User Guide

Introduction

This guide will show you how to set up the Mellivora honeypot management web app on an Ubuntu cloud device. In this case we are using Digital Ocean, but the stages should be similar for other cloud hosts (or even personal systems).

This guide assumes that you know how to use SSH and are somewhat experienced using Linux commands.

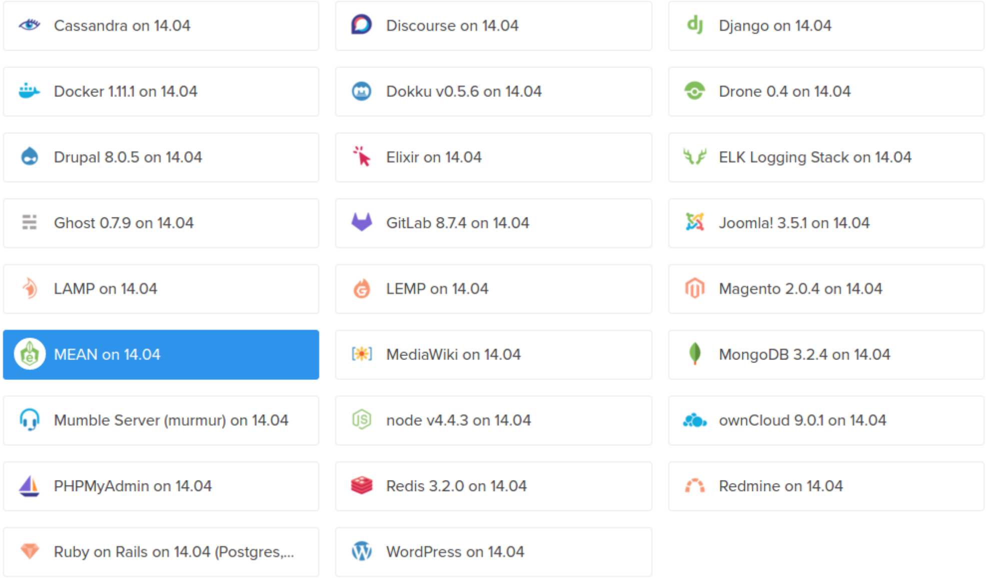
Starting the Server

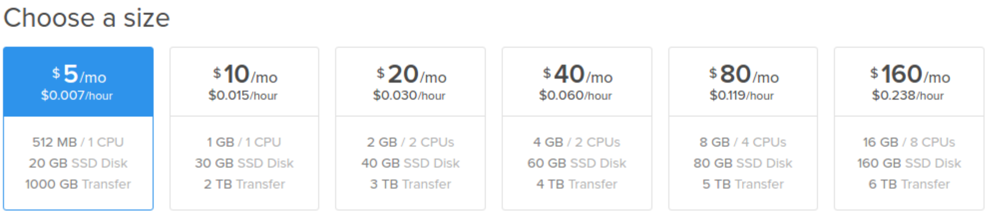
If you are using Digital Ocean, login and select the ‘Create Droplet’ button at the top right hand side of the screen:

From here, select One-click apps, and then 'MEAN on 14.04', this means we are creating a cloud server with Express.js and Node.js already pre-installed, if this is not available you can follow the installation guides to install the needed dependencies at:

Node.js: https://nodejs.org/en/download/package-manager/

Express.js: http://expressjs.com/en/starter/installing.html



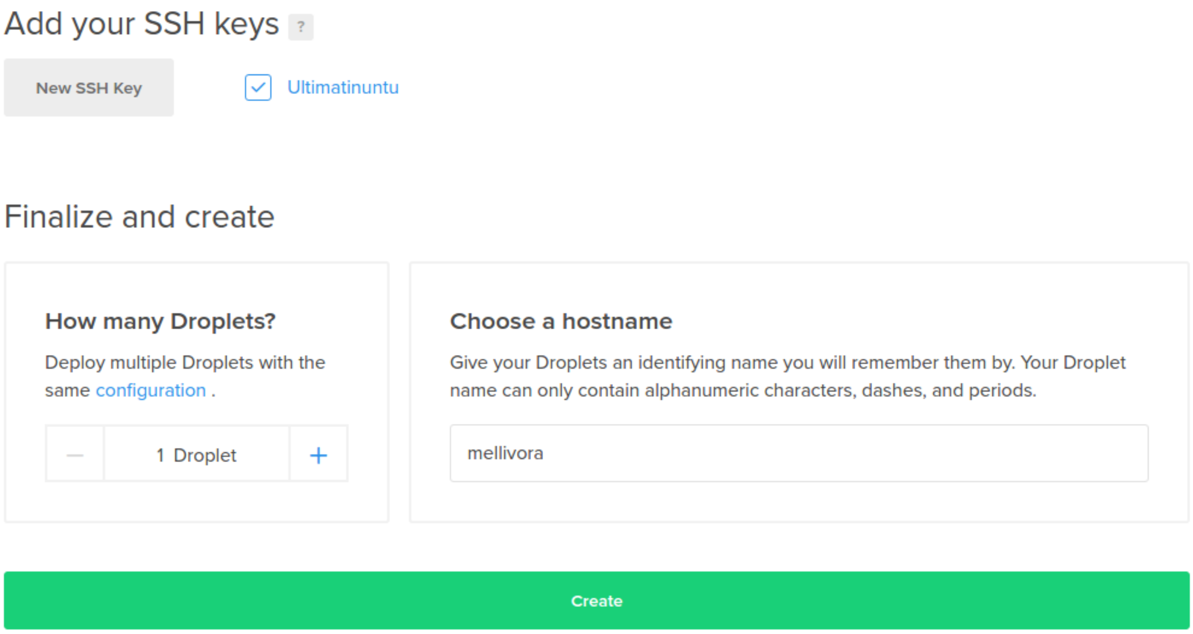


Choose a size

$5/mo	$10/mo	$20/mo	$40/mo	$80/mo	$160/mo
$0.007/hour	$0.015/hour	$0.030/hour	$0.060/hour	$0.119/hour	$0.238/hour
512 MB / 1 CPU	1 GB / 1 CPU	2 GB / 2 CPUs	4 GB / 2 CPUs	8 GB / 4 CPUs	16 GB / 8 CPUs
20 GB SSD Disk	30 GB SSD Disk	40 GB SSD Disk	60 GB SSD Disk	80 GB SSD Disk	160 GB SSD Disk
1000 GB Transfer	2 TB Transfer	3 TB Transfer	4 TB Transfer	5 TB Transfer	6 TB Transfer

Choose the size of the server you want, in this case we have selected the smallest, and then any region you would like. We are basing ours in London.

Next add your SSH key, give it a name, and click 'Create'.



Add your SSH keys ?

New SSH Key

☑ Ultimatinuntu

Finalize and create

How many Droplets?

Deploy multiple Droplets with the same configuration .

— 1 Droplet +

Choose a hostname

Give your Droplets an identifying name you will remember them by. Your Droplet name can only contain alphanumeric characters, dashes, and periods.

mellivora

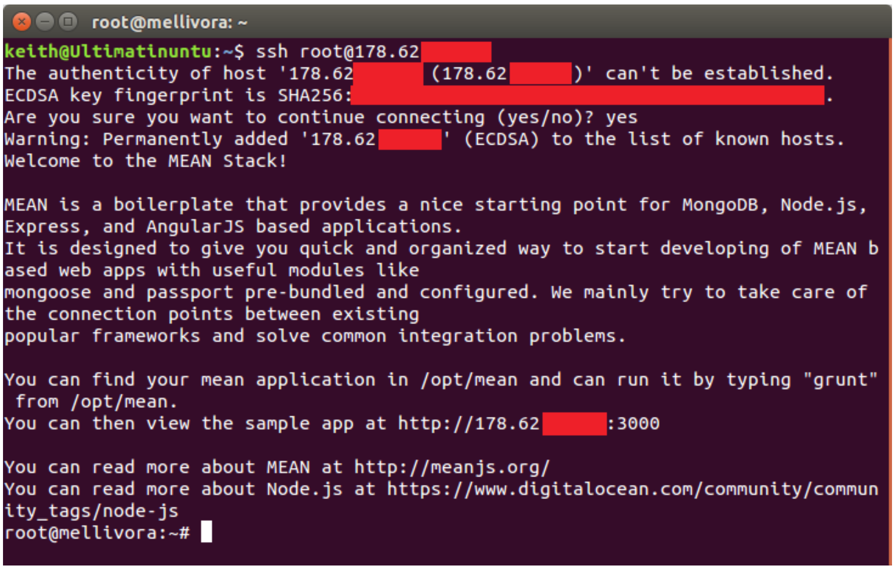
Create

Now that your server has been created, note down its public IP.

Next, open up a new terminal window on the system you have the associated SSH key configured with, and type:

```
$ ssh root@[Your IP]
```

Substitute [your IP] with the IP address of the new cloud server, confirm the ECDSA key fingerprint, and you should have SSH access:



root@mellivora: ~

```
keith@Ultimatinuntu:~$ ssh root@178.62[redacted]
The authenticity of host '178.62[redacted] (178.62[redacted])' can't be established.
ECDSA key fingerprint is SHA256:[redacted].
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '178.62[redacted]' (ECDSA) to the list of known hosts.
Welcome to the MEAN Stack!

MEAN is a boilerplate that provides a nice starting point for MongoDB, Node.js,
Express, and AngularJS based applications.
It is designed to give you quick and organized way to start developing of MEAN b
ased web apps with useful modules like
mongoose and passport pre-bundled and configured. We mainly try to take care of
the connection points between existing
popular frameworks and solve common integration problems.

You can find your mean application in /opt/mean and can run it by typing "grunt"
 from /opt/mean.
You can then view the sample app at http://178.62[redacted]:3000

You can read more about MEAN at http://meanjs.org/
You can read more about Node.js at https://www.digitalocean.com/community/commun
ity_tags/node-js
root@mellivora:~# █
```

Next it is highly recommended that you setup a new non-root account and configure a firewall to block any connections from unwanted locations, a guide to do this can be found at:

Configuring Non-root:
https://www.digitalocean.com/community/tutorials/initial-server-setup-with-ubuntu-14-04

Configuring the firewall:
https://www.digitalocean.com/community/tutorials/how-to-set-up-a-firewall-with-ufw-on-ubuntu-14-04

Once the firewall has been configured, either copy the Mellivora folder into the cloud server's directory or copy the latest version from the private git repository if you have access using the command:

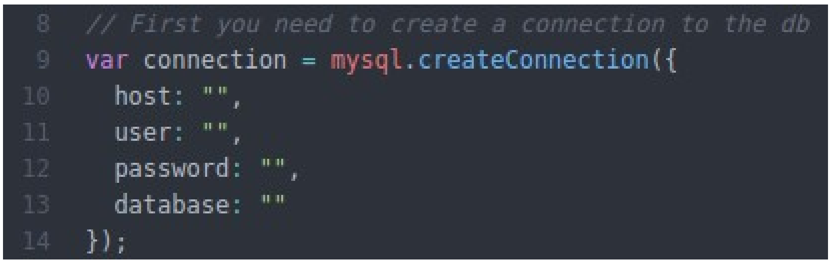
```
$ git clone https://github.com/HoneyDev/mellivora.git
```

Next, enter into the new mellivora directory:

```
$ cd mellivora
```

Note on MySQL Servers

If you are using your own SQL server, open the file ‘server.js’ with a text editor (such as nano), and edit the MySQL options on line 10 accordingly:



```
 8  // First you need to create a connection to the db
 9  var connection = mysql.createConnection({
10    host: "",
11    user: "",
12    password: "",
13    database: ""
14  });
```

Edit the same settings in the ‘mysql.js’ file on line 6.

If you plan on using my (the project creator's) MySQL server you will have to get into contact and send the IP of your server to: keith.short32@gmail.com

Once you are in the ‘mellivora’ directory, run:

```
$ npm install
```

This will install all required Node.js dependencies.

Once this is finished, it’s time to run the app, type:

```
$ node server.js
```

If everything went okay, a couple of tasks should run updating the MySQL database with new IPs and creating JSON files, one line should also say:

```
Mellivora is listening at http://localhost:3000
```

You can now run the app by opening up your browser and navigating to:

http://[Your IP]:3000

Substituting [Your IP] for the IP address of the cloud server.

Alternatively, if you would like to access the site locally (such as if you are running it from a home system, go to:

http://localhost:3000