
MLH localhost
Workshop
App Deployment & Security with DigitalOcean
MLH localhost
DigitalOcean

Our Mission is to Empower Hackers.

65,000+
HACKERS

12,000+
PROJECTS CREATED

400+
CITIES

We hope you learn something awesome today!
Find more resources: http://mlh.io/

Table of Contents

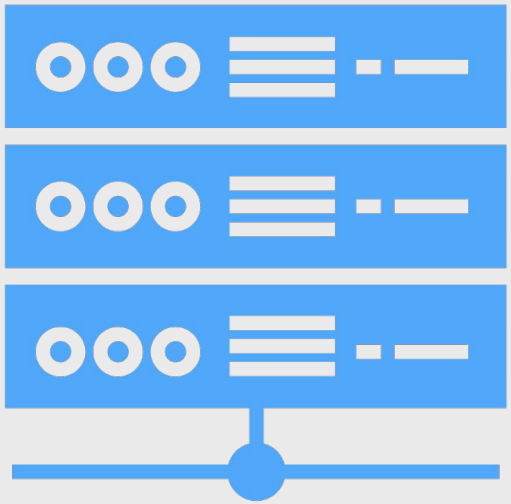
What will you learn today?

1. How to deploy an application to a DigitalOcean Droplet
2. How to make your application more secure and available
3. What DevOps is, and why it's so important to software development

What's the Cloud?

Every website or application you visit over the internet is stored somewhere: **a server**.

Your university will have a server room; businesses can create their own servers; you can even create local servers on your own devices with tools like **Express** and **Flask**.



What's the Cloud?

Some companies offer servers for hosting or processing power.

Due to the increase in internet connection speeds, you can access these servers in milliseconds, anywhere in the world.

This is known as **Cloud Computing**.

What's the Cloud?

DigitalOcean is a company that provides cloud infrastructure.

It allows us to create **Droplets** - virtual servers that you can use to host a web application.

In this workshop, we're going to create a web application, and deploy it on a DigitalOcean Droplet.

Table of Contents

Operation BeachClean!

We're going to deploy an application to DigitalOcean. What does it do?

It's a web application that will allow us to:

- Find beach cleanup events
- Search by beach and location
- Allow users to subscribe to upcoming beach cleanups

What are you going to deploy?

Find yourself a partner, and ask each other these questions:

If you could build a website, what would you create?

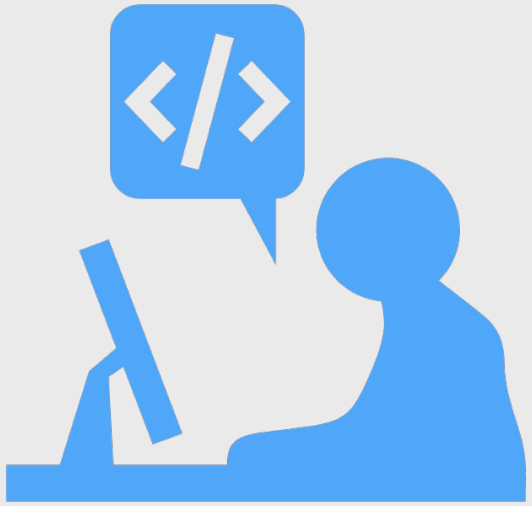
Let’s talk about DevOps.

What is DevOps?

When you think about making software, applications, or websites, sometimes you only think about the code itself - a software engineer writing lots of code to build something.

But there's **so much more** to developing!

That where Development and Operations (DevOps) comes in.



What is DevOps?

What kind of things is a DevOps engineer responsible for?

Deployment: Launching code into the real world! This could mean hosting a website on a server, or creating installation files for desktop software.

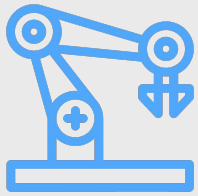
Maintenance: Code needs to be updated, and fixed when it breaks - a DevOps engineer safeguards these processes.

What is DevOps?

Monitoring: A DevOps engineer keeps track of the system health - are the servers running smoothly?

Security: Are the devices protected from internal and external attack?

Automation: For all of these jobs - can you do them automatically to save human time?

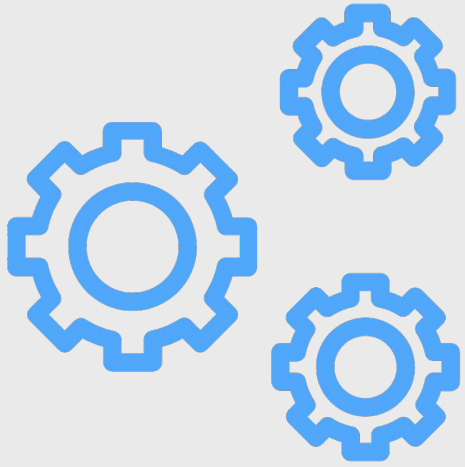


DevOps and DigitalOcean

We're going to deploy our application to DigitalOcean and learn some vital DevOps skills along the way.

By the end of the workshop, you'll know how to:

- Deploy code to a DigitalOcean Droplet.
- Keep your Droplet secure.
- Monitor the health of your Droplets.



Let’s get started!

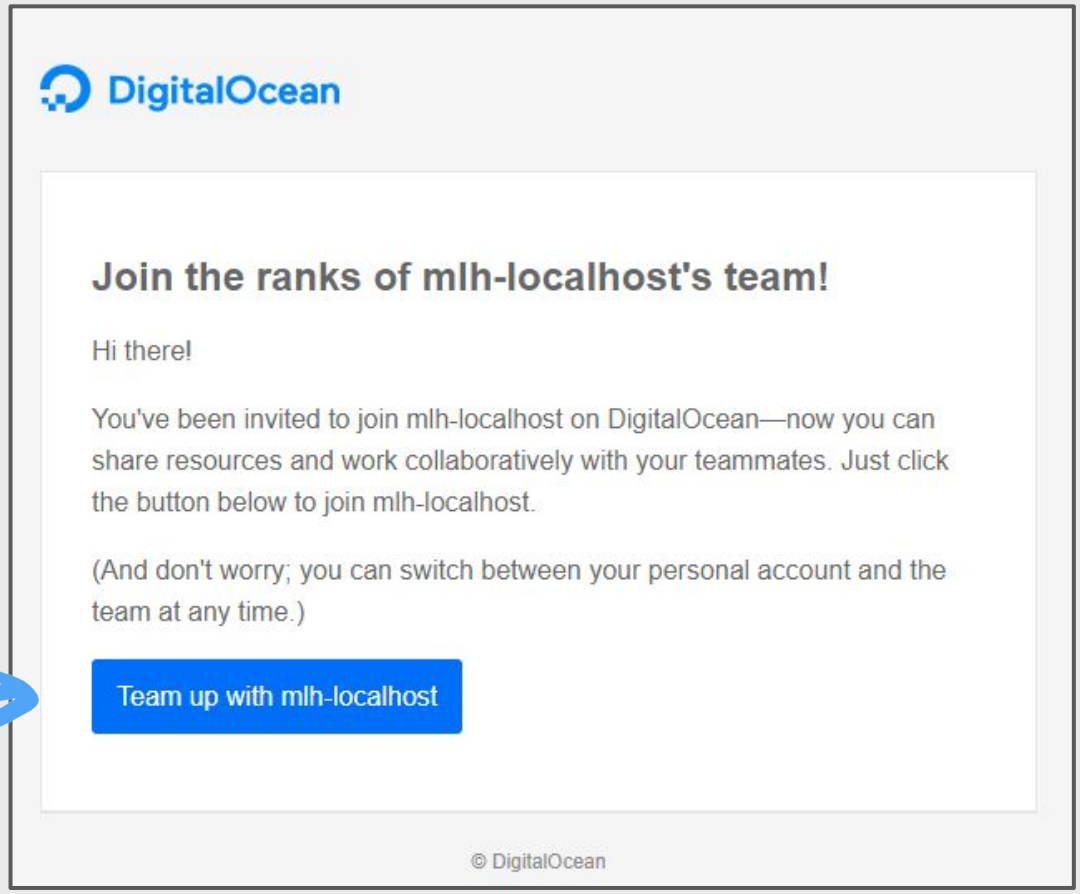
Table of Contents

Join the DigitalOcean team

Head to your email inbox and you'll have an invitation to join a DigitalOcean team for this workshop.

Join your Workshop's Team

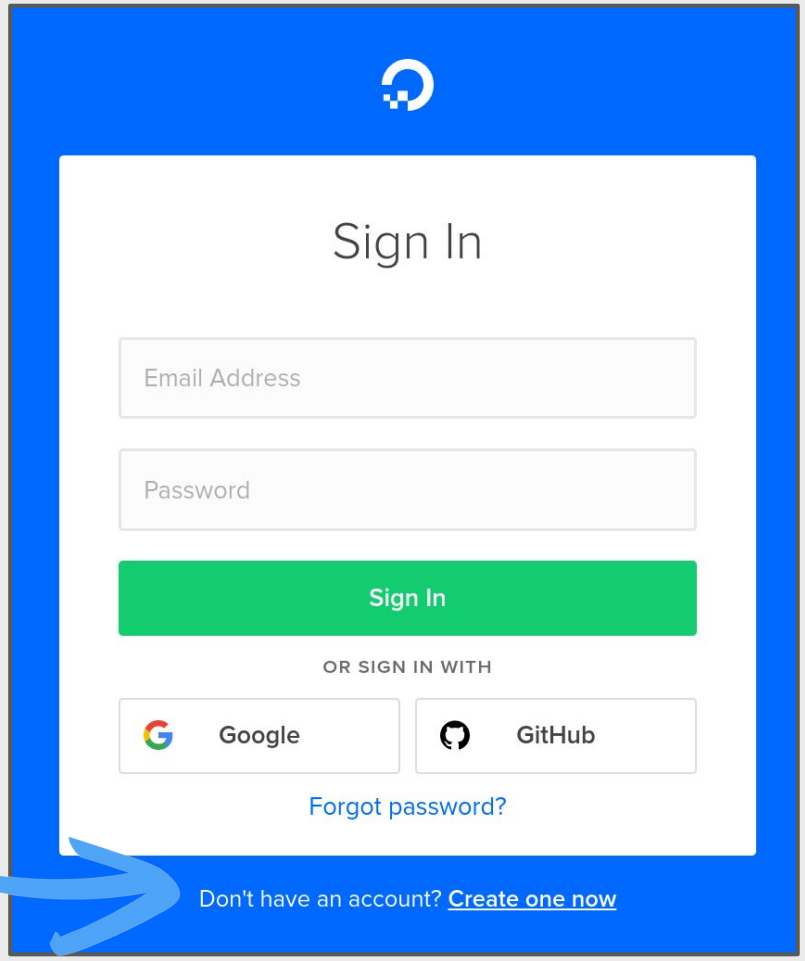
Click the **Team up** button to begin!



Create a DigitalOcean account

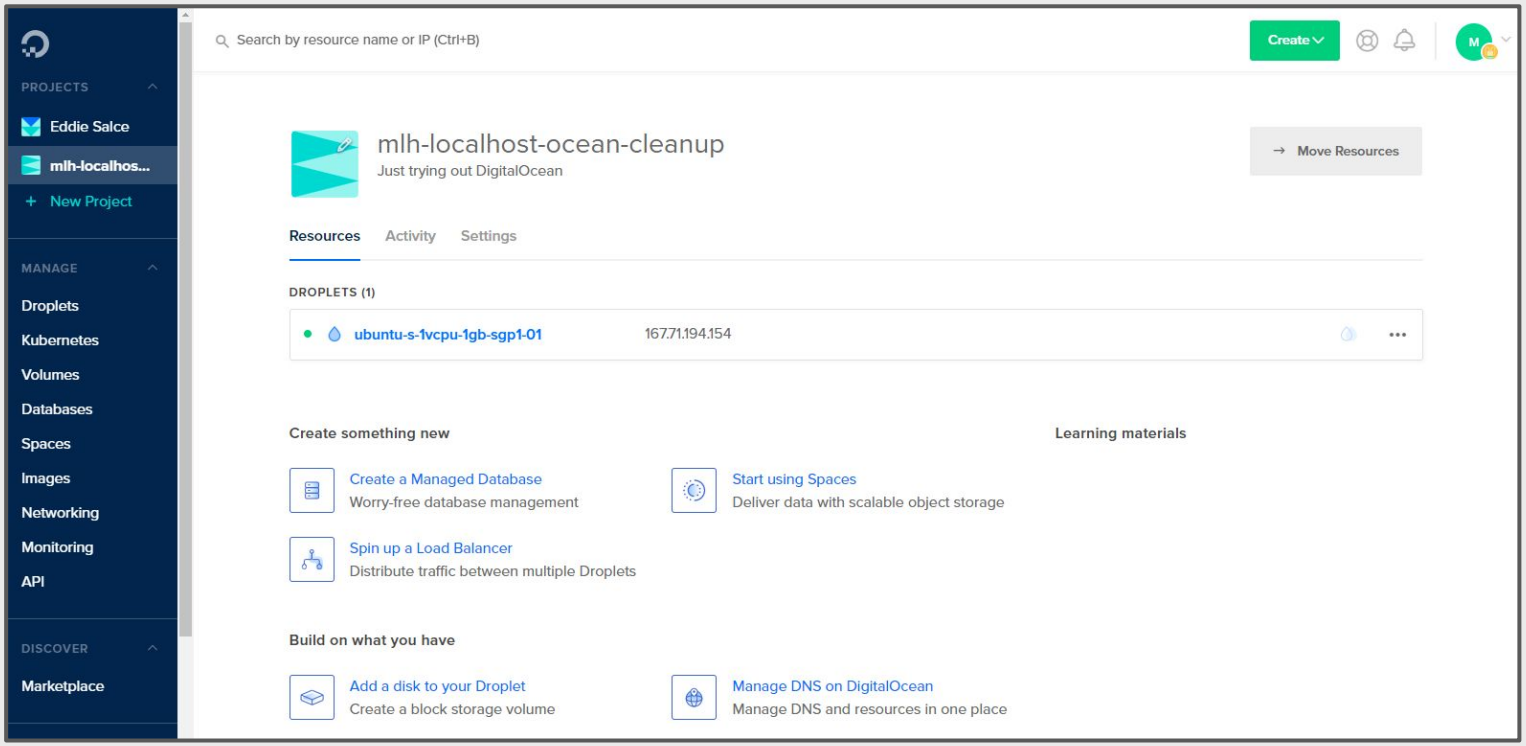
Click on **Create** to make a DigitalOcean account.

You can use your Google account or GitHub, or create one with your email address.

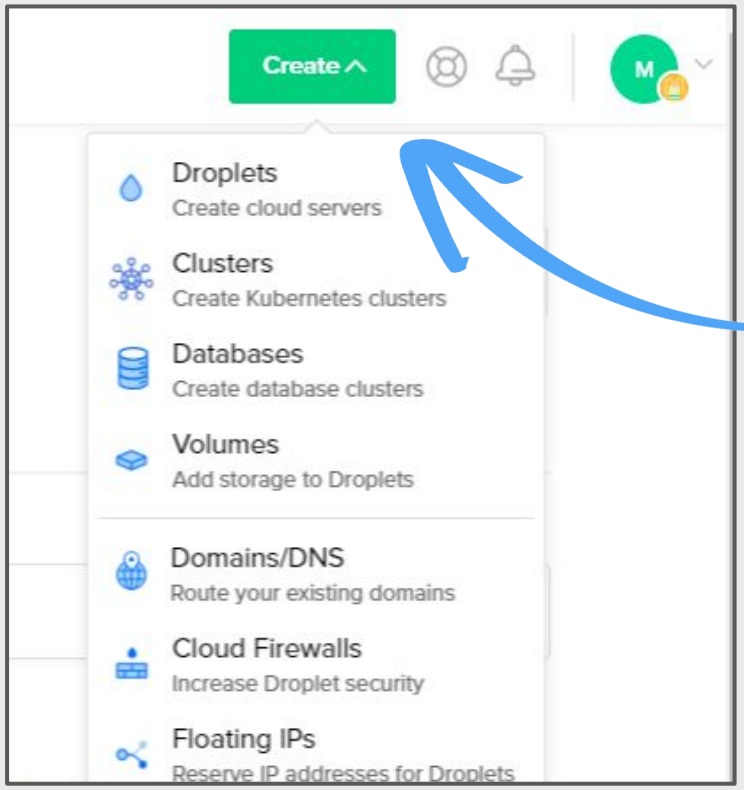


The DigitalOcean Dashboard

The link will open the DigitalOcean Dashboard. Here's where you can keep track of our projects and Droplets.



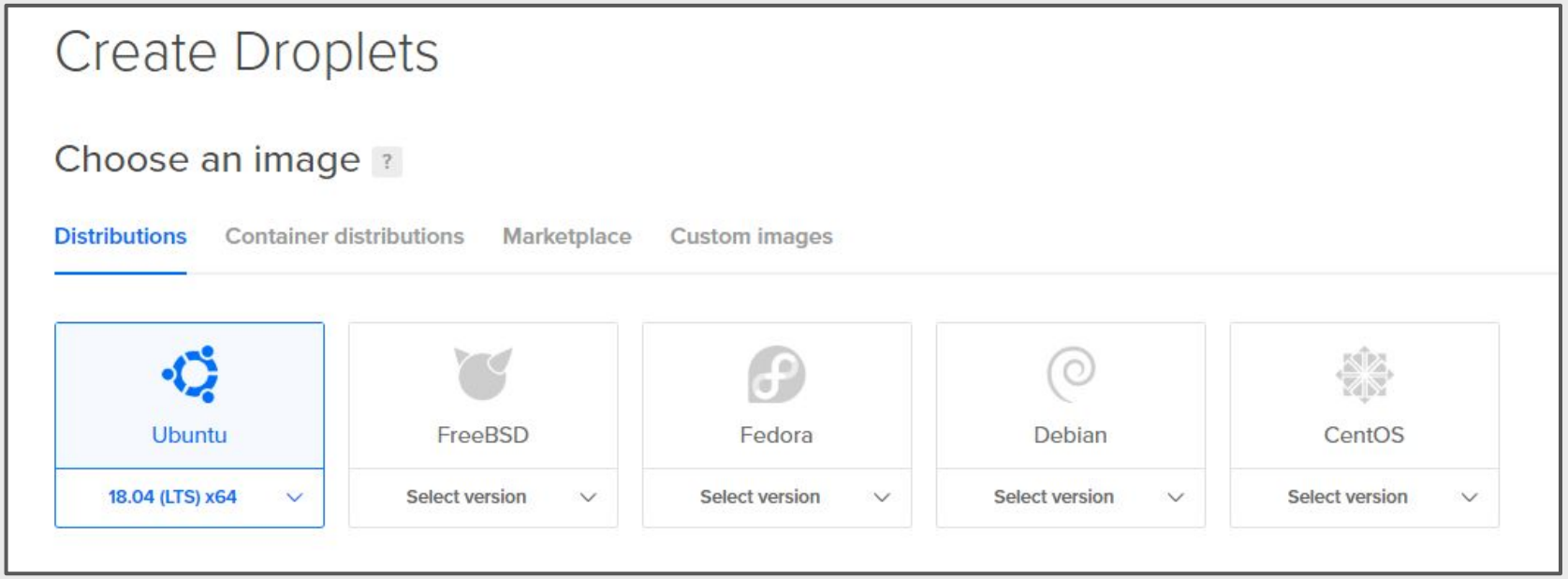
Create a new Droplet



Click the green **Create** button and then select **Droplets**.

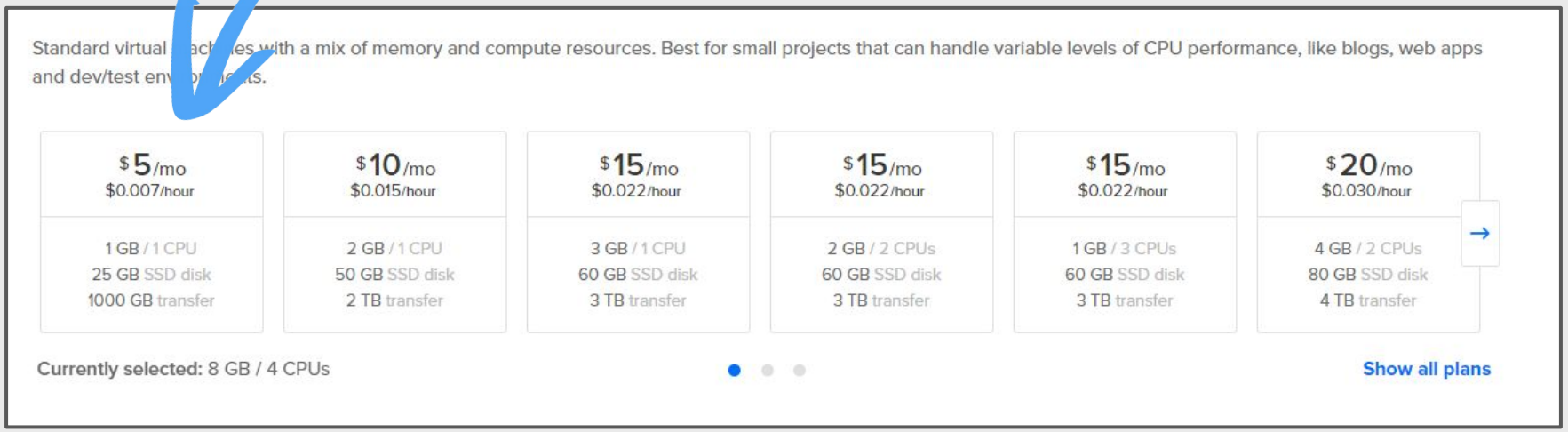
Choose an Operating System

We can select which operating system you want our Droplet to have. Select **Ubuntu 18.04** if it's not already highlighted.



Select a Plan

Scroll down and select the smallest plan - **1GB / 1 CPU**. That will be enough for our needs!



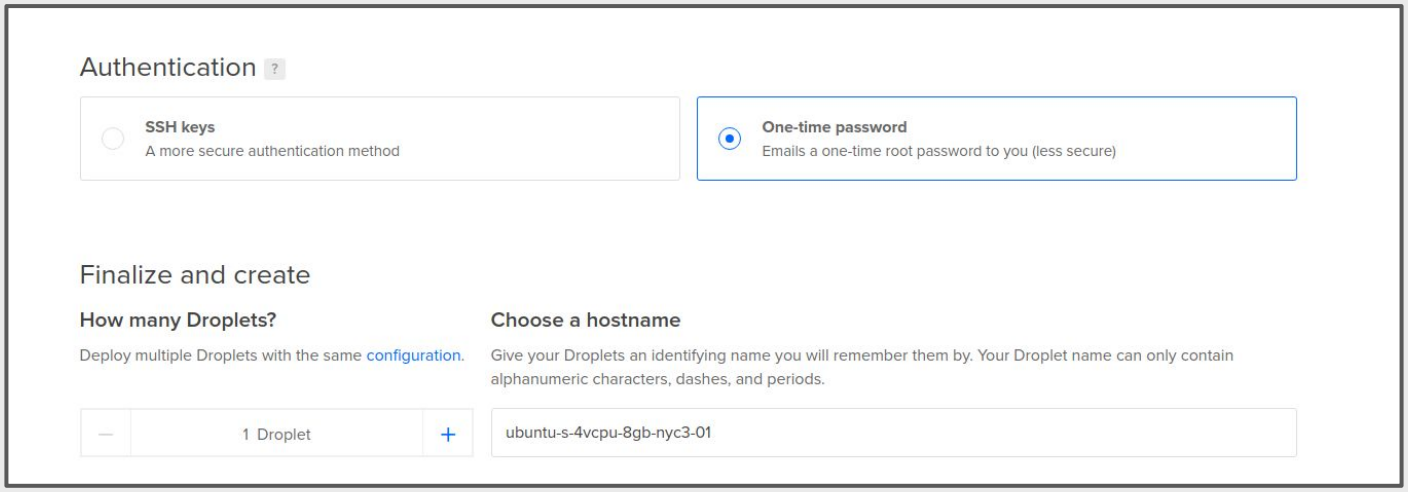
Datacenter Region

Select the Datacenter region which is closest to you.

Creating a Droplet

Under authentication, select **One-time password**.

Lastly, give your Droplet a name to identify it as yours, such as your first name initial and last name.



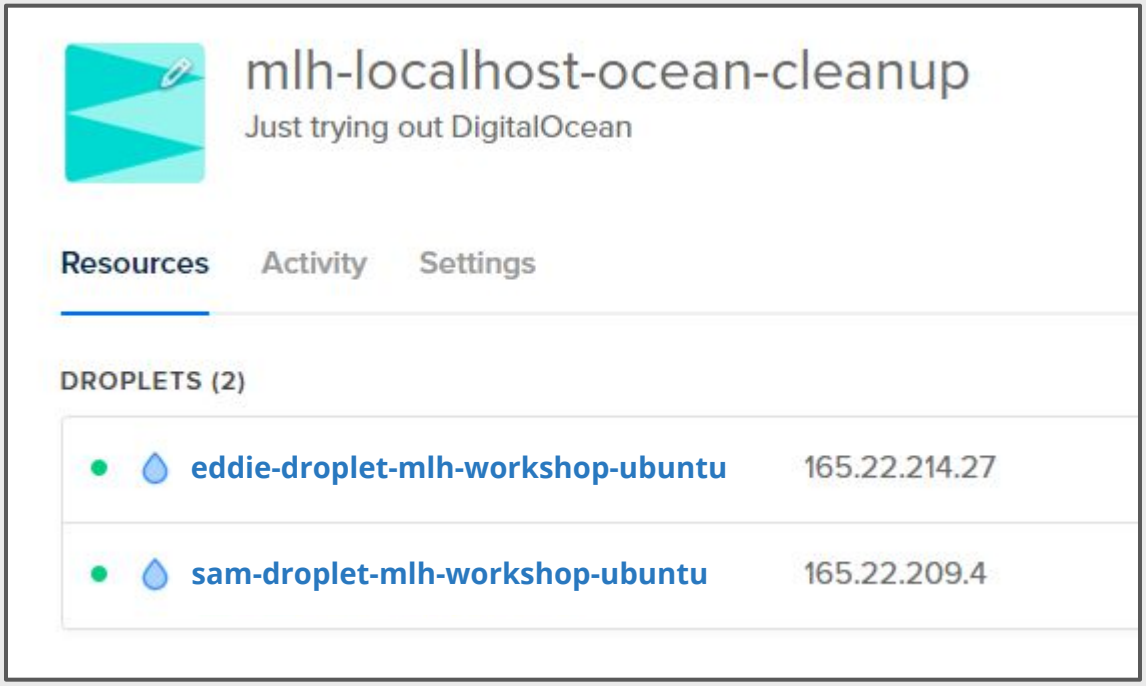
Assign to a Project

We can leave the rest of the settings on their default values.

Scroll to the bottom, and click **Create Droplet**!

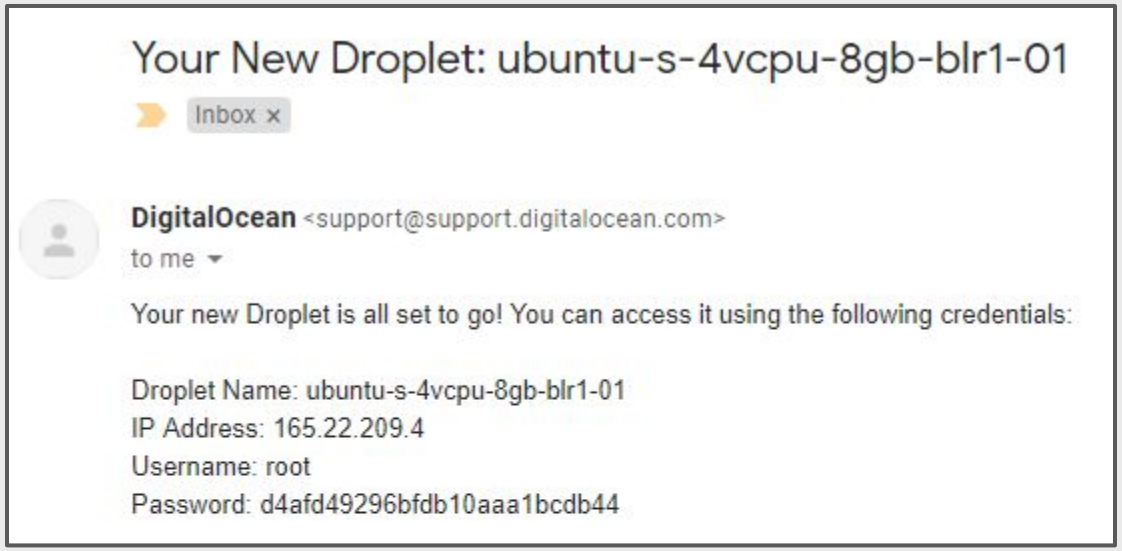
Back on the Project Dashboard, you can see that your Droplet has been created.

Now you can start using it!



Get your Droplet Details

Check your email and you should receive an email containing the details of your newly created Droplet.

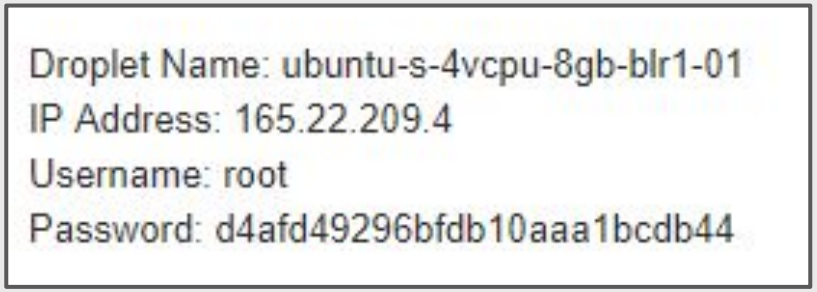


Get your Droplet Details

Droplet Name: How you identify your Droplet on the DigitalOcean dashboard.

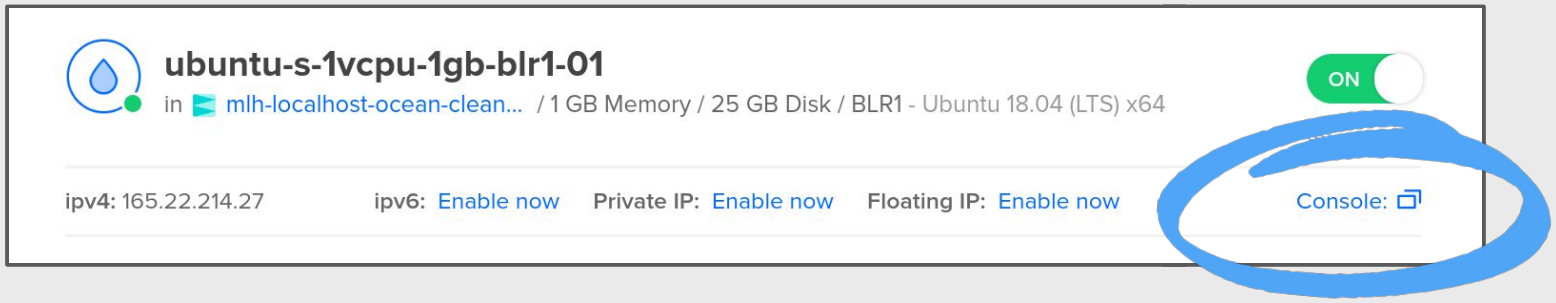
IP Address: Your Droplet's address on the internet.

Username/Password: How you can log in to your Droplet with an administrator (root) account.



Connecting to your Droplet

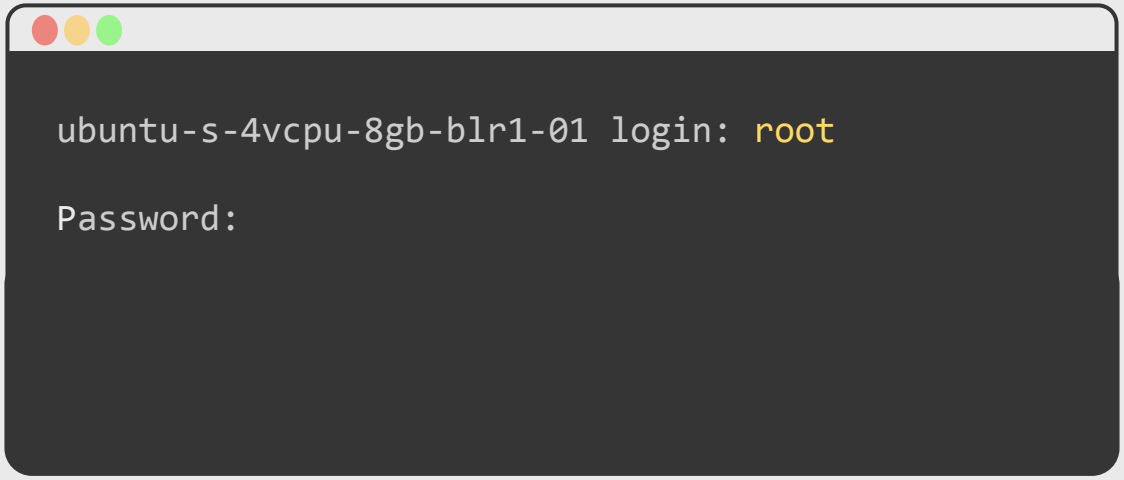
On the DigitalOcean dashboard, select your new Droplet, and click **Console**.



Connecting to your Droplet

Enter the username **Root**. Enter the password you received in the email.

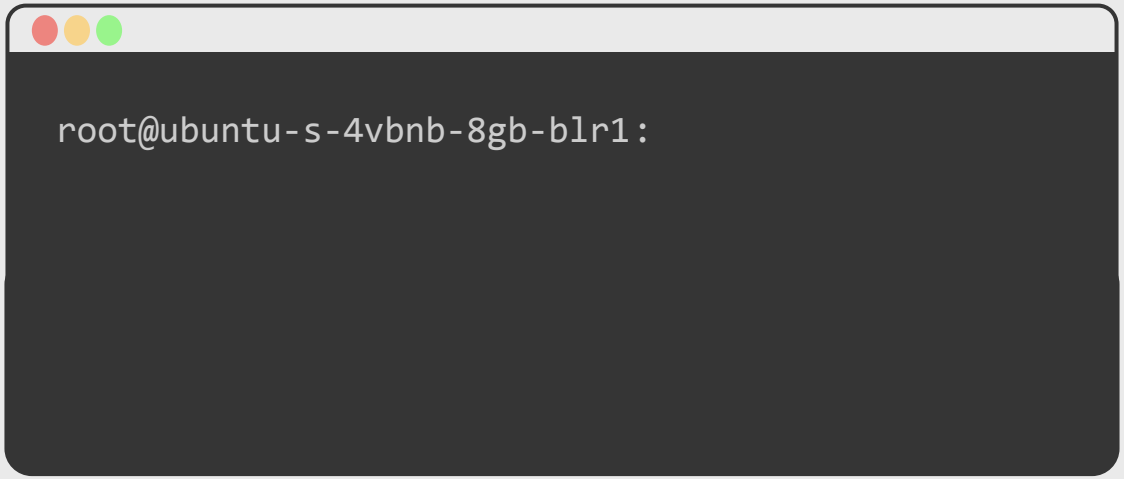
For your first time, you'll be asked to change your password. Make it something unique to you!



Connecting to your Droplet

Check out what happened to your Console. You're now inside the **Root** (administrator) account of your Droplet!

This means that you are completely controlling the Droplet from your own device. Awesome!



Adding new Users

An important DevOps responsibility is Security.

One way you can implement this is **Access Control**: only giving users the privileges and power that they need, and no more.

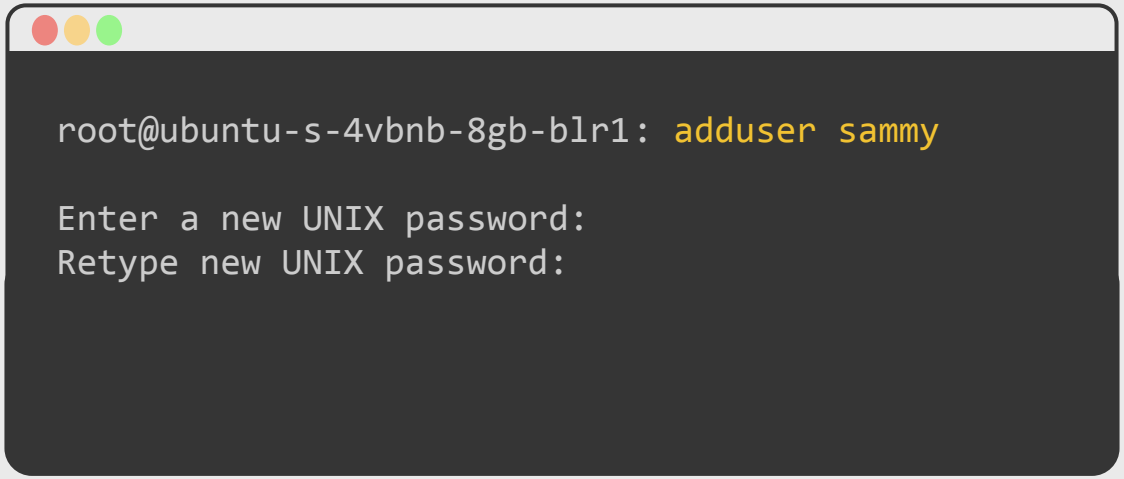
The Root account can do everything to our Droplet - you don't want this power falling into the wrong hands!

Let's make a user account that doesn't have quite as much power as Root.

Creating a new Account

Let's create a new user account. Type in the **adduser** command with a name and press Enter.

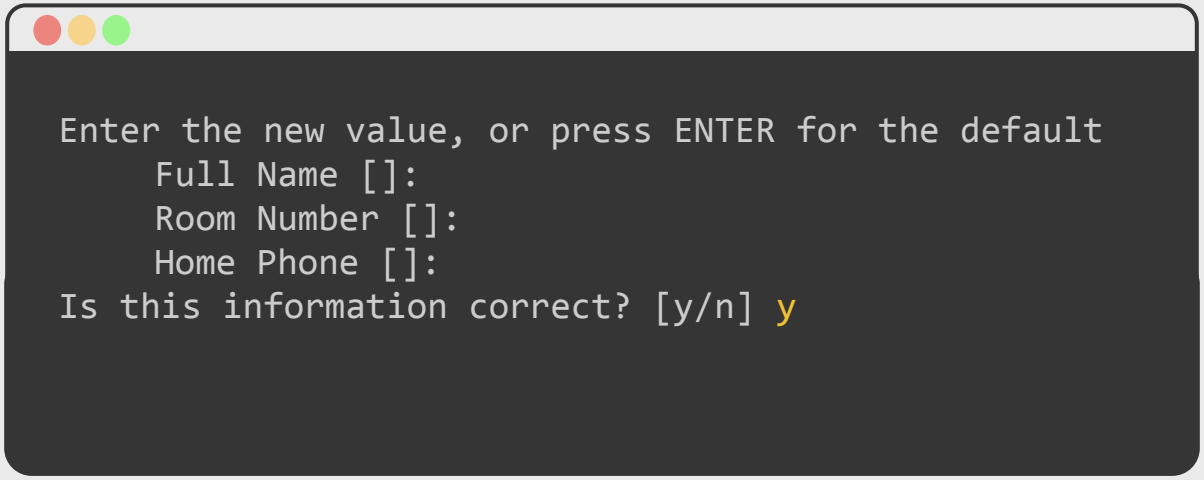
Give your account a password (different to the Root password you just created!) and verify it.



Creating a new Account

The Console will ask for some personal details - you can leave these blank and press Enter to continue.

At the end, type **y** to confirm.



Creating a new Account

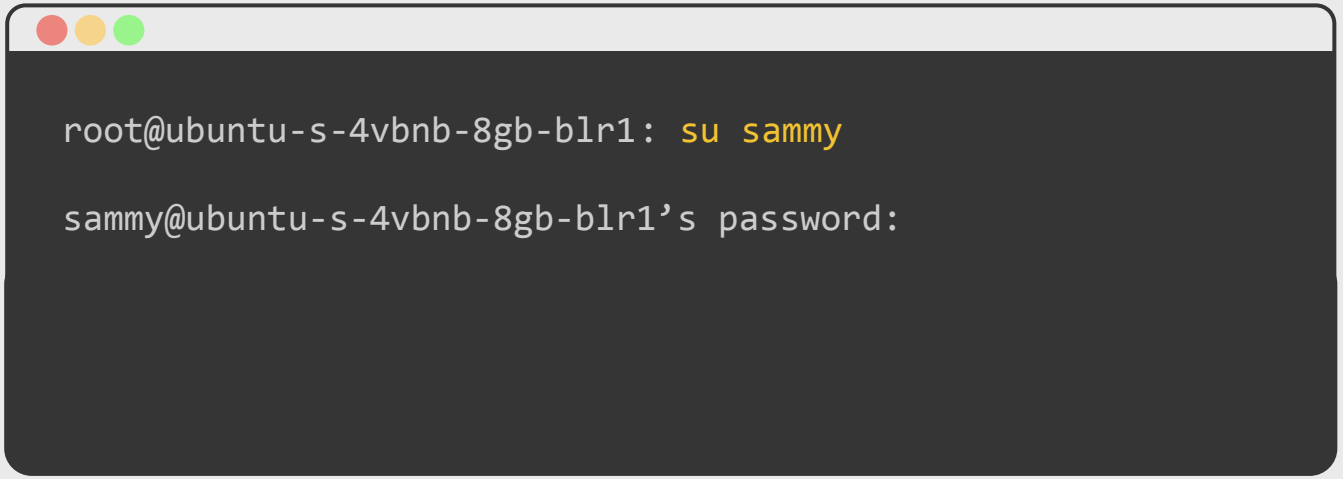
Next, you'll give your new account some minimal administrative privileges so that you can install new packages and updates.

Type in the following command, replacing **sammy** with your account name.

Creating a new Account

Now, log in to your newly created account. Use the **su** command (Switch User) to log in to your new account.

Replace **sammy** with the name you gave your account.



The Advanced Package Tool

Next, we're going to install some software on our Droplet using the Advanced Package Tool.

We use the **apt** command to install or update a package (piece of software).

By using APT, you ensure that we're installing the package from a trusted repository, and that any updates you install will be secure.

Installing Updates

Let's make sure our Droplet is up to date and has all of the packages it needs. Use the **apt** command, and enter your account password.

This might take a minute or two!

```
sammy@ubuntu-s-4vbnb-8gb-blr1: sudo apt update

sammy@ubuntu-s-4vbnb-8gb-blr1: sudo apt upgrade
```

Why do we update *and* upgrade?

Update returns a list of available packages and new versions, then **upgrade** actually installs them.

Installing Node.js

We need to install a couple of packages that our application requires to be able to run.

The first is Node.js, a server-side JavaScript environment.

Installing Node.js

To install Node.js, use the **apt install** command.

If it asks for permission, type **y** for yes.

```
sammy@ubuntu-s-4vbnb-8gb-blr1: sudo apt install nodejs

After this operation, 27MB of additional disk space will be
used. Do you want to continue? [Y/n]: y
```

Install Node Package Manager

Let’s do the same thing again to install Node Package Manager.

```
sammy@ubuntu-s-4vbnb-8gb-blr1: sudo apt install npm

After this operation, 27MB of additional disk space will be
used. Do you want to continue? [Y/n]: y
```

Get the Application Code

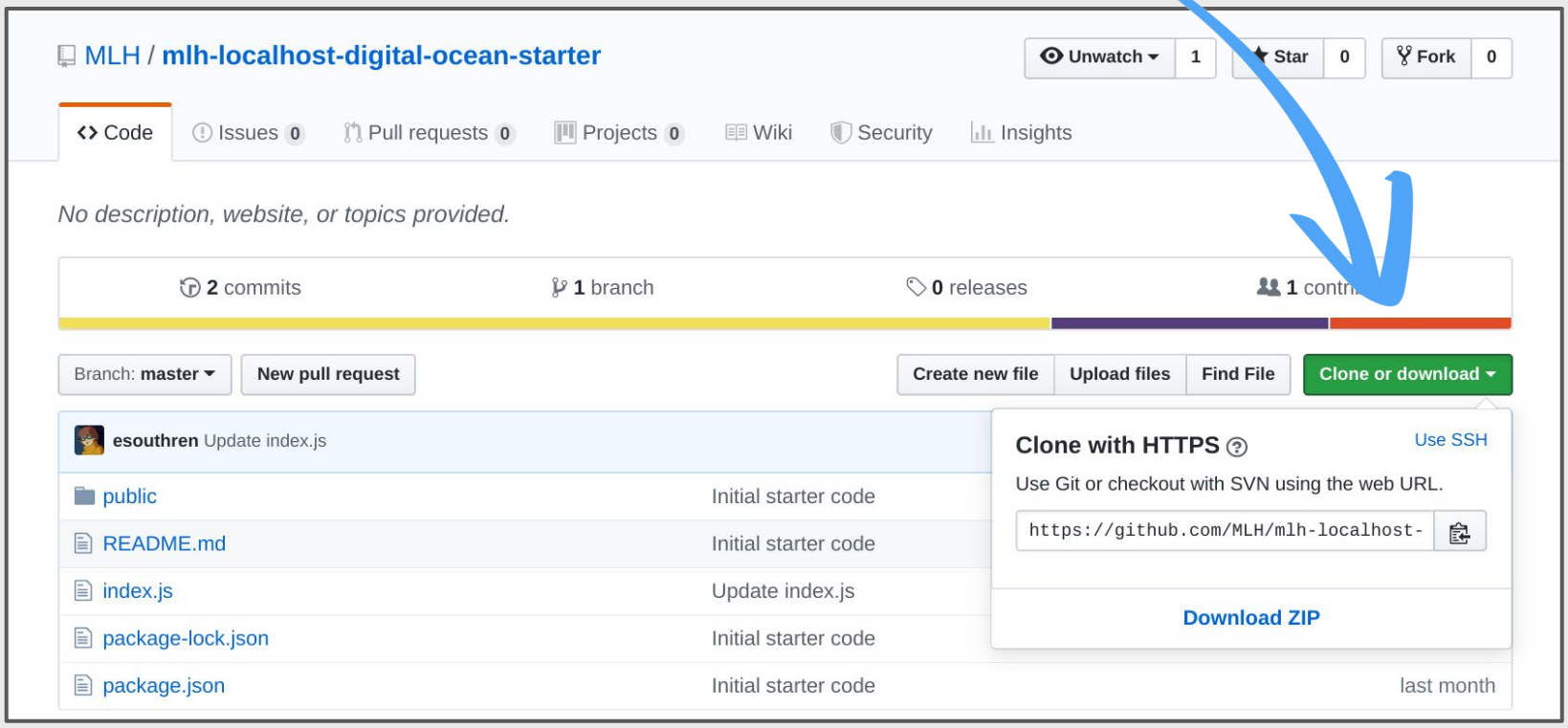
Next you need to add the application code to our Droplet.

Head to this GitHub repository, where the code is stored:

http://mlhlocal.host/digitalocean-project

Get the Application Code

Click the green **Clone or Download** button and copy the URL link.



Using Git for Version Control

GitHub is a tool for storing and sharing code online.

It uses Git as a form of **version control**. That means, keeping track of changes to the code over time.

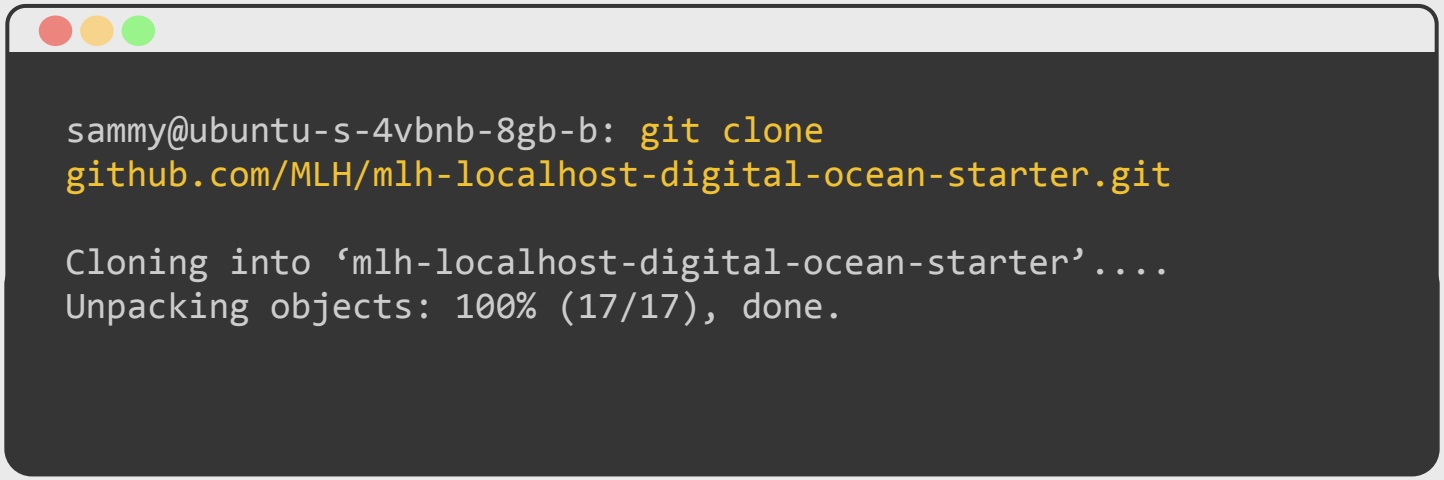
Why is this important to DevOps?

It allows us to have one central source for our code. We can then do things like automatically **deploy** this code when we want to update our application.

Copy the Project Files

Back in your terminal window, you'll use the **`git clone`** command to copy this repository into our droplet.

Type **`git clone`** then paste (Ctrl + V) the GitHub link you copied.



Install and Launch the App

You have everything you need!

Let's install the application and launch it.

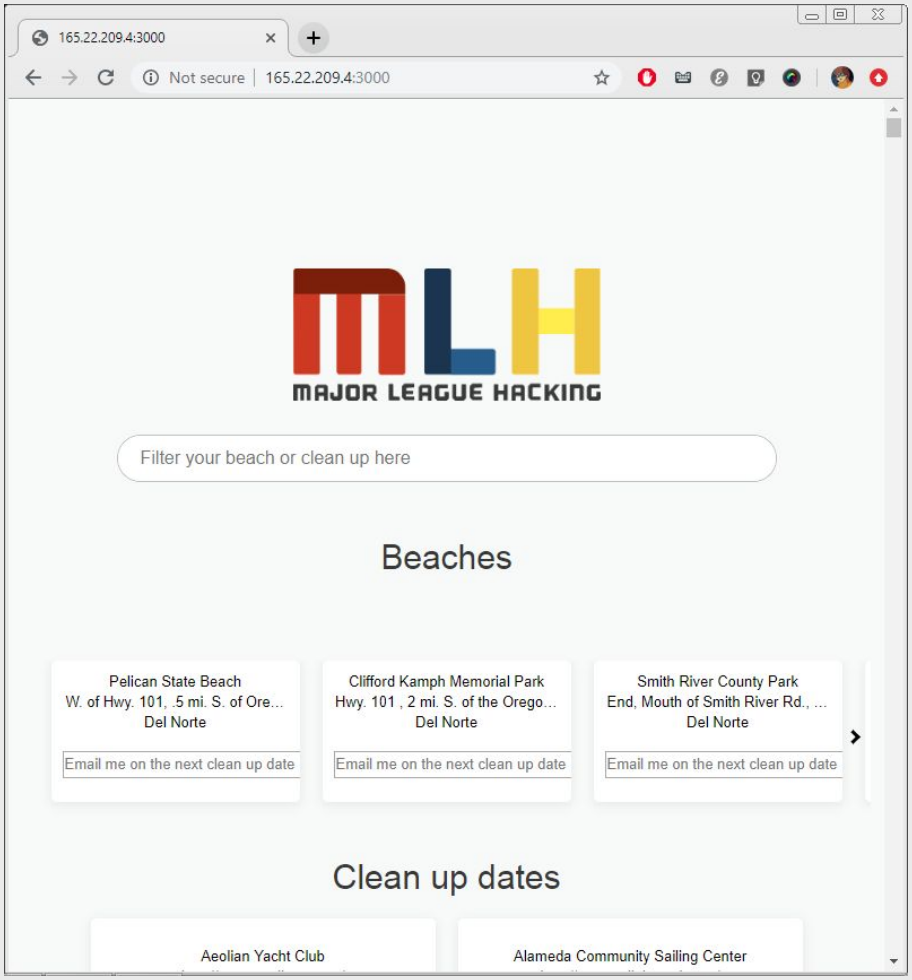
```
sammy@ubuntu-s-4vbnb-8gb-blr1: cd mlh-localhost-digital-ocean

sammy@ubuntu-s-4vbnb-8gb-blr1/mlh-localhost-digital-ocean: npm install

sammy@ubuntu-s-4vbnb-8gb-blr1/mlh-localhost-digital-ocean: npm start

DigitalOcean app listening on port 3000!
```

Accessing the App

On your device, open a new browser window and head to your Droplet's URL, on port 3000 - substitute your Droplet's IP address.

198.51.100.0:3000

Accessing the App



Success!

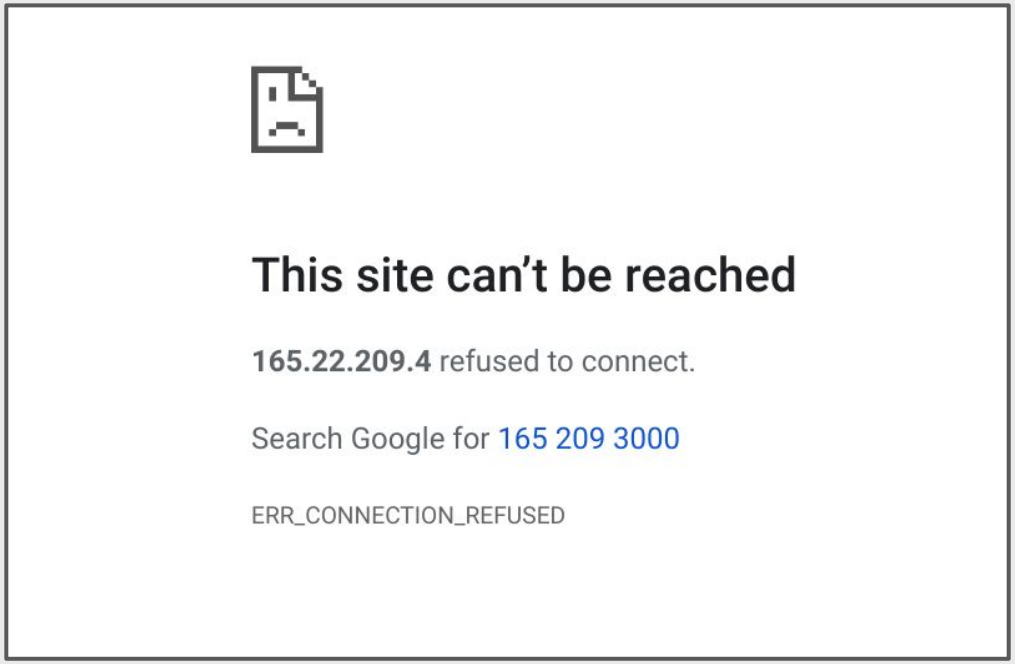
You are accessing the application that's being hosted in your DigitalOcean Droplet!

Table of Contents

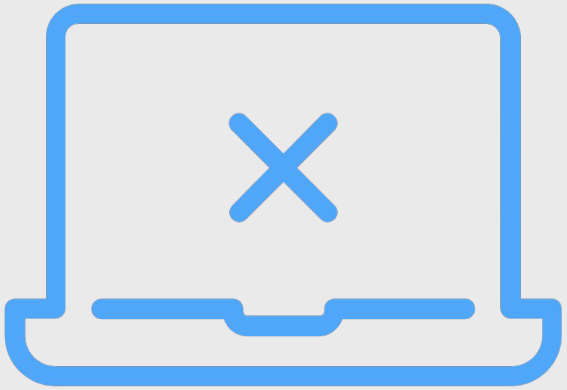
Let’s talk about some common DevOps features.

Server Availability

Have you ever gone to a website and seen this error message?



Server Availability



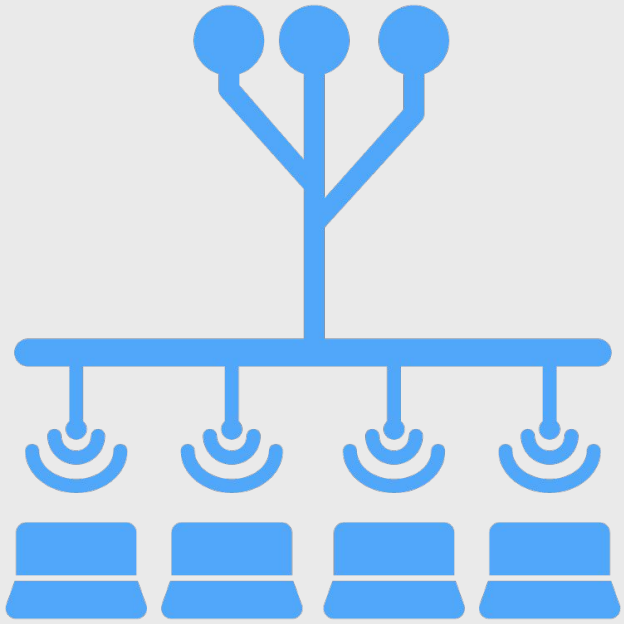
The user can't get to a website: a DevOps nightmare!

What can cause this?

- Servers going down or failing.
- An overload of server requests: too many people trying to connect at once.

The Solution

Thankfully, DigitalOcean can offer a solution to this: **Load Balancing**.



Load Balancing

Load balancing distributes network traffic across multiple Droplets. If one server fails, or is overloaded, the user can still make a successful connection.

Load balancers give us three things:

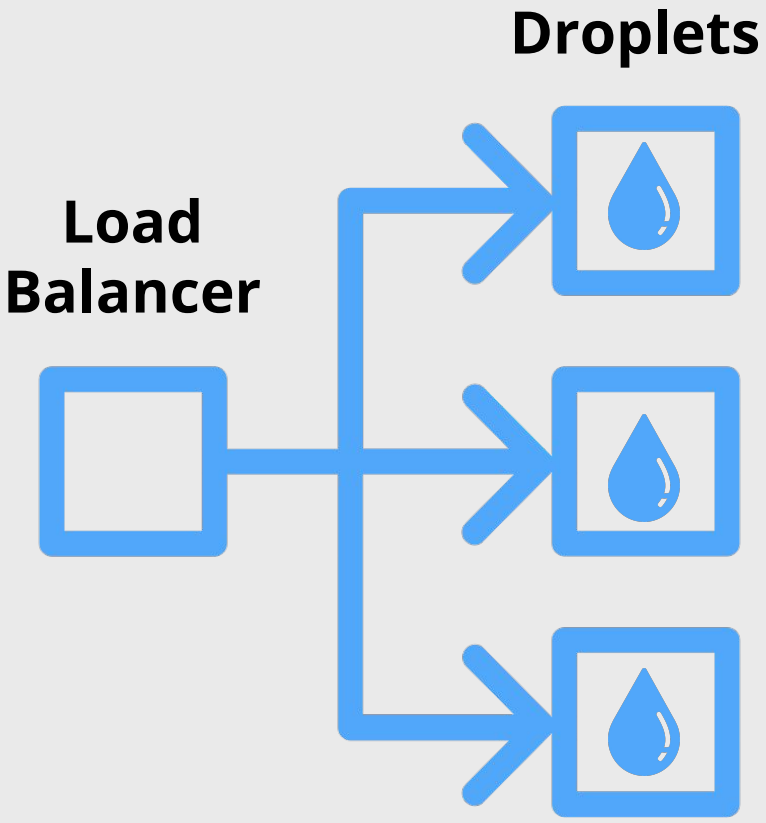
- **Availability:** Load balancers check to see which Droplets are healthy before sending requests to them.
- **Flexibility:** You can make changes to applications running on our servers without worrying that a user won't be able to access the server while you do it.
- **Performance:** You can share the traffic workload across multiple Droplets - so one Droplet doesn't get overwhelmed.

Load Balancing

How does it work?

Instead of directly accessing the Droplet, you go to the Load Balancer's network address.

The Load Balancer then selects the best Droplet to fulfill the request.



Load Balancer Health Checks

The Load Balancer will select a **healthy** droplet.

To measure healthiness, the Load Balancer will **ping** our Droplet periodically to check for a successful response.

If it gets a successful response, it will record that the Droplet is currently in a healthy status.

Key Term

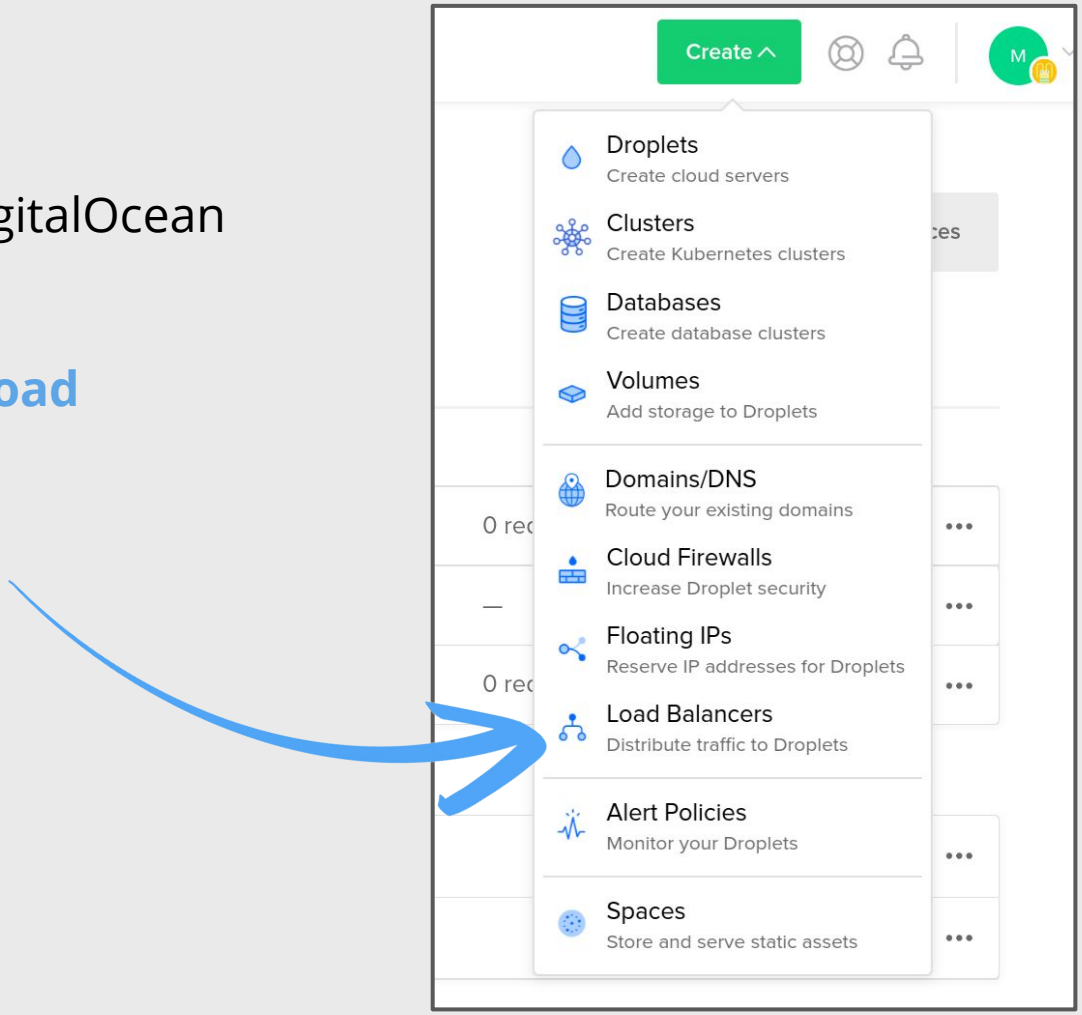
Ping: A utility that sends a request to a network device and waits for a response. It measures the time taken and if any data was lost.

Creating a Load Balancer

Let's make one!

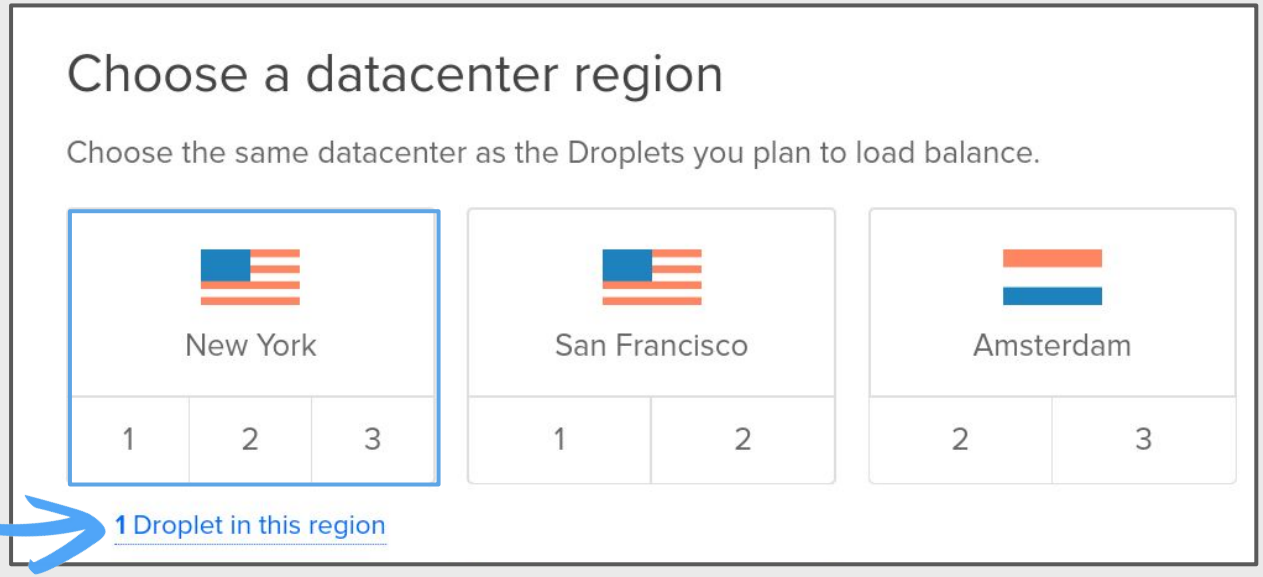
Head back to the DigitalOcean dashboard.

Click **Create**, then **Load Balancers**.



Creating a Load Balancer

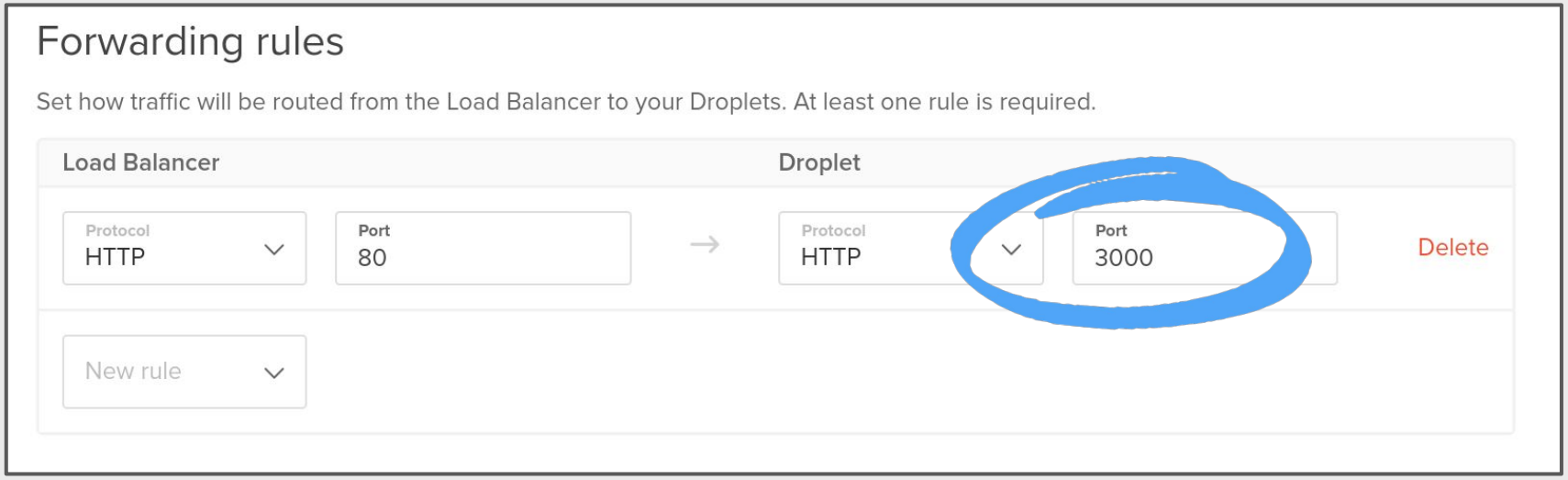
Select the Datacenter region that contains your droplets.



Creating a Load Balancer

We want the Load Balancer to redirect to Port 3000, which our application is running on.

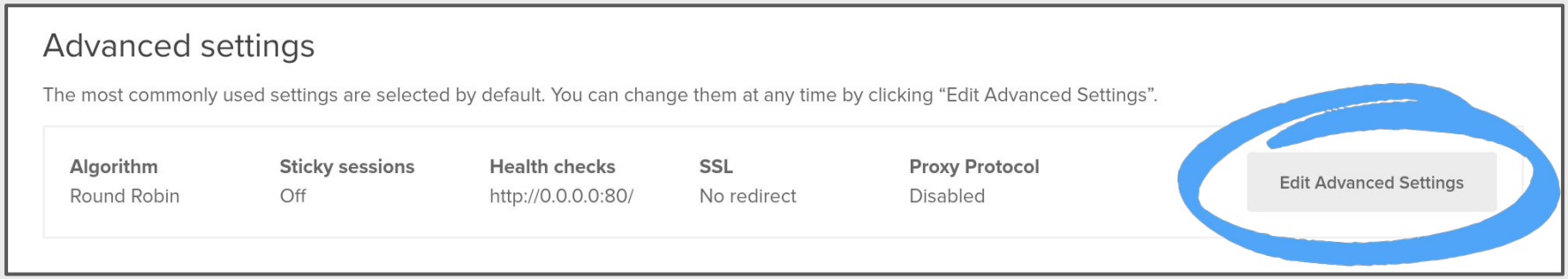
Change the Droplet port to **3000**.

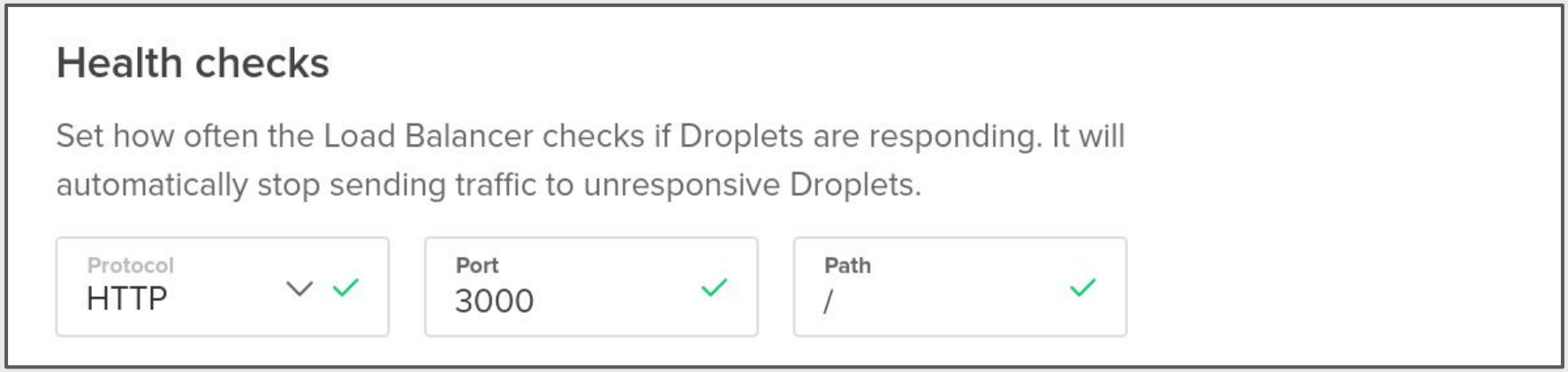


Creating a Load Balancer

Scroll down to Advanced Settings and click **Edit**.

Under Health Checks, change the Port to **3000**.

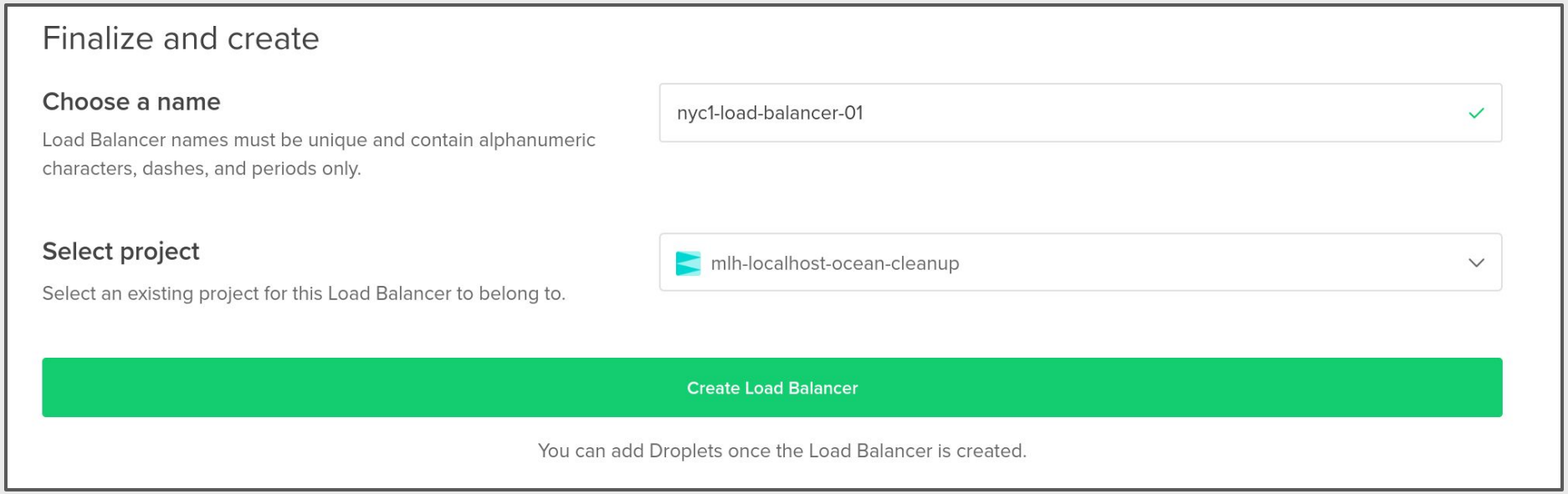




Creating a Load Balancer

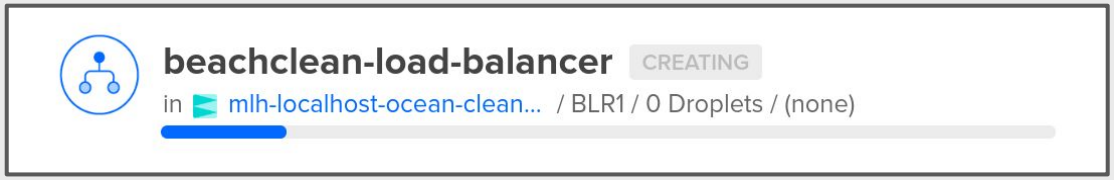
Give your Load Balancer a name, and make sure your project matches the workshop you are taking part in.

Lastly, click **Create Load Balancer**!



Creating a Load Balancer

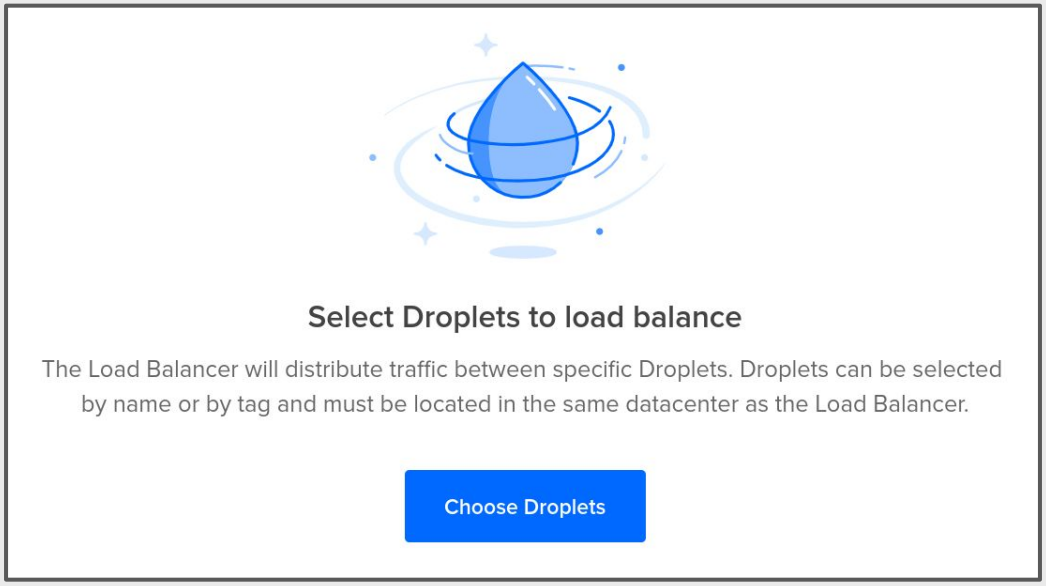
Your load balancer will take a few minutes to be created.



Adding Droplets

Now that you have a Load Balancer, you need to add some Droplets to it.

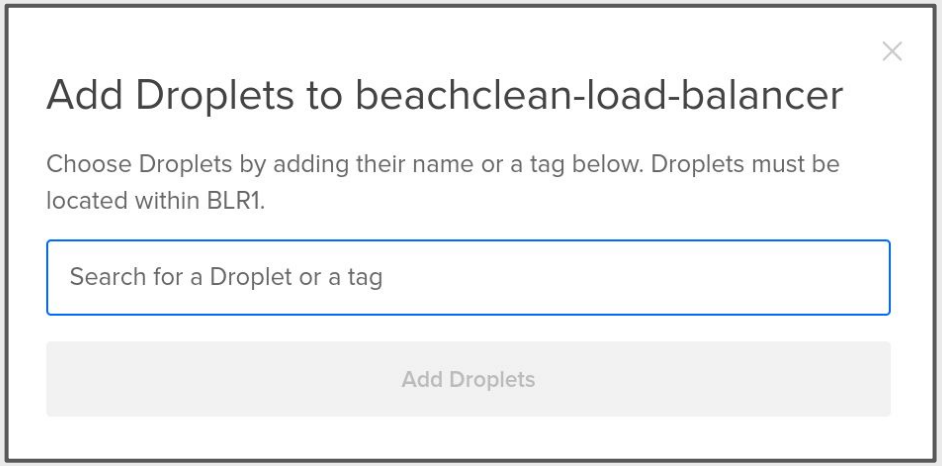
Click **Choose Droplets**.



Adding Droplets

Add your Droplet to the Load Balancer by searching for its name.

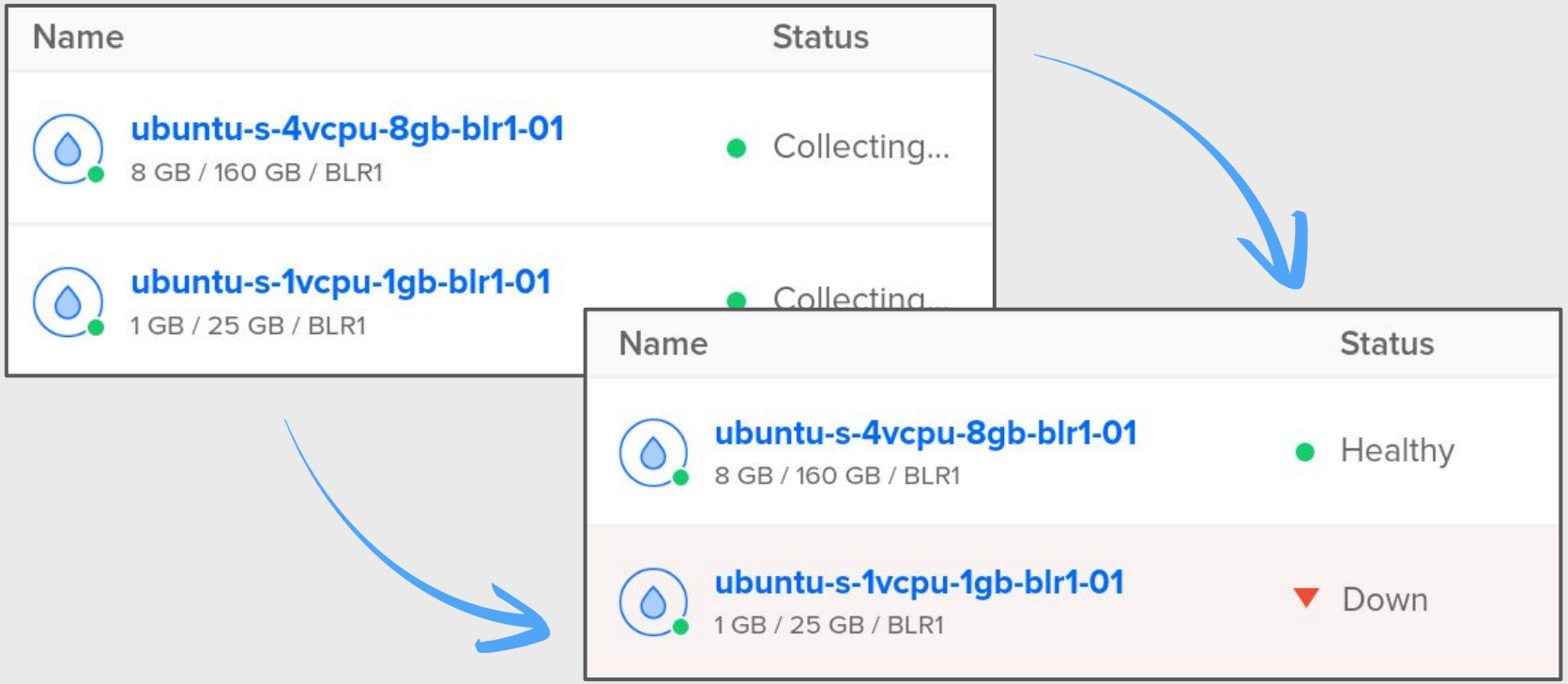
Add two or three more from other workshop participants!



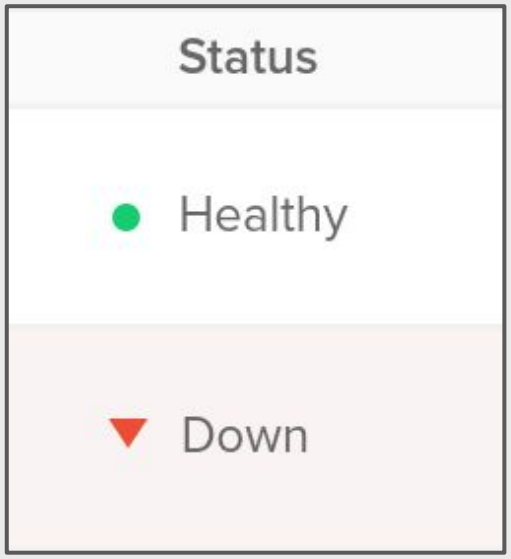
Adding Droplets

Wait a few minutes for your Droplets to be configured.

The status will change from **Collecting** to **Healthy** or **Down**.



Accessing the Load Balancer



What would cause a Droplet to be **Down**?

- If the Droplet is switched off
- If our application isn't running on Port 3000

We need at least one **Healthy** Droplet to be able to access our Load Balancer.

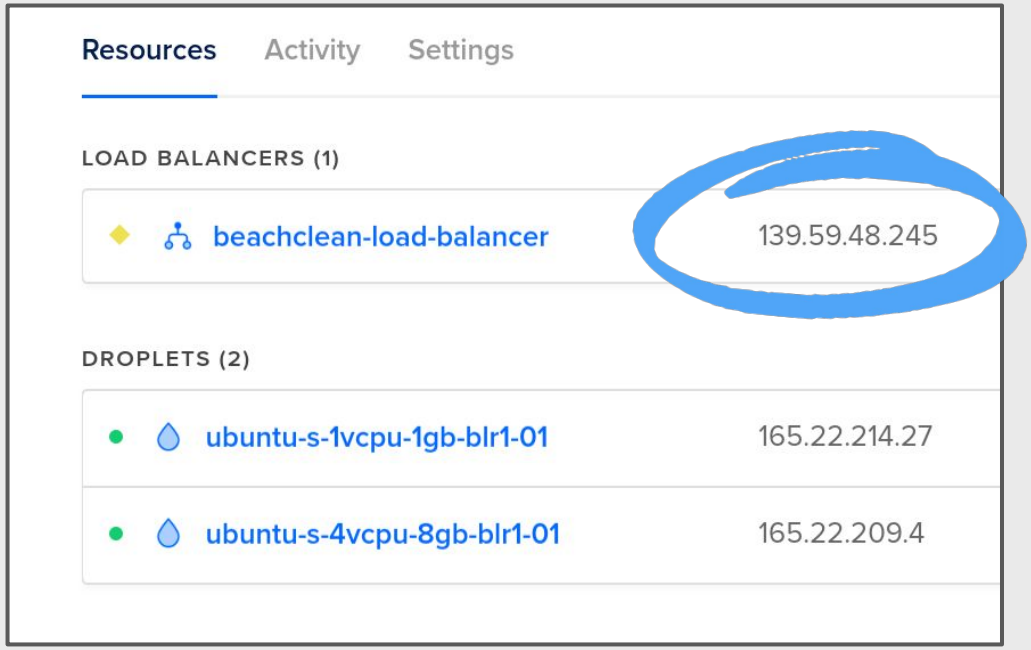
If yours is down, try logging into it and making sure the application is running.

Accessing the Load Balancer

Let's access our Load Balancer!

Back on the Dashboard, you can find the IP Address of our Load Balancer.

Type it into your address bar and check out the Application!



DevOps Monitoring

A vital part of DevOps is monitoring the servers that host our software.

We can track things such as:

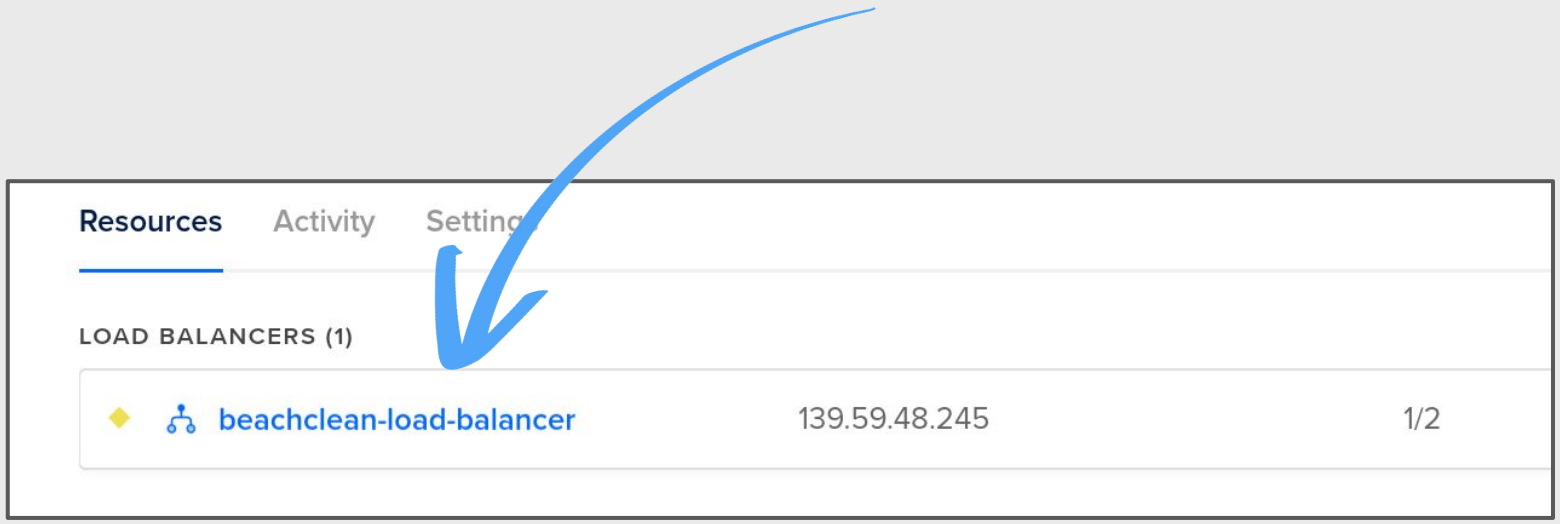
- Server memory usage
- Incoming/Outgoing network traffic
- HTTP Requests

DevOps Monitoring

Let's take a look at some of the DigitalOcean monitoring tools.

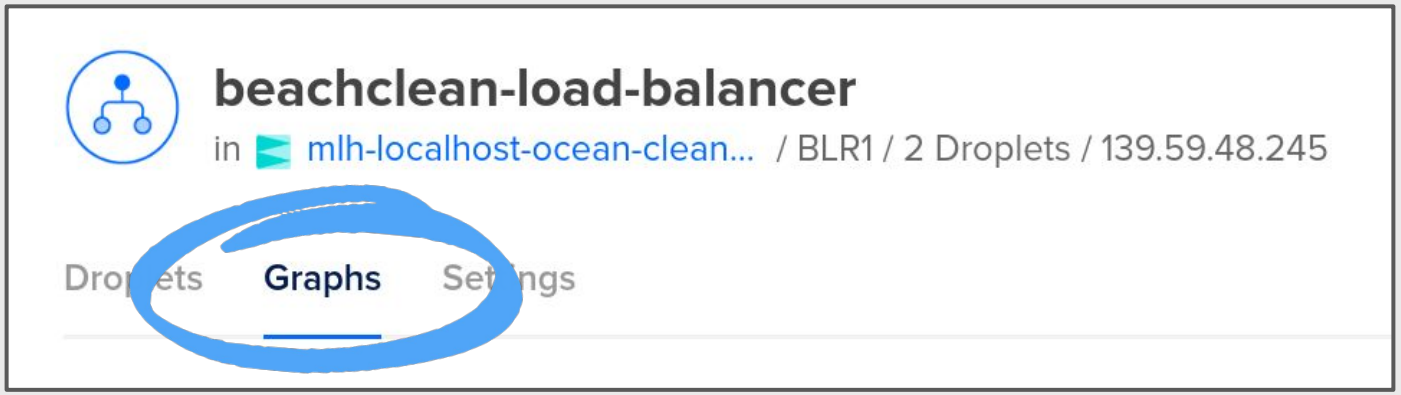
Then you'll add some custom tools of our own!

On your project dashboard, click on your Load Balancer.



DevOps Monitoring

Select **Graphs** from the tab menu.

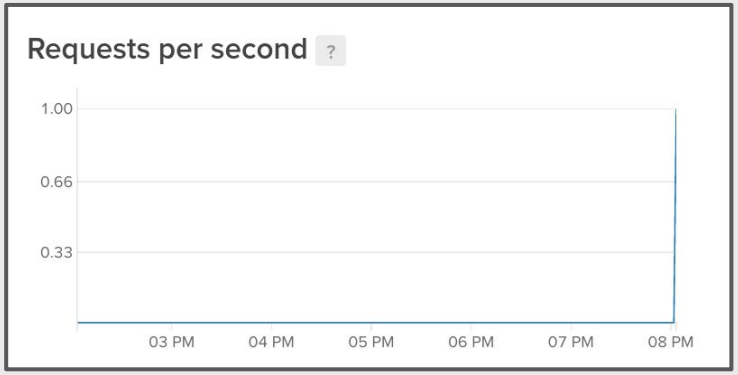


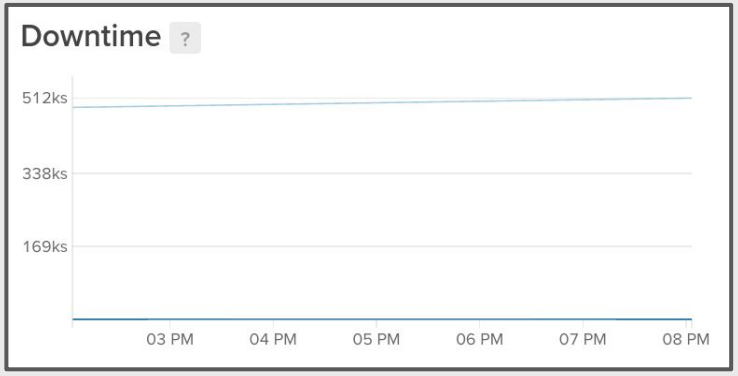
DevOps Monitoring

Let's see what you can monitor for our Load Balancer.

Requests per second: How many people are trying to access our Load Balancer every second? Here, you can see spikes that represent connections.

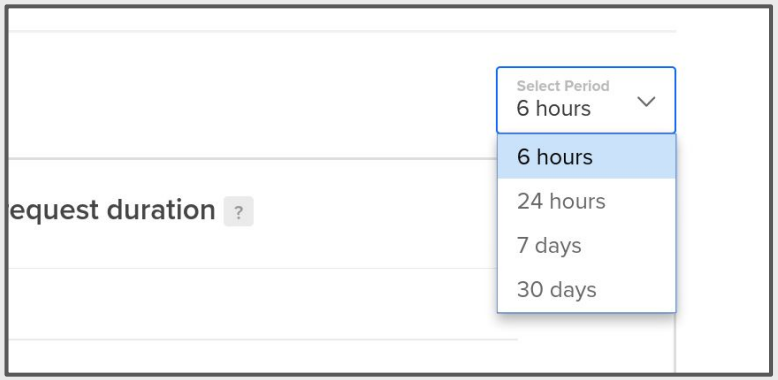
Downtime: How often is our server down, and for how long?





DevOps Monitoring

We can click **Select Period** and monitor activity from the last 6 hours to the last 30 days.



Alert Policies

An important aspect of monitoring is being alerted to potential issues before they become big problems.

With DigitalOcean you can create **Alert Policies**.

With an Alert Policy you define a threshold for a metric such as CPU or Memory usage.

When the threshold is breached, it sends us an alert via email.

Key Term

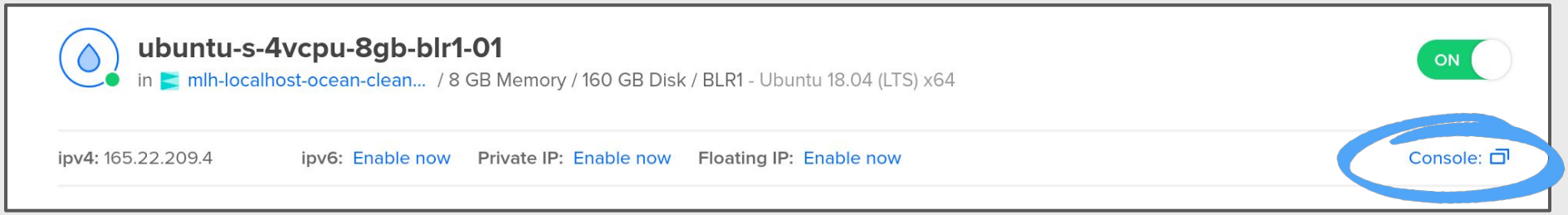
Metric: A measurable unit, such as percentage of memory being used, or number of requests received per second.

Installing the Metrics Agent

To monitor a Droplet, you first need to install a Metrics Agent on it.

This provides some more in-depth monitoring tools.

To get started, click on your Droplet and open the **Console**.



Installing the Metrics Agent

If your app is currently running, you can stop it by pressing Ctrl+C.

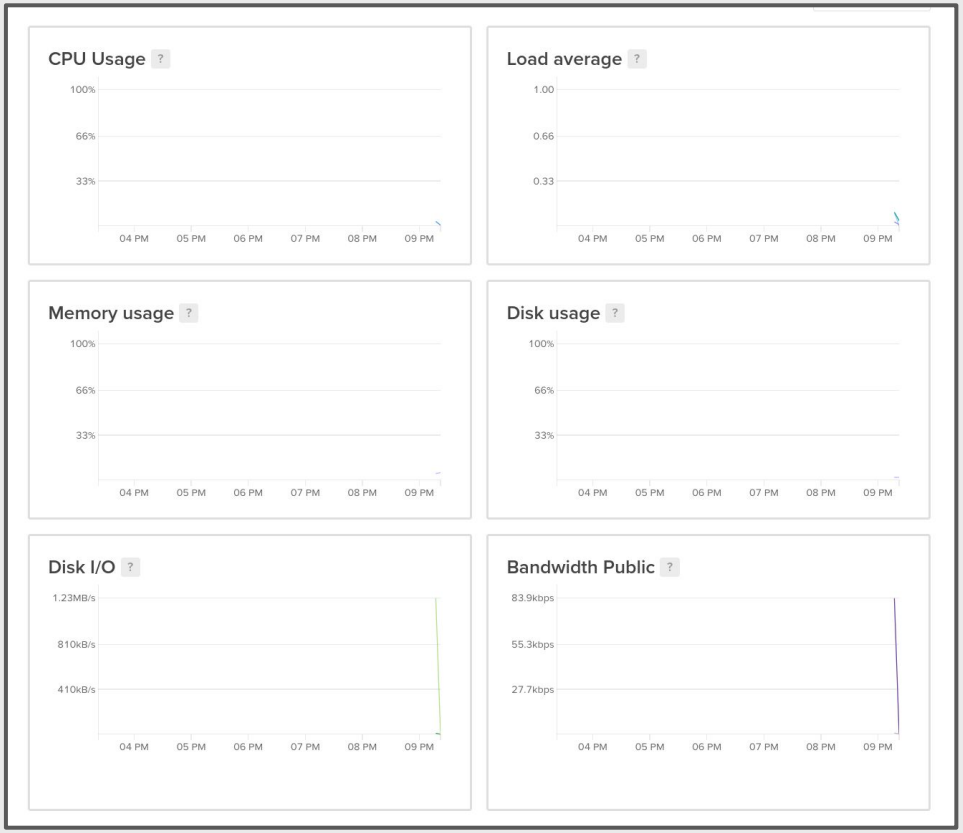
Use the **Curl** command to download the Metrics Agent installation script and use the **Bash** command to run it.

Remember to start your application again with **npm start**!

```
Digital Ocean App running on Port 3000!   <Ctrl+C>

sammy@your-droplet> curl -sSL
https://repos.insights.digitaloceanspaces.com/install.sh -o
/tmp/install.sh

sammy@your-droplet> sudo bash /tmp/install.sh

sammy@your-droplet> npm start
```

Metrics Agent Graphs

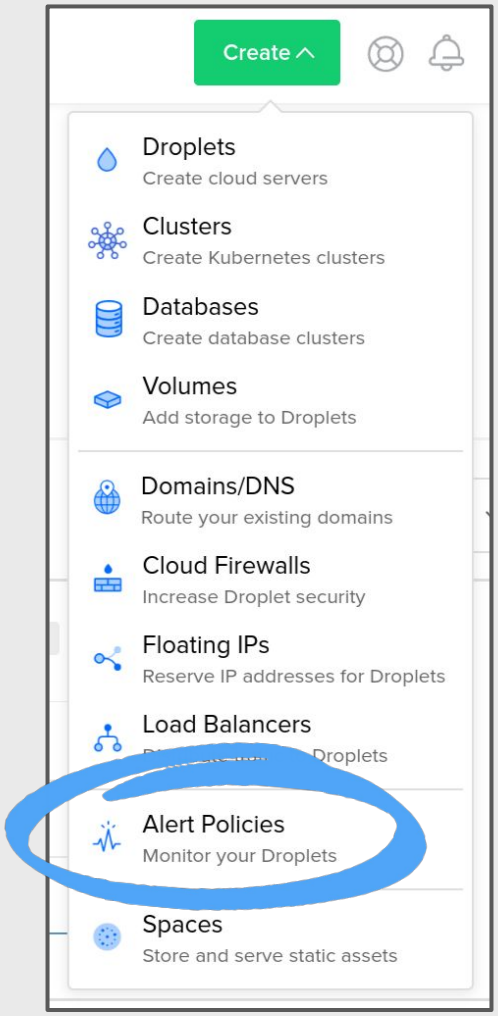
If you click on your Droplet dashboard, you'll see that you now have some more Monitoring graphs.



Create an Alert Policy

Now you can create an Alert Policy for our Droplet.

To create an Alert Policy, click **Create**, then **Alert Policies**.



Create an Alert Policy

Let's define the metric and threshold for our Alert Policy.

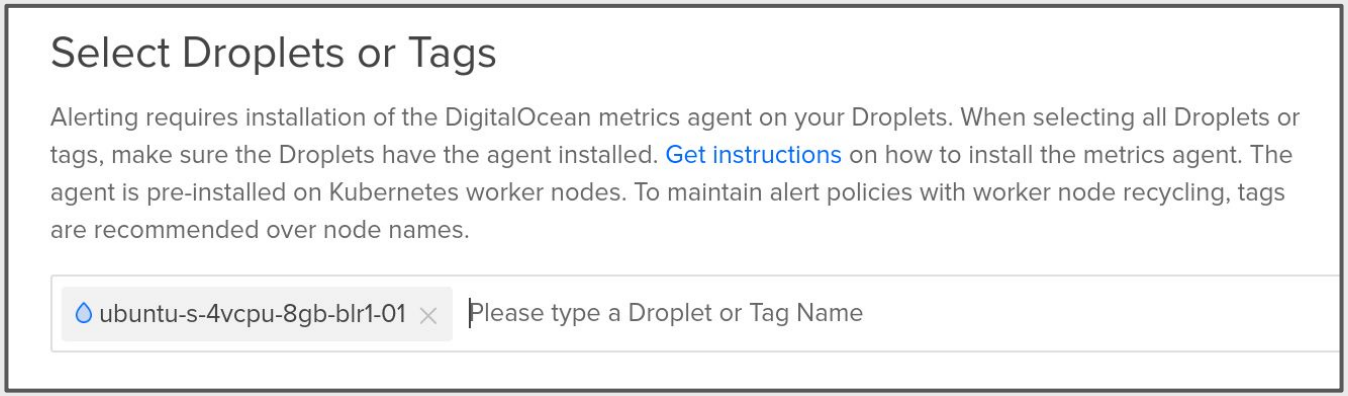
We want to be notified when the **CPU** usage of our Droplet **is above 70%** for more than **five minutes**.

Key Term

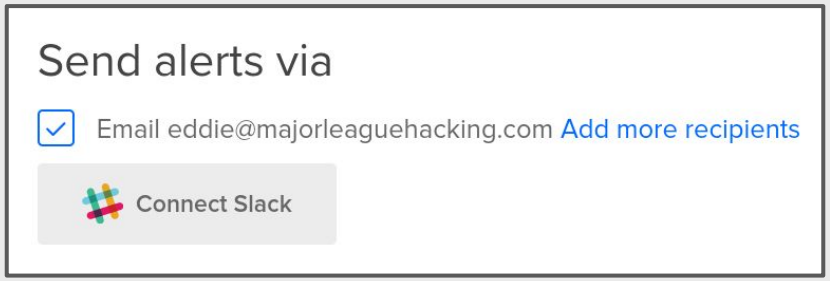
CPU (Central Processing Unit): The circuit in a computer that performs logical and mathematical operations.

Create an Alert Policy

Search for your Droplet name and add it to the Alert Policy.



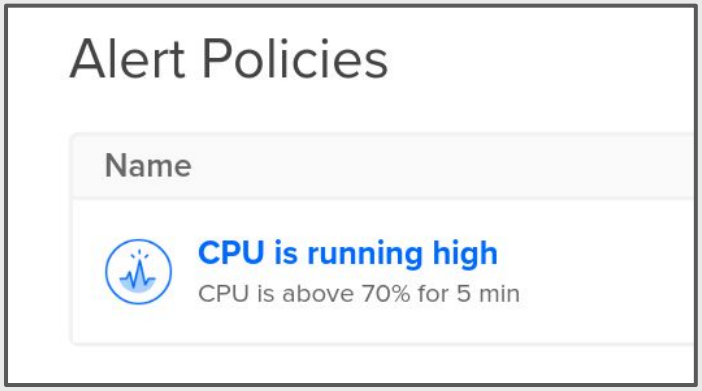
Create an Alert Policy



Click Add more recipients and add your email address.

Lastly, click **Create Alert Policy**.

Create an Alert Policy



Our Alert Policy is live!

Now, if our Droplet is using more than 70% of its CPU, you'll receive a notification via email.

This mean you can take preemptive action before it causes any problems.

DevOps Principles: Security

Let's focus on another one of the core DevOps principles: **Security**.

Our application is online and could be subject to remote attack over a network.

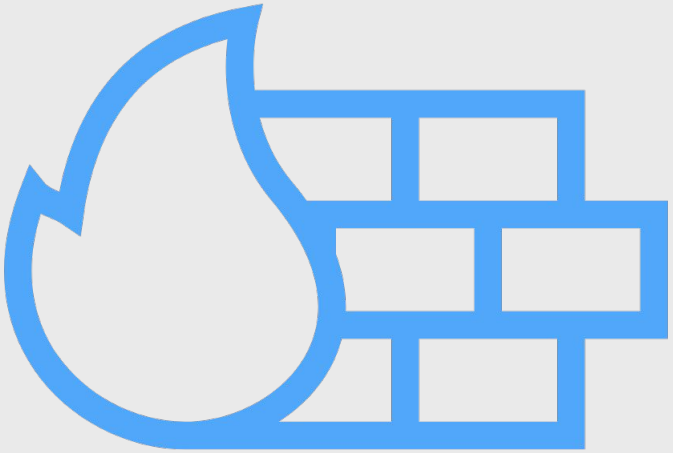
How can we combat this?

One way is through creating a **Firewall**.

Adding a Firewall

Firewalls control what traffic is allowed to access a server.

If the traffic isn't allowed, it's dropped by the Firewall.



Key Term

Firewall: A security service that monitors incoming and outgoing network traffic and uses rules to block unwanted traffic.

Ports

Your app has an IP address - its network home.

Each IP address has 2^{16} (more than 65,000!) possible **ports** that can connect to this IP address.

A port is a gateway to a network address and helps us identify what kind of information is being sent.

Some ports are reserved - for example, port 80 represents HTTP, or the world wide web.

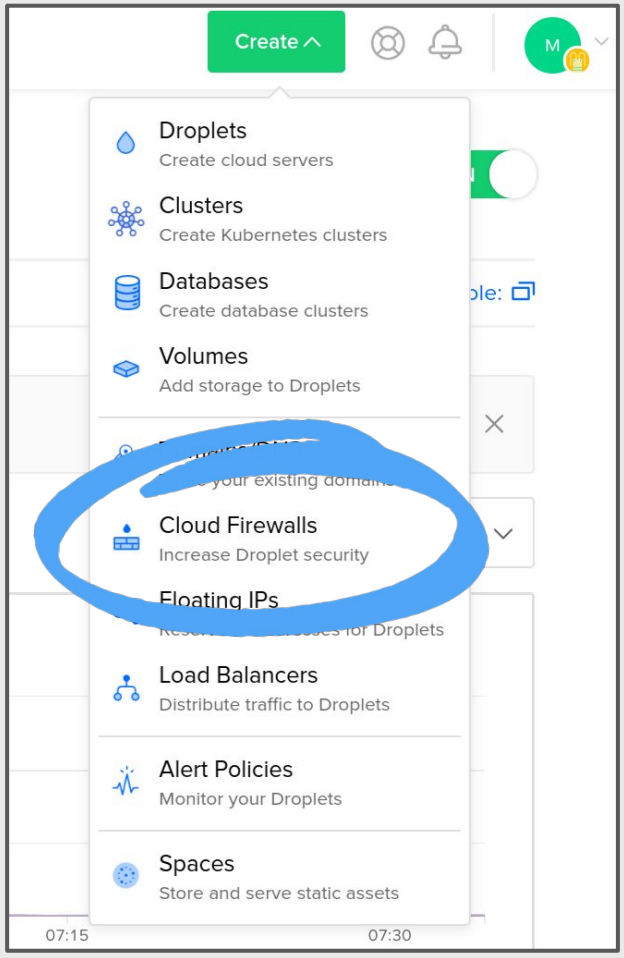
We can write rules for Firewalls which can block or allow access to specific ports.

Adding a Firewall

First, we'll demonstrate what a Firewall can do by creating one with **no rules**: so all traffic will be blocked!

Then, we'll create a rule so that users can only access our app's port, 3000.

Adding a Firewall



Start by heading to the DigitalOcean dashboard and click **Create** -> **Cloud Firewall**.

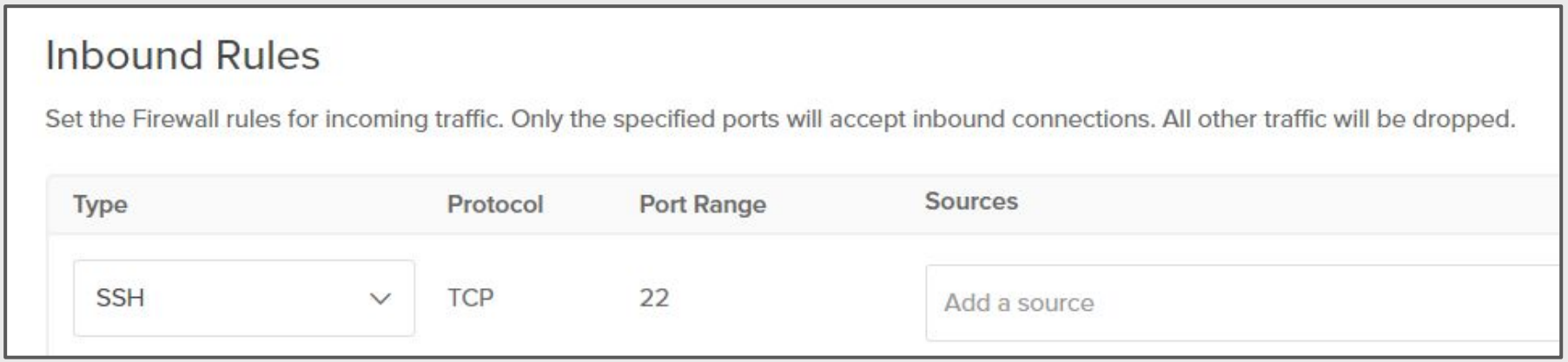
Adding a Firewall

First of all, give your firewall a name.

Configuring the Firewall

In the **Inbound Rules** section, remove IPv4 and IPv6 from **Sources**.

This means that our Firewall isn't going to let any traffic though!



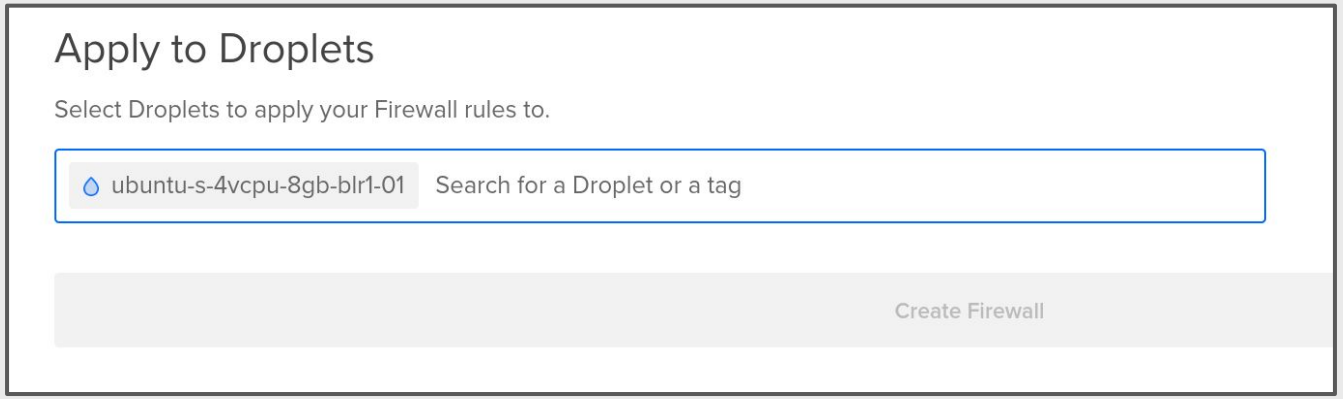
Key Term

IPv4/IPv6: Different network protocols which represent IP addresses. IPv4 uses 32 bits to store an address, giving 4.3 billion different possible addresses - and we're running out! IPv6 was introduced, which uses 128 bits per address.

Configuring the Firewall

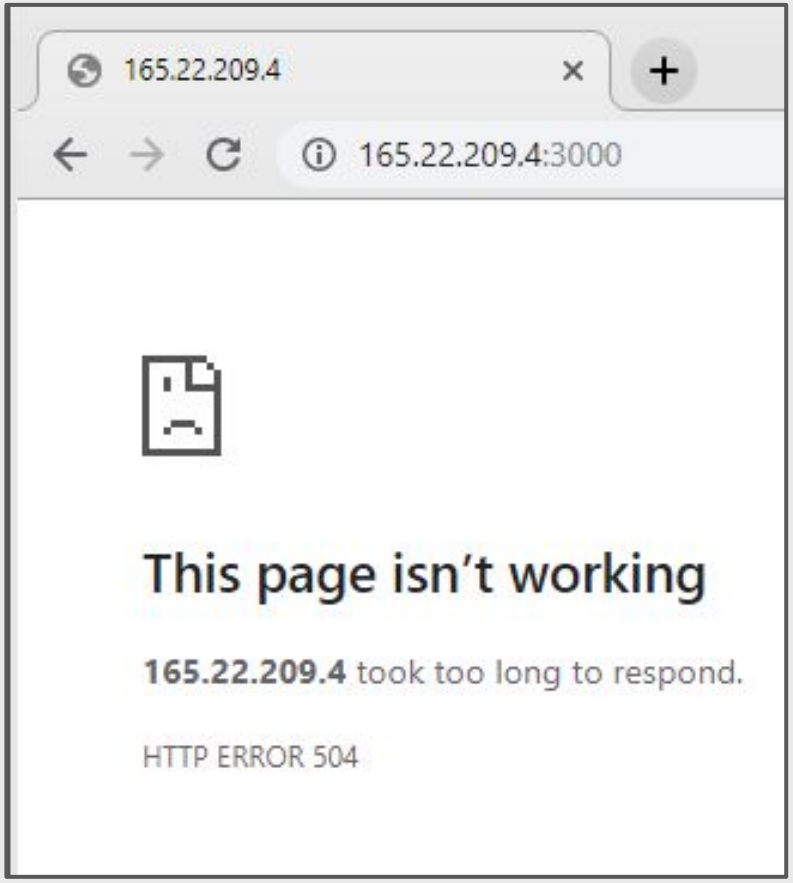
Lastly, select the Droplets you'd like to apply the Firewall to.

Search for your Droplet, and click **Create Firewall**.



Return to your application's IP address in your browser and try to load it again. What happens?

Testing the Firewall

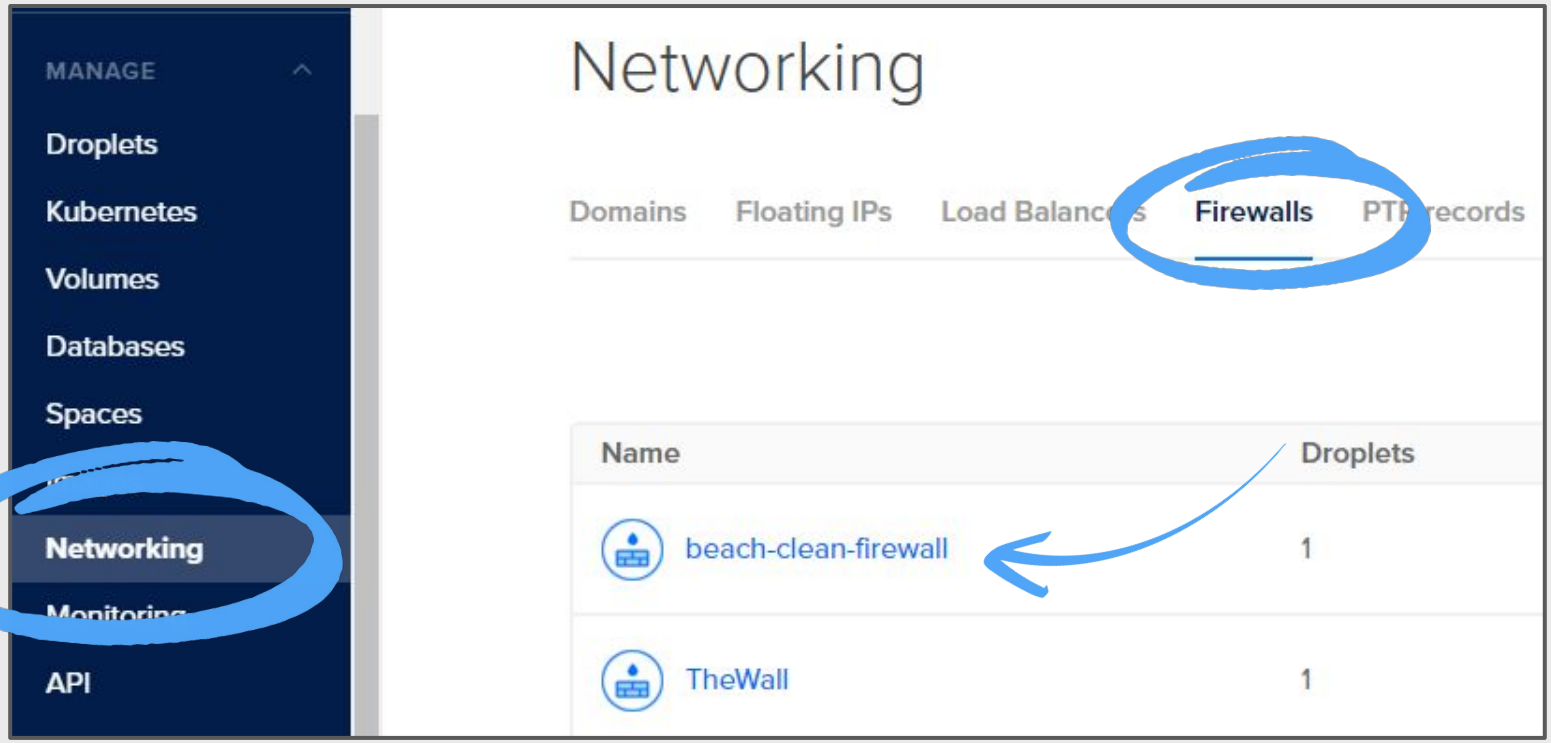


Our Firewall is successfully blocking all traffic, meaning that any requests made to our application are blocked!

We've demonstrated that our Firewall works, but let's add some rules so we can let some traffic through and still access our app.

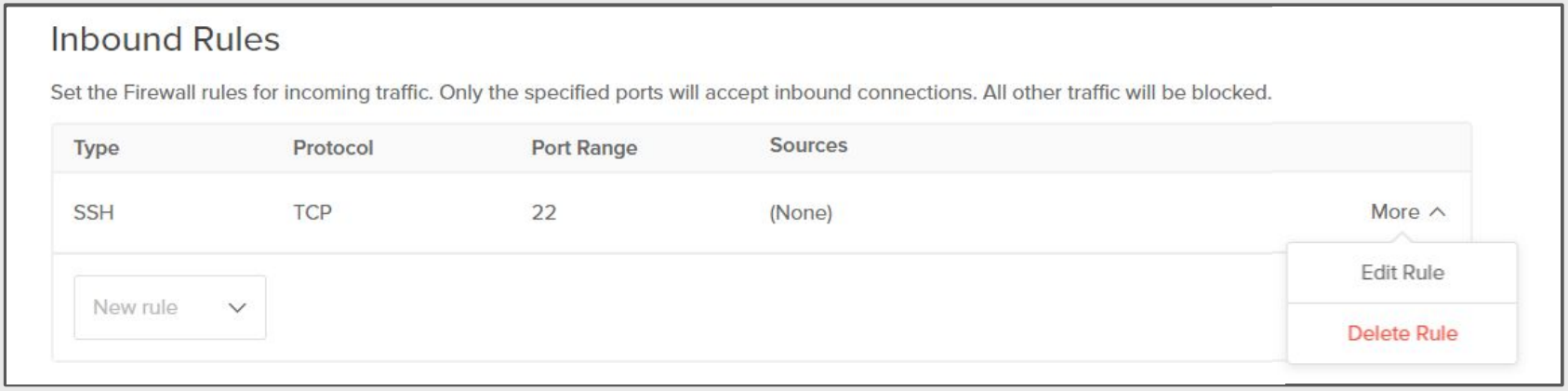
Re-Configuring the Firewall

On your Dashboard, click the Networking option, then select the Firewalls tab. Click on the Firewall you just made.



Re-Configuring the Firewall

In the **Inbound Rules** section, select **More** and then **Edit Rule**.

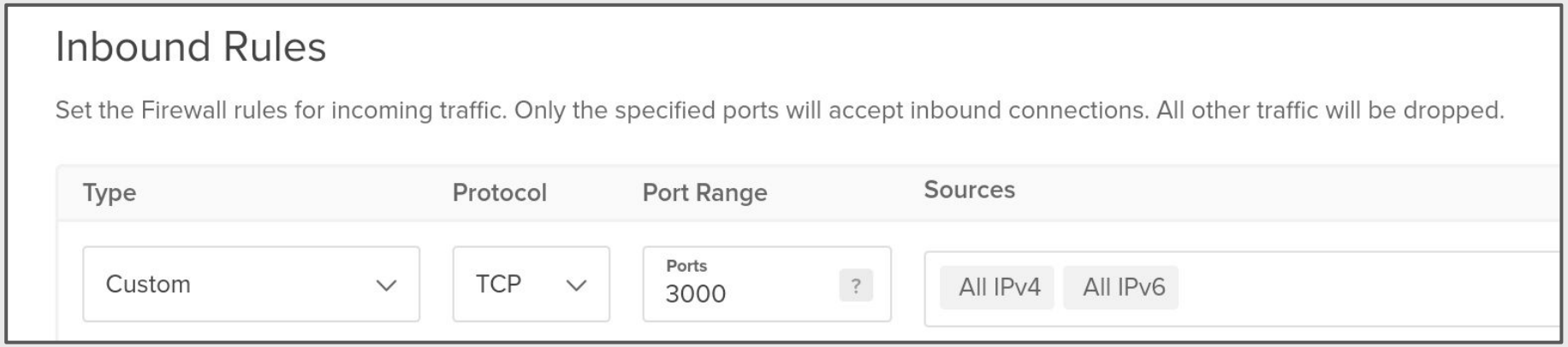


Re-Configuring the Firewall

Change the Type to **Custom** and the Ports to **3000**.

This means that our Firewall will only allow traffic to access our server if it travels through Port 3000.

In Sources, add **All IPv4** and **All IPv6**.



Re-Configuring the Firewall

Add another rule, and select **SSH**.

This will allow us to connect to our application remotely via secure shell,

Accept the default values and click **Save**.

Testing the Firewall

Now try refreshing your app.

Your Firewall rule has updated and allowed traffic on Port 3000, so your app is accessible again!

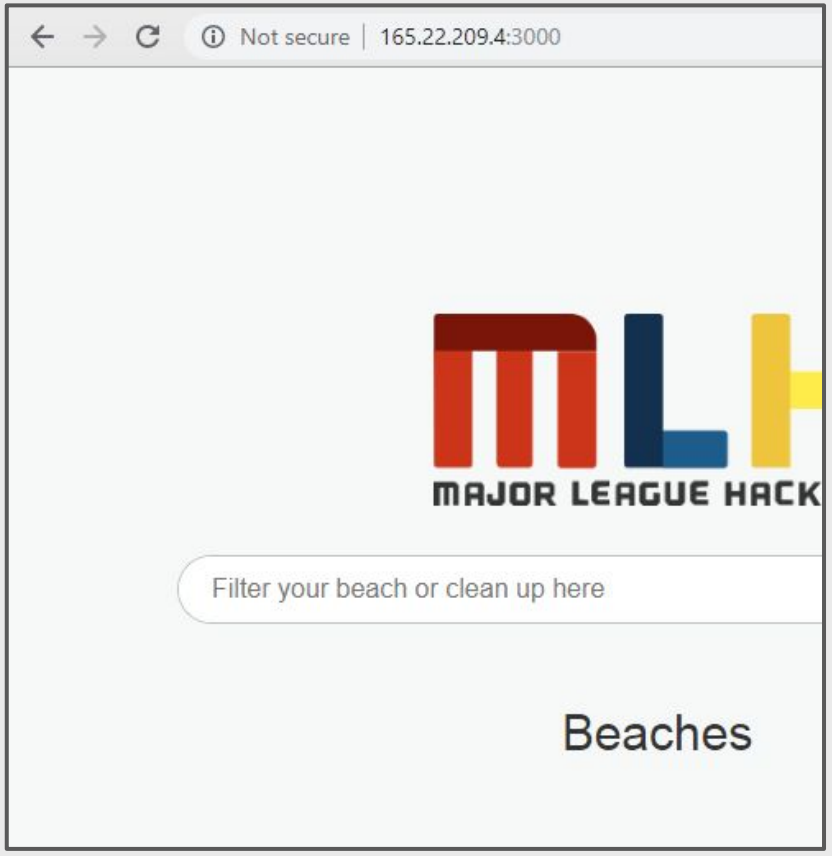


Table of Contents

Let's Recap

What did you learn?

1. DigitalOcean provides Cloud Computing services and allows us to host applications in Droplets.
2. DevOps is the practise of deploying, securing and updating software.
3. We can add features such as Firewalls, Load Balancers and Alert Policies to our Droplets.

What did you learn today?

We created a fun quiz to test your knowledge and see what you learned from this workshop.

http://mlhlocal.host/quiz

Learning shouldn't stop when the workshop ends...

Check your email for access to:

- These workshop slides
- Practice problems to keep learning
- Deeper dives into key topics
- Instructions to join the community
- More opportunities from MLH!

Next Steps

1. Investigate other Alert Policies you could add to your Droplet.
2. Research **Automatic Deployment** - where the app would reload every time you make a change to a GitHub repository.
3. Try changing your Alert Policy notifications from email to Slack!


MLH localhost
Workshop
App Deployment & Security with DigitalOcean
MLH localhost
DigitalOcean