This guide walks through how to deploy and host your Gatsby site on a DigitalOcean Droplet with Ubuntu and Nginx.

DigitalOcean provides a cloud platform to deploy, manage, and scale applications of any size, removing infrastructure friction and providing predictability so developers and their teams can spend more time building better software.

DigitalOcean's product droplets are scalable compute IaaS (Infrastructure as a Service) or a VPS (Virtual Private Server) on the cloud which has great reliability and scalability. They come with varied price ranges ideal for small apps to giant enterprise-level apps.

They provide service to select from various Unix-based distributions and select your technology-based platform with preinstalled pre-requisites from the marketplace. This guide will walk through the specific options that work best for deploying a Gatsby site with DigitalOcean.

A Droplet can be spun up in less than a minute for as little as $5/month.

Prerequisites

- A Gatsby site living in a Git repository (GitHub, GitLab, or any Git cloud)
- A DigitalOcean Droplet with a non-root user configured with `sudo` group (example: Ubuntu 18.04)
- A custom domain name for your Gatsby site to help with configuring HTTPS

How to deploy your Gatsby site to DigitalOcean

Install Node.js, npm and Gatsby-CLI onto your droplet

Follow these instructions for installs on an Ubuntu droplet.

1. Log in to your droplet as a non-root user.
2. Install Node.js

   ```
   sudo apt-get update
   sudo apt-get install nodejs
   ```

3. Install npm

   ```
   sudo apt-get install npm
   ```

   To view the version of Node.js and npm installed, run,

   ```
   nodejs -v
   npm -v
   ```

4. To install the latest stable Node.js release using the `n` package (Required),

   ```
   sudo npm install -g n
   sudo n stable
   ```

   > ***Note:*** *If you check the version now, you would see the older versions of Node.js and npm from the cache. You can either exit and restart your terminal or refresh the cache by running the following commands:*

```
hash nodejs
hash npm
```

5. Install the Gatsby CLI globally. This will be useful ahead in building the Gatsby site for production.

```
sudo npm install -g gatsby-cli
```

Clone your repository to the droplet

The next step is to clone the repository containing your Gatsby app (Replace `<your-github-repo-site>` with your GitHub repository link)

```
git clone <your-github-repo-site>
```

> ***Note:*** *Copy the path where your* `<your-github-repo-site>` *is cloned, for future reference.*

```
pwd
```

In case of a warning related to "Permission denied", check if `<your non-root user>` has `sudo` privileges. Or before cloning your repository, change permissions for `<your non-root user>` to access the `.config` directory of under `/home/<your non-root user>/` :

```
cd ~/
sudo chown -R $(whoami) .config
```

> ***Note:*** *This guide will refer to the cloned directory as* `<my-gatsby-app>` *for simplicity; you should replace it with your repo directory name.*

Generate your Gatsby site for production

The static files will be hosted publicly on the droplet. The `gatsby build` command provides utility to build the site and generate the static files in the `/public` .

> ***Note:*** *Go to the path where* `<my-gatsby-app>` *is. You can use the copied path for reference in a previous step.*

1. Install dependencies.

```
cd <my-gatsby-app>
sudo npm install
```

2. Run build to generate static files.

```
sudo gatsby build
```

Install Nginx Web Server to host the site and open firewall to accept HTTP and HTTPS requests

To host a website or static files onto a Linux-based server/VPS, a web-server like Apache or Nginx is required.

Nginx is web-server. It provides the infrastructure code for handling client requests from the World Wide Web, along with features like a load balancer, mail proxy, and HTTP Cache.

1. Install Nginx.

   ```
   sudo apt-get install nginx
   ```

2. Configure firewall settings of the droplet to listen to HTTP and HTTPS requests on port 80 and 443 respectively.

   ```
   sudo ufw allow 'Nginx HTTP'
   sudo ufw allow 'Nginx HTTPS'
   ```

   > ***Note:*** *You may have to allow access to port 443 explicitly using* `sudo ufw allow 443/tcp`.

3. To check the access,

   ```
   sudo ufw app list
   ```

4. If `ufw` status is disabled/inactive, you can enable it with the following command:

   ```
   sudo ufw enable
   ```

   Allow the OpenSSH if not already done, to not disconnect from your droplet.

   ```
   sudo ufw allow 'OpenSSH'
   ```

Configure Nginx to point to your Gatsby site's `/public` directory and add your domain

Change the root directory configuration of Nginx in the default server block file

1. Go to `/etc/nginx/sites-available/`

   ```
   cd /etc/nginx/sites-available/
   ```

2. Open the file `default` in Vim (shortcut cheat sheet)

   ```
   sudo vim default
   ```

3. Edit the file and make the following changes for below-mentioned fields, leave the rest of the fields as is. Your exact path may vary, but it may resemble `/home/<your non-root user>/<my-gatsby-app>/public`.

   ```
   server {
     root <path to my-gatsby-app>/public;
   ```

```
  index index.html index.htm index.nginx-debian.html;

  server_name <your-domain-name>;

  location / {
    try_files $uri $uri/ =404;
  }
}
```

4. Restart the Nginx service

   ```
   sudo systemctl restart nginx
   ```

   You should now be able to view your built Gatsby site at your DigitalOcean IP address, before configuring a domain.

5. Configure your domain to point to the IP address of your droplet. Go to the Advanced DNS settings in your domain provider's console and put an `A` record that points to the IP address of the droplet.

6. By this time, you can view your site live at `<your-domain>`.

Configuring HTTPS for your Gatsby site

Follow the below steps to configure your site with a free SSL/TLS certificate from Lets Encrypt using their Certbot CLI tool.

1. Install Certbot onto your droplet.

   You'll need to install Certbot using `snapd`. To do so, run the following commands:

   ```
   sudo snap install core; sudo snap refresh core
   sudo apt-get remove certbot
   ```

   Run the following command to install Certbot.

   ```
   sudo snap install --classic certbot
   ```

   Run the following command to start using Certbot.

   ```
   sudo ln -s /snap/bin/certbot /usr/bin/certbot
   ```

2. Generate the certificate. Certbot will automatically edit and configure the Nginx config file and point to the certificate file.

   Run the following command:

   ```
   sudo certbot --nginx
   ```

3. If you are using Certbot for the first time on this droplet then you will be prompted to enter your e-mail for recovery purposes.

4. Agree to the license agreement on prompt.

> ***Note:*** *You will be prompted to select the domain for which you want to generate the certificate. Select the domain configured in a previous step.*

> ***Note:*** *You will be prompted to choose the option to redirect HTTP requests to HTTPS, which you may choose on your needs. (It is recommended to choose to redirect HTTP to HTTPS)*

5. Restart the Nginx service.

```
sudo systemctl restart nginx
```

6. Now, you can access your site at `<your-domain>` with a secure connection.

View your Gatsby site live

Once you've followed along with all the steps and configuration properly, you can visit your site live at `<your-domain>`.

Whenever there's an update to your site, run a `sudo gatsby build` in the root of your `<my-gatsby-app>` and your changes will be live.

Congratulations! You've deployed your Gatsby App on a DigitalOcean droplet along with configuring HTTPS for it.

Additional resources

There's a lot more to learn about DigitalOcean's Droplets, Ubuntu configurations, and Nginx. Below are some links which could be useful in achieving the prerequisites of this post:

- Microblog - Create a new non-root user with `sudo` privileges on Ubuntu-based DigitalOcean Droplet configured with SSH
- Official DigitalOcean Docs
- Official Nginx Docs
- Configuring HTTPS Servers with Nginx
- How To Install Nginx on Ubuntu 18.04
- How To Secure Nginx with Let's Encrypt on Ubuntu 18.04
- Installing Certbot with Nginx