cafeandrew.com

Setting up an FTP server on your Amazon Linux AMI

Andy Halvorsen

Step #1: Install vsftpd

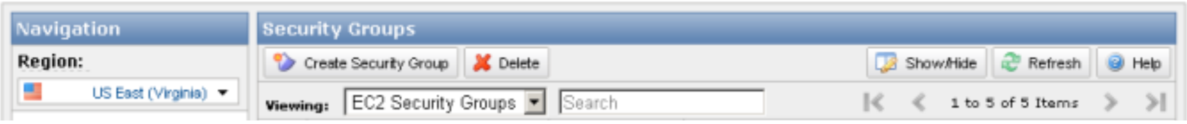
SSH to your EC2 server. Type:

```
> sudo yum install vsftpd
```

This should install vsftpd.

Step #2: Open up the FTP ports on your EC2 instance

Next, you'll need to open up the FTP ports on your EC2 server. Log in to the AWS EC2 Management Console and select Security Groups from the navigation tree on the left. Select the security group assigned to your EC2 instance. Select the Inbound tab and add port range 20-21:

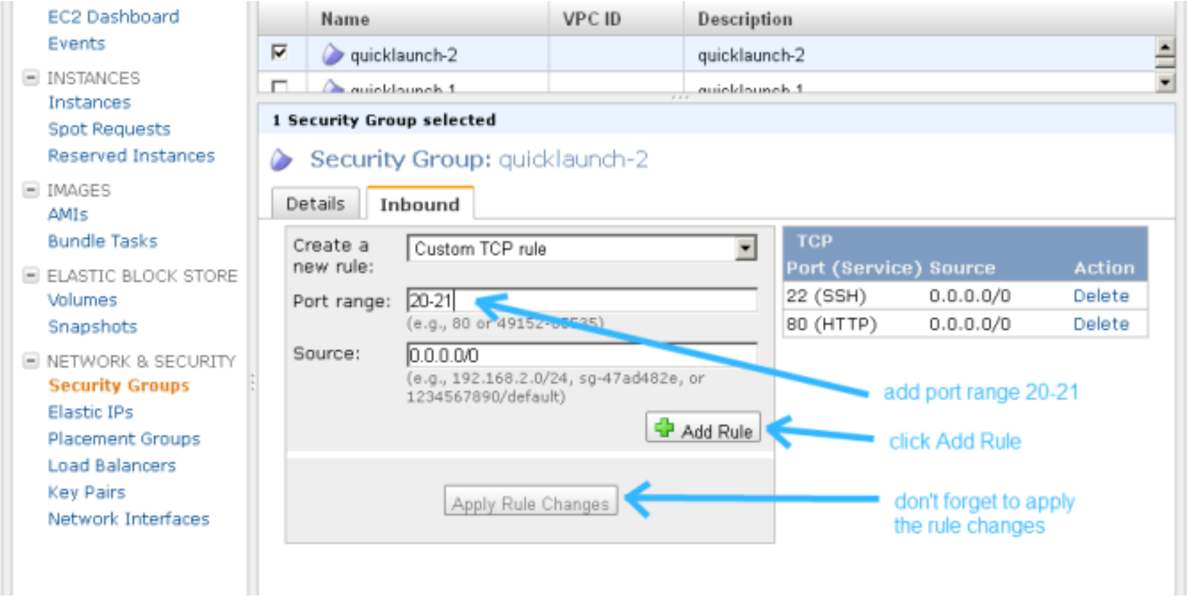


Region: US East (Virginia)

Viewing: EC2 Security Groups

Search

1 to 5 of 5 Items



	Name	VPC ID	Description
☑	quicklaunch-2		quicklaunch-2
□	quicklaunch-1		quicklaunch-1

1 Security Group selected

Security Group: quicklaunch-2

Create a new rule: Custom TCP rule

Port range: 20-21
(e.g., 80 or 49152-65535)

Source: 0.0.0.0/0
(e.g., 192.168.2.0/24, sg-47ad482e, or 1234567890/default)

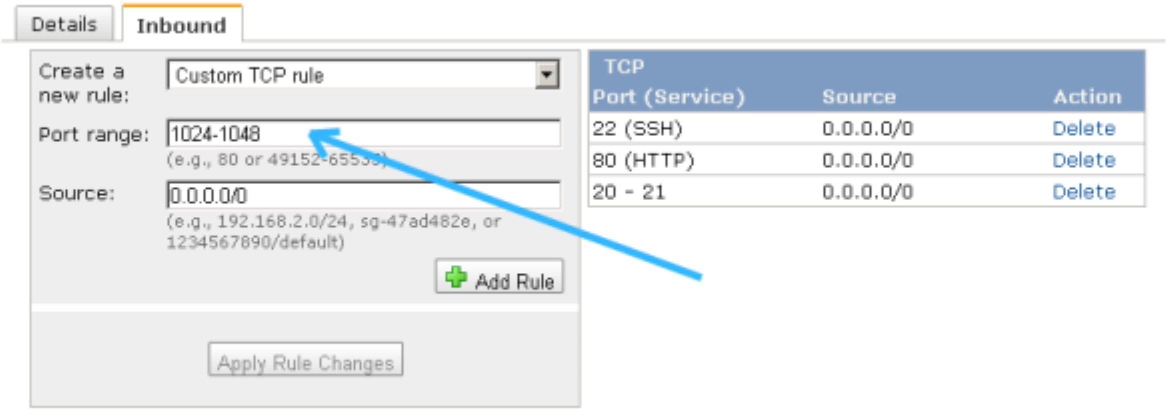
TCP Port (Service)	Source	Action
22 (SSH)	0.0.0.0/0	Delete
80 (HTTP)	0.0.0.0/0	Delete

add port range 20-21

click Add Rule

don't forget to apply the rule changes

Also add port range 1024-1048:



Create a new rule: Custom TCP rule

Port range: 1024-1048
(e.g., 80 or 49152-65535)

Source: 0.0.0.0/0
(e.g., 192.168.2.0/24, sg-47ad482e, or 1234567890/default)

TCP Port (Service)	Source	Action
22 (SSH)	0.0.0.0/0	Delete
80 (HTTP)	0.0.0.0/0	Delete
20 - 21	0.0.0.0/0	Delete

Step #3: Make updates to the vsftpd.conf file

Edit your vsftpd conf file by typing:

```
> sudo vi /etc/vsftpd/vsftpd.conf
```

Disable anonymous FTP by changing this line:

```
anonymous_enable=YES
```

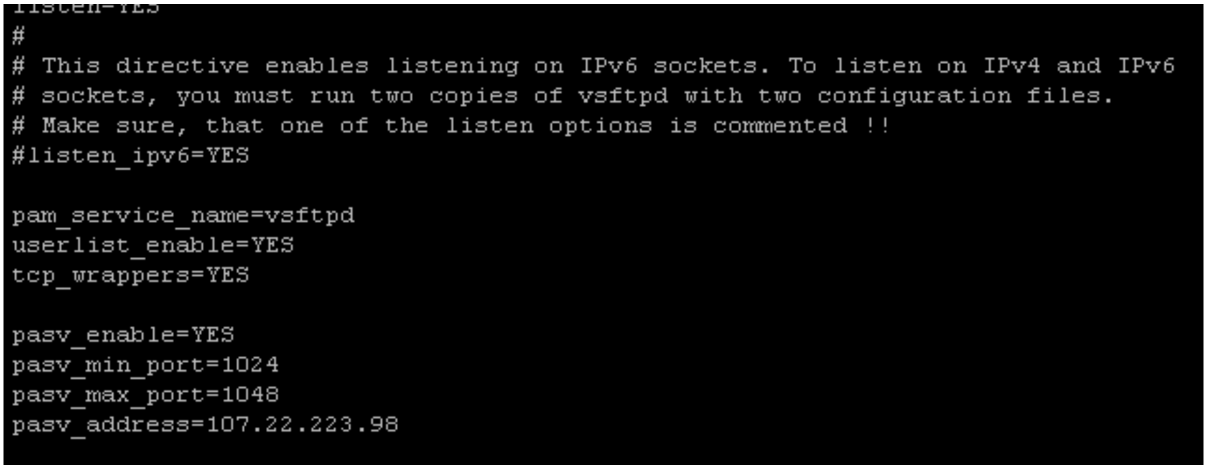
to

```
anonymous_enable=NO
```

Then add the following lines to the bottom of the vsftpd.conf file:

```
pasv_enable=YES
pasv_min_port=1024
pasv_max_port=1048
pasv_address=<Public IP of your instance>
```

Your vsftpd.conf file should look something like the following – except make sure to replace the pasv_address with your public facing IP address:



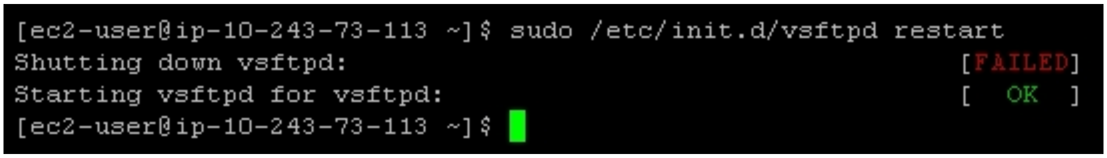
```
listen=YES
#
# This directive enables listening on IPv6 sockets. To listen on IPv4 and IPv6
# sockets, you must run two copies of vsftpd with two configuration files.
# Make sure, that one of the listen options is commented !!
#listen_ipv6=YES

pam_service_name=vsftpd
userlist_enable=YES
tcp_wrappers=YES

pasv_enable=YES
pasv_min_port=1024
pasv_max_port=1048
pasv_address=107.22.223.98
```

Step #4: Restart vsftpd

Restart vsftpd by typing:

```
> sudo /etc/init.d/vsftpd restart
```

You should see a message that looks like:



```
[ec2-user@ip-10-243-73-113 ~]$ sudo /etc/init.d/vsftpd restart
Shutting down vsftpd:                                      [FAILED]
Starting vsftpd for vsftpd:                                [  OK  ]
[ec2-user@ip-10-243-73-113 ~]$
```

Step #5: Create an FTP user

If you take a peek at /etc/vsftpd/user_list, you'll see the following:

```
# vsftpd userlist
# If userlist_deny=NO, only allow users in this file
# If userlist_deny=YES (default), never allow users in this file, and
# do not even prompt for a password.
# Note that the default vsftpd pam config also checks /etc/vsftpd/ftpusers
# for users that are denied.
root
bin
daemon
adm
lp
sync
shutdown
```

```
halt
mail
news
uucp
operator
games
nobody
```

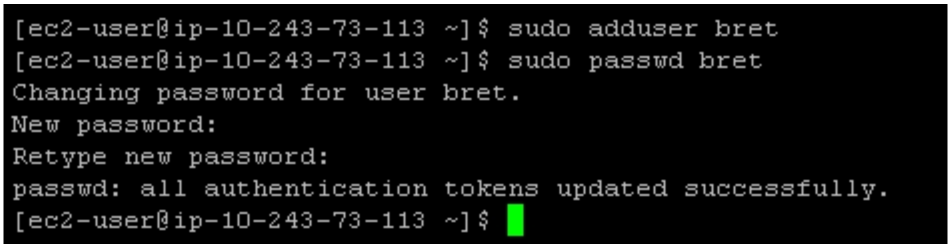
This is basically saying, “Don’t allow these users FTP access.” vsftpd will allow FTP access to any user not on this list.

So, in order to create a new FTP account, you may need to create a new user on your server. (Or, if you already have a user account that’s not listed in /etc/vsftpd/user_list, you can skip to the next step.)

Creating a new user on an EC2 instance is pretty simple. For example, to create the user ‘bret’, type:

```
> sudo adduser bret
> sudo passwd bret
```

Here’s what it will look like:



```
[ec2-user@ip-10-243-73-113 ~]$ sudo adduser bret
[ec2-user@ip-10-243-73-113 ~]$ sudo passwd bret
Changing password for user bret.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
[ec2-user@ip-10-243-73-113 ~]$
```

Step #6: Restricting users to their home directories

At this point, your FTP users are not restricted to their home directories. That’s not very secure, but we can fix it pretty easily.

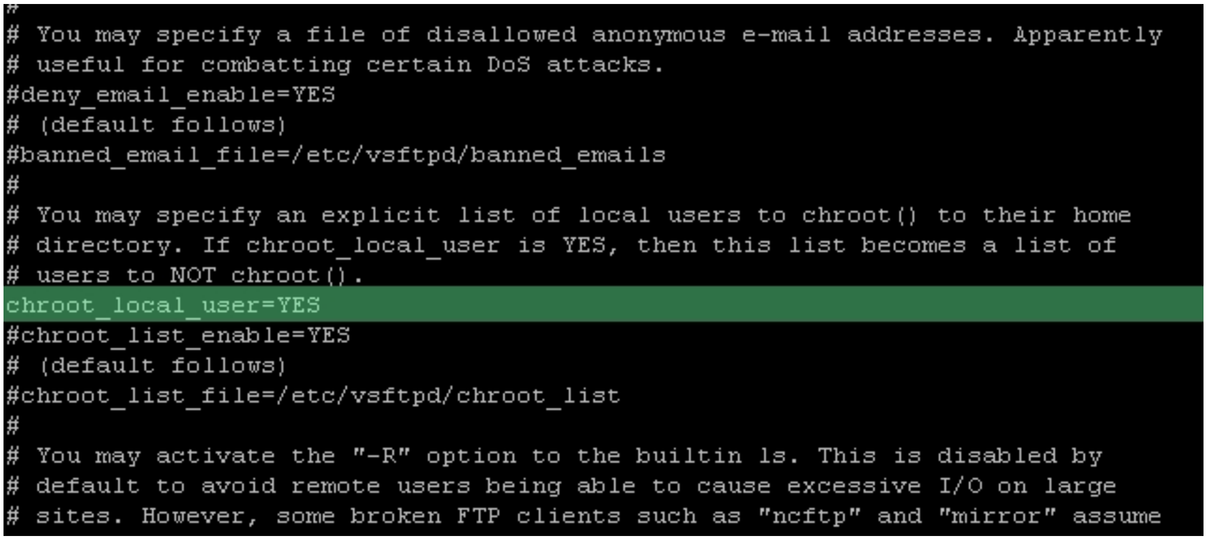
Edit your vsftpd conf file again by typing:

```
> sudo vi /etc/vsftpd/vsftpd.conf
```

Un-comment out the line:

```
chroot_local_user=YES
```

It should look like this once you’re done:



```
#
# You may specify a file of disallowed anonymous e-mail addresses. Apparently
# useful for combatting certain DoS attacks.
#deny_email_enable=YES
# (default follows)
#banned_email_file=/etc/vsftpd/banned_emails
#
# You may specify an explicit list of local users to chroot() to their home
# directory. If chroot_local_user is YES, then this list becomes a list of
# users to NOT chroot().
chroot_local_user=YES
#chroot_list_enable=YES
# (default follows)
#chroot_list_file=/etc/vsftpd/chroot_list
#
# You may activate the "-R" option to the builtin ls. This is disabled by
# default to avoid remote users being able to cause excessive I/O on large
# sites. However, some broken FTP clients such as "ncftp" and "mirror" assume
```

Restart the vsftpd server again like so:

```
> sudo /etc/init.d/vsftpd restart
```

All done!

Appendix A: Surviving a reboot

vsftpd doesn’t automatically start when your server boots. If you’re like me, that means that after rebooting your EC2 instance, you’ll feel a moment of terror when FTP seems to be broken – but in reality, it’s just not running!. Here’s a handy way to fix that:

```
> sudo chkconfig --level 345 vsftpd on
```