Part 1. Creating a AWS EC2 Instance.

Lunch Instance

In this step, we will create a AWS EC2 micro instance running Ubuntu. For launching new instances, AWS allows you to choose from a number of server locations. Since I assume you are mostly interested in using Shadowsocks in China, I'd suggest choosing the Asia Pacific (Tokyo) region because it's probably the best international EC2 server location for China.

First, let's go to the AWS Management Console and log in with your AWS account. After that, you need to select the Tokyo region from the dropdown menu in the top navigation bar. Then you click on the "EC2" link (on the upper left position on that page).

Once you are in the EC2 Management Console, click the blue button "Launch Instance".

Then you will see a list of Amazon Machine Images (AMIs) for you to choose. Find the one titled "Ubuntu Server 14.04 LTS (HVM) and click on "Select". (If you are reading this article when there's already a newer version of Ubuntu server available, just choose the newer one).

Then you will be asked to select the instance type. Select the first one, t2.micro, because it's the cheapest one and it's good enough or a Shadowsocks server. Also, it might be free tier eligible, which means you can use it free of charge for up to 1 year!

Then you click on the blue button down below labeled "Review and Launch". On the next screen, you will see an warning saying your security group is open to the world.

Security Group

Find the link "Edit security groups" (see picture below) and click on it. Let's create a new security group called "shadowsocks". You can click on the "Add Rule" button to add new rules. For **inbound**, add ports **22** and **8000 - 8388** which are open to 0.0.0.0/0. For outbound, add ports **0-65535** and open to 0.0.0.0/0. Then click on the blue button down below labelled "Review and Launch".

KeyPair

You can click on the blue button labeled "Launch". You will see a popup window like the one below that asks you about key pair. If you have already created a key pair before and that key pair is still available on your computer, you can select "using existing key pair". Otherwise, you can create a new key pair. I will skip the key

pair creation step here. Make sure you save your new key pair in a secure place.

After you're done with key pair, click on the blue button "Launch Instance". If everything is fine, you will see a message saying 'Your Instance are now launching". Click on the "View Instances" button down below to see your newly created instance.

You will see your new instance listed on that page. If you click on the checkbox to the left of that instance, you will see detailed information below the instance list, such as Instance ID, Private DNS, Private IPs etc. Among those, we will need two pieces of information: Public IP and Private IP. Copy and paste these two IPs for later use.

Optional: Elastic Public IP

Although the Public IP will work, I'd suggest you get a new Elastic IP and use that IP instead of the default public IP for the new server. That's because in the future if you stop the instance and then start it again, you might find the public IP address has changed, and then you have to change your Shadowsocks client's configuration accordingly. Therefore, a static IP will will be better. because it's stable. To setup a elastic IP, click on the "Elastic IPs" link under "Network & Security" in the left navigation bar in the AWS EC2 console. Then click on "Allocate New Address" and then you can associate the new IP with your new instance. From then on, use this new IP as the Public IP for the server and don't use the public IP that come with your new instance.

Part 2. Connecting to the New EC2 Instance to Setup Shadowsocks

With the checkbox for our new instance checked, click on the button on the top of the page labeled "Connect". You will see a popup showing you detailed information on how to connect to your instance using SSH.

Now, let's SSH to your newly created Ubuntu instance using the SSH command provide in that window. Let me skip instructions on how to SSH to a server from your computer etc. because that would be another long tutorial. Basically, you can use the Terminal application on a Mac or Putty on a Windows machine. The SSH commend will look something like this:

ssh -i "my*key*pair.pem" ubuntu@{your*public*IP} Once you have SSHed to the server, run the following commands:

First:

```
$ sudo su
```

Then let's update our machine:

`$ apt-get update`

Then let's install shadowsocks using these two commands:

```
$ apt-get install python-pip
$ pip install shadowsocks
```

Config File for Shadowsocks

Now let's create a config file for Shadowsocks. Let's use the vi editor to create a new file:1*

`$ vi /etc/shadowsocks.json`

This file should have the following contents:

```
{
    "server":"your_private_ip",
     "server_port":8000,
     "local_port":1080,
     "password":"your_passwd",
     "timeout":600,
     "method":"aes-256-cfb"
}
```

You need to replace "your private ip" which the private IP address of your EC2 instance (NOT the public IP!) and replace "your_passwd" with a real and better password.

Now let's try starting the shadowsocks server with the following command:

`$ ssserver -c /etc/shadowsocks.json -d start`

If this command brings error like this

```
AttributeError: /usr/lib/x86_64-linux-gnu/
libcrypto.so.1.1: undefined symbol:
EVP_CIPHER_CTX_cleanup
```

Follow 2* to fix the problem.

If you want to double check if everything is fine, you can check on Shadowsocks' logfile like this:

`$ less /var/log/shadowsocks.log`

If there's no error message in the log file, you are OK. In the future, if you want to stop the server, just run

`ssserver -c /etc/shadowsocks.json -d stop`

If you want to restart the server, run "ssserver -c /etc/shadowsocks.json -d restart".

Then you are done! Feel free to surf the internet.

1*: If any permission error occurs, use `sudo`

2*: The EVP*CIPHER*CTX_cleanup is outdated. Find openssl.py, probably in

```
/home/ubuntu/.local/lib/python2.7/site-packages/shadowsocks/crypto/openssl.py
```

use

`vi /home/ubuntu/.local/lib/python2.7/site-packages/shadowsocks/crypto/openssl.py`

to open the file. Then go to line 52, change

`libcrypto.EVP_CIPHER_CTX_cleanup.argtypes = (c_void_p,)`

to

`libcrypto.EVP_CIPHER_CTX_reset.argtypes = (c_void_p,)`

go to line 111, change

`libcrypto.EVP_CIPHER_CTX_cleanup(self._ctx)`

to

`libcrypto.EVP_CIPHER_CTX_reset(self._ctx)`

save and quit.