to define layouts that can be used on other pages throughout the site.

7.3 Gemfile

The Gemfile in the repo is only necessary for testing building the website without having to push to the repo and having github rebuild the website (which can be really time consuming, and clutter the website). Jekyll can be run locally on your machine to test building the website from source and viewed locally.