

IOTA 0.0.1 noob guide

1.) Get a server

We are going to use a Ubuntu 14.04 server for this, the following section explains how to create a Ubuntu 14.04 instance on Amazon AWS (the offer free tier for new users), if you dont want to use Amazon AWS and know how to get a server somewhere else you can just skip this section.

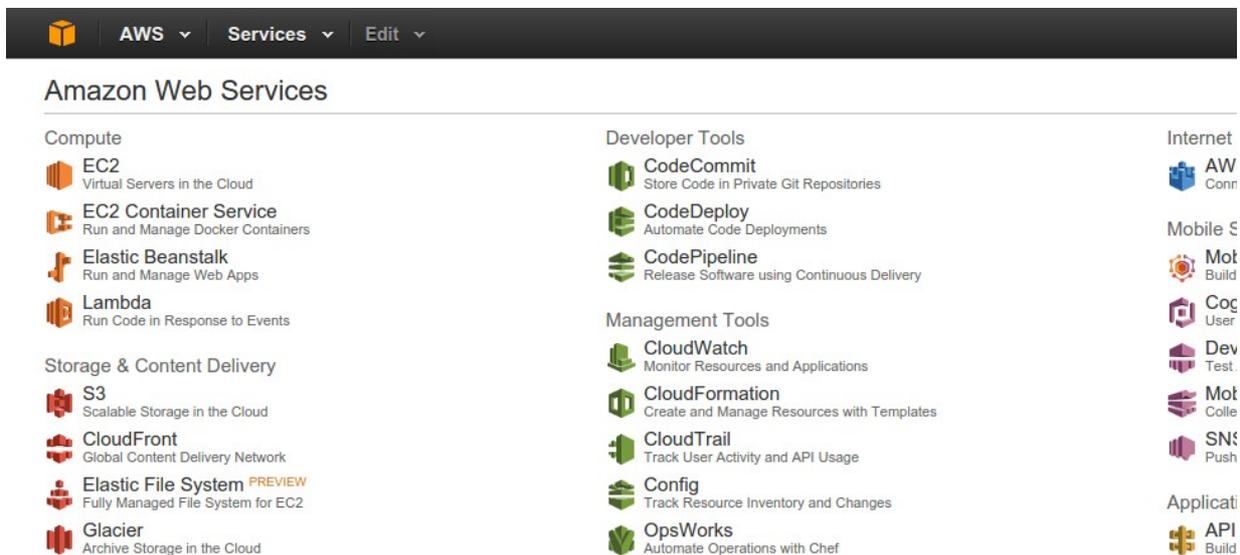
Create an Account

Go to <https://aws.amazon.com> and register a new account if you dont already have one.

Create a new Instance

Btw: Instance is just a fancy word for virtual server.

Log into your „AWS managment console“ you should see something like this:



click on „EC2“



This brings you to your EC2 Dashboard, in the top right corner of this Page you can select the Location of where you want to create new Instances choose whatever Location you like best.

Next click on Launch instance

Create Instance

To start using Amazon EC2 you will want to launch a virtual server, known as an Amazon EC2 instance.

Launch Instance

Select Ubuntu Server 14.04



Ubuntu Server 14.04 LTS (HVM), SSD Volume Type -
Ubuntu
Free tier eligible
Ubuntu Server 14.04 LTS (HVM), EBS General Purpose (SSD) V
Root device type: ebs Virtualization type: hvm

On the next screen you can select the Type of the Instance we are going to use the t2.micro for this.

| | | | | | |
|-------------------------------------|-----------------|--------------------------------|---|---|--|
| <input checked="" type="checkbox"/> | General purpose | t2.micro Free tier eligible | 1 | 1 | |
|-------------------------------------|-----------------|--------------------------------|---|---|--|

Select it and click on “Next: Configure Instance Details”

On Page „step 3“ the only import thing is to set Auto-assign Public IP to enable

Auto-assign Public IP ⓘ **Enable** ▼

Dont change anything else, click on „Next: Add Storage“

Step 4: Add storage

Default settings will be fine, click on „Next: Tag Instance“

Step 5: Tag Instance

No need to add Tags, click on „Next: Configure Security Group“

Step 6: Configure Security Group

Now this is important, Amazon blocks all traffic that is not specifically allowed in this Setting.

IOTA nodes communicate via port 999 udp so we have to add a rule to allow this, also SSH will be used to access this instance so leave the default, should look **exactly** like this:

| Type <small>i</small> | Protocol <small>i</small> | Port Range <small>i</small> | Source <small>i</small> | |
|----------------------------------|---------------------------|-----------------------------|-------------------------------------|------------------|
| SSH <small>v</small> | TCP | 22 | Anywhere <small>v</small> 0.0.0.0/0 | <small>x</small> |
| Custom UDP Rule <small>v</small> | UDP | 999 | Anywhere <small>v</small> 0.0.0.0/0 | <small>x</small> |

Click on review and launch and then launch.

Ec2 will ask you to create a new key pair to access your server, select „create a new key pair“ give it a nice name and download the file (will be a .pem file)

Note: The selected key pair will be added to the set of keys authorized for this instance. Learn more about [removing existing key pairs from a public AMI](#).

Create a new key pair v

Key pair name

iota_node

Download Key Pair

then click on „launch instances“, wait for it to launch and click „View instances“

you will be sent back to the ec2 dashboard where you can view your instance, here you can also stop your instance if you want to, If you select the instance you can see its public ip, should look something like this:

EC2 Dashboard

Events

Tags

Reports

Limits

INSTANCES

Instances

Spot Requests

Reserved Instances

Scheduled Instances

Commands

Dedicated Hosts

IMAGES

AMIs

Bundle Tasks

ELASTIC BLOCK STORE

Launch Instance Connect Actions v

Filter by tags and attributes or search by keyword ? 1 to 1 of 1

| Name | Instance ID | Instance Type | Availability Zone | Instance State | Status |
|------|-------------|---------------|-------------------|----------------|--------|
| | i-55021b8c | t2.micro | us-west-2a | running | |

Instance: i-55021b8c Public IP: 52.33.48.178

Description Status Checks Monitoring Tags

Instance ID i-55021b8c Public DNS -

(Yeah thats my IP up there, don't try connecting to it i threw that instance away already)

congrats, you now have a ubuntu 14.04 instance with a public IP

2.) Learn how to access your server

If you use a Linux based system this is rather simple, open a Terminal window and type

```
user@host:~$ ssh -i </path/to/.pem> ubuntu@<IP>
```

Obviously replace insert the corret path to your .pem file and your IP

If you use windows you have to download the Bitvise SSH Client from

<https://www.bitvise.com/ssh-client-download>

Start the SSH client and import your .pem file in the „Client Key Manager“, set your Host IP , Port 22 and Username „ubuntu“ and Connect.

A nice black command-line interface will open and File Explorer where you can upload and download files to/from your server. We will need both things.

Sorry no screenshot for this, actually i dont have a windows pc, I hope this is correct :D

If you use Mac then....google...sorry, maybe the same as linux but i am not sure.

3.) Install Oracle Java 8

In the command line from step 2 enter the following commands:

```
sudo apt-add-repository ppa:webupd8team/java  
sudo apt-get update  
sudo apt-get install oracle-java8-installer
```

4.) Upload IOTA

First lets make a new folder on the server by typing

```
mkdir iota
```

If nothing happens if you press enter thats fine.

Now use the Bitwise SFTP to upload the contents of the IOTA-0.0.1 zip to your server, upload the files into /home/ubuntu/iota

Back to command-line interface

If you cd (change directory) into your iota folder, and list all the files you should see this:

```
ubuntu@ip-172-30-0-52:~$ cd iota
```

```
ubuntu@ip-172-30-0-52:~/iota$ ls
```

```
configuration.iota Iota.jar nodes.iota transactions.iota
```

```
ubuntu@ip-172-30-0-52:~/iota$
```

you can now start your IOTA node by typing

```
sudo java -jar Iota.jar
```

inside the iota directory

This will start iota in foreground, this is good for testing but it will not continue running if you close the terminal, to start it in background type

```
sudo java -jar Iota.jar &
```

and you will get something like

```
ubuntu@ip-172-30-0-52:~/iota$ sudo java -jar Iota.jar &
```

```
[1] 2930
```

(actual number is probably different, thats the process id)

you can stop iota by executing

```
kill 2930 ( use YOUR process id )
```

TIPP: if you never used Linux and dont know how to use the Console this might help:

http://linuxcommand.org/learning_the_shell.php

Congrats you now own an IOTA node



Now go and internet the shit out of everything, #IOTA