Deploying Node.js App to DigitalOcean Server

Hayk Simonyan
Senior Full Stack Engineer
34 articles + Follow

March 23, 2023

Open Immersive Reader

If you're looking to deploy a Node.js application on a server then this is a very simple guide for you to deploy it step-by-step with me. You can sign up to the Digital Ocean with the link that I will provide and get 100$ free credit or just use any cloud provider that you prefer.

We will set up a secure server together. I have divided the process into 10 small steps so it's easy to follow along:

1. Setup SSH keys
2. Setup a digital ocean account and droplet (server)
3. SSH into a server
4. Install Node.js & Git on the server
5. Clone your git repository
6. Keep your app always running with the PM2 process manager
7. Enable ufw firewall

8. Use Nginx as a reverse proxy to run your app on port 80
9. Create a domain and connect it to your server
10. Create SSL Certificate and enable HTTPS

1. Setup SSH keys

If you're a Windows user then here's **How to generate SSH key in Windows**. Otherwise, if you're using macOS or Linux it's very simple. First, open your terminal and navigate to the *.ssh* folder

> cd ~/.ssh

run this command with your email address

> ssh-keygen -C "**your_email_address@gmail.com**"

it will ask you for the key name, you can select the default name **id_rsa** but I will recommend setting up a separate named key for your server (like **id_rsa_digitalocean**)

> Enter file in which to save the key (/Users/admin/.ssh/id_rsa): /Users/admin/.ssh/id_rsa_digitalocean

you can enter a passphrase for extra security or just leave it empty and hit enter

> Enter passphrase (empty for no passphrase):

> Enter same passphrase again:

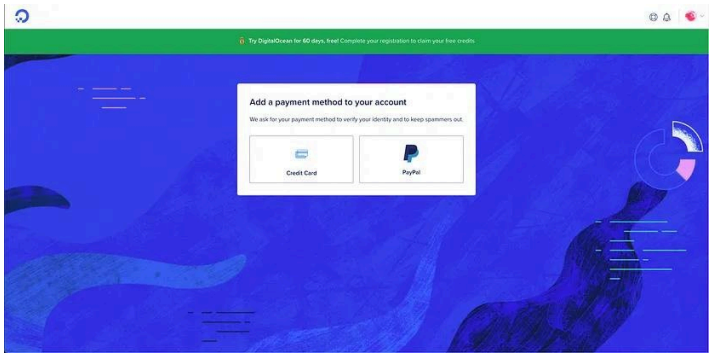
that's it, we now have SSH keys to create a connection with our server. Just to double-check run **ls** in **~/.ssh** folder and make sure that you see **id_rsa_digitalocean** and **id_rsa_digitalocean.pub** keys.

If you're not sure how SSH works, here's my very simple explanation on this topic **What is SSH and how does it**

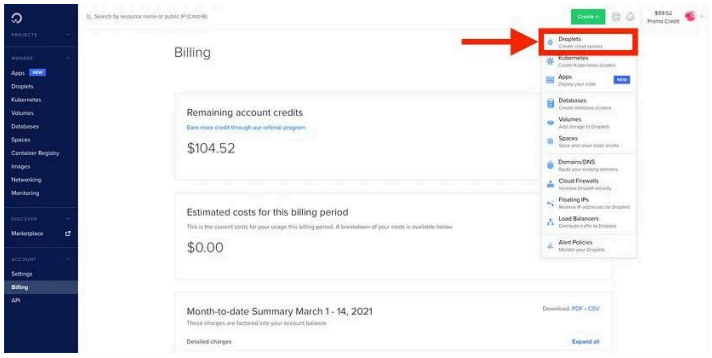
work?

2. Setup a DigitalOcean account and droplet (Server)

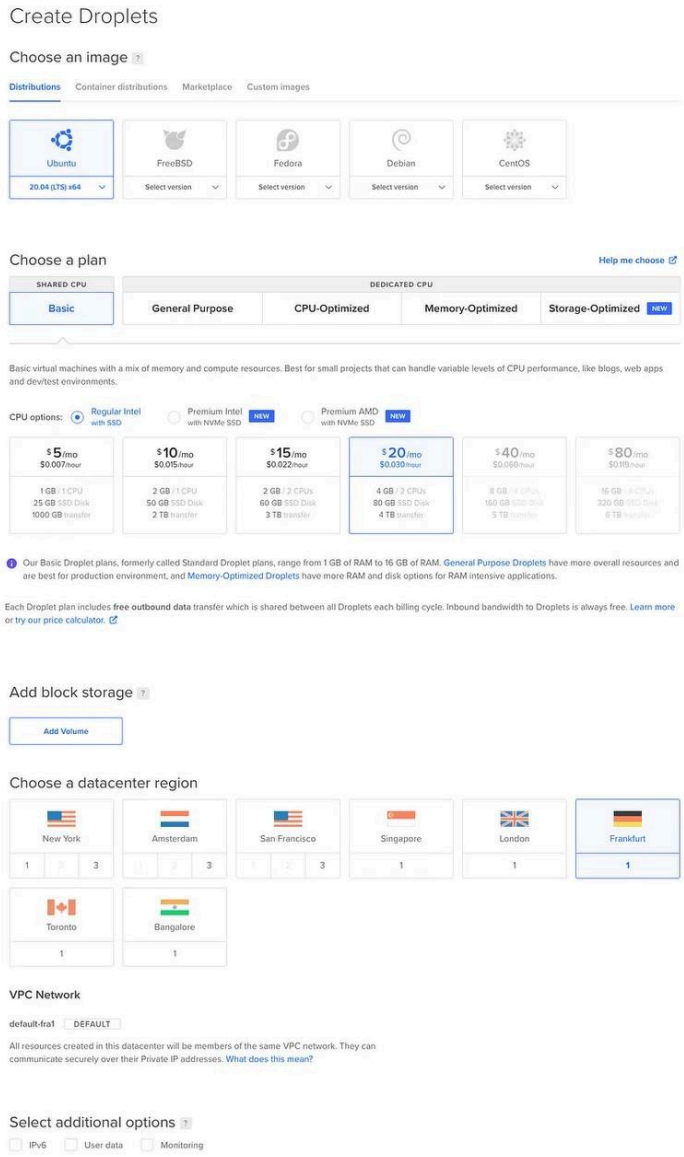
Create your account at DigitalOcean if you don't have one (use this link for 100$ free credit if you want). Once you sign up it will ask you to provide a payment method



After adding a payment method you'll be redirected to the **Billing** page



Simply click on **Create/Droplet** from the top right corner. Droplet is just a server, it's a Linux-based virtual machine.

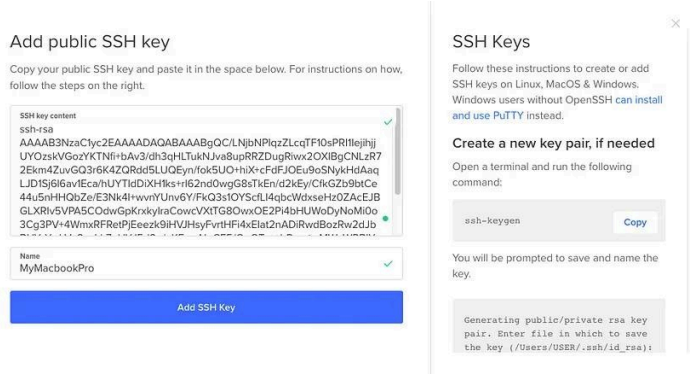


Choose OS, plan as per your needs, and select a region that is closer to your users.

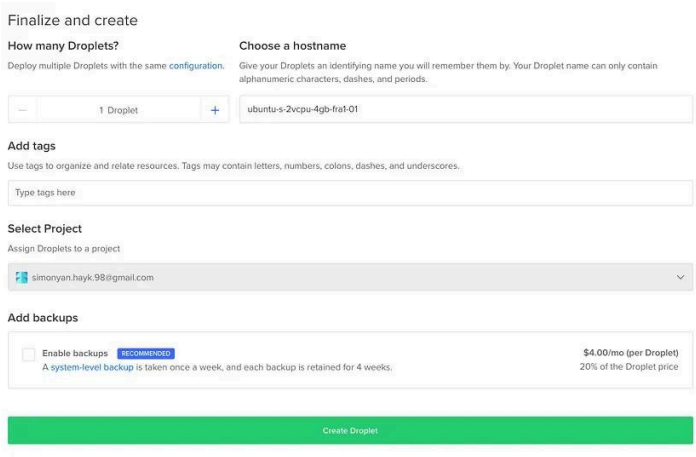
And now for the authentication part, we'll need the public key (**id_rsa_digitalocean.pub**) that we've just created. Copy the public key with the following command

> pbcopy < ~/.ssh/id_rsa_digitalocean.pub

Click on "New SSH Key" and paste your public key, "Add SSH Key"



Choose a hostname and click on “Create Droplet”



We’ve successfully created a droplet now.

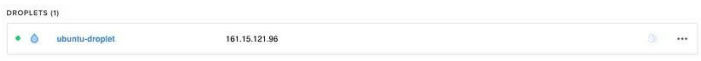
3. SSH into a server

Let’s SSH into your server. First, go to your terminal and add identity to access the server (this time we need the private key)

> ssh-add ~/.ssh/id_rsa_digitalocean

> Identity added:
> /Users/admin/.ssh/id_rsa_digitalocean

You’ll be redirected to the newly created droplet, copy the droplet ID (in my case 161.15.121.96), and log in to your server with that ID and the root user.



> ssh root@161.15.121.96

> Welcome to Ubuntu 20.04.1 LTS (GNU/Linux 5.4.0–51-generic x86_64)

> root@ubuntu-droplet:~#

We're now in our digital ocean server.

4. Install Node.js & Git on the server

Now that we're inside of the server let's install Node and Git on it.

- Installing Node.js

I'm installing version 14, if you need another version just change the 14 to that version

> root@ubuntu-droplet:~# curl -sL https://deb.nodesource.com/setup_14.x | sudo -E bash -

and then

> root@ubuntu-droplet:~# sudo apt install nodejs

check if it's installed now

> root@ubuntu-droplet:~# node --version

> v14.16.0

> root@ubuntu-droplet:~# npm --version

> 6.14.11

- Installing Git

> sudo apt install git

check if it's installed

> root@ubuntu-droplet:~# git --version

> git version 2.25.1

5. Clone your git repository

> root@ubuntu-droplet:~# git clone your_project_link

Install dependencies

> root@ubuntu-droplet:~# cd your_project

> root@ubuntu-droplet:~/your_project:~# npm i

6. Keep your app always running with the PM2 process manager

Install PM2 process manager globally on your server

> root@ubuntu-droplet:~# sudo npm i pm2 -g

now try to run your app with PM2

> root@ubuntu-droplet:~# cd your_project

> root@ubuntu-droplet:~/your_project# pm2 start server.js (or app.js depending on your main file)

> [PM2] Starting /root/money-manager-api/server.js in fork_mode (1 instance)

> [PM2] Done.

Now our application is always running in the background.

Set your app to start when even the server is rebooted

> root@ubuntu-droplet:~/your_project# pm2 startup ubuntu

7. Enable ufw firewall

Now let's enable the ufw firewall which will enable SSH (port 22), HTTP (port 80), HTTPS (port 443). Check firewall status, by default it should be inactive

> root@ubuntu-droplet:~# ufw status

> Status: inactive

Enable it with this command

> root@ubuntu-droplet:~# ufw enable

> Command may disrupt existing ssh connections. Proceed with operation (y|n)? y

> Firewall is active and enabled on system startup

Allow SSH, HTTP, and HTTPS

> root@ubuntu-droplet:~# ufw allow ssh

> root@ubuntu-droplet:~# ufw allow http

> root@ubuntu-droplet:~# ufw allow https

Now check the ufw status it must be active with SSH, HTTP, HTTPS allowed

> root@ubuntu-droplet:~# ufw status

> Status: active

> To Action From

> — — — — — —

> 22/tcp ALLOW Anywhere

> 80/tcp ALLOW Anywhere

> 443/tcp ALLOW Anywhere

> 22/tcp (v6) ALLOW Anywhere (v6)

> 80/tcp (v6) ALLOW Anywhere (v6)

> 443/tcp (v6) ALLOW Anywhere (v6)

8. Use Nginx as a reverse proxy

Install Nginx on the server with this command

> root@ubuntu-droplet:~# sudo apt install nginx

> Do you want to continue? [Y/n] y

Now let's edit the config file in server /etc/nginx/sites-available/default and set up a reverse proxy so that when we go to port 80 it will load our app that is running on port 5000 (or change 5000 if you're using another port). And while we're here you can also set up your domain name if you're going to connect this server to a domain.

> root@ubuntu-droplet:~# sudo nano /etc/nginx/sites-available/default

this will open that file for editing, find **server_name** and **location** lines here and replace them

```
server_name yourdomain.com www.yourdomain.com;

location / {
        # First attempt to serve request as
file, then
        # as directory, then fall back to
displaying a 404.
```

proxy_pass http://localhost:5000; **#or** your app port

```
    proxy_http_version 1.1;
            proxy_set_header Upgrade $http_upgrade;
            proxy_set_header Connection 'upgrade';
            proxy_set_header Host $host;
            proxy_cache_bypass $http_upgrade;
    }
```

Once you're done press **Ctrl+X** it will ask you if you want to save type **y** and **enter.** Check if the Nginx config file is ok and restart Nginx

> root@ubuntu-droplet:~# sudo nginx -t

> nginx: the configuration file /etc/nginx/nginx.conf syntax is ok

> nginx: configuration file /etc/nginx/nginx.conf test is successful

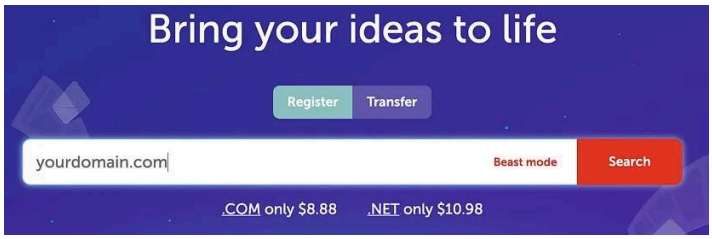
> root@ubuntu-droplet:~# sudo service nginx restart

Now you can make a request to your IP address (161.15.121.96) with port 80 and it will redirect you to whatever port your app is running on (in my case 5000). In fact, all other ports now are disabled except 80 and 443.

9. Create a domain and connect it to your server

I'll be using **namecheap** for domain registration and I'll suggest you go with namecheap, but you can also use **freenom.com** for a free domain or any other domain provider.

Let's go to the namecheap and register a new domain. Search for a domain that you want, add it to your cart, and go to the checkout.

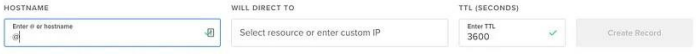


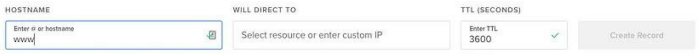
Thank you for your purchase!

Once you're done with domain registration go to the Networking tab on the digital ocean

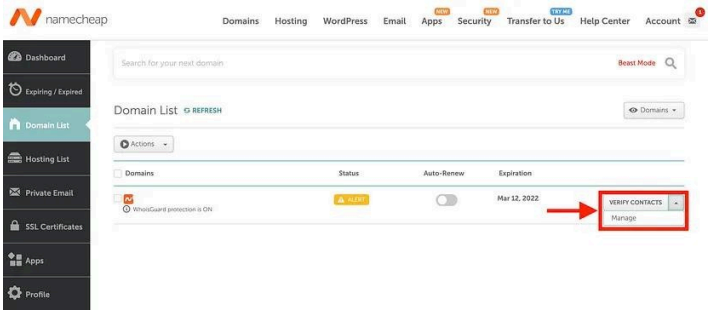
Enter your domain and click “Add Domain”.

Point the root (@) and www to your droplet here and “Create Record”.

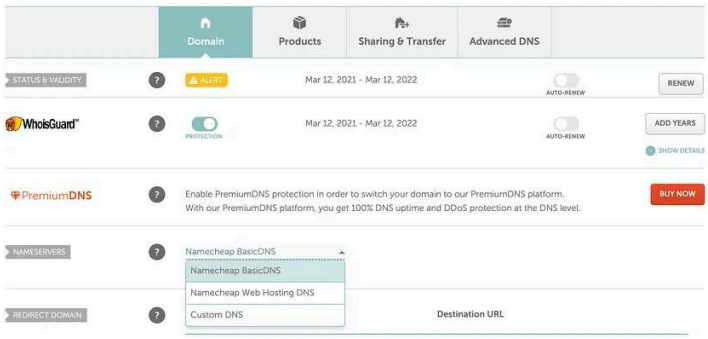




Now go back to Namecheap **Account/Domain List** and click “Manage” on your domain



Select Custom DNS here, add ns1.digitalocean.com, ns2.digitalocean.com., ns3.digitalocean.com, and save. (keep in mind that **DNS server update may take up to 48 hours to take effect**)



10. Create SSL Certificate and enable HTTPS

Just go through these commands one-by-one

> root@ubuntu-droplet:~# sudo add-apt-repository ppa:certbot/certbot

> root@ubuntu-droplet:~# sudo apt update

> root@ubuntu-droplet:~# sudo apt-get install python3-certbot-nginx

> root@ubuntu-droplet:~# sudo certbot --nginx -d yourdomain.com -d www.yourdomain.com

for the last command it will ask for your email address and to select the appropriate number [1–2] just type 2 and hit enter to enable HTTPS.

Now if you visit your domain you'll see the secure icon on the left which means that the connection is secure and HTTPS is enabled.

> Congratulations 🎉

> You've successfully deployed your Node application to Digital Ocean server

#deployment #digitalocean #nodejs #deploying #droplet

Full Stack Web Developer | Software Engineer | ***Turning Business Challenges into Digital Solutions***

what if we use mongodb data base in our projects ? how do we proceed then ?

Like | Reply · 2 Replies

Hayk Simonyan • 3rd+ 6mo •••
Senior Full Stack Engineer

kheireddine terfa in that case I would set up mongodb atlas and connect to the server through the atlas URI. But if you want it to be on the same server (which i don't recommend) you need to ssh into the droplet and install mongodb there

Like · 1 | Reply

kheireddine terfa • 3rd+ 6mo •••
Full Stack Web Developer | Software Engineer | ***Turning Business Challenges into Digital Solutions***

Hayk Simonyan okey , thank you Hayk Simonyan

Like | Reply

Luan Albineli Pinto • 3rd+ 3mo •••
Full Stack Engineer at Stuart

Worked like a charm! Thanks!

Like | Reply

Show 3 more comments

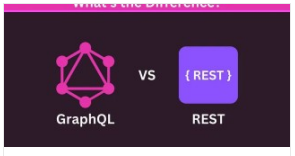
Hayk Simonyan
Senior Full Stack Engineer

+ Follow

More from Hayk Simonyan

Beginner's Guide to Prompt Engineering with ChatGPT
Hayk Simonyan on LinkedIn

Functional Programming Simplified
Hayk Simonyan on LinkedIn



REST vs GraphQL
Hayk Simonyan on LinkedIn

See all 34 articles