CREATE A STATIC WEBSITE USING JEKYLL

Deploying: Overview

In this course, you'll learn how to deploy a static site to the Internet.

What exactly do we mean by *deploy*, or *deploying*?

Deploying means making content or software accessible and available for use.

The website that you'll deploy in this course will be public, or "live", on the Internet. It will be accessible to anyone (with open Internet access), at anytime, anywhere in the world.

By the end of this course, you will have:

1. Generated a static site
2. Deployed it to the Internet
3. Given the website a custom domain name

Let's begin!

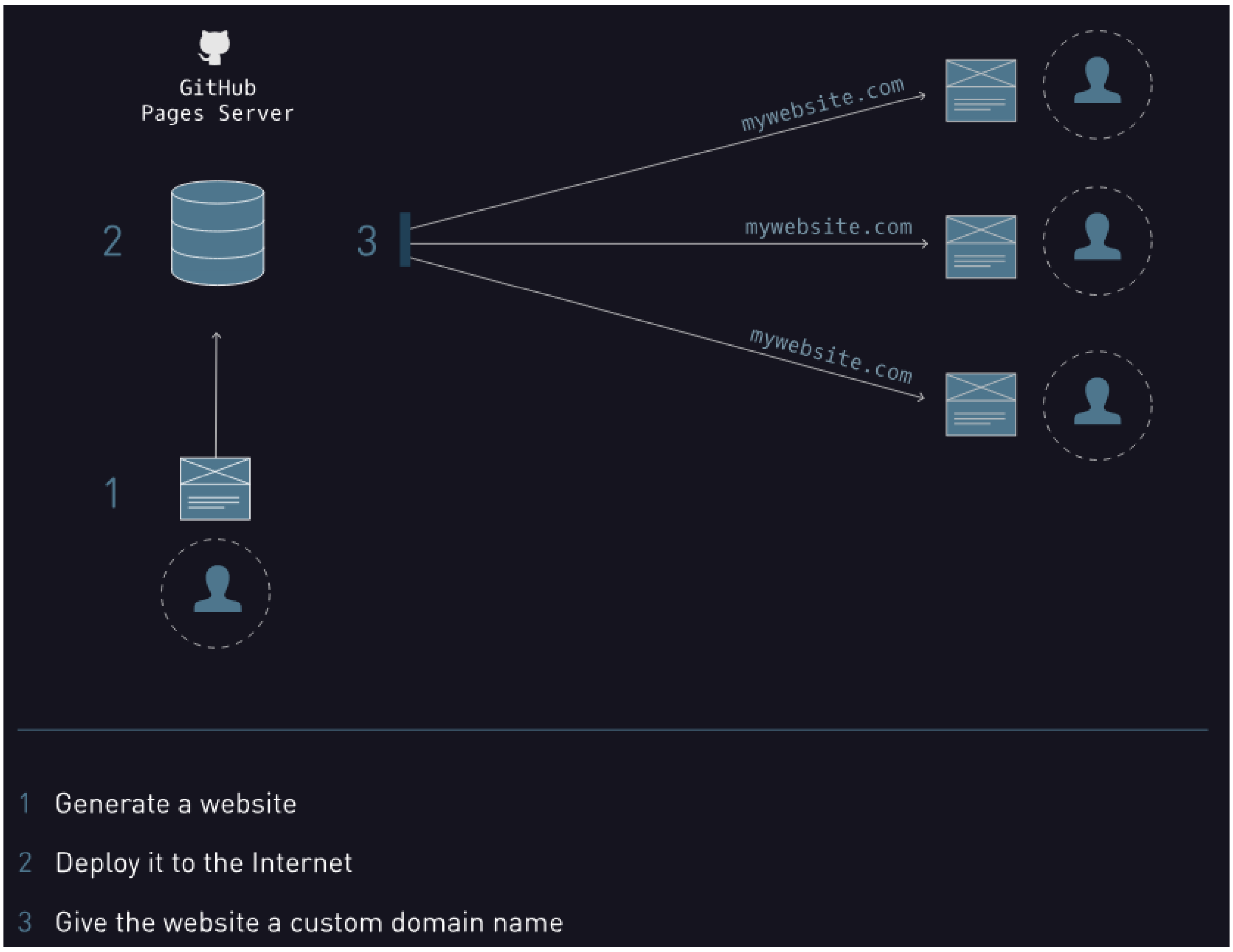
Instructions

The diagram to the right illustrates the three main steps we'll cover in this course — (1) generating the site, (2) deploying it to the public Internet, and (3) assigning it a custom domain name. We'll walk through each of these steps in the order they appear on the diagram.

Note: We'll make extensive use of the command line and Git in this course. If you need to review, check out the corresponding Codecademy courses:

Learn the Command Line

Learn Git



Jekyll: Overview

The focus of this course will be on the process of deploying a website, not on actually creating a website.

We'll use a popular tool known as *Jekyll* to quickly generate a website. This will help keep the focus on the deployment process and quickly provide you with content to deploy, rather than focusing solely on website creation.

Jekyll is a simple static site generator. Using Jekyll is a very common way of generating a "ready-to-publish static website" within seconds. You can learn more about Jekyll here.

An important note:

The reason this course uses Jekyll is so that we can generate the static website quickly and focus on deploying it. However, we understand that you may not want to use the Jekyll-generated content.

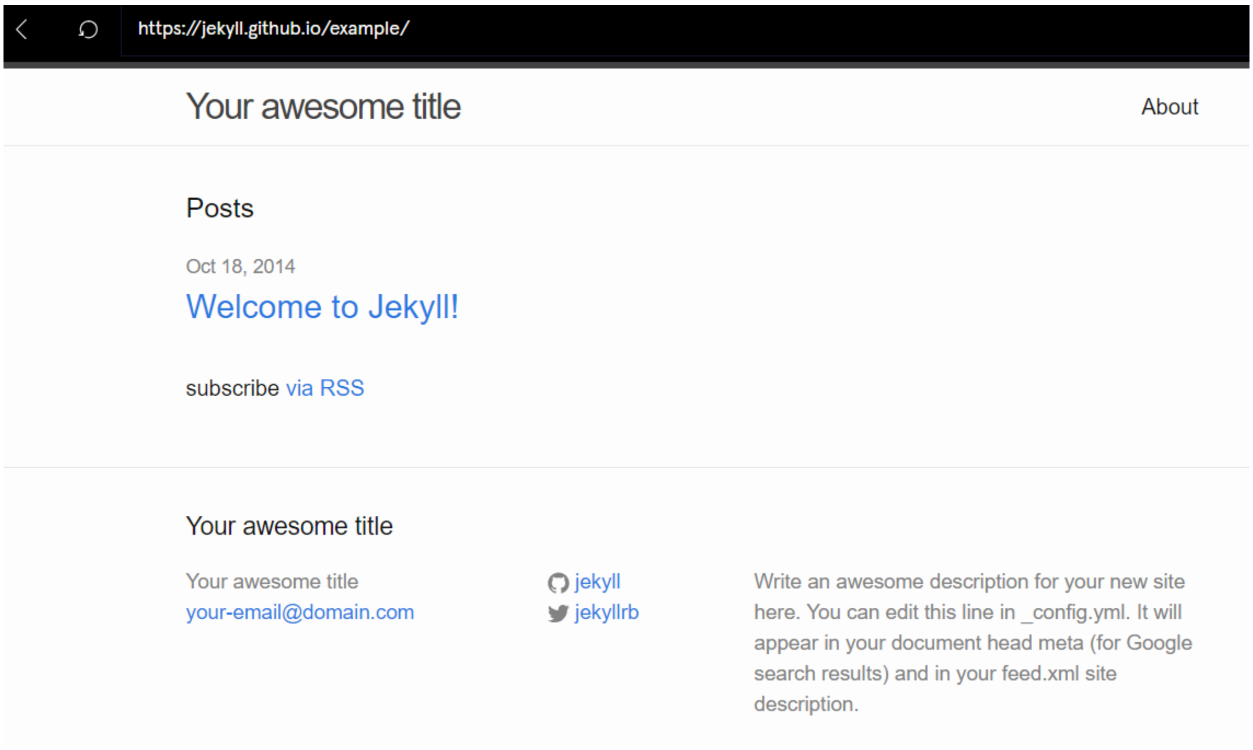
In that case, it is possible to follow all of the steps outlined in this course with your *own* content — just make sure that your HTML is inside of a file called **index.html**. As you'll see, even Jekyll uses a file called **index.html**.

If instead you'd like to learn about creating static sites (starting from scratch), check out our Make a Website course.

Instructions

Take a look at the sample site in the browser to the right.

By the end of this unit, you'll have your own Jekyll-generated website ready to deploy.


https://jekyll.github.io/example/

Your awesome title

About

Posts

Oct 18, 2014

Welcome to Jekyll!

subscribe via RSS

Your awesome title

Your awesome title
your-email@domain.com

jekyll
jekyllrb

Write an awesome description for your new site here. You can edit this line in _config.yml. It will appear in your document head meta (for Google search results) and in your feed.xml site description.

Installing Jekyll

Before we can generate a website, we must install Jekyll.

Jekyll is a Ruby gem (also known as a RubyGem) and can be installed from the command line. Don't worry, knowledge of Ruby is *not* required in order to complete this course.

Once Jekyll is installed, we can use it to generate your website.

Instructions

Checkpoint 1 Passed

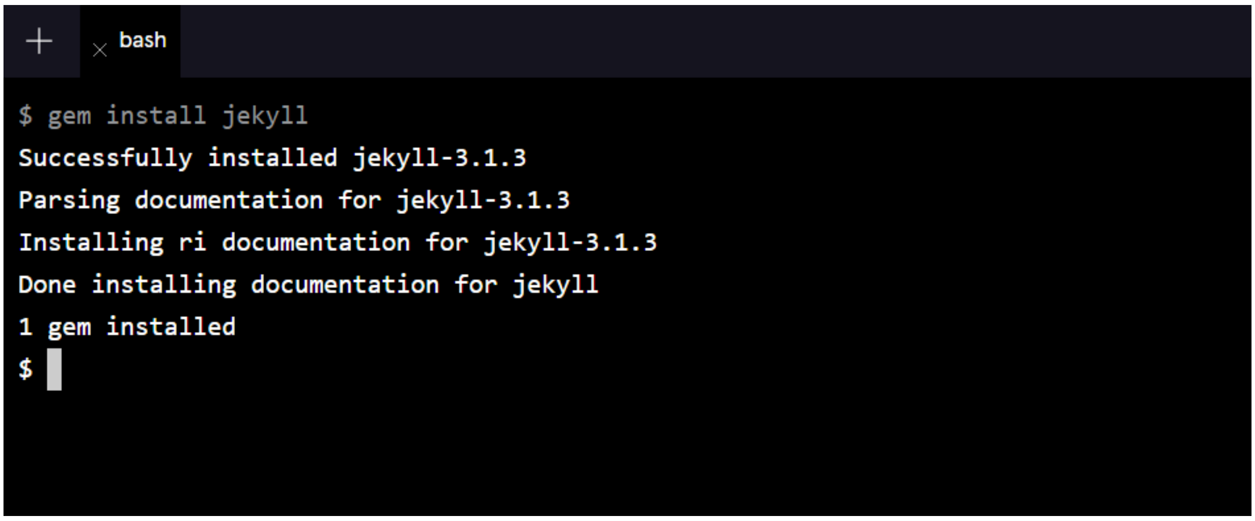
1.

Install Jekyll by typing the following command in the terminal:

```
gem install jekyll
```

Wait for the gem to finish installing before moving onto the next exercise. A successful install will display the following:

```
Successfully installed jekyll-3.1.3
Parsing documentation for jekyll-3.1.3
Installing ri documentation for jekyll-3.1.3
Done installing documentation for jekyll
1 gem installed
```



```
$ gem install jekyll
Successfully installed jekyll-3.1.3
Parsing documentation for jekyll-3.1.3
Installing ri documentation for jekyll-3.1.3
Done installing documentation for jekyll
1 gem installed
$
```

Generate a Static Site

Great! Now that Jekyll is installed, let's generate your website.

To do so, we'll use Jekyll's `new` command and specify a directory name. The directory will contain all of your site's default content that can be customized later.

For example, to generate a website in a directory called `my-portfolio-site`, we can type:

```
jekyll new my-portfolio-site
```

The command will create a directory called `my-portfolio-site` and fill it with Jekyll's site content.

Note: We'll use the directory name `my-portfolio-site` in examples throughout the course. Don't worry about the details of the directory structure just yet, we'll give an overview in the upcoming exercises.

Instructions

1.

In the terminal, use Jekyll's `new` command to generate your static website with a directory named `personal-website`.

2.

Use the `ls` command to verify that the directory was successfully generated.

Note: The first character of the `ls` command is a lowercase L, *not* the number 1.

Preview Your Site Locally

Want to see what your site currently looks like? You can use Jekyll to view your site *locally*.

On the web, a server hosts your site's files and makes your website available for everyone to see.

However, viewing a website locally means that you're viewing the site on your *own* computer (hence the term "locally" or "local"). The site is not, however, available on the public Internet. Instead, your computer is acting as the server that hosts your site.

You can view your site locally by using Jekyll's `serve` command, like so:

```
jekyll serve
```

This command starts a local server that will server the files to your computer. The `serve` command will also come in handy when you want to preview changes you make to your site.

By default, the address for the local server that *Jekyll's* `serve` command starts is `http://localhost:4000/`.

Instructions

1.

To view your site locally, you *must first navigate to your site's directory*, using the `cd` command.

Navigate to your site's directory using the `cd` command.
Hint

Your site's directory that you created in the previous exercise is `personal-website`.
Checkpoint 2 Passed

2.

Next, use the `serve` command to start a local server.

Then, navigate to http://localhost:4000 in the browser to view your site.

Jekyll's Directory Structure

The website that Jekyll generates differs from a website that you'd create on your own. It offers a standard directory structure, as well as components that help speed up development.

It's important to understand Jekyll's default directory structure and contents of your site:

1. **_config.yml** - This is a configuration file that contains many values that need to be edited only once. These values are used across your site, for example, your site's title, your e-mail, and more. Note that this is a **.yml** file, which you can learn more about here.
2. **_includes/** - This directory contains all the partials (code templates that keep you from repeating your code over and over) that your site uses to load common components, like the header and the footer.
3. **_posts/** - This directory is where blog posts are stored. New blog posts can be added and will be rendered with the site's styling, as long as the file name follows Jekyll's standard naming convention.
4. **_layouts/** - This directory contains templates that are used to style certain types of posts within the site. For example, new blog posts will use the HTML layout defined in `post.html`.

You can learn more about the Jekyll directory structure here.

Instructions

Use the file navigator to the right to browse Jekyll's default directory structure and contents.

Note: Don’t edit any of the content in the files at this moment. If you make edits and accidentally make a mistake, you run the risk of an unsuccessful deploy.

Review

Great work! Let’s review what you accomplished in this unit:

1. Installed Jekyll
2. Used Jekyll to generate a static site
3. Started a local server to view your site

Why was this important? The course will focus on the deployment process, but first we needed content to deploy. Jekyll quickly provided us with that content and now we can focus on the deployment process.

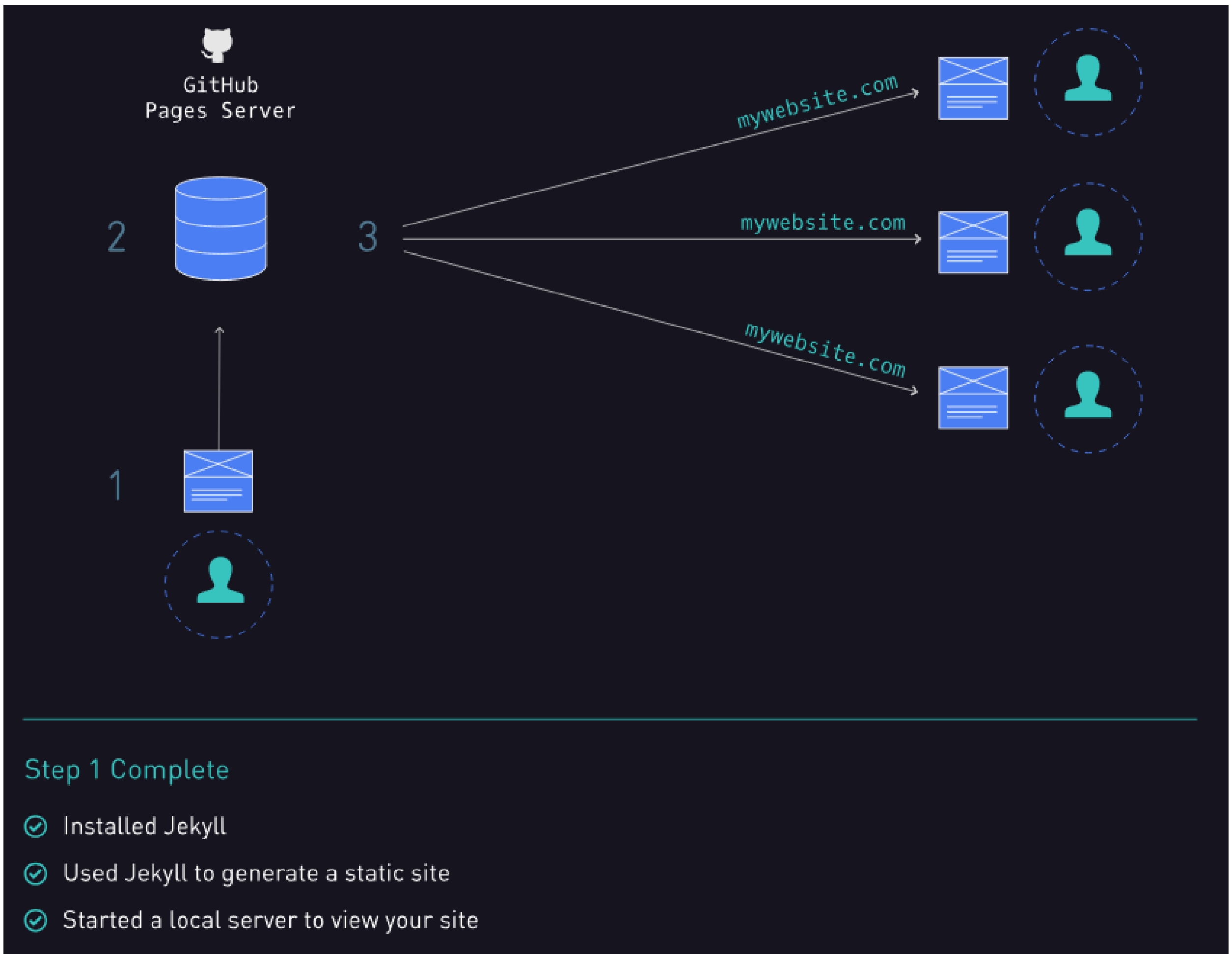
Instructions

Take a look at the diagram to the right. In this unit, you successfully accomplished the first step: using Jekyll to generate a website.

We also provided you with handy Jekyll commands used to preview your site along with an overview of how your site is structured.

Unfortunately, individuals on the Internet around the world still do not yet have access to your content.

In the next unit, we’ll move to the second step: taking the website you generated and publishing it to the rest of the world!



Step 1 Complete

- ✓ Installed Jekyll
- ✓ Used Jekyll to generate a static site
- ✓ Started a local server to view your site