



2. Setting up a VPS:

Server settings:

The server settings depend on the load your site will be under and the complexity of the project. Since I will be deploying a simple site, I chose the most minimal settings.

OS Marketplace Custom images

Version: 22.10 x64

CPU options:

- ☒ Regular (Disk type: SSD)
- ☐ Premium Intel (Disk: NVMe SSD)
- ☐ Premium AMD (Disk: NVMe SSD)

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

Show all plans

For the authentication method, I chose a password.

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

Create root password *

.....

Once our server is up and running, we can start connecting to it.

test

Image	Size	Region	IPv4	IPv6	Private IP	VPC
Ubuntu 22.10 x64	1 vCPU 1GB / 25GB Disk (\$6/mo) Resize	FRA1	164.90.218.8	Enable	10.114.0.2	default-fra1

Copy the IPv4 or IPv6 address, then open a terminal and enter the following command:

Ssh root@164.90.218.8(instead of 164.90.218.8 put your IPv4)

After this, it will ask you to enter the password that you set when creating the VPS:

```
Warning: Permanently added '164.90.218.8' (ED25519) to the list of known hosts.
root@164.90.218.8's password: [?]
```

When you successfully log in, you will see something similar to this:

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

System information as of Sat Jul  8 16:06:59 UTC 2023

System load:  0.0      Users logged in:      0
Usage of /:   6.8% of 24.06GB IPv4 address for eth0: 164.90.218.8
Memory usage: 18%      IPv4 address for eth0: 10.19.0.5
Swap usage:   0%       IPv4 address for eth1: 10.114.0.2
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@test:~#
```

From there, we have two ways to transfer our project folder. If you have it in a zip file on your local computer, you can use these commands to transfer the zip file with your project to the server.

```
#if you have authentication method thought password
scp your-archive.zip root@164.90.218.8:~/
#if you have authentication method thought ssh key
scp -i ~/.ssh your-archive.zip root@164.90.218.8:~/
```

However, I suggest that you use GitHub.

Log in to GitHub and use “git clone” to put your project in the global directory. For example, I will use my Node.js project called “qr_code_hunt_challenge”.

```
root@test:~# git clone https://github.com/ImmuneFOMO/qr_code_hunt_challenge.git
Cloning into 'qr_code_hunt_challenge'...
remote: Enumerating objects: 18, done.
remote: Counting objects: 100% (18/18), done.
remote: Compressing objects: 100% (11/11), done.
remote: Total 18 (delta 0), reused 12 (delta 0), pack-reused 0
Receiving objects: 100% (18/18), 6.83 KiB | 1.71 MiB/s, done.
root@test:~# ls
qr_code_hunt_challenge  snap
root@test:~#
```

Then, we need to install packages. Use the following commands on our VPS:

```
#to update everything
sudo apt update
#after it shows us that we have packages that we can upgrade
```

```
sudo apt upgrade
#when you get Do you want to continue? [Y/n], you just write Y
```

After this, go to the git repository using this link and choose the version of Node.js you need. In my case, I will use the latest LTS version.

Press the copy button next to the commands and insert them in our VPS:



```
Node.js LTS (v18.x):

Using Ubuntu

curl -fsSL https://deb.nodesource.com/setup_lts.x | sudo -E bash - &&\
sudo apt-get install -y nodejs
```

```
root@test:~# curl -fsSL https://deb.nodesource.com/setup_lts.x | sudo -E bash - &&\
sudo apt-get install -y nodejs
```

```
root@test:~# node -v
v18.16.1
root@test:~#
```

If you encounter any errors:

```
sudo apt-get update
sudo apt-get install -f
sudo apt-get clean
sudo apt-get install -y nodejs
sudo apt
```

Then:

```
#to install vim, zip, nginx
sudo apt install vim zip nginx
```

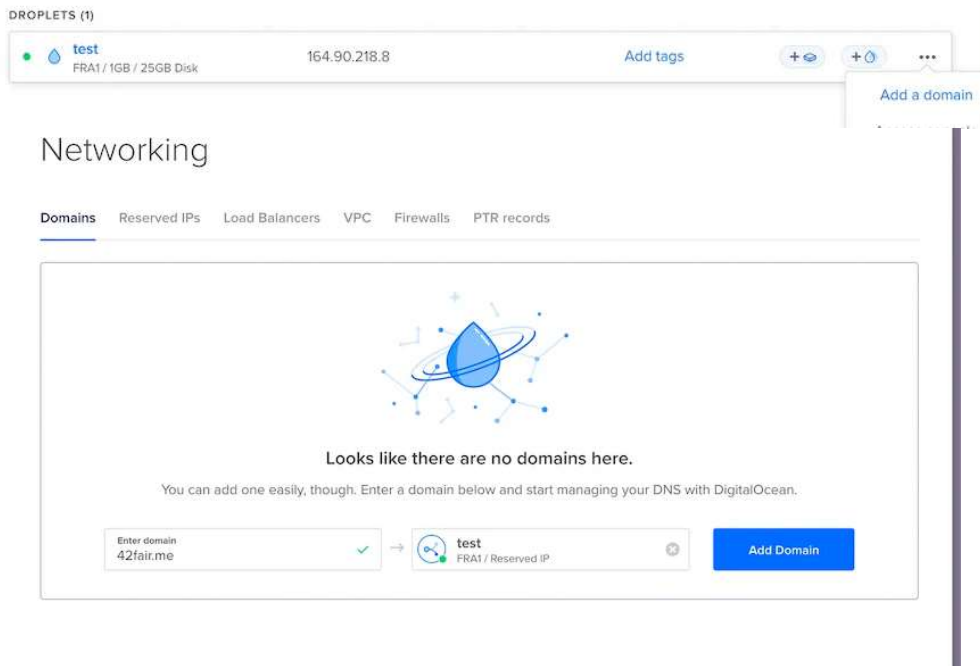
Now, we need to make some changes in the firewall settings:

```
sudo ufw allow 'OpenSSH'
sudo ufw allow 'Nginx HTTP'
sudo ufw enable
#after all this commands you need to write
sudo ufw status
#and you will get something like this
Status: active
```

To	Action	From
--	-----	----
OpenSSH	ALLOW	Anywhere
Nginx HTTP	ALLOW	Anywhere
OpenSSH (v6)	ALLOW	Anywhere (v6)
Nginx HTTP (v6)	ALLOW	Anywhere (v6)

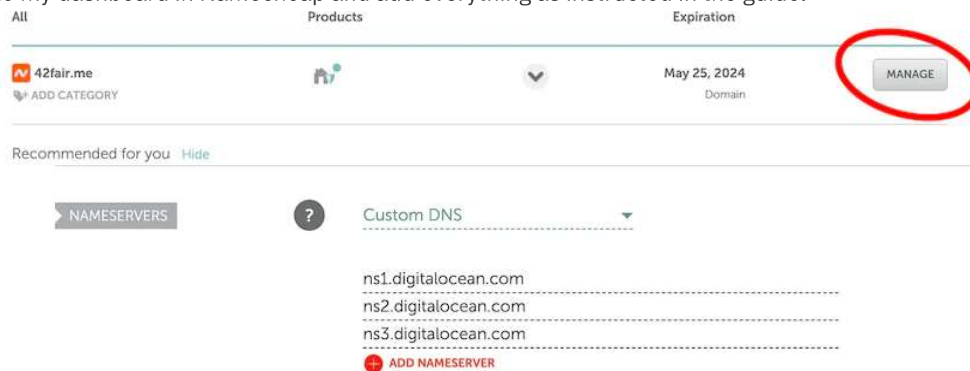
3. Setting up a domain:

First, we need to input our domain on the VPS hosting service. In my case, it is DigitalOcean. Go to the dashboard where you can check your VPS and start:



When you insert the domain name and click “Add Domain,” you need to point to DigitalOcean in your nameservers. You can follow the guide from DigitalOcean using this link.

In my case, I go to my dashboard in Namecheap and add everything as instructed in the guide:



Second, we need to create a directory:

```
sudo mkdir /var/www/your_domain.com
sudo chown -R $USER:$USER /var/www/your_domain.com/
sudo vim /etc/nginx/sites-available/your_domain.com
```

The last command will open Vim, and here we need to insert this Nginx configuration:

```
server {
    server_name your_domain.com www.your_domain.com;

    location / {
        proxy_pass http://localhost:3000;#here you need to change to your 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;
```

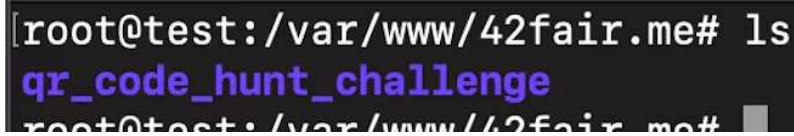
```
}
}
```

Then, use these commands:

```
rm -rf /etc/nginx/sites-available/default
rm -rf /etc/nginx/sites-enabled/default
sudo ln -s /etc/nginx/sites-available/your_domain.com /etc/nginx/sites-enabled/your_domain.com
#by next command you can check if everything okay
nginx -t
#now we need to reload our nginx
nginx -s reload
```

Now, we need to move our Node.js project to another directory:

```
cd /var/www/your_domain.com
mv ~/name_of_your_project_folder
```

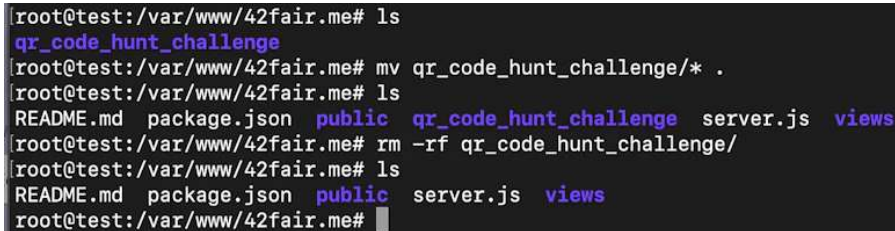


```
[root@test:/var/www/42fair.me# ls
qr_code_hunt_challenge
root@test:/var/www/42fair.me#
```

After using “ls,” you will see the folder or “.zip” file in your directory. If you just moved the “.zip” file, you need to unzip it using the command “**unzip your_file.zip.**”

Now, we need to move all files from this directory:

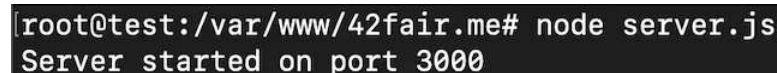
```
mv name_of_your_project_folder/* .
#and now we can delete this directory
rm -rf name_of_your_project_folder
```



```
root@test:/var/www/42fair.me# ls
qr_code_hunt_challenge
root@test:/var/www/42fair.me# mv qr_code_hunt_challenge/* .
root@test:/var/www/42fair.me# ls
README.md package.json public qr_code_hunt_challenge server.js views
root@test:/var/www/42fair.me# rm -rf qr_code_hunt_challenge/
root@test:/var/www/42fair.me# ls
README.md package.json public server.js views
root@test:/var/www/42fair.me#
```

After that, use these commands:

```
npm install
#and if you need some packages you can install now, in my case i will use:
npm init
npm install express
npm install ejs
npm install path
#and when you installed everything that you need, you can start your node.js
node server.js
```



```
[root@test:/var/www/42fair.me# node server.js
Server started on port 3000
```

To make this site work without stopping:

```
sudo npm install pm2@latest -g

pm2 startup systemd

pm2 start /var/www/your_domain.com/server.js

pm2 save

pm2 status
#after last command you should see
```



```
root@test:/var/www/42fair.me# pm2 status
```

id	name	namespace	version	mode	pid	uptime	u	status	cpu	mem	user	watching
0	server	default	1.0.0	fork	4090	2m	0	online	0%	53.7mb	root	disabled

```
root@test:/var/www/42fair.me#
```

If everything is okay, start using these commands:

```
sudo systemctl start pm2-root

sudo systemctl reboot
```

4. Setting up HTTPS:

To use HTTPS on our app, use these commands:

```
sudo ufw allow 'Nginx Full'

sudo apt install certbot python3-certbot-nginx

sudo certbot --nginx -d your_domain.com -d www.your_domain.com

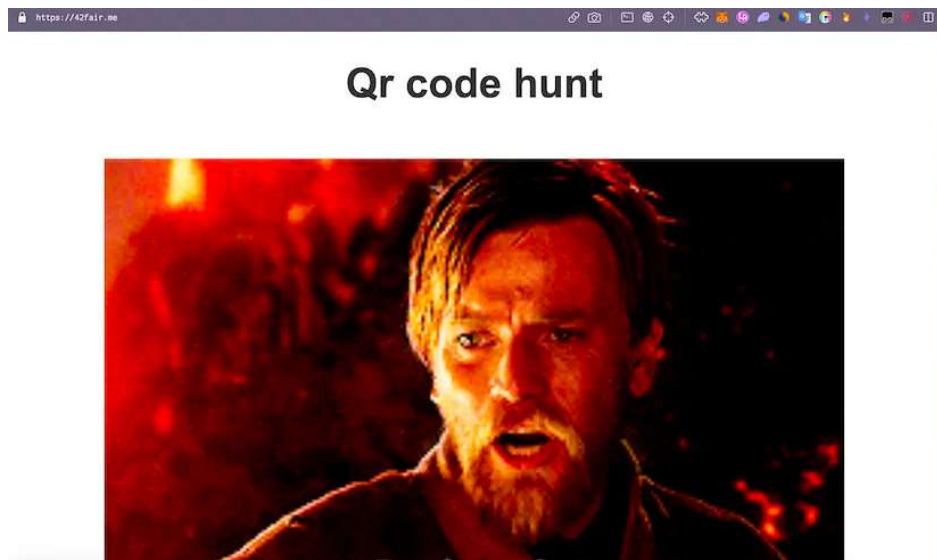
sudo systemctl status certbot.timer

sudo certbot renew --dry-run

sudo ufw delete allow 'Nginx HTTP'

#and if you upload some file on your app
sudo vim /etc/nginx/nginx.conf
#and here you need to insert in http {}
client_max_body_size 1000M; #here you can input any amount of megabytes

#and after all this
nginx -t && nginx -s reload
```

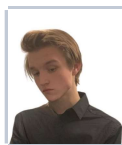


Now, our Node.js app is deployed and reachable through our domain, and everything is working.

Finishing Up

I hope you understood the instructions correctly. If you have any thoughts on this article, I will be waiting for your comments. If I really helped you, I kindly ask you to subscribe to my **LinkedIn** and provide feedback on this article.

Also published [here](#).



by Anatolii Zhadan [@zhadan](#).

Anatoly - Tech Enthusiast | Programmer | Web Dev | Learning C @42 Lisbon 🚀 Medium:
https://medium.com/@Anatolii_Zhadan

[Read my stories](#)

Comments

Add Comment