TASK

Hosting and Maintaining Apps on DigitalOcean

Visit our website

Introduction

WELCOME TO THE TASK ABOUT HOSTING AND MAINTAINING APPS ON DIGITALOCEAN!

By the end of this task, you will have a practical understanding of how to host and maintain a web application on a DigitalOcean Droplet. Through the course of the task, you will also gain an understanding of what web servers are and how to secure them.

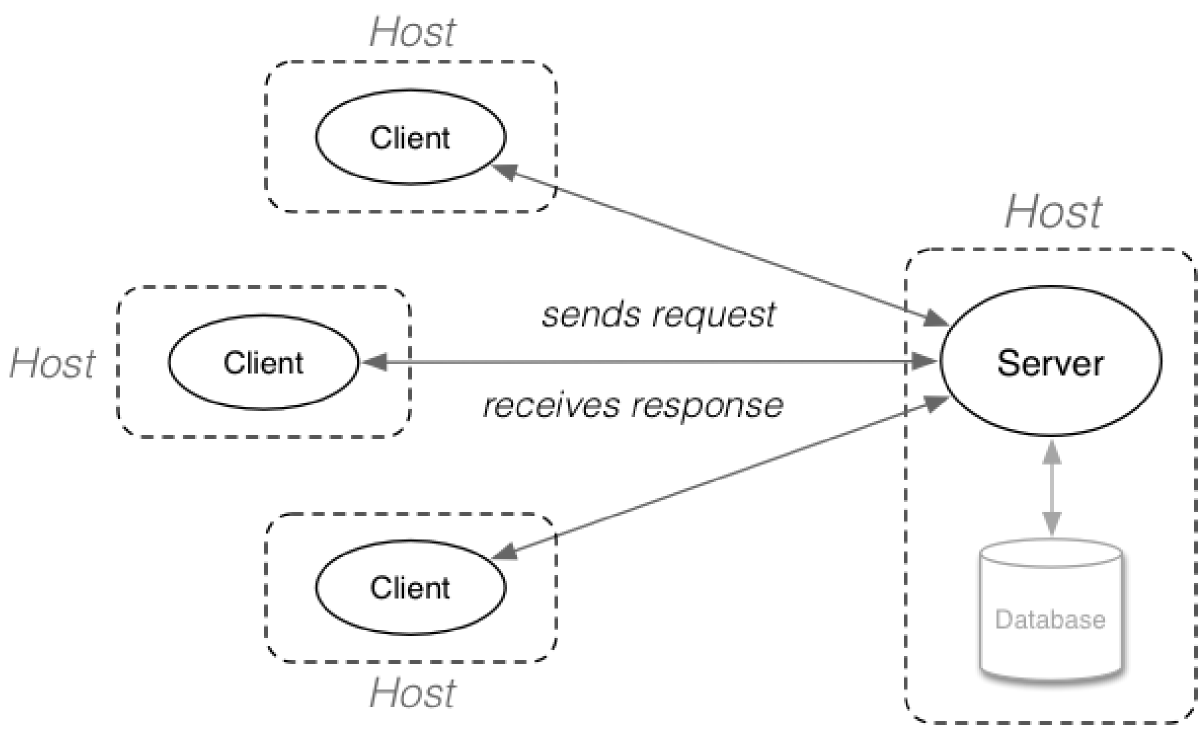
The practical experience gained through this task can also be applied to hosting and maintaining applications on servers and virtual machines from providers other than DigitalOcean.

WHAT IS CLIENT-SERVER ARCHITECTURE?

Client-server architecture, also known as a client-server model, is a network architecture that breaks down tasks and workloads between clients and servers that reside on the same system or are linked by a computer network such as an intranet or the Internet.

Client-server architecture typically features multiple workstations, PCs, or other devices belonging to users, connected to a central server via an Internet connection or other network connection. The **client** sends a request for data, and the **server** accepts and processes the request, sending the requested data back to the user who needs it on the client-side.

For the purpose of this bootcamp we will be focusing on how this model relates to web development in particular.



The above illustration shows how clients interact with a server and how that server interacts with the database so that data is processed by a server and served back to the clients.

Hosting & Web Servers

WHAT IS HOSTING (OR WEB HOSTING)?

Hosting (or Web Hosting) is the process of storing application code and/or data on a **server** or a virtual machine, which can then be accessed by **clients** over the Internet, typically over the HTTP or HTTPS protocols.

A web hosting service provides all the facilities necessary for hosting a web site. Typically, a hosting service provider provides such a service, which is basically a computing and storage infrastructure accessible over the Internet. DigitalOcean is an example of such a service provider. Others you may have heard of include **Azure** and **AWS**.

WHAT IS A WEB SERVER?

A web server is a piece of software (or a program) that can serve requests from clients for a web page using HTTP/HTTPS or in a more modern approach through APIs. The web pages can be written in HTML and JavaScript, as well as other languages like PHP, Java, etc.

Types of Web Servers

Web Servers are broadly categorised into two types based on the operating systems they run on. They are:

1. Linux web servers
2. Windows web servers

Examples of Web Servers

The following are some examples of popular web servers and the operating systems they are available for:

Name	Linux	Windows
Apache	✓	✓
NGINX	✓	✓
Lighttpd	✓	
Oracle WebLogic	✓	✓
Wildfly	✓	
Litespeed	✓	
Microsoft IIS		✓

Hosting a Web App

SETTING UP A WEB SERVER

In the following sections, you will set up the tools for executing commands over **Secure Shell (SSH)**.

Pre-requisites

In order to successfully create a Droplet and access it you will require the following:

- A DigitalOcean account.

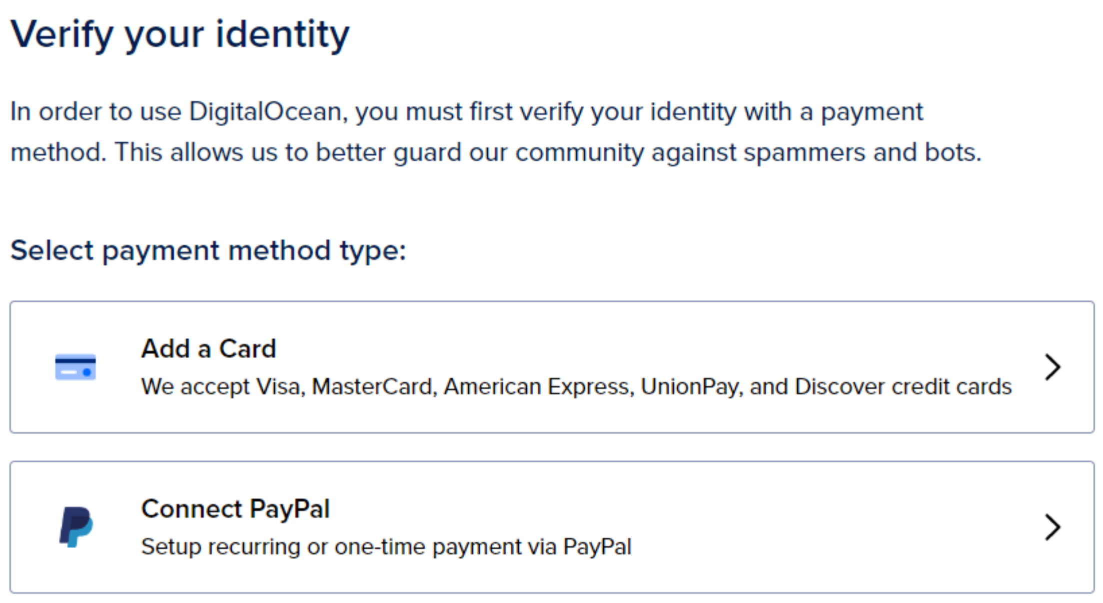
GitHub Student Developer Pack

Important: As a HyperionDev student, you are entitled to a **GitHub Student Developer Pack**. This will give you access to real-world tools used in the industry to give yourself a head start for when you're within the industry.

Go to the **GitHub Student Development Pack Guide** in the Installations folder in your Dropbox and follow the instructions to set up your access if you have not already done so.

Using and accessing DigitalOcean

1. Create a DigitalOcean account on the **registrations page**.
2. Verify your identity using a bank card or PayPal (options vary according to region).



Verify your identity

In order to use DigitalOcean, you must first verify your identity with a payment method. This allows us to better guard our community against spammers and bots.

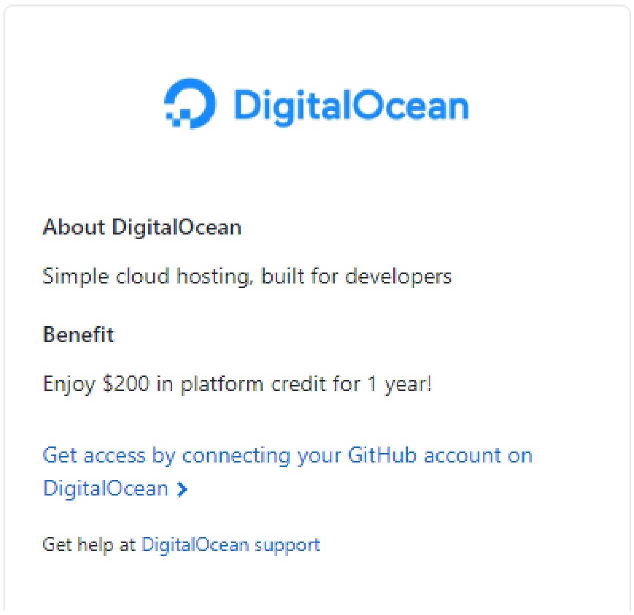
Select payment method type:

Add a Card
We accept Visa, MasterCard, American Express, UnionPay, and Discover credit cards

Connect PayPal
Setup recurring or one-time payment via PayPal

3. Go to **Github Education** to activate your DigitalOcean credit ($ 200). You will need to sign into your GitHub account for this step.

4. Scroll down to the DigitalOcean tile and click “Get access by connecting your GitHub account on DigitalOcean”.



DigitalOcean

About DigitalOcean

Simple cloud hosting, built for developers

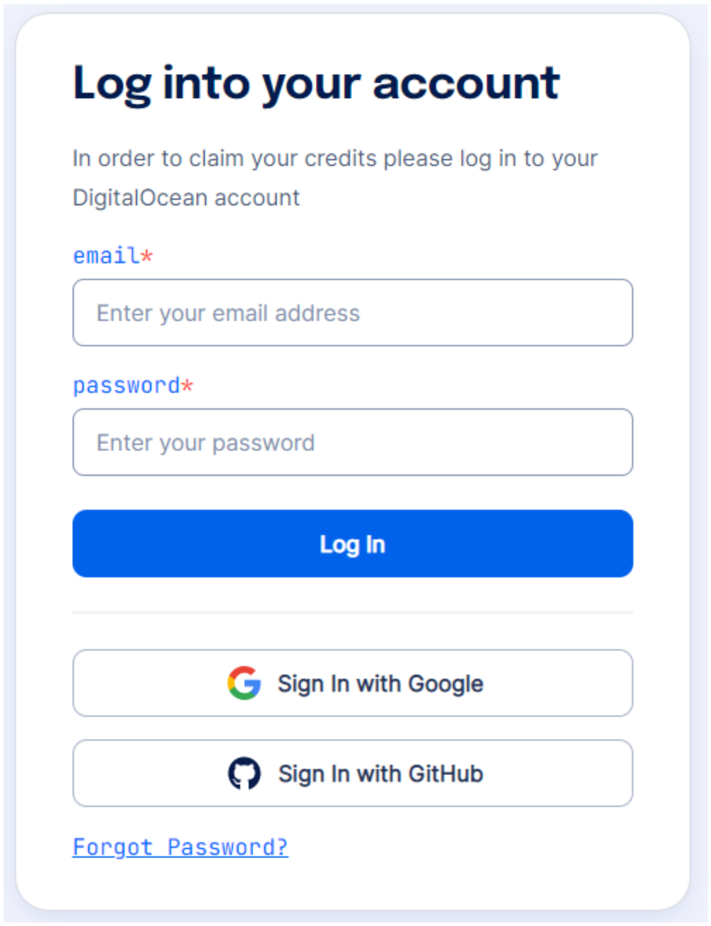
Benefit

Enjoy $200 in platform credit for 1 year!

Get access by connecting your GitHub account on DigitalOcean >

Get help at DigitalOcean support

5. Log into your Digital Ocean account.



Log into your account

In order to claim your credits please log in to your DigitalOcean account

email*

Enter your email address

password*

Enter your password

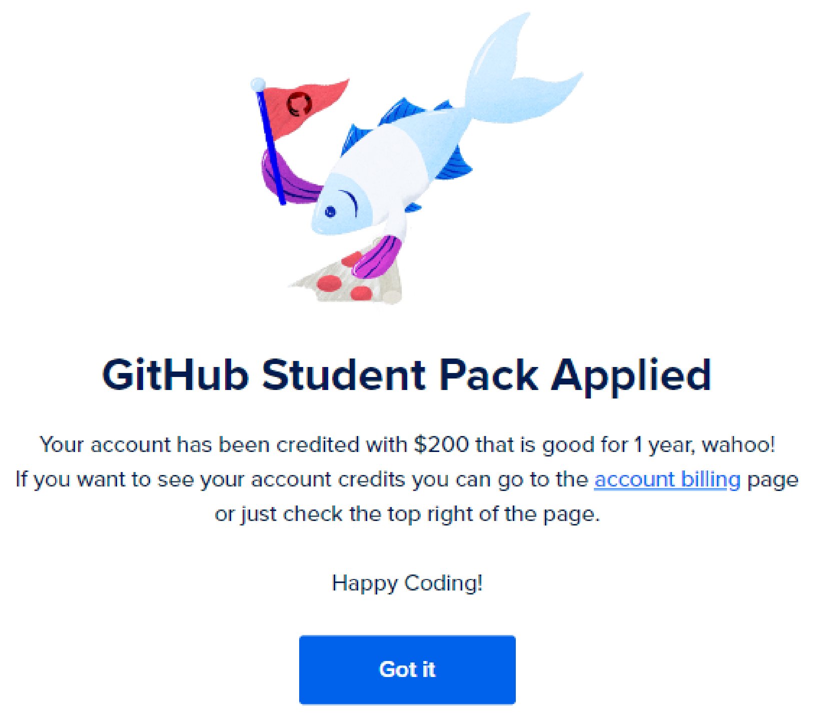
Log In

Sign In with Google

Sign In with GitHub

Forgot Password?

6. You should receive confirmation that you have been credited $200.

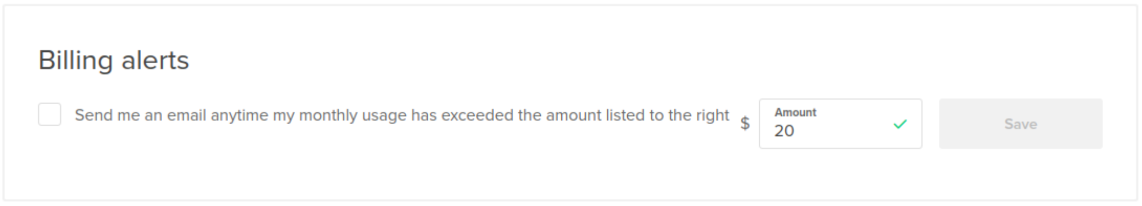


GitHub Student Pack Applied

Your account has been credited with $200 that is good for 1 year, wahoo!
If you want to see your account credits you can go to the account billing page or just check the top right of the page.

Happy Coding!

7. You may set up a **billing alert** to ensure you do not spend more than the amount credited to your account.



Billing alerts

Send me an email anytime my monthly usage has exceeded the amount listed to the right $ Amount 20

Creating a Project

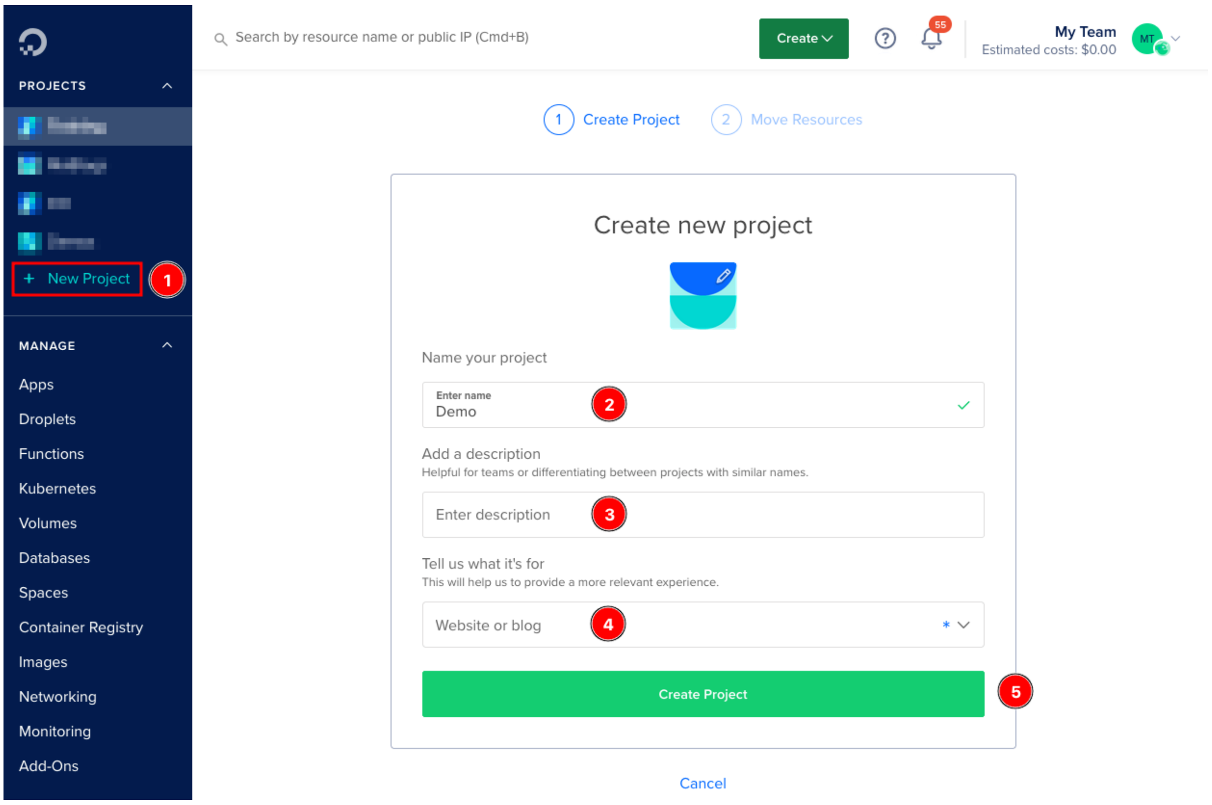
DigitalOcean provides a mechanism to organise resources such as Droplets, Spaces etc into a group called a **Project**. A use case of the Project can be to group resources that belong to a certain environment such as development, staging or production. A Project is required in order to create resources.

A new account includes a **Default Project** which can be used. One can also create additional projects. You may use either the default project or your own new project that you create.

The following steps show how to create a Project in DigitalOcean.

1. On the dashboard, click **+ New Project** on the left to begin.
2. Enter a name for the project; in this example, it's called **Demo**

3. Give a description (optional)
4. Select **Website or blog** under “Tell us what it’s for”
5. Click **Create Project**



Name your project
Enter name
Demo

Add a description
Helpful for teams or differentiating between projects with similar names.
Enter description

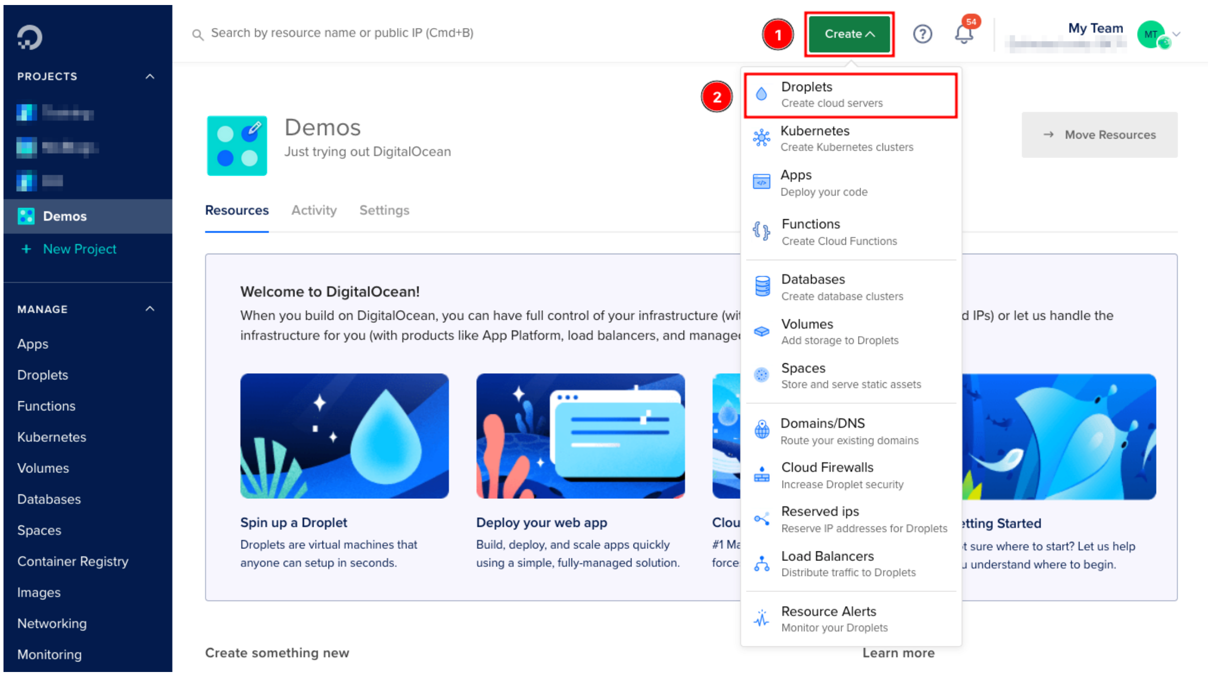
Tell us what it's for
This will help us to provide a more relevant experience.
Website or blog

Creating a Droplet

Follow the steps below to create a Droplet. Leave any fields not specified to their defaults.

1. Start by accessing the DigitalOcean control panel at https://cloud.digitalocean.com/.

2. Click the **Create** button (1) and select **Droplets** (2) from the dropdown.



Welcome to DigitalOcean!

When you build on DigitalOcean, you can have full control of your infrastructure (wi d IPs) or let us handle the infrastructure for you (with products like App Platform, load balancers, and manage

Spin up a Droplet

Droplets are virtual machines that anyone can setup in seconds.

Deploy your web app

Build, deploy, and scale apps quickly using a simple, fully-managed solution.

Clou

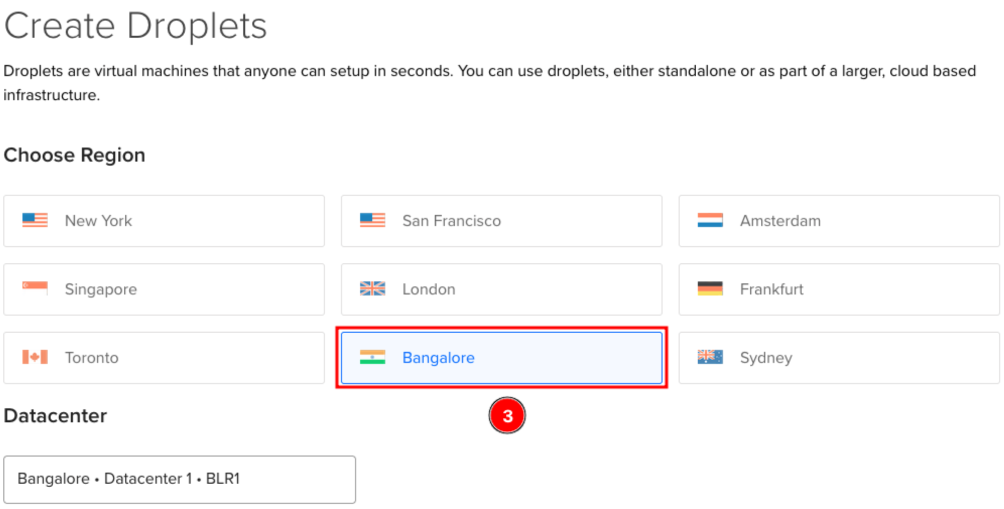
#1 Ma

force

tting Started

t sure where to start? Let us help u understand where to begin.

3. Select a Region (3) closest to you. In this example, **Bangalore** is selected.



Create Droplets

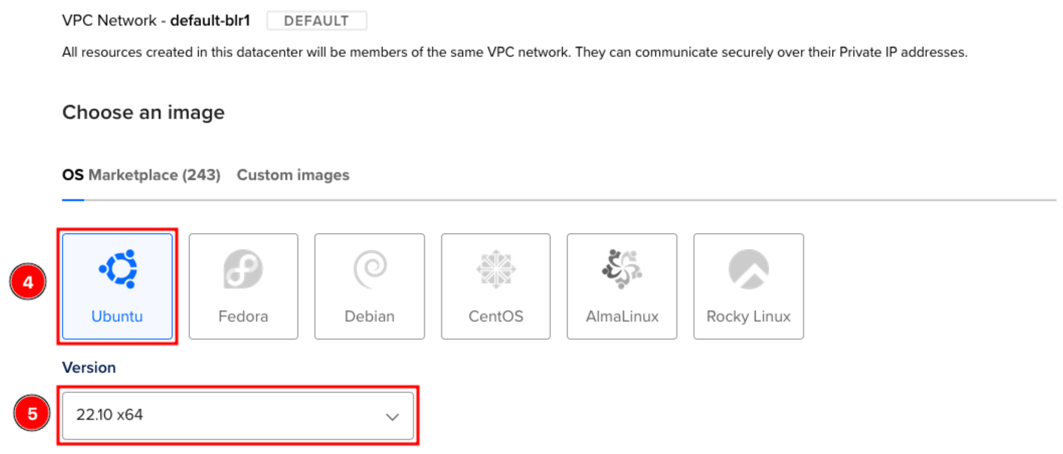
Droplets are virtual machines that anyone can setup in seconds. You can use droplets, either standalone or as part of a larger, cloud based infrastructure.

Choose Region

Datacenter

Bangalore • Datacenter 1 • BLR1

4. Next, choose an image. For this task, select the OS as **Ubuntu** (4). The most recent version will be pre-selected - use this and don't change it. .



VPC Network - **default-blr1** DEFAULT

All resources created in this datacenter will be members of the same VPC network. They can communicate securely over their Private IP addresses.

Choose an image

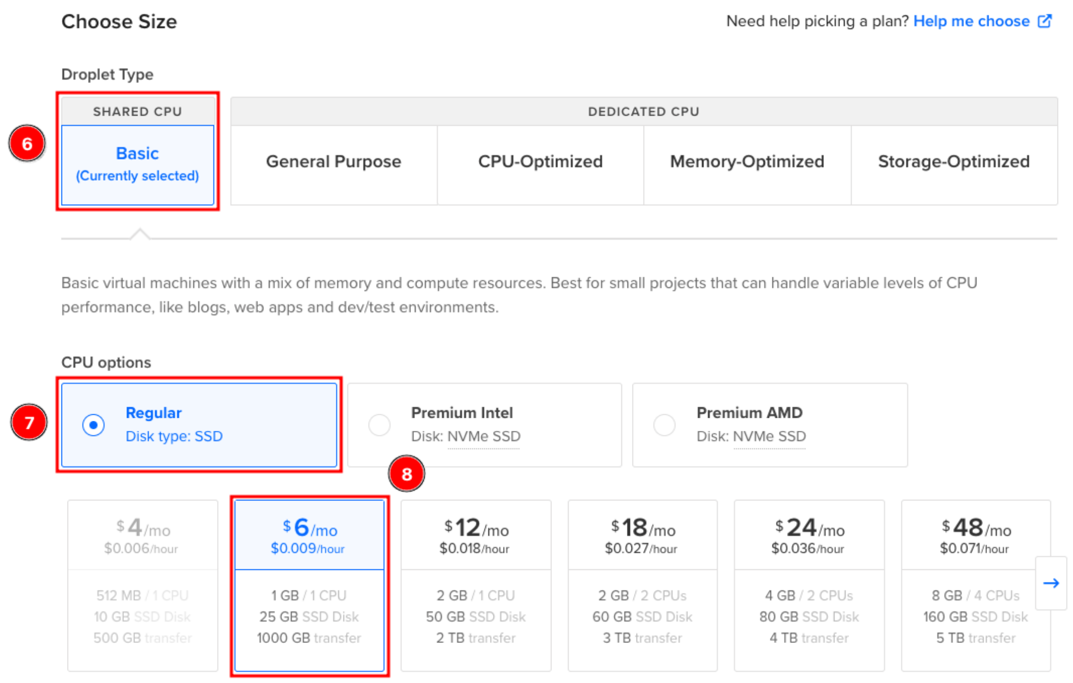
OS Marketplace (243) | Custom images

4 [✓ Ubuntu] Fedora Debian CentOS AlmaLinux Rocky Linux

Version

5 22.10 x64

5. For the size of the Droplet, select **SHARED CPU Basic** (6) and click **Regular** (7) for CPU options. Then select the **$6/mo** option (8), which provides 1 GB RAM, 1 CPU, 25GB SSD, and 1000 GB transfer.



Choose Size

Need help picking a plan? Help me choose

Droplet Type

SHARED CPU	DEDICATED CPU			
6 Basic (Currently selected)	General Purpose	CPU-Optimized	Memory-Optimized	Storage-Optimized

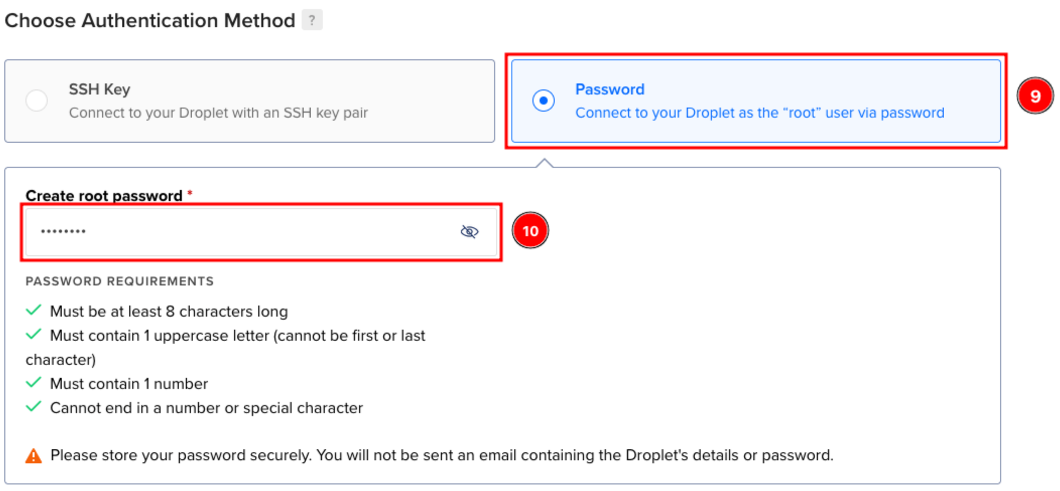
Basic virtual machines with a mix of memory and compute resources. Best for small projects that can handle variable levels of CPU performance, like blogs, web apps and dev/test environments.

CPU options

7 [✓ Regular Disk type: SSD] Premium Intel Disk: NVMe SSD Premium AMD Disk: NVMe SSD

$4/mo $0.006/hour	8 $6/mo $0.009/hour	$12/mo $0.018/hour	$18/mo $0.027/hour	$24/mo $0.036/hour	$48/mo $0.071/hour
512 MB / 1 CPU 10 GB SSD Disk 500 GB transfer	1 GB / 1 CPU 25 GB SSD Disk 1000 GB transfer	2 GB / 1 CPU 50 GB SSD Disk 2 TB transfer	2 GB / 2 CPUs 60 GB SSD Disk 3 TB transfer	4 GB / 2 CPUs 80 GB SSD Disk 4 TB transfer	8 GB / 4 CPUs 160 GB SSD Disk 5 TB transfer

6. Select **Password** (9) as the authentication method and enter a password of your choice (10).


Choose Authentication Method ?

SSH Key
Connect to your Droplet with an SSH key pair

Password
Connect to your Droplet as the "root" user via password

9

Create root password *
••••••••

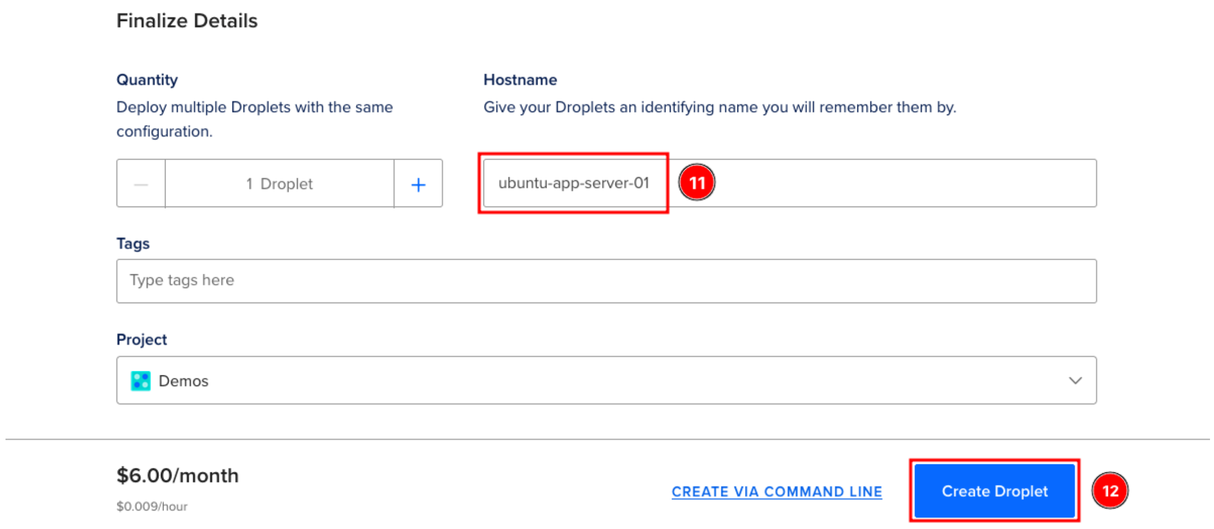
10

PASSWORD REQUIREMENTS
✓ Must be at least 8 characters long
✓ Must contain 1 uppercase letter (cannot be first or last character)
✓ Must contain 1 number
✓ Cannot end in a number or special character

⚠ Please store your password securely. You will not be sent an email containing the Droplet's details or password.

Take note: Remember to **make a note of this password,** as DigitalOcean will not send any email or notification containing the password.

7. Finally, we will set a hostname for easy identification. Type **ubuntu-app-server-01** in the hostname box (11). Leave the rest of the fields set to their defaults and click **Create Droplet** (12).


Finalize Details

Quantity
Deploy multiple Droplets with the same configuration.

− 1 Droplet +

Hostname
Give your Droplets an identifying name you will remember them by.

ubuntu-app-server-01

11

Tags
Type tags here

Project
Demos

$6.00/month
$0.009/hour

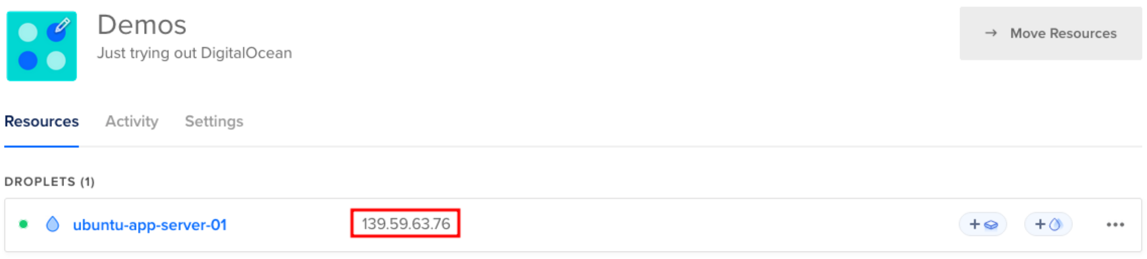
CREATE VIA COMMAND LINE

Create Droplet

12

8. This will provision a Droplet for you. On successful provisioning, you will see the public **IP address** of the Droplet. **Make a note of it** for use in the next

step.



ubuntu-app-server-01 139.59.63.76

This completes the creation of a Droplet.

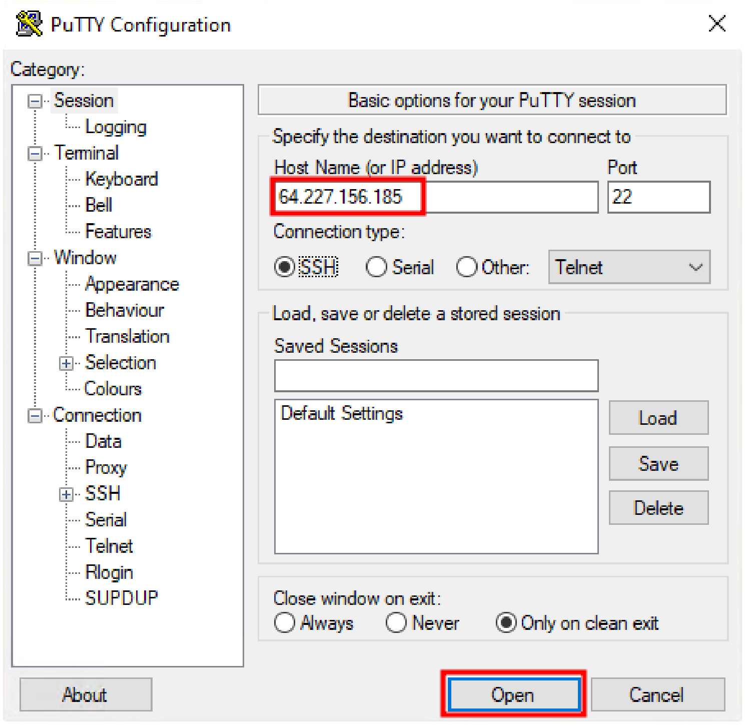
Accessing the Droplet

There are two ways to access your Droplet - using a terminal, or using the Droplet's built-in console.

Access via PuTTY

If you use Windows, **download PuTTY** and install it by double-clicking the downloaded msi file. This is so that we can SSH into your Ubuntu Linux virtual machine on DigitalOcean later on. Windows does not have native SSH support, so it is necessary to use a tool like PuTTY.

Next, launch PuTTY, enter the IP address of the Droplet in the **Host Name (or IP Address)** box as shown in the screenshot below and click **Open**.



Specify the destination you want to connect to

Host Name (or IP address): 64.227.156.185

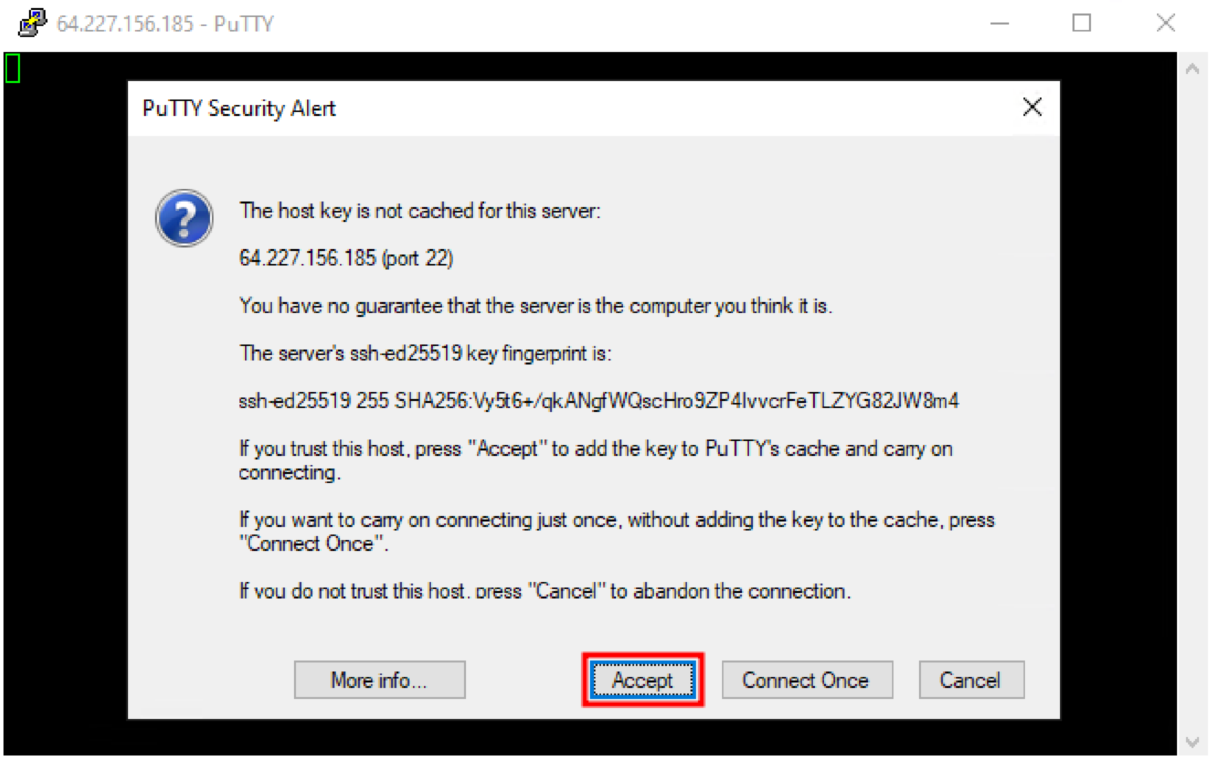
Port: 22

Connection type: [✓ SSH] Serial Other: Telnet

Saved Sessions: Default Settings

Close window on exit: Always Never [✓ Only on clean exit]

If this is the first time logging in, it will prompt whether you trust the host - click **Accept**.



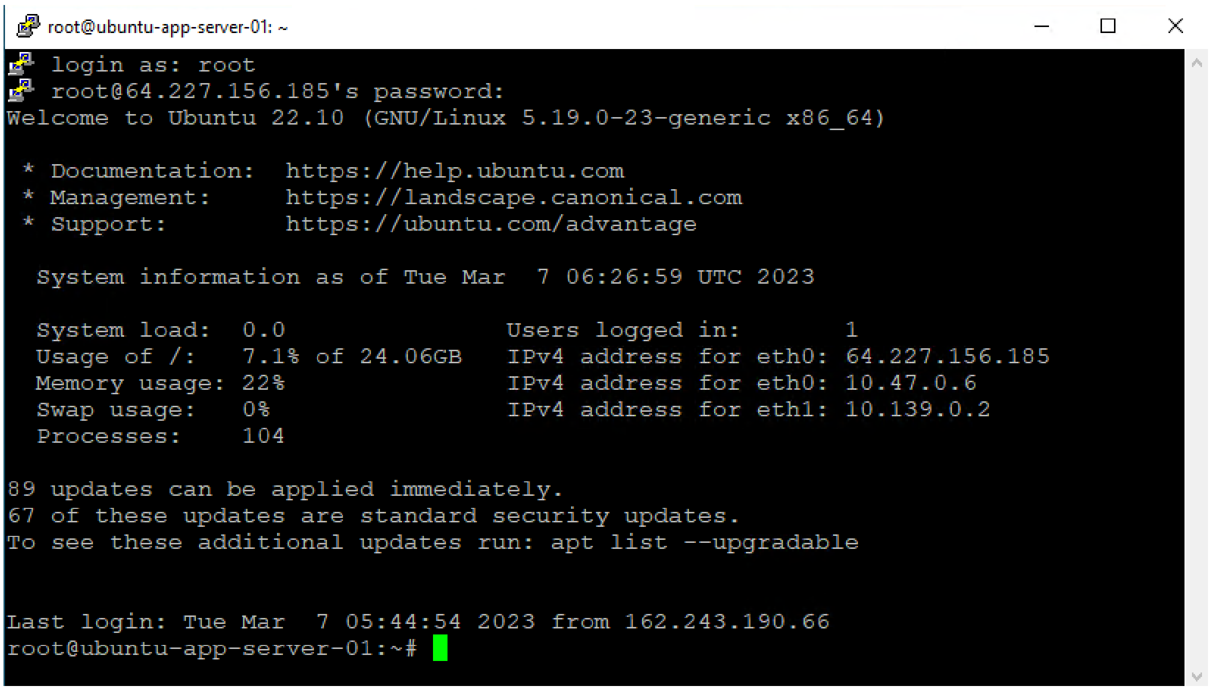
```
64.227.156.185 - PuTTY

PuTTY Security Alert

The host key is not cached for this server:
64.227.156.185 (port 22)
You have no guarantee that the server is the computer you think it is.
The server's ssh-ed25519 key fingerprint is:
ssh-ed25519 255 SHA256:Vy5t6+/qkANgfWQscHro9ZP4lvvcrFeTLZYG82JW8m4
If you trust this host, press "Accept" to add the key to PuTTY's cache and carry on
connecting.
If you want to carry on connecting just once, without adding the key to the cache, press
"Connect Once".
If you do not trust this host, press "Cancel" to abandon the connection.

More info...    Accept    Connect Once    Cancel
```

Finally, login using the username and password.



```
root@ubuntu-app-server-01: ~
login as: root
root@64.227.156.185's password:
Welcome to Ubuntu 22.10 (GNU/Linux 5.19.0-23-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Tue Mar  7 06:26:59 UTC 2023

  System load:  0.0               Users logged in:       1
  Usage of /:   7.1% of 24.06GB   IPv4 address for eth0: 64.227.156.185
  Memory usage: 22%               IPv4 address for eth0: 10.47.0.6
  Swap usage:   0%                IPv4 address for eth1: 10.139.0.2
  Processes:    104

89 updates can be applied immediately.
67 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable


Last login: Tue Mar  7 05:44:54 2023 from 162.243.190.66
root@ubuntu-app-server-01:~#
```

Access via the terminal

The Droplet can also be accessed through a Linux / macOS terminal.

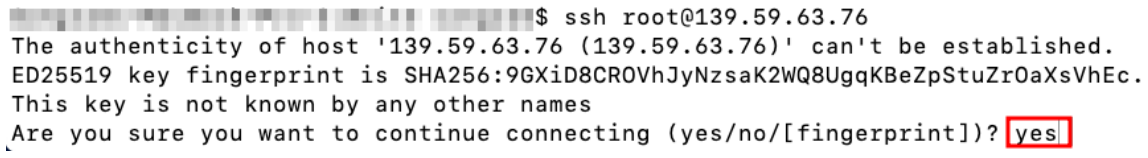
The following steps demonstrate how to access your Droplet using a terminal.

1. Open the terminal.

2. Type the following command on the terminal and press enter. Remember to **replace the IP with your actual Droplet IP** that you noted down earlier.

```
ssh root@139.59.63.76
```

3. When prompted about the authenticity of the host, type **yes** and press enter.



```
[redacted]$ ssh root@139.59.63.76
The authenticity of host '139.59.63.76 (139.59.63.76)' can't be established.
ED25519 key fingerprint is SHA256:9GXiD8CROVhJyNzsaK2WQ8UgqKBeZpStuZrOaXsVhEc.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
```

4. At the next prompt, type your password and press enter to complete the connection. Note that when you type the password it won't be visible.

Well done - you have now successfully logged into the Droplet!

Access via the built-in console

Droplets also provide a built-in access console to connect to the Droplet. You can also use this method with SSH. The following steps demonstrate how to use the built-in access console.

1. Ensure you are in the project dashboard (click on the project name on the left (1)) and click on the horizontal ellipsis menu (three dots **...**) to the right of the Droplet name (2). Then, select **Access Console** (3) from the list.

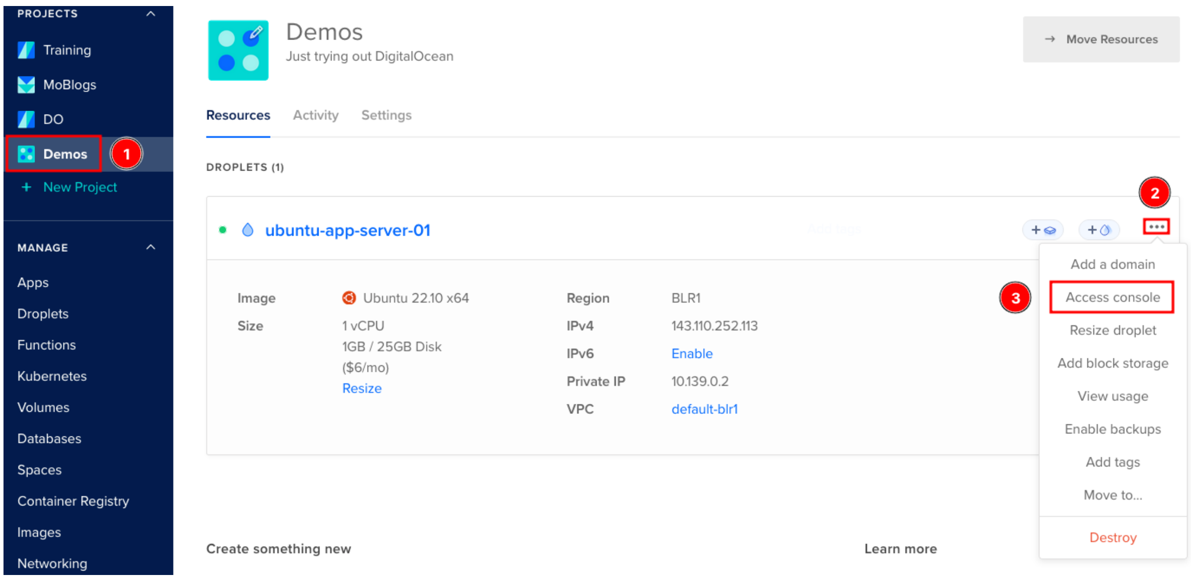
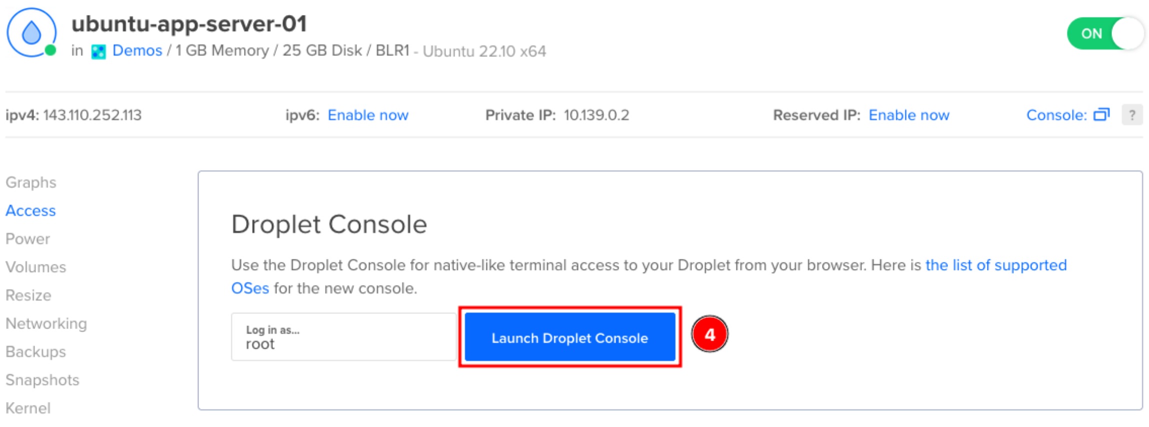
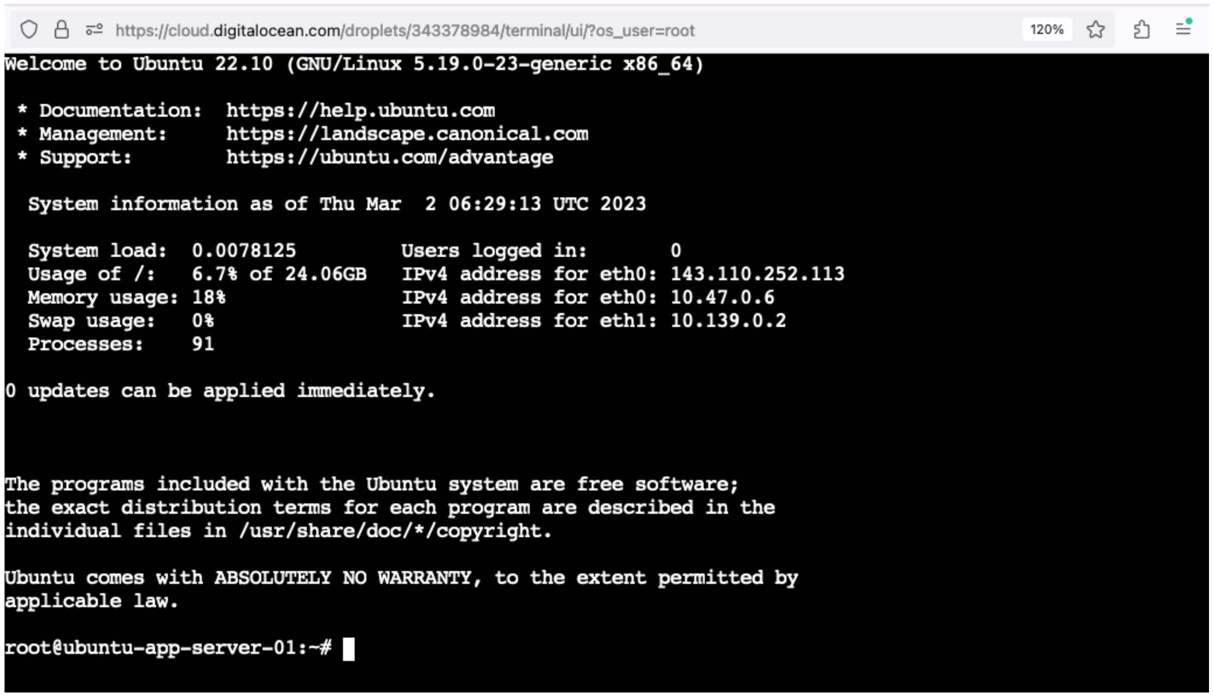


Image	Ubuntu 22.10 x64	Region	BLR1
Size	1 vCPU 1GB / 25GB Disk ($6/mo) Resize	IPv4	143.110.252.113
		IPv6	Enable
		Private IP	10.139.0.2
		VPC	default-blr1

2. On the next page, click **Launch Droplet Console** (4).



3. This will launch a new window, similar to the screenshot below, with access to the Droplet. If the window does not pop up, check if a pop-up blocker is enabled on your browser.


https://cloud.digitalocean.com/droplets/343378984/terminal/ui/?os_user=root
120%

```
Welcome to Ubuntu 22.10 (GNU/Linux 5.19.0-23-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Thu Mar  2 06:29:13 UTC 2023

  System load:  0.0078125         Users logged in:       0
  Usage of /:   6.7% of 24.06GB   IPv4 address for eth0: 143.110.252.113
  Memory usage: 18%               IPv4 address for eth0: 10.47.0.6
  Swap usage:   0%                IPv4 address for eth1: 10.139.0.2
  Processes:    91

0 updates can be applied immediately.


The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

root@ubuntu-app-server-01:~# 
```

All the steps in this task have been demonstrated using the built-in access console.

Setting up basic security

Creating a non-root user

A **root user (also known as a superuser)** has powerful privileges; it is not recommended to use root for day-to-day operations as one can make destructive changes accidentally. To provide context to those unfamiliar with this concept, root privileges on Linux are equivalent to the admin privileges on Windows.

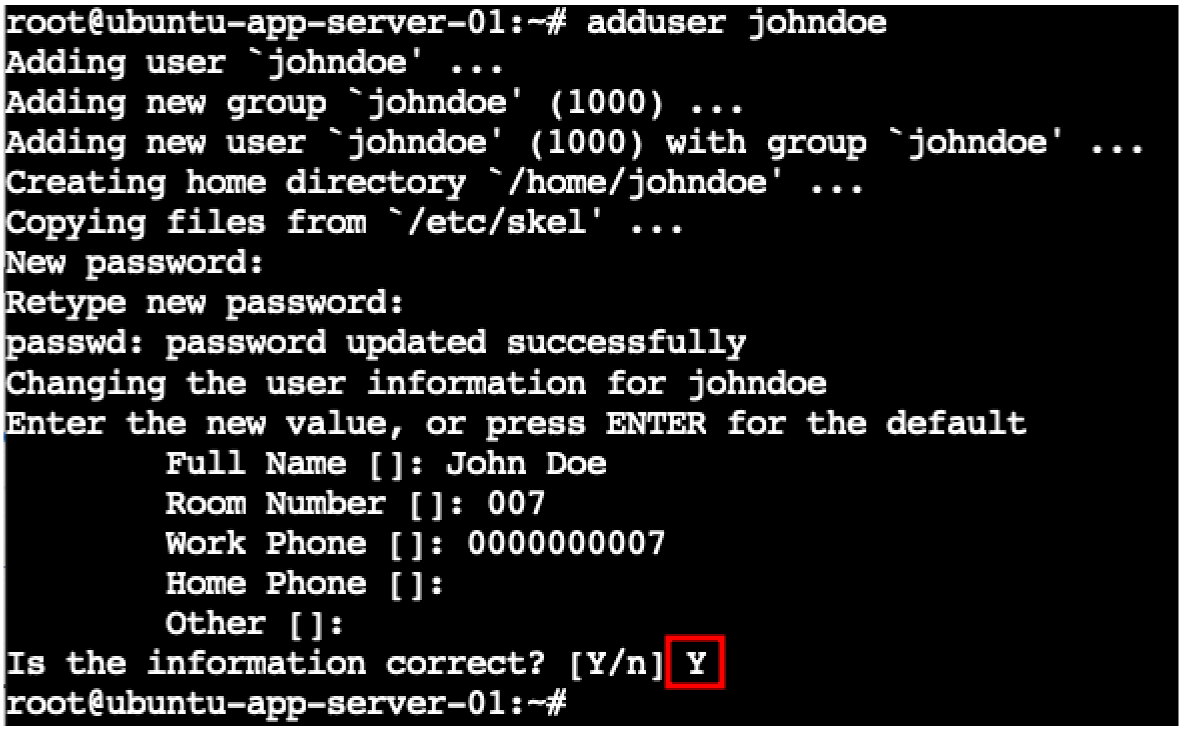
In this step, we will create a non-root user and provide the user with **sudo** privileges which may be needed to gain temporary root privileges for certain operations such as installing packages.

Follow the steps below in sequence. The example uses *johndoe* as an example username, but you are encouraged to replace it with a username of your choice.

1. To add a user, type **adduser [USERNAME]** and press enter.

   ```
   adduser johndoe
   ```

   This will prompt for a password for the user (provide a secure password and make sure you have **noted this down** somewhere safe) along with details such as Full Name, Phone number, etc which can be skipped by pressing enter. Finally, type **Y** to confirm the details.



```
root@ubuntu-app-server-01:~# adduser johndoe
Adding user `johndoe' ...
Adding new group `johndoe' (1000) ...
Adding new user `johndoe' (1000) with group `johndoe' ...
Creating home directory `/home/johndoe' ...
Copying files from `/etc/skel' ...
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for johndoe
Enter the new value, or press ENTER for the default
        Full Name []: John Doe
        Room Number []: 007
        Work Phone []: 0000000007
        Home Phone []:
        Other []:
Is the information correct? [Y/n] Y
root@ubuntu-app-server-01:~#
```

2. Next, we will add this user to the sudo group so that, when required, the user can perform operations that require elevated privileges. We do this using the **usermod** command. Type the following command:

```
usermod -aG sudo johndoe
```

3. Switch to the non-root user using the following command:

```
su - johndoe
```

   If you are using terminal or PuTTY, logout and login as the non-root user to complete the process.

Setting up a firewall

A firewall is a network security device that monitors incoming and outgoing network traffic and allows or blocks data packets based on a set of rules. **Ports** are virtual start or end points for network connections. Different port numbers are used for different protocols and types of traffic; for example, port 20 is for File Transfer Protocol (FTP) and port 22 is for SSH connections. Firewall rules can be created to permit or deny network traffic based on the ports the traffic is using.

Ubuntu operating systems use the UFW package to configure standard firewall settings. The firewall ensures only certain types of connections to specific services are allowed - in other words **only necessary ports are allowed**.

1. Start by checking the current firewall status. Type the following command:

```
sudo ufw status
```

   This should show the status as inactive.

2. Add the **OpenSSH** application to the firewall (UFW) allow list. This is to allow SSH connections to the Droplet. To do so run the following command:

```
sudo ufw allow OpenSSH
```

Take note:

Failure to add OpenSSH to the allow list before enabling the firewall may result in a Droplet that you are **locked out** of and can no longer SSH into**.** That said, the built-in console on Digital Ocean should continue to allow you to access your Droplet, and if you ever completely lose access to it, the DigitalOcean support team can often help you recover it.

3. Next, enable the firewall with the following command:

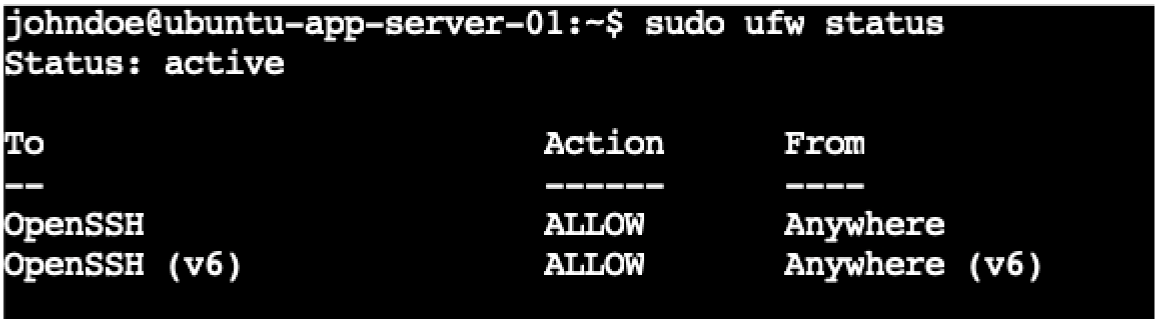
```
sudo ufw enable
```

   When prompted type **y** to proceed with the operation.

4. Check the firewall status with the command:

```
sudo ufw status
```

   You should receive an output similar to the screenshot below.



```
johndoe@ubuntu-app-server-01:~$ sudo ufw status
Status: active

To                         Action      From
--                         ------      ----
OpenSSH                    ALLOW       Anywhere
OpenSSH (v6)               ALLOW       Anywhere (v6)
```

Installing prerequisite packages

1. Update the package manager cache.

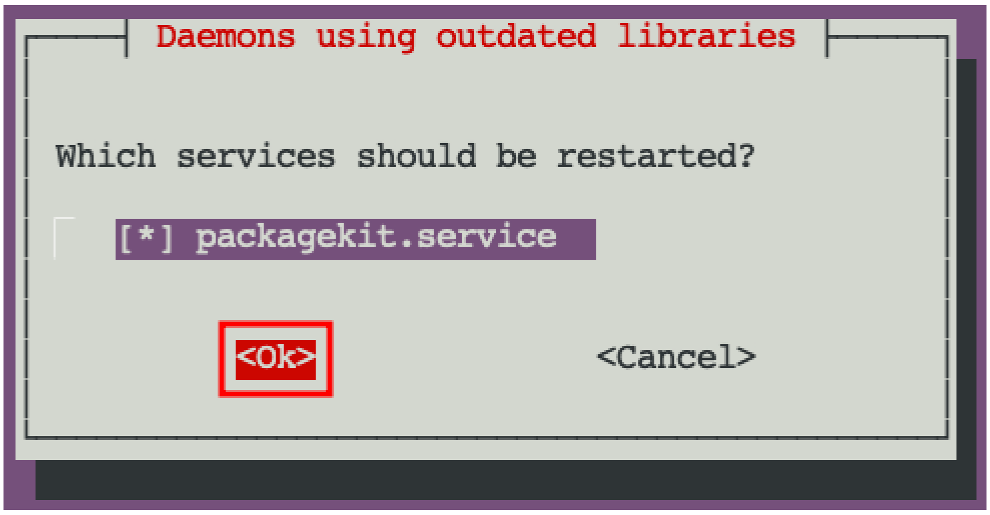
```
sudo apt update
```

   If prompted, enter the username's password.

2. Install the git and curl tools.

```
sudo apt install -y git curl
```

   You may be prompted with a service restart screen similar to the one below. Click **Ok**.



```
Daemons using outdated libraries

Which services should be restarted?

[*] packagekit.service

<Ok>                    <Cancel>
```

Installing Apache on the Droplet

Introduction to Apache HTTP Server

As discussed earlier, in order to serve HTML pages we need a web server that can process the requests. **Apache HTTP Server** is our choice of web server for this task.

The Apache HTTP Server project is part of the Apache Software Foundation. Apache HTTP Server is the most popular web server for serving some of the busiest websites out there.

Among the popular features are:

- Support for server-side programming languages such as PHP, Python, Perl, etc.
- HTTP/2 support
- Extensibility to the core functionality using dynamic modules
- SSL/TLS support, various authentication mechanisms
- URL rewriting
- Virtual Hosts, which allows a single installation of Apache to service multiple websites.
- Reverse proxy
- IPv6 compatibility
- Compression and decompression using **gzip**

In Ubuntu the package name is **apache2**, whereas in **RPM-based distributions** it's called **httpd**.

Installing Apache

Run the following command in order to install Apache.

```
sudo apt install -y apache2
```

Allow Apache through the firewall

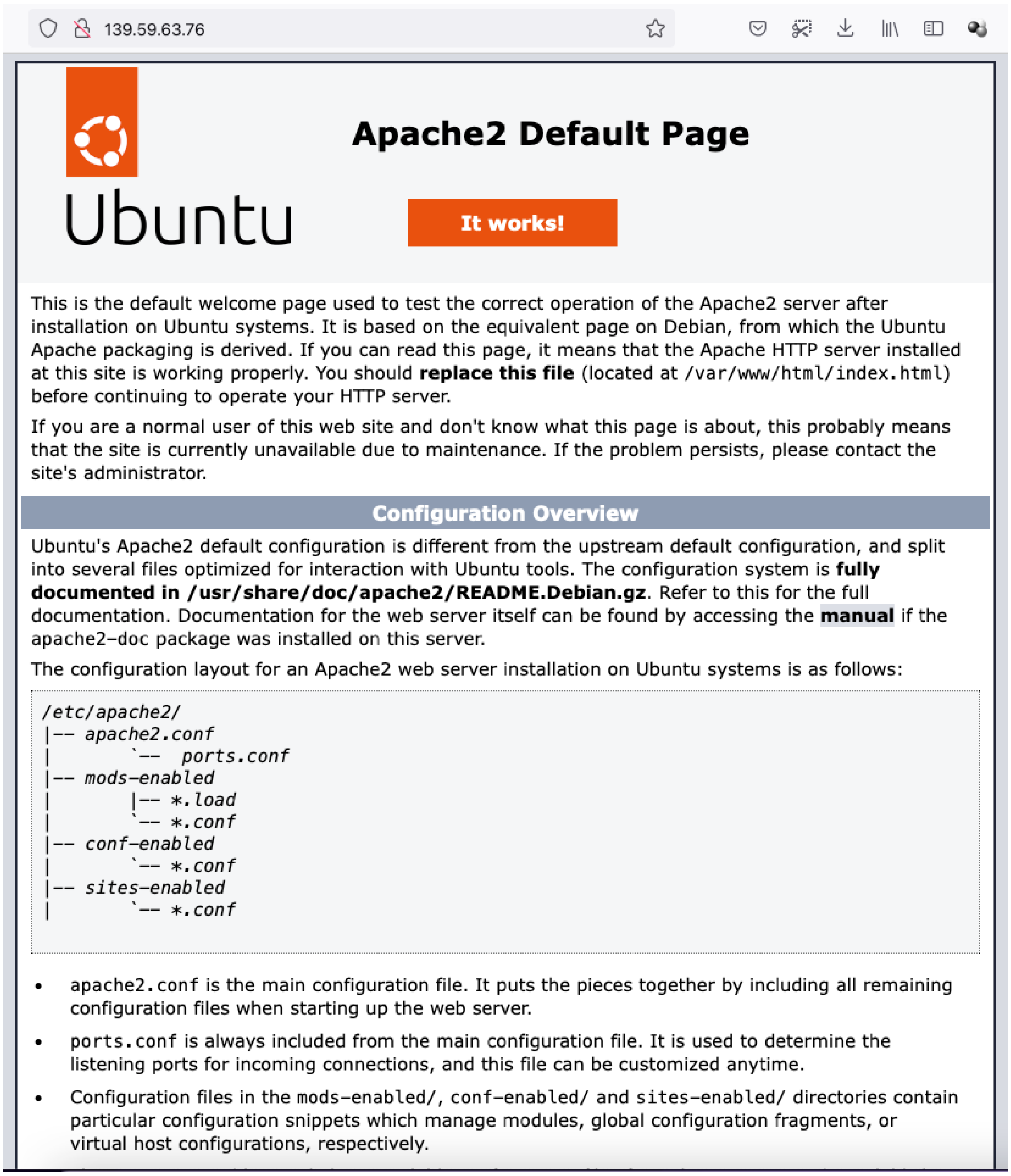
Now that Apache is installed we have to allow traffic to it. This is done by adding Apache to the allowed list in UFW (firewall).

```
sudo ufw allow in "Apache"
```

Check the firewall status to ensure Apache is added to the list by running the following command:

```
sudo ufw status
```

Finally, let's verify that the web server works. Open a browser and paste the IP address of the Droplet on the address bar. You should see the Apache2 Default Page.



This is the default welcome page used to test the correct operation of the Apache2 server after installation on Ubuntu systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is derived. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

Configuration Overview

Ubuntu's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Ubuntu tools. The configuration system is **fully documented in /usr/share/doc/apache2/README.Debian.gz**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the apache2-doc package was installed on this server.

The configuration layout for an Apache2 web server installation on Ubuntu systems is as follows:

```
/etc/apache2/
|-- apache2.conf
|        `--  ports.conf
|-- mods-enabled
|        |-- *.load
|        `-- *.conf
|-- conf-enabled
|        `-- *.conf
|-- sites-enabled
|        `-- *.conf
```

- `apache2.conf` is the main configuration file. It puts the pieces together by including all remaining configuration files when starting up the web server.
- `ports.conf` is always included from the main configuration file. It is used to determine the listening ports for incoming connections, and this file can be customized anytime.
- Configuration files in the `mods-enabled/`, `conf-enabled/` and `sites-enabled/` directories contain particular configuration snippets which manage modules, global configuration fragments, or virtual host configurations, respectively.

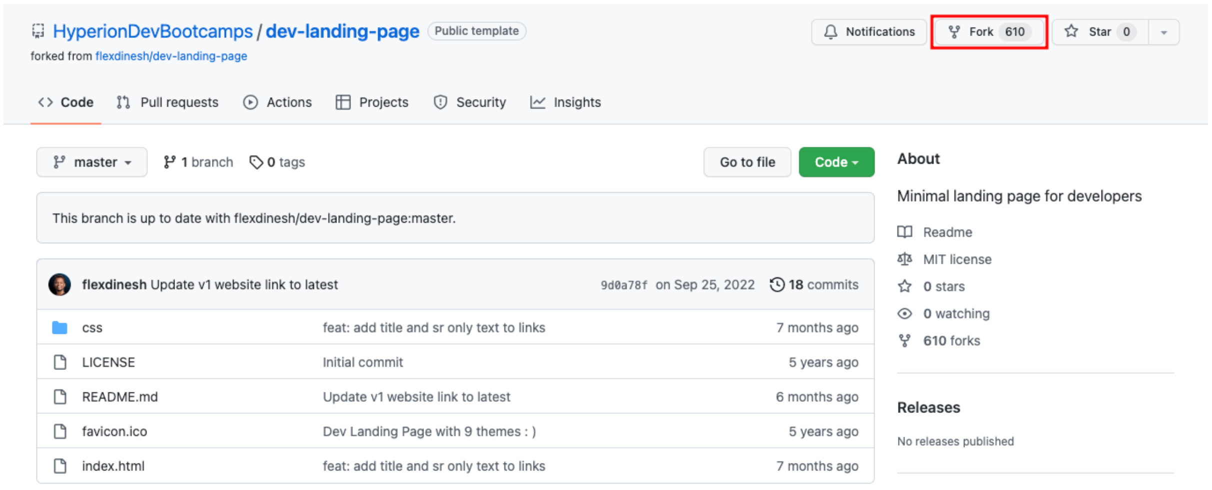
You have now completed the basic web server configuration steps and are ready to host your application!

HOSTING A WEB APP

For this step of the task, you may choose any website you have previously created, and use this as the site you are going to host.

Pre-requisites

- A working web server.
- Access to code hosted on GitHub. If you do not have a code of your own you can fork a copy of the sample code provided here which is published under the MIT licence for public use. To fork, navigate to https://github.com/HyperionDevBootcamps/dev-landing-page and click **Fork**. If prompted, sign in to GitHub.



This branch is up to date with flexdinesh/dev-landing-page:master.

flexdinesh Update v1 website link to latest 9d0a78f on Sep 25, 2022 18 commits

css	feat: add title and sr only text to links	7 months ago
LICENSE	Initial commit	5 years ago
README.md	Update v1 website link to latest	6 months ago
favicon.ico	Dev Landing Page with 9 themes :)	5 years ago
index.html	feat: add title and sr only text to links	7 months ago

Copying the code into the Droplet

In order to copy the code into the Droplet we will use Git. We will copy the code into the default root directory used by Apache which is /var/www/html.

Apache uses a **DocumentRoot** parameter in its configuration, which is the directory from where it serves files.

1. Run the following command to navigate to the **default Apache directory**.

```
cd /var/www/html
```

2. Remove the default index.html file with the command:

```
sudo rm index.html
```

3. In order to copy the code from GitHub into the folder run the following command. When you do the practical task using your own code, you will need to replace the GitHub link with your own link or use the forked link if you are using the sample code provided by us (note this is a single command and there is a dot at the end of the command):

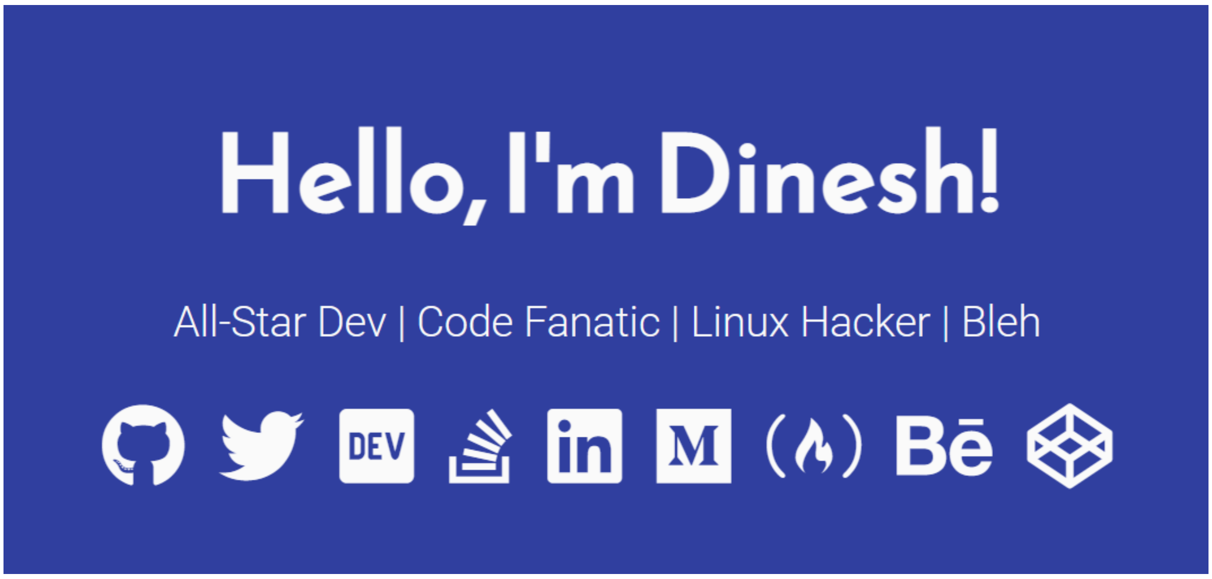
```
sudo git clone
https://github.com/HyperionDevBootcamps/dev-landing-page .
```

4. Verify files have been copied with the command:

```
ls -l
```

Verify the website works

Navigate to the browser and refresh the page. You should now see the website load. The example given shows the webpage from the public-access repo we used earlier, but remember that when you do the Compulsory Task later, you will be using your own previously created webpage and so what you see at this step will differ.



Hello, I'm Dinesh!

All-Star Dev | Code Fanatic | Linux Hacker | Bleh

MAINTAINING A WEB APP

Regular backups

Backups are absolutely crucial to any server, and the same is true for web servers hosting websites. Generally, backups are performed for the following files:

- Configuration files used by the web server
- Website code
- Databases used by websites
- Files uploaded to a website that are stored on the web server

When code is stored on GitHub, a backup of it from the web server is not necessary but it may come in handy in case someone deletes the **repo (repository)** completely.

Supplementary backups guide

In a professional environment, i.e. when you are actually using cloud hosting as part of your job, it will be important to remember to make backups. **We do NOT recommend you do this as part of this task**, because it is important to delete your hosted site after your task has been reviewed, to ensure that you don't exceed the free trial period or run out of free credit and get charged for the hosting. However, we have included a guide to creating backups using DigitalOcean spaces as a supplementary reading in your Dropbox folder along with this task, for general enrichment.

Updating your website

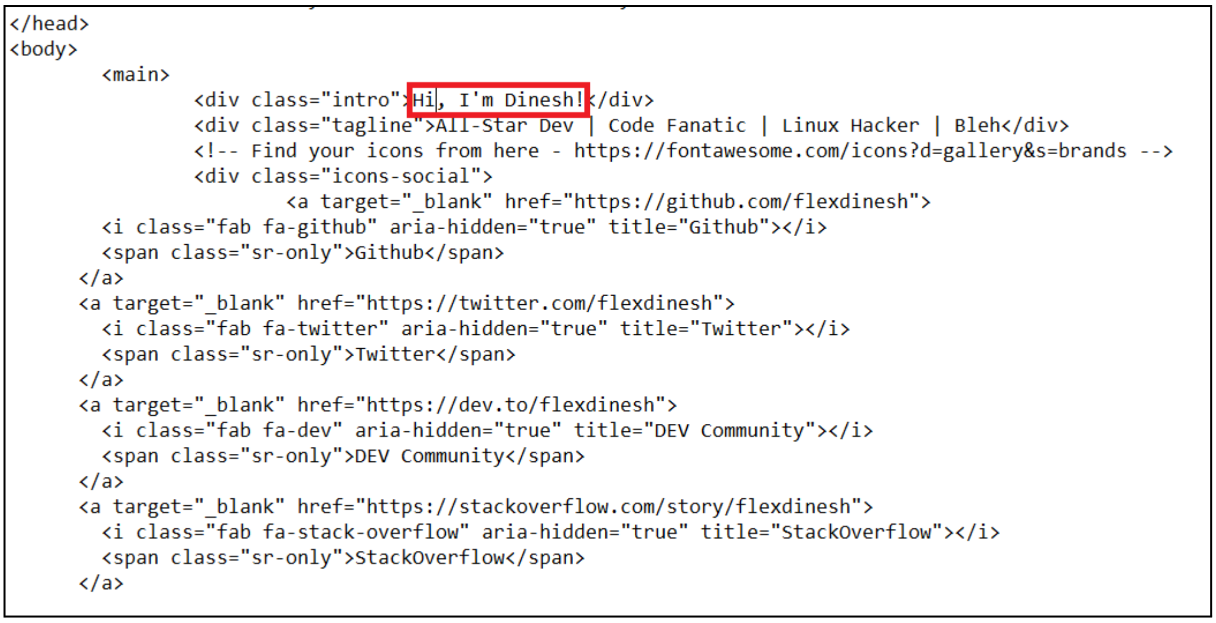
A regular maintenance task is to update the code hosted on the web server when it has been updated on GitHub, such as when new pages are added, content has been updated/modified, or the theme/template has been changed.

In this maintenance task you will learn how to update the code when it is stored in GitHub.

Using GitHub to update your website

To update the code in the web server, we can use Git. However, this will only work if Git was used to clone the code initially.

Let's first make a small change in our code (in GitHub). For example, in the **index.html** file pictured below, we go to the heading and change the word **Hello** to the word **Hi**, so the new text becomes **Hi, I'm Dinesh!** You can make a similar change to your own index.html file to test updating your code. Then commit and push the change.



```
</head>
<body>
        <main>
                <div class="intro">Hi, I'm Dinesh!</div>
                <div class="tagline">All-Star Dev | Code Fanatic | Linux Hacker | Bleh</div>
                <!-- Find your icons from here - https://fontawesome.com/icons?d=gallery&s=brands -->
                <div class="icons-social">
                        <a target="_blank" href="https://github.com/flexdinesh">
        <i class="fab fa-github" aria-hidden="true" title="Github"></i>
        <span class="sr-only">Github</span>
      </a>
      <a target="_blank" href="https://twitter.com/flexdinesh">
        <i class="fab fa-twitter" aria-hidden="true" title="Twitter"></i>
        <span class="sr-only">Twitter</span>
      </a>
      <a target="_blank" href="https://dev.to/flexdinesh">
        <i class="fab fa-dev" aria-hidden="true" title="DEV Community"></i>
        <span class="sr-only">DEV Community</span>
      </a>
      <a target="_blank" href="https://stackoverflow.com/story/flexdinesh">
        <i class="fab fa-stack-overflow" aria-hidden="true" title="StackOverflow"></i>
        <span class="sr-only">StackOverflow</span>
      </a>
```

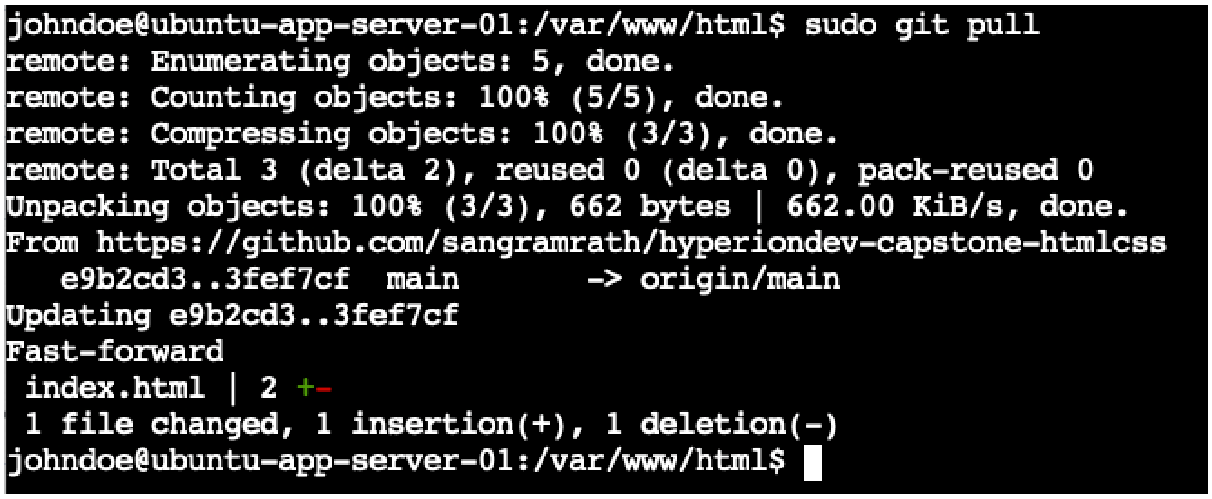
Next, we navigate to the folder where the code was initially cloned, which in our case was /var/www/html.

```
cd /var/www/html
```

Run the git pull command to update.

```
sudo git pull
```

The output should look similar to the screenshot below.



```
johndoe@ubuntu-app-server-01:/var/www/html$ sudo git pull
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 3 (delta 2), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), 662 bytes | 662.00 KiB/s, done.
From https://github.com/sangramrath/hyperiondev-capstone-htmlcss
   e9b2cd3..3fef7cf  main       -> origin/main
Updating e9b2cd3..3fef7cf
Fast-forward
 index.html | 2 +-
 1 file changed, 1 insertion(+), 1 deletion(-)
johndoe@ubuntu-app-server-01:/var/www/html$
```

You can review the effect of the changes by refreshing the website.

Performance

Page speed

One of the popular performance metrics for websites is page load speed. This is the time taken between the request for a page and the point when the contents of the page are completely loaded.

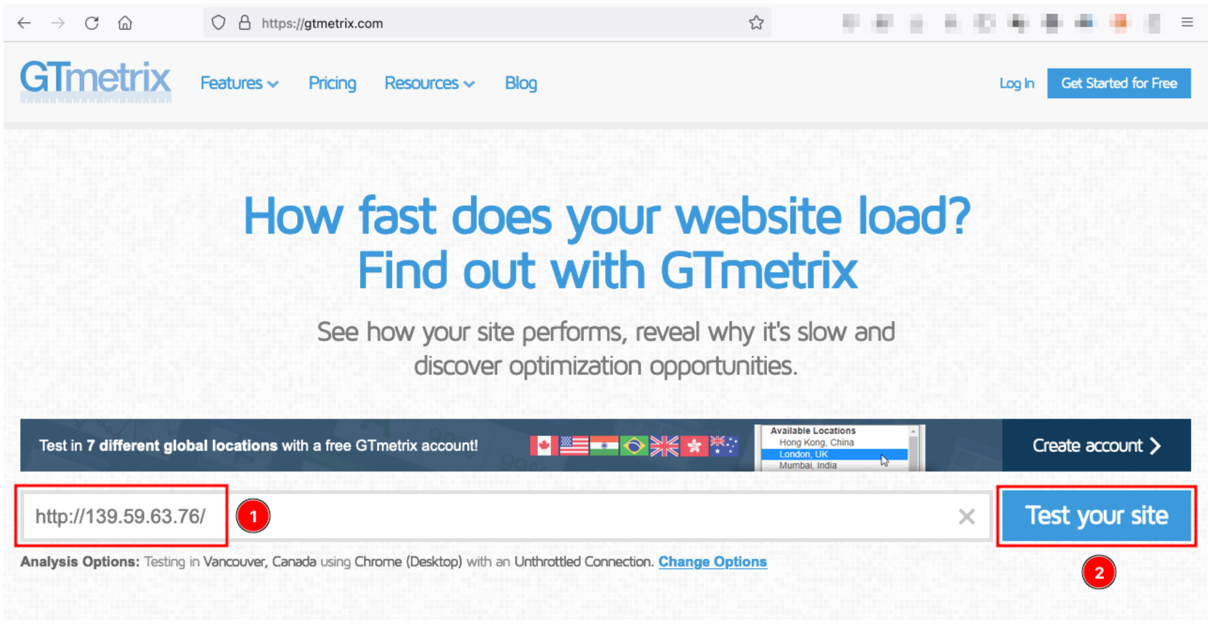
There are many tools (online) that can help you check your website performance using a scoring model, including the page speed score. Some of the popular tools are:

- GTmetrix
- Pingdom
- Google PageSpeed tools

These tools also provide recommendations on how to improve your website's performance.

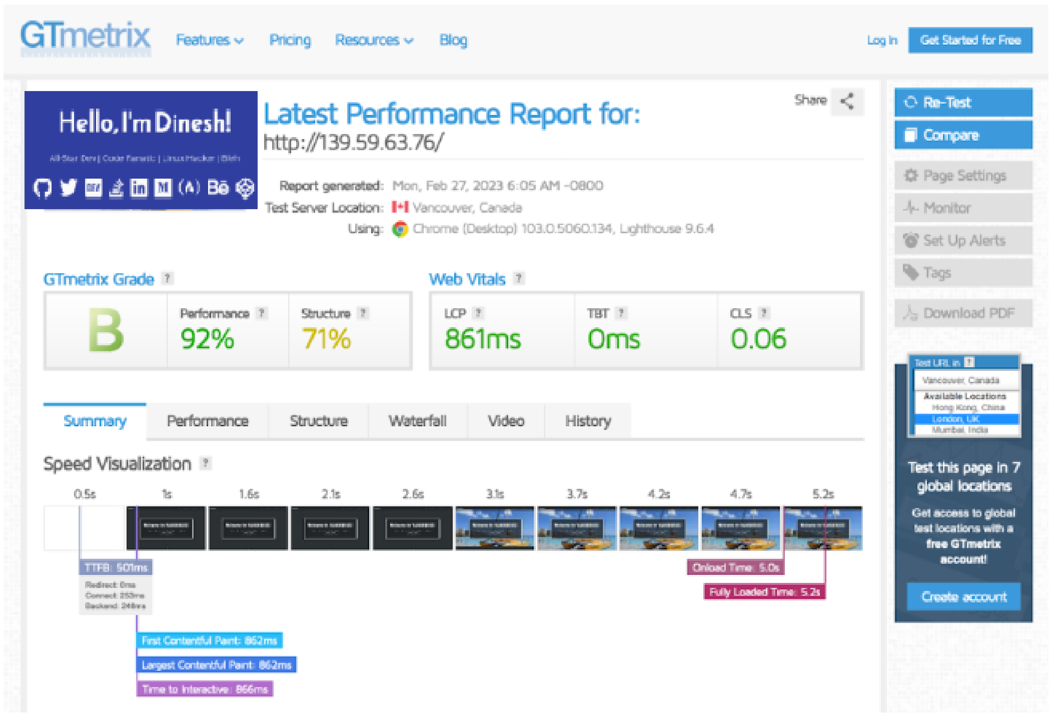
In this task, we will use the GTmetrix tool to check our website performance.

1. Open GTmetrix at **https://gtmetrix.com/**.
2. Enter the IP address of the Droplet (or the domain name) in the **Enter URL to Analyze** field box (1) and click **Test your site** (2).



Analysis Options: Testing in Vancouver, Canada using Chrome (Desktop) with an Unthrottled Connection. Change Options

3. In a few minutes you should see a report similar to the screenshots below.



Latest Performance Report for:
http://139.59.63.76/

Report generated: Mon, Feb 27, 2023 6:05 AM -0800
Test Server Location: Vancouver, Canada
Using: Chrome (Desktop) 103.0.5060.134, Lighthouse 9.6.4

GTmetrix Grade: B — Performance 92% — Structure 71%

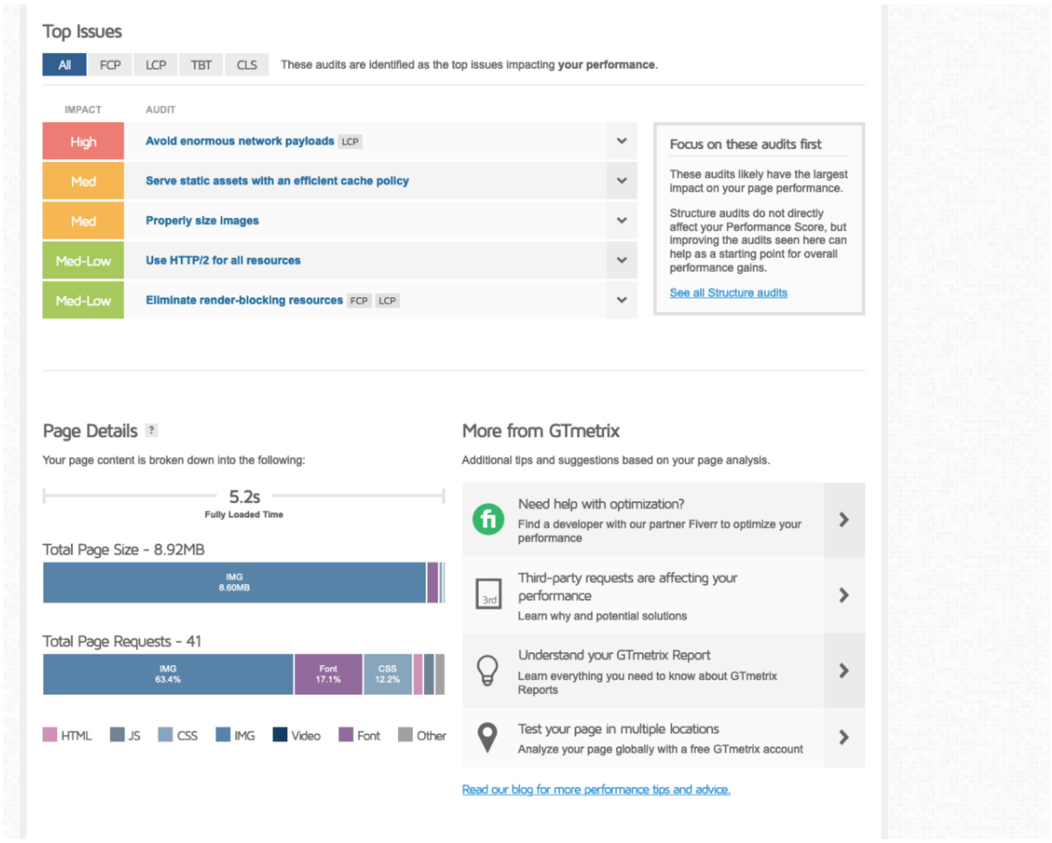
Web Vitals: LCP 861ms — TBT 0ms — CLS 0.06

Summary | Performance | Structure | Waterfall | Video | History

Speed Visualization

0.5s 1s 1.6s 2.1s 2.6s 3.1s 3.7s 4.2s 4.7s 5.2s

TTFB: 501ms (Redirect: 0ms, Connect: 253ms, Backend: 248ms)
First Contentful Paint: 862ms
Largest Contentful Paint: 862ms
Time to Interactive: 866ms
Onload Time: 5.0s
Fully Loaded Time: 5.2s



Top Issues

These audits are identified as the top issues impacting **your performance**.

IMPACT	AUDIT
High	Avoid enormous network payloads LCP
Med	Serve static assets with an efficient cache policy
Med	Properly size images
Med-Low	Use HTTP/2 for all resources
Med-Low	Eliminate render-blocking resources FCP LCP

Focus on these audits first

These audits likely have the largest impact on your page performance.

Structure audits do not directly affect your Performance Score, but improving the audits seen here can help as a starting point for overall performance gains.

Page Details

Your page content is broken down into the following:

5.2s Fully Loaded Time

Total Page Size - 8.92MB (IMG 8.60MB)

Total Page Requests - 41 (IMG 63.4%, Font 17.1%, CSS 12.2%)

HTML, JS, CSS, IMG, Video, Font, Other

More from GTmetrix

Additional tips and suggestions based on your page analysis.

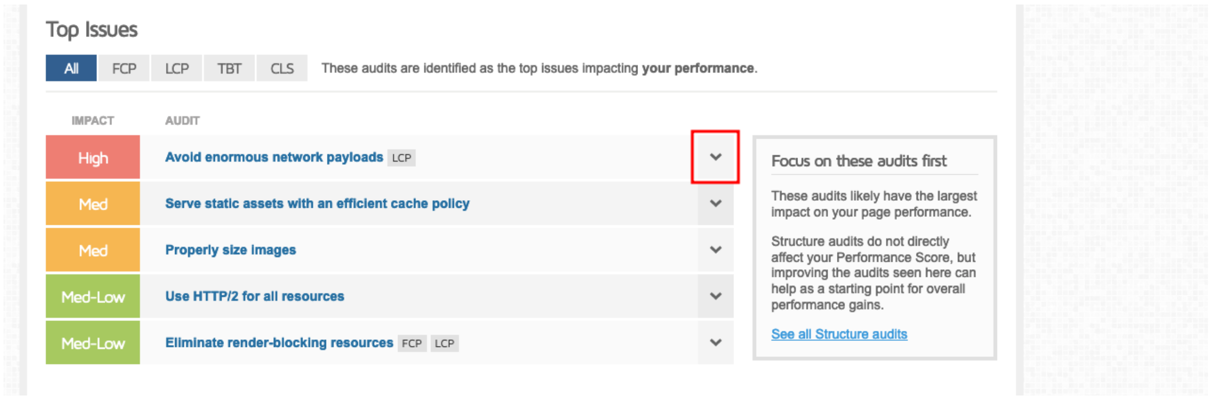
Need help with optimization? Find a developer with our partner Fiverr to optimize your performance

Third-party requests are affecting your performance. Learn why and potential solutions

Understand your GTmetrix Report. Learn everything you need to know about GTmetrix Reports

Test your page in multiple locations. Analyze your page globally with a free GTmetrix account

4. Feel free to look at the top issues and see recommendations on how to improve the scores. For example, under top issues, look for an issue with high impact and expand the issue to view the details.



Top Issues

All | FCP | LCP | TBT | CLS — These audits are identified as the top issues impacting **your performance**.

IMPACT	AUDIT
High	**Avoid enormous network payloads** LCP
Med	**Serve static assets with an efficient cache policy**
Med	**Properly size images**
Med-Low	**Use HTTP/2 for all resources**
Med-Low	**Eliminate render-blocking resources** FCP LCP

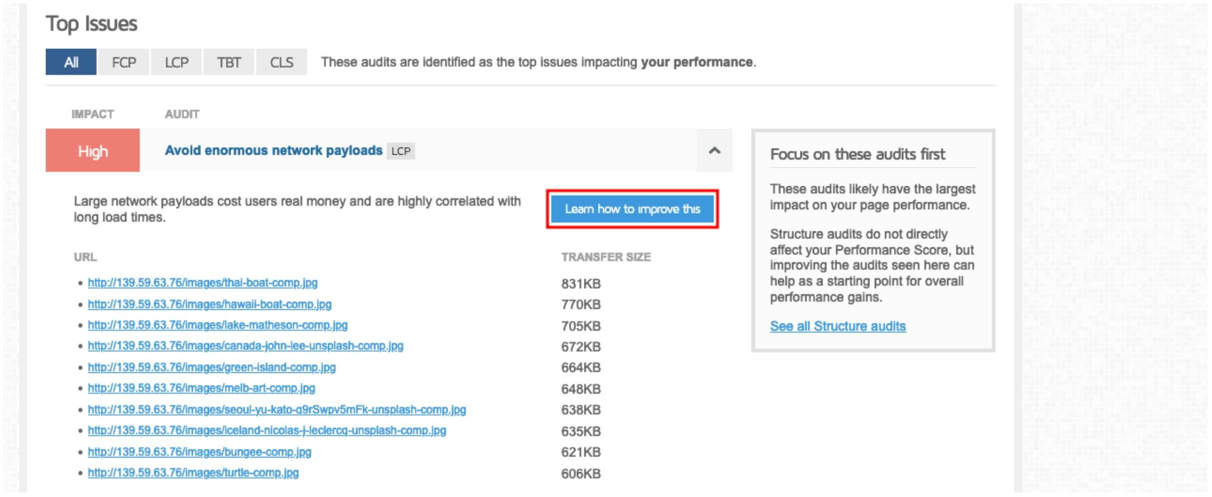
Focus on these audits first

These audits likely have the largest impact on your page performance.

Structure audits do not directly affect your Performance Score, but improving the audits seen here can help as a starting point for overall performance gains.

See all Structure audits

5. Now take a look at the URLs that are causing the issue along with a button that directs you to a link where you can learn how to improve this issue. (Note that your results will obviously differ from the screenshot as you're using different code.)



Top Issues

All | FCP | LCP | TBT | CLS — These audits are identified as the top issues impacting **your performance**.

IMPACT	AUDIT
High	**Avoid enormous network payloads** LCP

Large network payloads cost users real money and are highly correlated with long load times.

URL	TRANSFER SIZE
http://139.59.63.76/images/thai-boat-comp.jpg	831KB
http://139.59.63.76/images/hawaii-boat-comp.jpg	770KB
http://139.59.63.76/images/lake-matheson-comp.jpg	705KB
http://139.59.63.76/images/canada-john-lee-unsplash-comp.jpg	672KB
http://139.59.63.76/images/green-island-comp.jpg	664KB
http://139.59.63.76/images/melb-art-comp.jpg	648KB
http://139.59.63.76/images/seoul-yu-kato-q9rSwpv5mFk-unsplash-comp.jpg	638KB
http://139.59.63.76/images/iceland-nicolas-j-leclercq-unsplash-comp.jpg	635KB
http://139.59.63.76/images/bungee-comp.jpg	621KB
http://139.59.63.76/images/turtle-comp.jpg	606KB

Focus on these audits first

These audits likely have the largest impact on your page performance.

Structure audits do not directly affect your Performance Score, but improving the audits seen here can help as a starting point for overall performance gains.

See all Structure audits

Fixing issues like this is not a part of this task and diving further into this is out of the current scope, but forms an interesting area for self-study.

Security updates

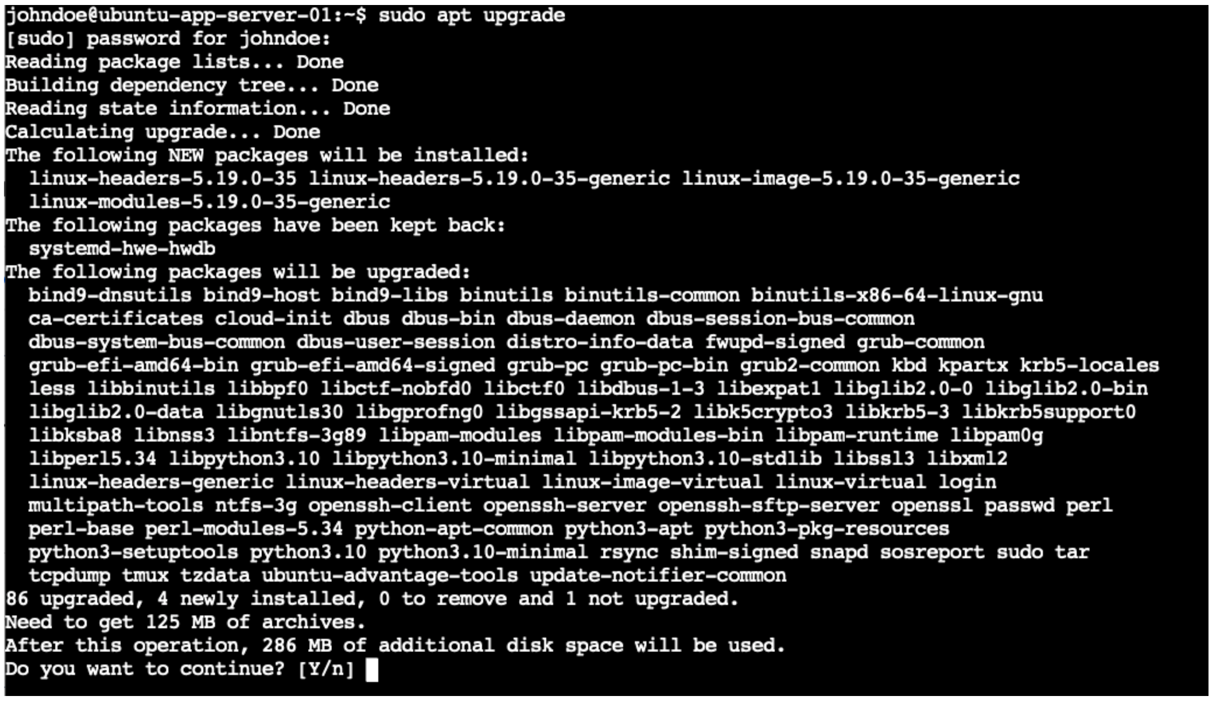
Another common maintenance task that is performed regularly on web servers is updating the operating system and its packages. Sometimes we update specific packages only, based on security advisories that are released from time to time.

It is highly recommended that you test the updates on a dev/staging server before applying them to production websites. This is very important for dynamic websites that use server-side technologies where the code has dependencies on versions.

Start by updating the package manager cache. Run the upgrade command to see which packages can be upgraded.

```
sudo apt upgrade
```

Notice that we are not using the **-y** switch, as we want to review the packages before upgrading. You will see an output similar to the screenshot below:



```
johndoe@ubuntu-app-server-01:~$ sudo apt upgrade
[sudo] password for johndoe:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Calculating upgrade... Done
The following NEW packages will be installed:
  linux-headers-5.19.0-35 linux-headers-5.19.0-35-generic linux-image-5.19.0-35-generic
  linux-modules-5.19.0-35-generic
The following packages have been kept back:
  systemd-hwe-hwdb
The following packages will be upgraded:
  bind9-dnsutils bind9-host bind9-libs binutils binutils-common binutils-x86-64-linux-gnu
  ca-certificates cloud-init dbus dbus-bin dbus-daemon dbus-session-bus-common
  dbus-system-bus-common dbus-user-session distro-info-data fwupd-signed grub-common
  grub-efi-amd64-bin grub-efi-amd64-signed grub-pc grub-pc-bin grub2-common kbd kpartx krb5-locales
  less libbinutils libbpf0 libctf-nobfd0 libctf0 libdbus-1-3 libexpat1 libglib2.0-0 libglib2.0-bin
  libglib2.0-data libgnutls30 libgprofng0 libgssapi-krb5-2 libk5crypto3 libkrb5-3 libkrb5support0
  libksba8 libnss3 libntfs-3g89 libpam-modules libpam-modules-bin libpam-runtime libpam0g
  libperl5.34 libpython3.10 libpython3.10-minimal libpython3.10-stdlib libssl3 libxml2
  linux-headers-generic linux-headers-virtual linux-image-virtual linux-virtual login
  multipath-tools ntfs-3g openssh-client openssh-server openssh-sftp-server openssl passwd perl
  perl-base perl-modules-5.34 python-apt-common python3-apt python3-pkg-resources
  python3-setuptools python3.10 python3.10-minimal rsync shim-signed snapd sosreport sudo tar
  tcpdump tmux tzdata ubuntu-advantage-tools update-notifier-common
86 upgraded, 4 newly installed, 0 to remove and 1 not upgraded.
Need to get 125 MB of archives.
After this operation, 286 MB of additional disk space will be used.
Do you want to continue? [Y/n]
```

You can review the packages that will be upgraded and then type **Y** and press Enter if you decide to upgrade all packages. For this task, go ahead and do the same.

You may also upgrade specific packages only. For example:

```
sudo apt upgrade openssh-server -y
```

Since this is a single package, we can add **-y** to avoid the prompt.

ENRICHMENT: SECURING A WEB APP WITH SSL/TLS

Take note:

The steps below are for enrichment - they are **optional for this task** because they **require a domain** to execute successfully, and this incurs a cost**.** If you do want to purchase a domain, they can be purchased from providers such as name.com, Google Domains, etc.

What is an SSL/TLS certificate?

SSL/TLS are security protocols used to build and serve secure websites. SSL stands for **Secure Socket Layer** and TLS stands for **Transport Security Layer**. These protocols provide features such as privacy, data integrity, authentication, etc. TLS is an evolution of SSL and is the recommended protocol.

An SSL/TLS certificate contains the public key of the website, its identity information such as the owner, and some other related details.

Need for SSL/TLS

An SSL/TLS certificate allows a website to be served using HTTPS instead of HTTP, which is insecure.

SSL/TLS certificate providers

SSL/TLS certificates can be purchased from many hosting providers and Certificate Authorities.
There are also some free, non-profit and open certificate authorities such as **Let's Encrypt** that provide certificates (X.509) for TLS for free.

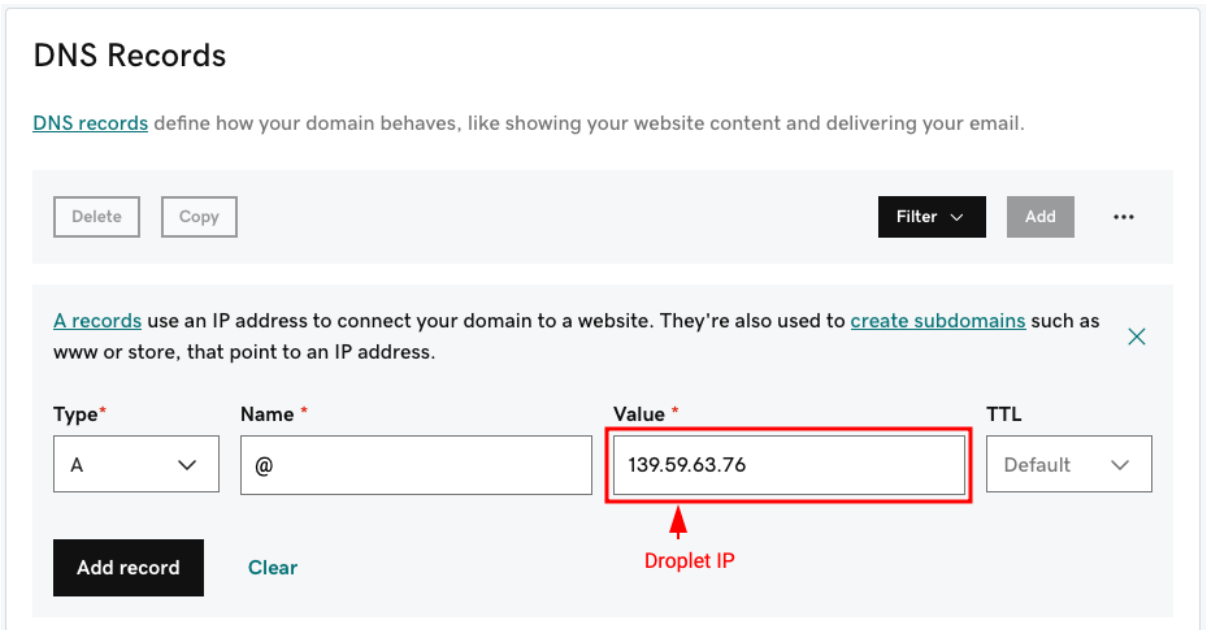
There are many ways to configure a website for SSL/TLS. The method mentioned below is just one of them.

Pre-requisites

- A working web server
- A domain or subdomain.
- A Droplet IP mapped to the domain (subdomain) through an A record. An example of this using GoDaddy is shown in the screenshot below:

Take note:

It is recommended to use a **Reserved IP** for the Droplet when mapping it to a domain. Reserved IPs remain even when a Droplet is destroyed and can be easily assigned to another Droplet or used for failover scenarios. This is an additional cost. Read more about it **here**.


DNS Records

DNS records define how your domain behaves, like showing your website content and delivering your email.

A records use an IP address to connect your domain to a website. They're also used to create subdomains such as www or store, that point to an IP address.

Type*	Name *	Value *	TTL
A	@	139.59.63.76	Default

Obtain & Install a free TLS certificate for Apache

For obtaining, installing, and reissuing the free certificate automatically we will use a tool called **certbot**. In this example, we are using a subdomain called **wanderlust.wavecafe.ml**. When running the commands use your own domain/subdomain.

1. Start by installing certbot. We use a tool called snap to install certbot.

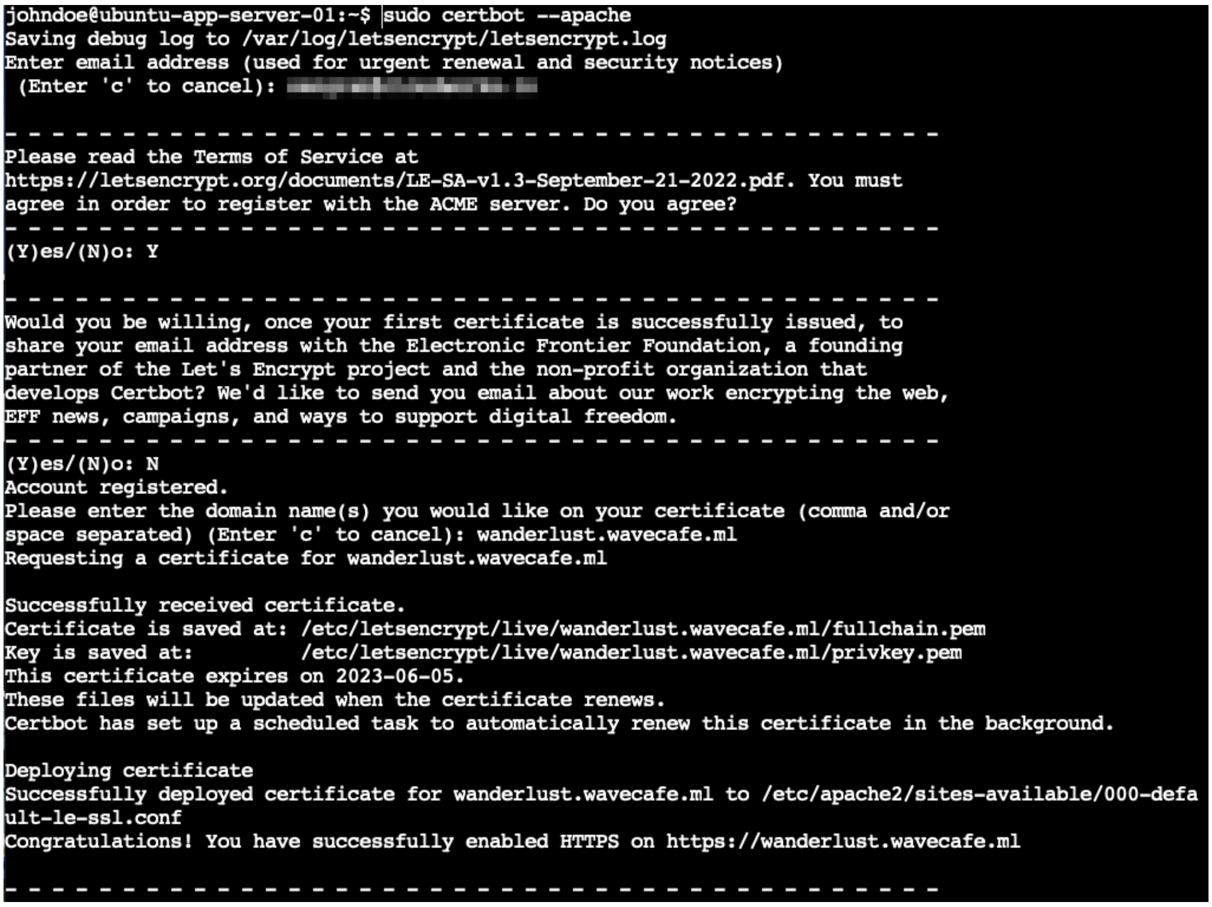
```
sudo snap install --classic certbot
```

2. Obtain and install a free certificate for your domain from **Let's Encrypt** by running the command below. A single command can do both tasks.

```
sudo certbot --apache
```

It will prompt for the following details (read carefully through the screenshot provided):

   a. Email address
   b. Agreement to Terms of Service (you must type **Y**)
   c. Willingness to share email with EFF (you can choose **No** for this)
   d. Domain name(s)



```
johndoe@ubuntu-app-server-01:~$ sudo certbot --apache
Saving debug log to /var/log/letsencrypt/letsencrypt.log
Enter email address (used for urgent renewal and security notices)
 (Enter 'c' to cancel): [redacted]

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
Please read the Terms of Service at
https://letsencrypt.org/documents/LE-SA-v1.3-September-21-2022.pdf. You must
agree in order to register with the ACME server. Do you agree?
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
(Y)es/(N)o: Y

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
Would you be willing, once your first certificate is successfully issued, to
share your email address with the Electronic Frontier Foundation, a founding
partner of the Let's Encrypt project and the non-profit organization that
develops Certbot? We'd like to send you email about our work encrypting the web,
EFF news, campaigns, and ways to support digital freedom.
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
(Y)es/(N)o: N
Account registered.
Please enter the domain name(s) you would like on your certificate (comma and/or
space separated) (Enter 'c' to cancel): wanderlust.wavecafe.ml
Requesting a certificate for wanderlust.wavecafe.ml

Successfully received certificate.
Certificate is saved at: /etc/letsencrypt/live/wanderlust.wavecafe.ml/fullchain.pem
Key is saved at:         /etc/letsencrypt/live/wanderlust.wavecafe.ml/privkey.pem
This certificate expires on 2023-06-05.
These files will be updated when the certificate renews.
Certbot has set up a scheduled task to automatically renew this certificate in the background.

Deploying certificate
Successfully deployed certificate for wanderlust.wavecafe.ml to /etc/apache2/sites-available/000-defa
ult-le-ssl.conf
Congratulations! You have successfully enabled HTTPS on https://wanderlust.wavecafe.ml

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
```

3. Open the HTTPS port for Apache in your firewall with the command:

```
sudo ufw allow in "Apache Secure"
```

4. You should now be able to access your website using HTTPS. When accessed through HTTP, it will redirect to HTTPS automatically.

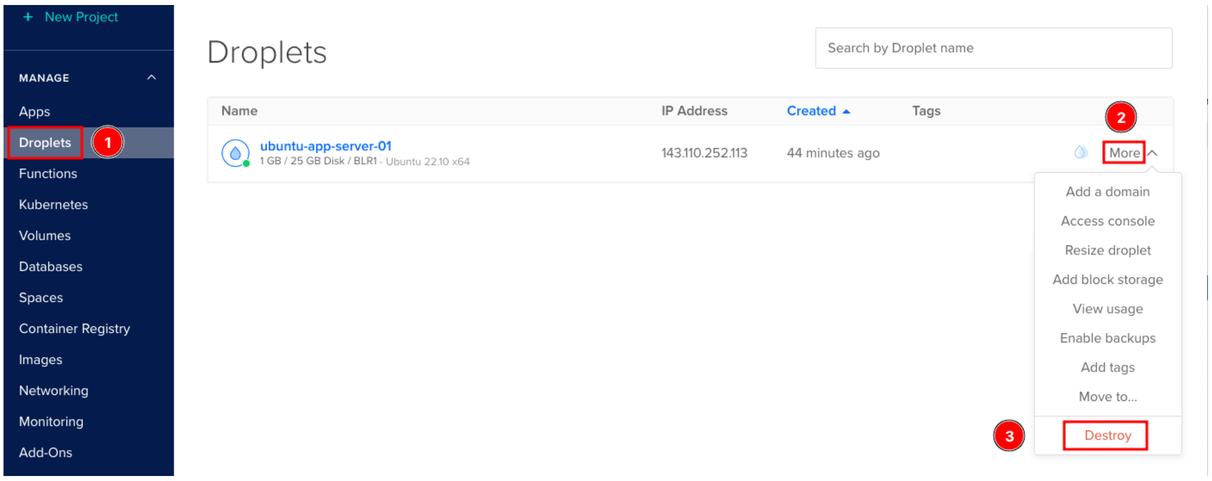
CLEANING UP RESOURCES

In order to avoid getting billed for resources after the free trial period ends or your GitHub Student development pack free credits are exhausted, **it is important that you delete all the resources created in this task after your compulsory task has been reviewed.**.

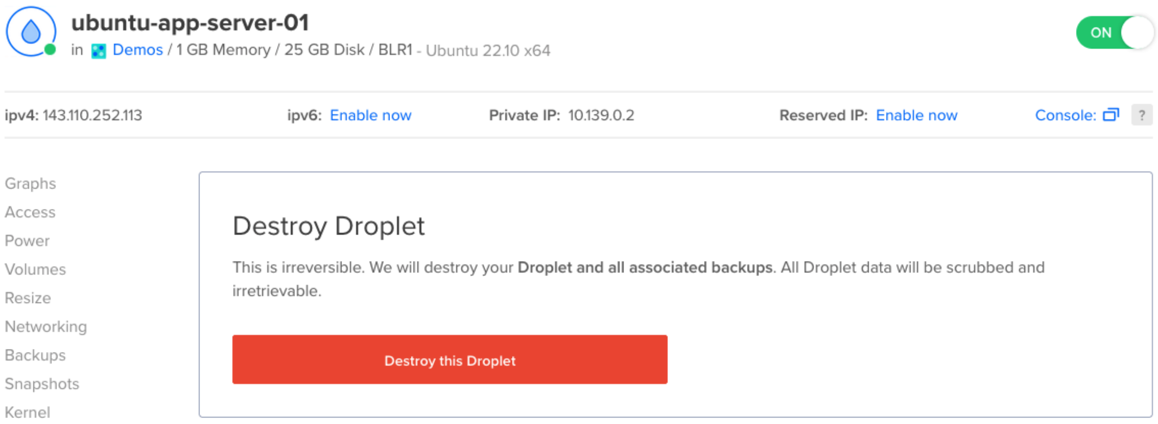
Deleting Droplets

Follow the instructions below to delete all Droplets created during the task (but remember to **only** do this after your task has been reviewed and marked).

1. Navigate to the Droplets dashboard by clicking on Droplets under Manage on the left (1), click More (2) next to the Droplet(s), and select Destroy (3).



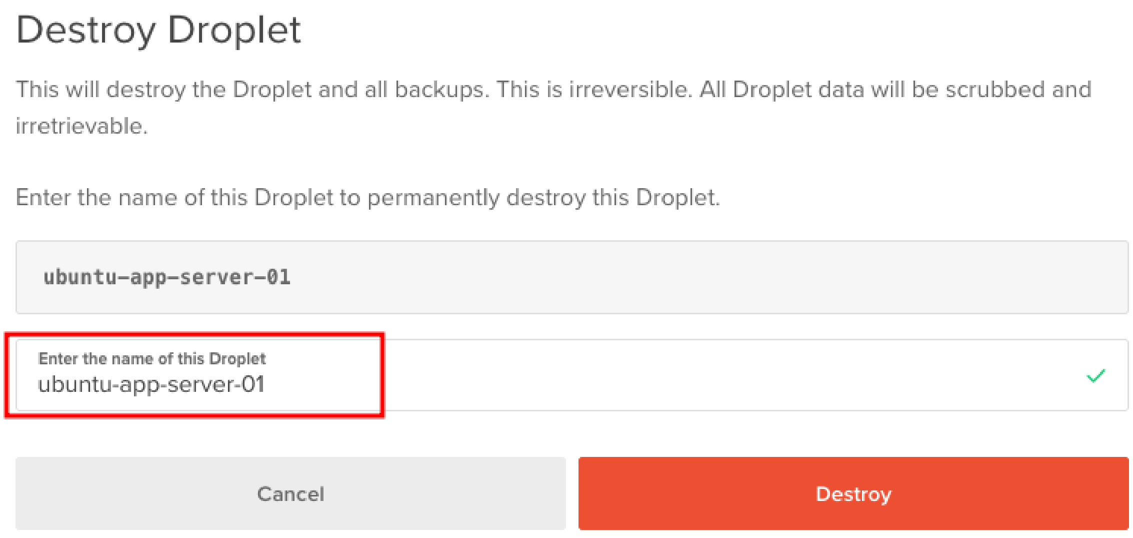
2. Click “Destroy this Droplet” again to confirm that this is what you want to do.



Destroy Droplet

This is irreversible. We will destroy your **Droplet and all associated backups**. All Droplet data will be scrubbed and irretrievable.

3. On the pop-up screen, confirm by typing the Droplet name and then click Destroy again.



Destroy Droplet

This will destroy the Droplet and all backups. This is irreversible. All Droplet data will be scrubbed and irretrievable.

Enter the name of this Droplet to permanently destroy this Droplet.

```
ubuntu-app-server-01
```

Enter the name of this Droplet
ubuntu-app-server-01

Compulsory Task 1

This compulsory task will test your understanding of the configuration of a web server, in this case, Apache. You will essentially work through what you have learned in the body of this document to serve a website from your Droplet.

The code for the website is located at:
https://github.com/HyperionDevBootcamps/hyperiondev-capstone-htmlcss

Follow these basic steps, consulting the body of the task for details, to host a website on your DigitalOcean Droplet:

- Start by cloning the code to your own GitHub account
- Next, create a Project with a new Droplet in it.
- Access your Droplet using SSH - you can either use the terminal option or the DigitalOcean access console option explained previously.
- Set up a user in Ubuntu on your Droplet, add the new user to the sudo group and login to the new user (ensure you're not using the Root user)
- Change to the new user you created. You can do this by using the "su" command. (for example:. su johndoe)
- Set up Apache and ensure it's running on your Droplet.
- Navigate to the following directory on Ubuntu: /var/www/html/
  You can do so by using the following command: cd /var/www/html/
- Ensure the following directory is displayed behind the dollar sign to ensure you're in the correct directory:
  yourusername@dropletIP:/var/www/html/.
- Remove the default index.html file from the html directory.
- Clone the code into the directory, by using the git clone command. Ensure a space and period is added at the end of the repository url, to clone the contents of the repository into the html directory, without creating a new directory.
- Verify the files have been cloned into the directory.
- Ensure your website is running and publicly visible.
- In your Dropbox, create a new text file called **address.txt**.

- Paste the IP address of your website into the text file and save changes for review.

After adding the text file to your Dropbox, ensure your Dropbox is synced before requesting a review.

ONCE YOUR TASK HAS BEEN REVIEWED, REMEMBER TO FOLLOW THE STEPS TO DELETE THE DROPLET.

Completed the task(s)?

Ask an expert code reviewer to review your work!

Review work

Rate us

Share your thoughts

HyperionDev strives to provide internationally-excellent course content that helps you achieve your learning outcomes.

Think that the content of this task, or this course as a whole, can be improved or think we've done a good job?

Click here to share your thoughts anonymously.