

# Buzztouch Server 2.0 with Amazon EC2

This is for those that want a step by step instructions on how to prepare an Amazon's EC2 instance for the Buzztouch server. This document only covers the amazon EC2 instance.

The Buzztouch team has already created a really good document on how to upload and configure the Buzztouch server itself. Please review that if you have any questions about buzztouch.

## **Buzztouch**

<http://www.buzztouch.com/BT-server/installation.pdf>

\*If you are new to Amazon please follow amazon's guide to create an account.

\*\*If you notice any errors please let me know: kmassey at crossingsokc.com. I'll be happy to

fix them or post your questions to the buzztouch forum for others to answer

\*\*\*Also, you can review the source sites at the end of this document.

So here we go!

## Amazon EC2 Setup

1. Login to AWS console.
2. Click EC2 on the top tabs
3. Click Launch Instance
4. Click Community AMIs
5. Type in Lamp in the search box (takes a bit to load)

Option I. a. Sort by platform

b. Scroll to Ubuntu

c. Look for the Ubuntu 11.04 AMI that is not bitnami related

Option II. a. In the search bar type: ami-9567a6fc

6. Select number of instances (1) should do.

7. Select Instance type (if this is your first time just select micro)

8. Availability zones really don't matter at the moment (just leave default)

9. Continue

10. Under instance details just leave these at default and press continue (you could enable termination protection but you can also do this later)

11. Continue

12. You can leave the name blank for now...we will name it later.

13. Continue

14. You now will be asked to create a key-pair. If you don't know what this is I highly

recommend you look it up. It will effect everything.

15. Select Create a new Key Pair (This is only necessary if you have never created one

before)

16. Name the key pair something special

17. click create and download key-pair

18. Once it's finished downloading move it to a folder you can remember.

a. Open up your terminal

b. cd to the folder you have the key placed in

c. sudo chmod 400 nameofkeypair

d. you may have to input your computer's password

19. You can leave the terminal open for now but minimize it

20. Once you have created the key pair it should advance you...if not continue

21. Select your security group. If you don't have one or want another one other than default

select create new security group.

22. Give it a name

23. For inbound rules here are the basics (There are two ways to add a new rule:

1. from dropdown and

2. just input the port range and click add rule. (It's up to you about how much security you want)

You must allow at a minimum port 80, 22, 21 for buzztouch.

a. All TCP

b. SSH - Port 22

c. HTTP - Port 80

d. MySQL - Port 3306

e. SMTP - Port 25

f. HTTPS - Port 443

g. FTP - Port 21

24. Continue

25. Review & Launch

26. Your instance is now launching

27. Close

28. Select Instances on left hand side

29. Your instance should say pending and it may already be up and running

30. Under name - it should say empty. Roll over it and click the pencil tool and lets rename it

something clever!

31 Under network & security on the left hand side select Elastic IPs

32. Under Addresses select - Allocate New Address

33. Select EC2 - click Yes, Allocate

34. Once you see your IP address - Select with a check mark and up at the top will say

Associate Address. Click it and you will see your EC2 named instance. Select it.

35. Your instance now is up and running and has an IP address.

36. At this point you need to go to your domain host and set your @ name record to point to this IP address. It may take a while for your domain host to register this new address. (NO

Worries we still have work to do while it works itself out)

## Configuring Your Amazon EC2 Server for the Web

If you were able to skip part one just keep in mind that this document is for those that have never setup a server like this before. Hopefully I have kept it simple and straightforward.

1. So now that the server is installed and we are waiting for the Name record to set we can login into our server via the Mac Terminal and start setting up the server. Let's just use our IP address for the time being.

2. Maximize the terminal that you had open from earlier (for those that skipped part 1 CD to your folder with the server's key pair via the terminal)

3. Once in the folder you will need to type the following:

```
ssh -i yourKeyName ubuntu@yourIPaddress(or your domain name without http://www)
```

ex. i

```
ssh -i mytestkeypair.pem ubuntu@55.55.555.555
```

ex. ii

```
ssh -i mytestkeypair.pem ubuntu@mydomain.com
```

You can also find this by navigating to the instances in your dashboard selecting it and clicking instance actions drop down and then click connect. It will always give you the ssh connection info.

\*Remember that this AMI always uses ubuntu for the login name

4. click enter and you should now be connecting to your server.

## Common Problems:

1. You don't have port 22 open
2. You have port 22 open but the operation is still timing out. Sorry, looks like you have a bad instance. Try stopping it and starting it under the instance actions tab. If that doesn't work terminate it in the instance actions and start over. That's one of the unfortunate things with Amazon. You can get a bad instance!
5. Now that you're in type: `cd /`
6. Let's configure apache. The following are simple commands. (you can copy/paste them into the terminal if you like)
7. `cd etc/apache2`
8. `sudo vi httpd.conf`
9. It should be blank:
  - a. press `i`
  - b. type: `ServerName domainname.com`
  - c. escape
  - d. type `:wq` then enter (this gets you out of the editor with saving)
10. `cd ..`
11. `sudo vi hosts`
12. It should have some text like this: `127.0.0.1 localhost`
  - a. press `i`
  - b. scroll past `127.0.0.1 localhost` and type: `yourdomain.com`
  - c. escape
  - d. type `:wq` then enter
13. `sudo service apache2 restart`

14. You should be able to navigate to your server now.
15. You're still not done. Here is what we have left to do.
  - a. Mysql
  - b. PHPmyadmin
  - c. FTP
  - d. SMTP - Email Server
  - e. Upload and Configure BuzzTouch
  - f. Secure the server with a simple .htaccess file

## MYSQL

1. `sudo apt-get update`
2. `sudo service mysql stop`
3. `sudo /usr/sbin/mysqld --skip-grant-tables --skip-networking &`
4. `mysql -u root`
5. `type: FLUSH PRIVILEGES;`
6. `SET PASSWORD FOR root@'localhost' = PASSWORD('mypassword');`
7. `UPDATE mysql.user SET Password=PASSWORD('mypassword') WHERE User='root';`
8. `FLUSH PRIVILEGES;`
9. `CREATE DATABASE yourdatabasename;`
10. `GRANT ALL PRIVILEGES ON *.* TO 'root'@'localhost' IDENTIFIED BY`

```
'mypassword' WITH GRANT OPTION;
```

11. \q

Install PhpMyAdmin

1. `sudo apt-get install phpmyadmin`

2. type y

3. Prompt pops up with two options select apache2 by pressing space

bar and then enter. Click OK to configure when it ask you to.

4. follow the prompts and enter the appropriate data such as

a. mysql useranme

b. mysql password

c. mysql database name (phpmyadmin)

d. phpmyadmin username

e. phpmyadmin password

5. click ok

6. If you ever have trouble and need to reconfigure it type the

following command.

```
sudo dpkg-reconfigure -plow phpmyadmin
```

7. Log into `yourdomain.com/phpmyadmin` and you will see the database

you create earlier.

8. You may need to restart apache:

```
sudo service apache2 restart
```

## FTP

Now that we are done with mysql and PhpMyadmin it's time to setup our

FTP protocol.

1. `cd /`

2. `sudo apt-get install vsftpd`

3. `sudo vi etc/vsftpd.conf`

4. Use the setting below: press i

```
anonymous_enable=NO
```

```
local_enable=YES
```

```
write_enable=YES
```

5. Press escape then type `:wp` then enter

6. `sudo vi shells`

7. Add the following by pressing i

```
/bin/sh
```

```
/bin/bash
```

```
/bin/false
```

8. Press escape then type `:wp` then enter to exit

9. `cd /`

10. `sudo mkdir home/ftp`

11 `sudo mkdir -p /home/ftp/ftpusername`

12. `sudo user add ftpusername -d /home/ftp/ftpusername/ -s /bin/false`

13. `sudo passwd ftp`

14. Enter your ftp password twice

15. `sudo service vsftpd restart`

16. `cd /`

17. `cd etc/apache2/sites-enabled`

18. `sudo vi 000-default`

19. press `i`

20. Change the root directory to `/home/ftp/ftpusername`

Locate all `AllowOverride None` and change `None` to `All` (this should be

done in three or four places) This is for redirecting your site

with `.htaccess`

a. `sudo service apache2 restart`

21. FTP is done... You should be able to enter your credentials into

programs like dreamweaver or filezilla.

22. Here is how the credentials should look.

host: `yourdomain.com`

username: `ftpusername`

password: `ftppassword you created`

port: `21`

Make sure its on active mode not passive

23. You should be completed at this point. You can now connect into

the `ftpuser` folder that you created earlier.

# You now need to setup Email

Using Gmail as a Sendmail Relay

It should be similar to other mail host. If you don't do this Amazon

will eventually email you and let you know that their default system

is sending to many emails and require you to make the fix.

1. First off make sure you can actually resolve gmail.com.

```
sudo apt-get install sendmail
```

```
sudo cp /etc/mail/sendmail.mc /etc/mail/sendmail.mc.orig
```

```
sudo vi /etc/mail/sendmail.mc
```

2. Add the following to the bottom of your virgin sendmail.mc

(shift+g to get to end of file):

```
define(`SMART_HOST',`smtp.gmail.com')dnl
```

```
define(`confAUTH_MECHANISMS', `EXTERNAL GSSAPI DIGEST-MD5  
CRAM-MD5
```

```
LOGIN PLAIN')dnl
```

```
FEATURE(`authinfo',`hash /etc/mail/auth/client-info')dnl
```

```
define(`CERT_DIR', `MAIL_SETTINGS_DIR`'certs')
```

```
define(`confCACERT_PATH', `CERT_DIR')
```

```
define(`confCACERT', `CERT_DIR/CAcert.pem')
```

```
define(`confSERVER_CERT', `CERT_DIR/mycert.pem')
```

```
define(`confSERVER_KEY', `CERT_DIR/mykey.pem')
```

```
define(`confCLIENT_CERT', `CERT_DIR/mycert.pem')
```

```
define(`confCLIENT_KEY', `CERT_DIR/mykey.pem')
```

3. Save the file.

4. `sudo mkdir /etc/mail/auth`

5. `sudo vi /etc/mail/auth/client-info`

6. Add the following line to the new client-info file replacing the

text between the greater than and less than signs (did I need to

spell that out?):

```
AuthInfo:smtp.gmail.com "U:root" "I:youremail@domain.com"
```

```
"P:yourpassword"
```

```
AuthInfo:smtp.gmail.com:465 "U:root"
```

```
"I:youremail@domain.com"
```

```
"P:yourpassword"
```

7. Save the file.

8. `sudo su`

```
cd /etc/mail/auth/
```

```
makemap hash client-info < client-info
```

```
chmod 700 /etc/mail/auth
```

```
chmod 600 /etc/mail/auth/*
```

```
mkdir /etc/mail/certs
```

```
cd /etc/mail/certs
```

9. `openssl dsaparam 1024 -out dsa1024 -out dsa1024.pem`

```
openssl req -x509 -nodes -days 3650 -newkey dsa:dsa1024.pem  
-out
```

```
/etc/mail/certs/mycert.pem -keyout  
/etc/mail/certs/mykey.pem  
  
ln -s /etc/mail/certs/mycert.pem /etc/mail/certs/CAcert.pem  
  
openssl req -x509 -new -days 3650 -key  
/etc/mail/certs/mykey.pem  
  
-out /etc/mail/certs/mycert.pem  
  
chmod 700 /etc/mail/certs  
  
chmod 600 /etc/mail/certs/*  
  
cd /etc/mail  
  
make  
  
10. /etc/init.d/sendmail reload  
  
11. exit  
  
12. You done with Sendmail  
  
13. sudo service apache2 restart
```

## A Few Extras You will need to install

BuzzTouch requires php curl to be installed.

1. sudo apt-get install curl php5-curl
2. sudo service apache2 restart

3. Once you setup BuzzTouch try to setup an application and configure it. If you get a JSON error or a curl error you may need to enable php-curl in the php.ini file

i. `cd etc/php5/apache2`

ii. `sudo vi php.ini`

iii. scroll down to the extension section

iv. type i then insert this line

```
extension=php_curl.dll
```

4. `sudo service apache2 restart`

## Upload and Configure BuzzTouch

So you finally made it to this point. You are now ready to follow the buzz touch guide to installing the server. It's pretty straightforward from here.

1. You may need to check the folder permission for

`home/ftp/ftpusername` if you are having trouble writing to the ftp

folder

2. Temporarily set it to

```
sudo chmod 777 home/ftp/ftpusername
```

3. When done uploading the server I assume that `chmod 755` will work

for this folder.

## .htaccess for security

You can take security as far as you want it. As for this document,

just make sure your directory isn't being shown by placing an

index.html in the root directory. You can then redirect all traffic

to your /BT-server

1. `cd /home/ftp/ftpuser`
2. `sudo vi .htaccess`
3. press `i`
4. Redirect `/index.html` <http://yoursite.com/BT-server/index.php>
5. escape
6. `:wq`

## Bonus

Make sure you make regular backups of your amazon ec2 instance. You can do this by selecting the instance and in the instance actions drop down at the top select create Image (EBS AMI). It will temporarily stop your instance but will create a backup image just in case your instance ever goes down. You can then launch another instance with this image and all your settings are still active. Just set the elastic IP to point to that instance and your up and rolling within a few minutes.

## Sources

PHPMyAdmin

<https://help.ubuntu.com/community/phpMyAdmin>

Vsftpd

<http://cviorel.easyblog.ro/2009/03/05/how-to-setup-vsftpd-ftp-onubuntu->

linux/

MYSQL

<https://help.ubuntu.com/community/ApacheMySQLPHP>

Amazon EC2

Sorry but this is my own trial and error. Try googling it!

Buzztouch

<http://www.buzztouch.com/BT-server/installation.pdf>