

Building Websites on the Cheap With Hugo

James Henstridge <james@jamesh.id.au>
Perth Linux Users Group, June 2022

Introduction

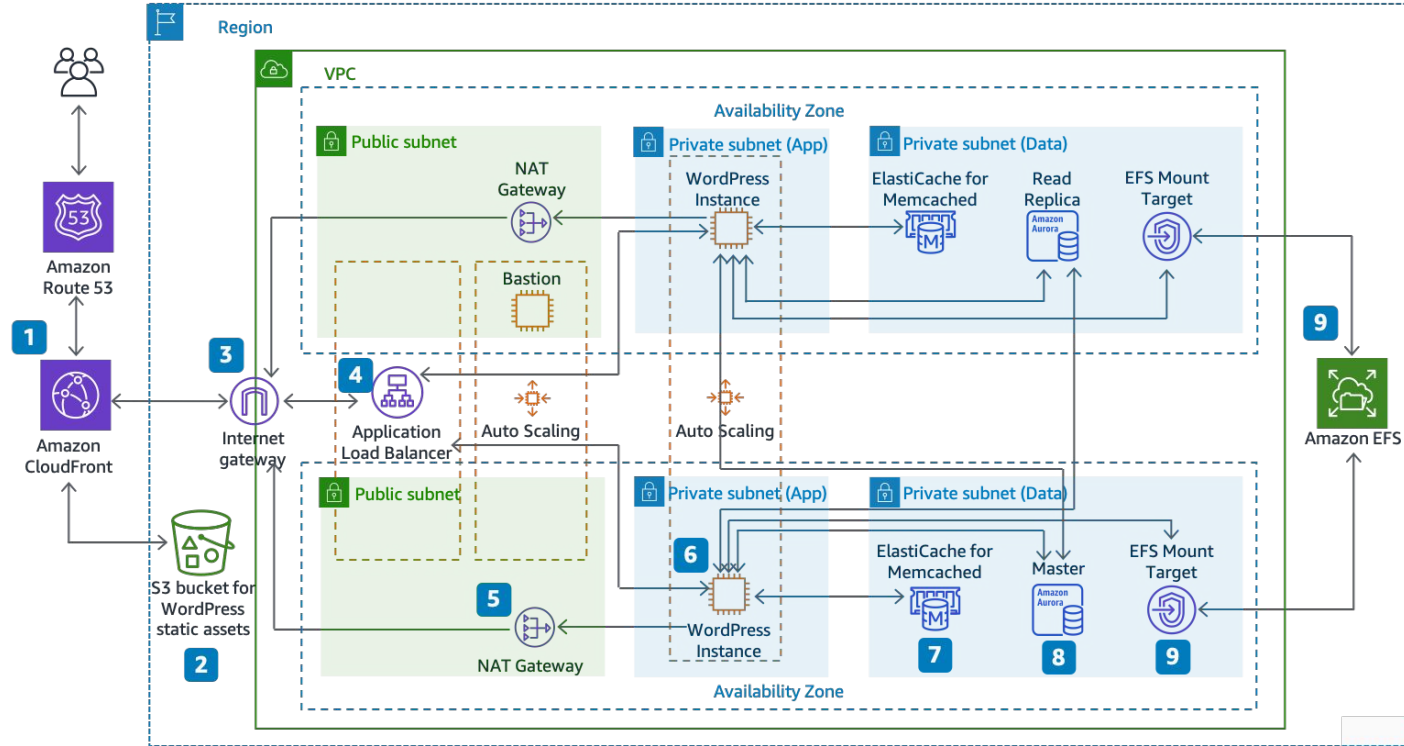
- Why not use something like WordPress?
- Benefits of static websites
- How to build a Hugo website
- How to host the site cheaply

Why not use WordPress?

- WordPress is great, but does a lot more than I need.
- As an Internet facing service, important to keep up to date.
 - Both the core and plugins
- Has an application specific backup/restore system
- More moving parts to scale up if I get popular.

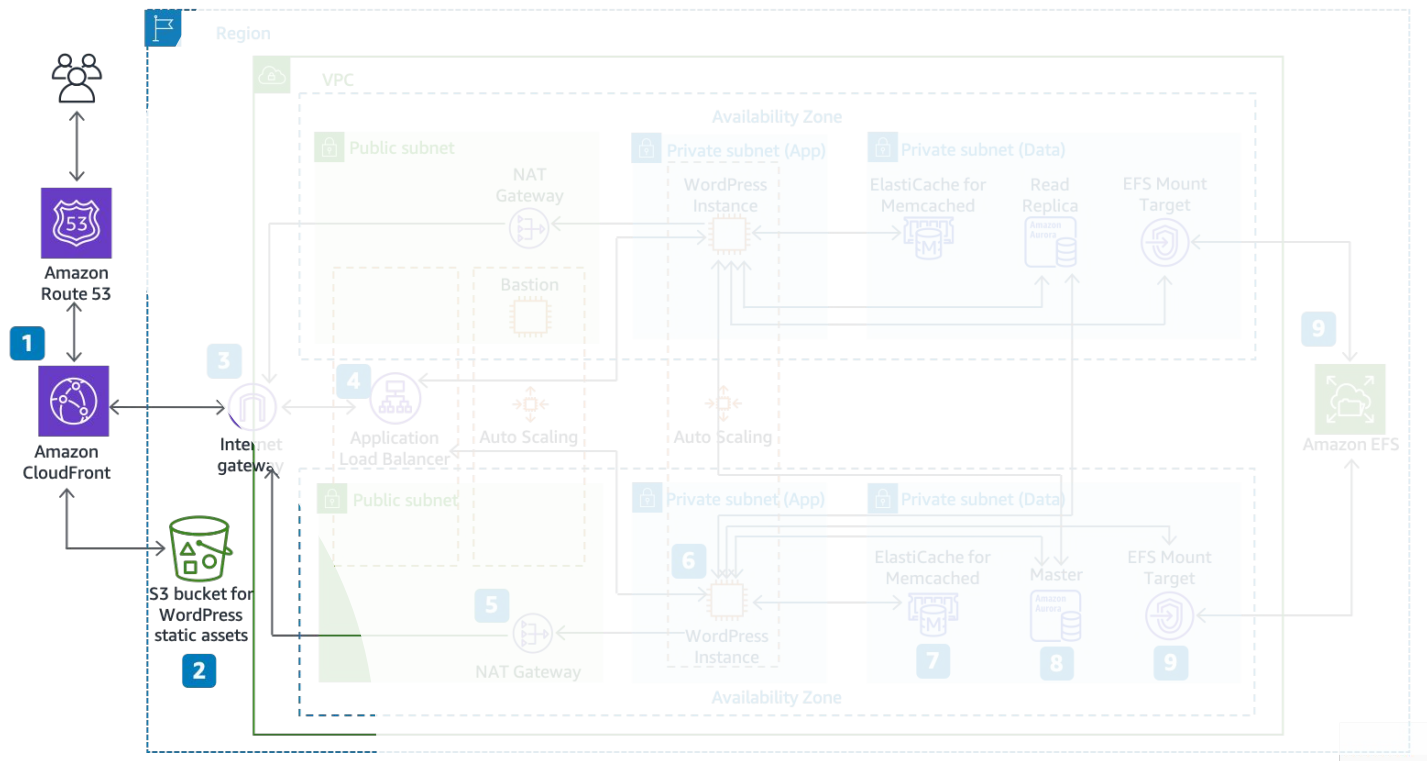
Vuln ID ❸	Summary ❹	CVSS Severity ❺
CVE-2011-1762	<p>A flaw exists in Wordpress related to the 'wp-admin/press-this.php' script improperly checking user permissions when publishing posts. This may allow a user with 'Contributor-level' privileges to post as if they had 'publish_posts' permission.</p> <p>Published: April 18, 2022; 1:15:11 PM -0400</p>	<p>V3.1: 6.5 MEDIUM</p> <p>V2.0: 4.0 MEDIUM</p>
CVE-2022-21664	<p>WordPress is a free and open-source content management system written in PHP and paired with a MariaDB database. Due to lack of proper sanitization in one of the classes, there's potential for unintended SQL queries to be executed. This has been patched in WordPress version 5.8.3. Older affected versions are also fixed via security release, that go back till 4.1.34. We strongly recommend that you keep auto-updates enabled. There are no known workarounds for this issue.</p> <p>Published: January 06, 2022; 6:15:08 PM -0500</p>	<p>V3.1: 8.8 HIGH</p> <p>V2.0: 6.5 MEDIUM</p>
CVE-2022-21663	<p>WordPress is a free and open-source content management system written in PHP and paired with a MariaDB database. On a multisite, users with Super Admin role can bypass explicit/additional hardening under certain conditions through object injection. This has been patched in WordPress version 5.8.3. Older affected versions are also fixed via security release, that go back till 3.7.37. We strongly recommend that you keep auto-updates enabled. There are no known workarounds for this issue.</p> <p>Published: January 06, 2022; 6:15:08 PM -0500</p>	<p>V3.1: 7.2 HIGH</p> <p>V2.0: 6.5 MEDIUM</p>
CVE-2022-21662	<p>WordPress is a free and open-source content management system written in PHP and paired with a MariaDB database. Low-privileged authenticated users (like author) in WordPress core are able to execute JavaScript/perform stored XSS attack, which can affect high-privileged users. This has been patched in WordPress version 5.8.3. Older affected versions are also fixed via security release, that go back till 3.7.37. We strongly recommend that you keep auto-updates enabled. There are no known workarounds for this issue.</p> <p>Published: January 06, 2022; 6:15:08 PM -0500</p>	<p>V3.1: 5.4 MEDIUM</p> <p>V2.0: 3.5 LOW</p>
CVE-2022-21661	<p>WordPress is a free and open-source content management system written in PHP and paired with a MariaDB database. Due to improper sanitization in WP_Query, there can be cases where SQL injection is possible through plugins or themes that use it in a certain way. This has been patched in WordPress version 5.8.3. Older affected versions are also fixed via security release, that go back till 3.7.37. We strongly recommend that you keep auto-updates enabled. There are no known workarounds for this vulnerability.</p> <p>Published: January 06, 2022; 6:15:07 PM -0500</p>	<p>V3.1: 7.5 HIGH</p> <p>V2.0: 5.0 MEDIUM</p>

Amazon's Best Practice for WordPress



Source: <https://docs.aws.amazon.com/whitepapers/latest/best-practices-wordpress/reference-architecture.html>

What's left for a static site



Static sites

Just a bunch of files served by the web server.

No dynamic code to keep up to date.

Requires the bare minimum from a hosting provider.

Static site generators

No one wants to write each page from scratch

Help build a site that can be easily navigated

Generate ancillary files like RSS feeds, sitemaps, etc

Hugo

Hugo can generate a website from a collection of markdown files

Powerful templating system built on top of Go's "text/template" package

Built-in support for compiling SASS style sheets.

Many themes available: 1000+ on <https://themes.gohugo.io/>

Convenient for local development

Install Hugo

On Ubuntu, the most recent version of Hugo can be installed as a snap:

```
sudo snap install --channel=extended hugo
```

Alternatively:

```
sudo apt install hugo
```

```
sudo dnf install hugo
```

```
sudo zypper install hugo
```

Set up a Hugo site

In an empty directory:

```
hugo new site mysite
cd mysite
git init .
```

Set up a theme:

```
git submodule add https://github.com/theNewDynamic/
                    gohugo-theme-ananke.git themes/ananke
echo 'theme = "ananke"' >> config.toml
echo 'enableGitInfo = true' >> config.toml
```

Hugo directory layout

“hugo new site” creates a number of files and directories:

config.toml	Site configuration
content/	Location for site pages
archetypes/	Defaults for new pages
layouts/	Templates used to format pages from content/ directory
themes/	Themes
static/	Content copied verbatim into site
data/	Structured data (can probably be ignored)

Pages

Pages can be a stand alone markdown file, or a markdown file and associated resources (e.g. embedded images)

Pages can be nested in “branch bundles”, with their own index page

Page	Page Bundle	Branch Bundle
foo.md	foo/ index.md image1.jpg image2.jpg ...	foo/ _index.md bar.md baz/ index.md

Page Format

Pages consist of front matter in YAML format followed by text in Markdown format:

```
---
```

```
yaml: ...
```

```
---
```

```
Page content
```

Yaml cheatsheet

Hierarchical structured data. Similar to JSON, but easier to write.

Simple values	Arrays
42 "string" true	- item1 - item2 - subitem
Dictionaries	Arrays of dictionaries
key1: value1 key2: value2 key3: subkey: value	- key1: value1 key2: value2 - key3: value3 key4: value4

Front matter metadata

`title`: page title

`date`: the date assigned to the page

`draft`: if true, then the page will not be included in the site

`url`: override the URL path for this page

`aliases`: array of URLs that should redirect to this page

Plus any custom metadata you want

Markdown cheatsheet

Heading

Subheading

Subsubheading

italic

****bold****

``monospace``

[link](http://example.com)

1. first

2. second

3. third

* first

* second

* third

...

Preformatted text

...

Shortcodes

Shortcodes are shortcuts for including common content in a page:

- Youtube videos: `{{< youtube "video-id" >}}`
- Twitter: `{{<tweet user="user" id="id" >}}`
- Syntax highlighting: `{{< highlight lang >}}` ... `{{</ highlight >}}`

Can also define custom shortcodes using the template system.

Building the site

Can be done by running “hugo” with no arguments. The following happens:

- All files from `static/` directory copied to `public/`
- Load all pages from `content/`
- Use templates to generate output in `public/` from the set of pages.
 -
- Generate resources referenced by templates (e.g. SASS compiled to CSS), and copy to `public/`

Contents of `public/` can be served by a web server.

Local development

Run “hugo server -D”

Point web browser at <http://localhost:1313/>

Edit pages in your favourite text editor

See pages updated in real time in browser

Commit when happy

Blog style URLs

Add the following to config.toml:

```
[permalinks]
blog = "/blog/:year/:month/:slug/"
```

All pages in content/blog/ will be assigned URLs based on the date in their frontmatter.

With an appropriate theme, <https://sitename/blog/> will show a paginated index of pages in the blog.

Related Pages

Add the following to config.toml:

```
[related]
threshold = 80
includeNewer = true
[[related.indices]]
name = "keywords"
weight = 100
```

Use keywords metadata in front matter to find related pages. Templates can use list of related pages when generating output.

Deploying a Hugo site

As an alternative to self-hosting, there are several companies that will build and publish your site from a repository hosted on Github/Gitlab/Bitbucket.

Continuous integration: push change to git ⇒ hugo build ⇒ publish site

Netlify

Sign up for account at <https://www.netlify.com/>

Free tier provides 300 build minutes and 100 GB of bandwidth per month.

Supports repos on Github, Gitlab, and Bitbucket.

Creates a free <https://sitename.netlify.app/> domain.

Supports custom domains, with automatic Let's Encrypt TLS certificates.

Will build “deploy previews” of pull requests: great for collaboration.

More info: <https://gohugo.io/hosting-and-deployment/hosting-on-netlify/>

Github

Github Pages can build a site from a repo hosted on Github (only public repos for free accounts though).

Use Github Actions to build site with Hugo and check build artifacts into a gh-pages branch of repo.

Creates a free <https://username.github.io/> domain.

Can configure custom domains with automatic Let's Encrypt TLS certificates.

More info: <https://gohugo.io/hosting-and-deployment/hosting-on-github/>

Gitlab

Gitlab can build sites from repos hosted on gitlab.com (even private repos)

Use a Gitlab CI pipeline is invoked to generate the site.

Creates a free <https://username.gitlab.io/> domain.

Supports custom domains, but doesn't automatically acquire TLS certificates.

- Can still use free Let's Encrypt certs, but must upload a new cert every 90 days.

More info: <https://gohugo.io/hosting-and-deployment/hosting-on-gitlab/>

Dynamic site features

Even if your site is mostly static, you might want a few dynamic features. Some of these can be integrated from third parties.

For example:

- Comments
- Contact forms

Comment systems

Disqus

- Hugo has built in support:
 - <https://gohugo.io/content-management/comments/#add-disqus>
- Show ads these days, and has started tracking users to target those ads.

Cusdis

- An open source alternative: <https://cusdis.com/>
- Either self hosted, or use their service.
- No Hugo integration, so requires some template hacking.

Contact forms

Netlify Forms offers 100 submissions a month in free tier, but only for Netlify hosted sites:

- <https://www.netlify.com/products/forms/>

Formspark provides 250 submissions in free tier, and is not tied to hosting provider:

- <https://formspark.io/>

What if I want more?

If you are self-hosting your site, the web server can serve static pages directly and forward other URLs to a web app.

If using a static site hosting service, consider putting dynamic features on their own domain.

Use something like AWS Lambda for infrequently run code.

If everything needs to be dynamic, then perhaps a static site generator is not for you.

Demo and Questions