Introduction to Hugo

Static Site Generator using Markdown

What is Hugo?

Think of Hugo like a **modern factory**

- Given Markdown and Templates, Hugo translates them into a Static Website.

```
Markdown + Templates → Hugo → Static Website
      ↓                  ↓              ↓
   Content           Processor    Ready Product
```

Hugo is a **static site generator** that transforms simple text files into lightning-fast websites.

- **Raw Materials**: Markdown files + templates
- **Factory Machine**: Hugo compiler (written in the Go language)
- **Finished Product**: Complete HTML/CSS/JS website

Hugo vs Marp

Similarities 📝

- Both use **Markdown** as primary content format
- Support YAML front matter (`---`) for metadata
- Can embed images, code blocks, and tables
- Cross-platform & open-source
- Factory-like page generation - template & compiled into target pages

Differences 🔍

Feature	Hugo 🌐	Marp 🎤
Purpose	Static site generator	Presentation slide generator
Output	Web pages (HTML/CSS/JS)	Slide decks (HTML/PDF/PPTX)
Theme	Website themes/templates	Slide themes/styles
Build	`hugo` command → website	`marp` CLI/extension → slides
Usage	Blogs, docs, portfolios	Talks, lectures, workshops

4

Restaurant Kitchen Analogy

Traditional Web (Dynamic)	Hugo (Static)
Cook-to-order restaurant	**Pre-made** meal factory
Server asks chef for each order	All meals prepared in advance
Slow (database queries)	Fast (files already ready)
Complex kitchen setup	Simple serving
More expensive to run	Cheaper to serve

Why Hugo?

- Speed
    - Websites load in milliseconds
    - No database queries needed
    - Files served directly

- Security

  - No server-side code to hack

  - No database vulnerabilities

  - Just static files

- Cost

  - Host for free (**GitHub Pages**, Netlify)

  - No server maintenance

  - Minimal resources needed

Hugo is Perfect For

- Documentation sites

- Blogs and portfolios

- Company websites

- Landing pages

Popular Sites Built with Hugo

- **GoHugo.io** - Hugo's own documentation
- **Kubernetes.io** - Container orchestration docs
- **1Password.com** - Password manager's blog
- **Netlify.com** - Hosting platform's site

From Markdown to Website

Input: Markdown File with YAML header

```
---
title: "My First Post"
date: 2024-01-15
---

# Welcome to My Blog

This is **bold text** and this is *italic*.

- List item 1
- List item 2
```

Hugo Project Structure

The `public/` folder is what gets uploaded to web servers (or to github.io)!

```
my-blog/
├── config.yaml              # Site configuration
├── content/                 # Your markdown files
│   ├── posts/
│   │   ├── first-post.md
│   │   └── second-post.md
│   └── about.md
├── themes/                  # Website themes
├── static/                  # Images, CSS, JS
├── layouts/                 # HTML templates
└── public/                  # Generated website (output)
```

Installation

```
# Windows (using Chocolatey)
choco install hugo-extended
# macOS (using Homebrew)
brew install hugo
# Linux/Ubuntu (using apt)
sudo apt install hugo
```

Check Hugo version

```
hugo version
```

The version should be 0.128 or later; if it is older than this version, install a new Hugo (Linux).

Install the newest Hugo.

```
# remove the non-extended .deb you installed
sudo apt remove -y hugo

# install the extended tarball
cd /tmp
VER=0.148.2
wget https://github.com/gohugoio/hugo/releases/download/v${VER}/hugo_extended_${VER}_linux-amd64.tar.gz
tar -xzf hugo_extended_${VER}_linux-amd64.tar.gz
# the archive contains 'hugo' binary
sudo mv hugo /usr/bin/hugo
hugo version   # should show +extended
```

Create New Site

```
Hugo new site my-awesome-blog
cd my-awesome-blog
```

Add themes

- Use `git submodule add` to add Hugo themes.

```
git init
git submodule add \
  https://github.com/theNewDynamic/gohugo-theme-ananke  themes/ananke
```

- Update config.toml configuration file.

```
echo "theme = 'ananke'" >> config.toml
```

Creating Content

Create Your First Post

```
hugo new posts/my-first-post.md
```

Edit the Content

```
---
title: "My First Post"
date: 2024-01-15T10:00:00Z
draft: false
---

# Hello World!
Hugo makes it **super easy** to create static websites!

## Features I Love:
- Fast compilation
- Beautiful themes
```

Start Development Server

```
hugo server —D
```

- `—D` flag includes draft posts
- Website available at `http://localhost:1313`
- **Live reload** - changes appear instantly!

Build for Production

```
hugo
```

This creates the `public/` folder with your complete website.

Deploy to GitHub Pages

- Create a GitHub.io repository or make a GitHub page for a repository.

- Upload the files in the `public/` directory into the GitHub Pages directory.

GitHub Actions Workflow (optional)

- We can use `GitHub Actions` to automatically publish Hugo content.

- Every git push automatically deploys your site!

- Copy the YAML in the `.github/workflows/hugo.yml`

- Make your Hugo project in GitHub; whenever you sync your file, it will automatically update the public directory.

```yaml
# .github/workflows/hugo.yml
name: Deploy Hugo site to Pages
on:
  push:
    branches: ["main"]
jobs:
  build:
    runs-on: ubuntu-latest
    steps:
      - uses: actions/checkout@v3
      - uses: peaceiris/actions-hugo@v2
        with:
          hugo-version: 'latest'
      - run: hugo --minify
      - uses: peaceiris/actions-gh-pages@v3
        with:
          github_token: ${{ secrets.GITHUB_TOKEN }}
          publish_dir: ./public
```

Hugo vs Other Solutions

Feature	Hugo	WordPress	React
Speed	+++	+	++
Learning Curve	Easy	Medium	Hard
Hosting Cost	Free	$$	$
Security	High	Medium	Medium
SEO	Excellent	Good	Good
Content Management	Markdown	GUI	Code

Resources

- **Official Documentation**

  - Hugo Documentation

  - Hugo Themes

- **Tutorials**

  - Hugo Quick Start

  - GitHub Pages Deploy Guide