DigitalOcean Tutorial

This is a tutorial on setting up a server on DigitalOcean, a cloud computing platform. We will go through the whole process, from creating an account to setting up a server and connecting to it.

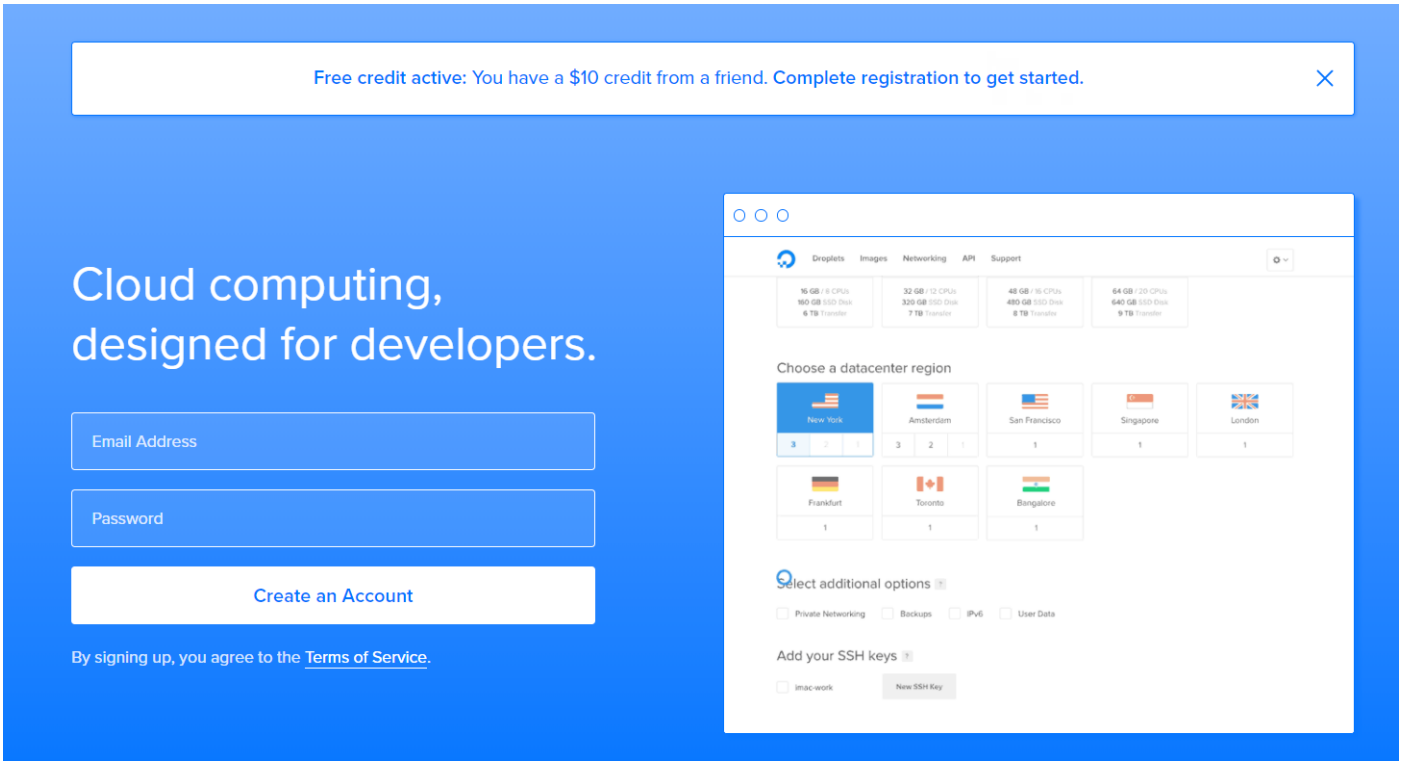
Are you looking for a guide on deploying your Flask application to your server? Check out this guide instead!

DigitalOcean

DigitalOcean is a cloud infrastructure provider focused on simplifying web infrastructure for software developers. It allows you to rent servers with different performance at different cost. For more detailed information, you may refer to the official website help page here.

Creating an account

You can sign up to DigitalOcean using our affiliate link. Doing so gives you a starting credit of $10, which allows you to rent a server for 2 months. Click this link to create your account and get the $10: https://m.do.co/c/d54c088544ed. If the link doesn't work, paste it into your browser.



After clicking the link, you should see a page like the above. Create your account at the left-bottom corner and you'll receive the $10 automatically. Beware that you'll be asked to provide payment info when creating the account, since all services (which you'll choose below) in DigitalOcean will be charged after your credit runs out.

Creating a Droplet

A server instance in DigitalOcean is called a `Droplet`. It's just a name that may vary in different platforms, for example, `Dyno` in **Heroku** and `EC2` in **AWS** (Amazon Web Services). Below are the steps to create a Droplet.

1. Choose an image

To create a Droplet, we must first specify an image. This essentially means choosing what

Operating System you want the server to run. We recommend you use an Ubuntu LTS (Long Term Support) distribution. For more info on Ubuntu life time, please refer to the official Ubuntu end of life page here. In our example, we'll use `Ubuntu 16.04.* x64`, which is an LTS distribution.

2. Choose a size

Next, we need to choose the specs for our server. In this tutorial, we'll be using the most basic tier of a Standard Droplet, which offers a single CPU with 512MB RAM, 20GB SSD and 1000 GB transfer at $5 per month. Generally, it's more than enough for running personal applications. You may also run several services in a single Droplet.

3. Choose a datacenter region

Generally, choosing a region that's **closest to your users** will make their interactions with your service faster. If your users are primarily in the United States, you could choose a United States-based Droplet.

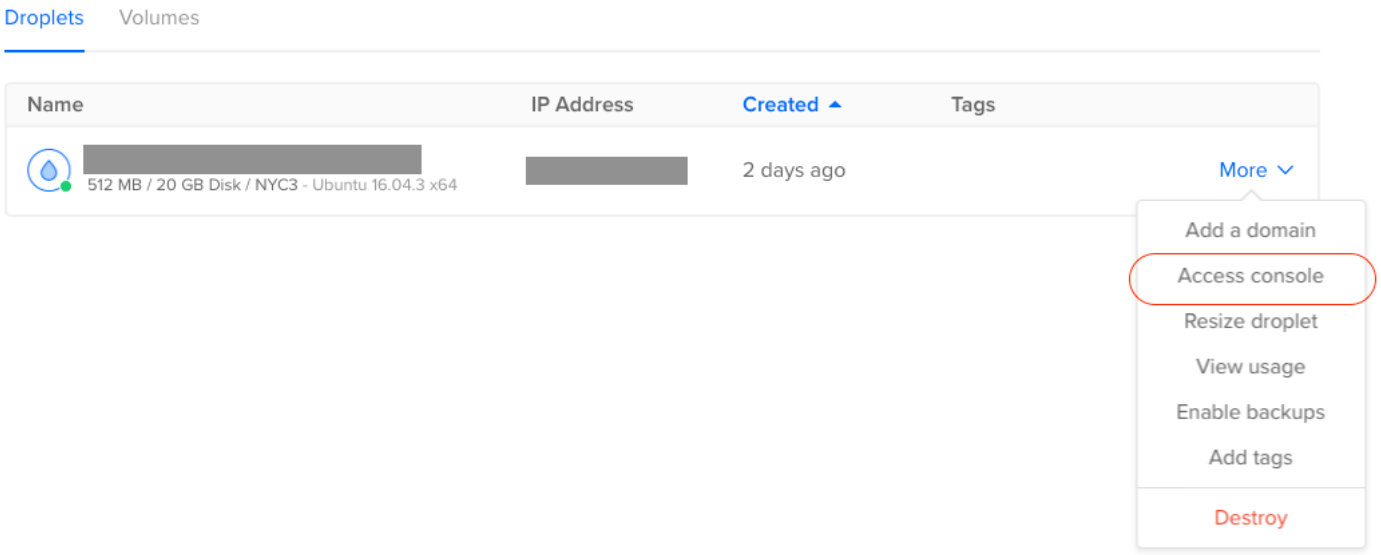
Other settings

In our example, we do not need to add any other services such as block storage or private network. So we will ignore these settings to keep our setup simple and cheap.

You may choose to use an SSH key or you can just leave it unchecked and a password will be generated for you. If you choose to use SSH key, then each time you want to login to the server, you will need to provide the key. Since you have to have access to the SSH key whenever you log in, it can be more secure but also a bit limiting.

Finally, you may change the name of your Droplet to something you like and then click **Create** to create and launch your Droplet.

Managing our droplet



Once you've created your Droplet, you will receive an e-mail containing your login (`root`) and initial password for it. Now click the "Access Console" option in the dropdown menu associated with your Droplet as shown in above image.

Changing your password

Now you will be shown a console connected to your Droplet. Use the login `root` and the password received in your email to login. Then you will be ask to provide your password again and change it to a new password.

Notice that all password field will not show any modification when you are typing. There is nothing wrong with your console, it's just a UNIX security feature. After changing your password, you are now logged in as the `root` user on your server.

If you have successfully followed the tutorial so far, then you have finished setting up your DigitalOcean server!

Deploying an application

After setting up our server, we may want to deploy an application onto it. This is covered in a separate tutorial: Python App Deployments with nginx and uWSGI.

We have organized the contents in such way because deployment is an independent process

from setting up your server, and it should work in similar ways for most platforms as discussed earlier in this document.

Thanks for reading!

— Jose and the Teclado team.