AWS Ubuntu 20 EC2 Postfix Install

In this tutorial, we will be installing Postfix as an outbound send-only email server on an Ubuntu 20 EC2.

Prerequisites

- an AWS Free Tier account
- AWS Ubuntu 20 EC2 instance
- Amazon SES (Simple Email Service) configured
- internet access

If you do not have an AWS account, you can access my **AWS Create Free Tier Account** tutorial **here**.

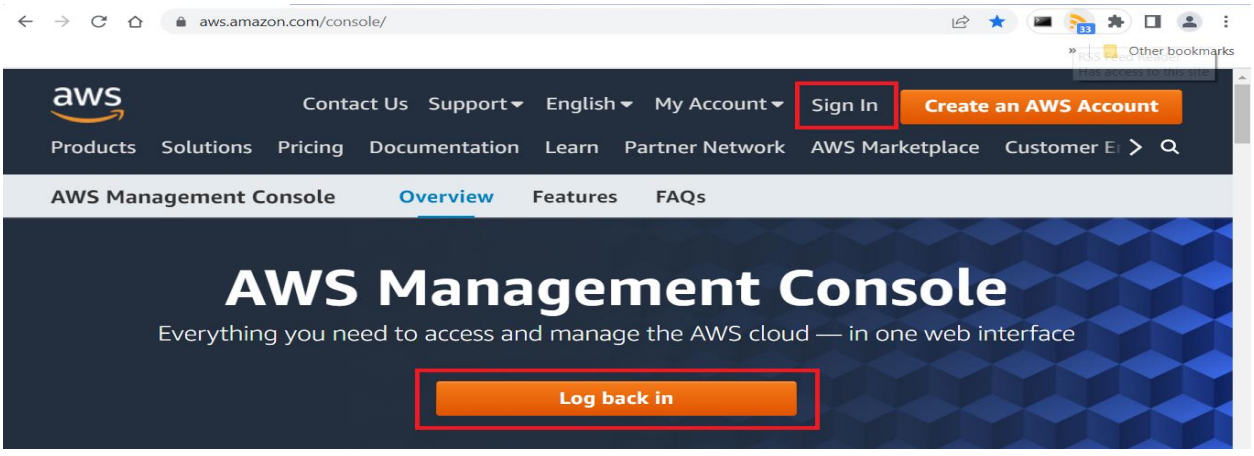
If you do not have an AWS Ubuntu 20 EC2 instance, my tutorial **Create AWS Ubuntu 20 EC2 Instance** is **here**.

Finally, if you have not yet configured the required components to use Amazon SES, complete my other tutorial **Amazon SES Configuration**, accessible **here**, then return to this tutorial.

Steps to complete tutorial:

Start Ubuntu 20 EC2 Instance

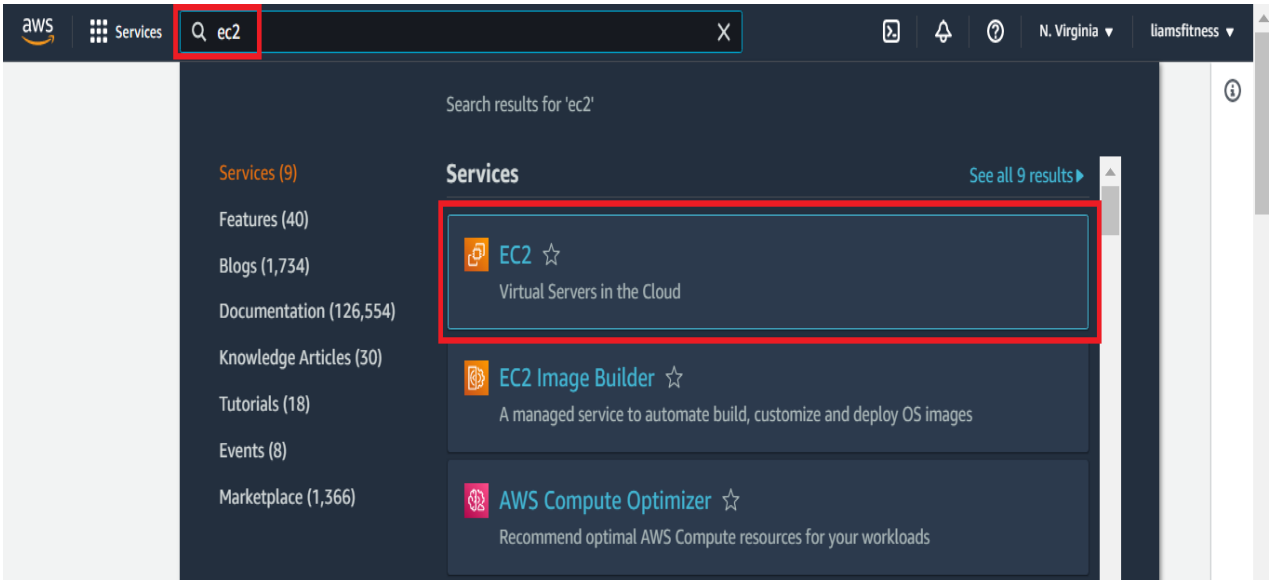
To begin, go to the following website, https://aws.amazon.com/console/ and log in to the console.



At the end of my **Create AWS Ubuntu 20 EC2 Instance** tutorial, I suggested shutting down your EC2 instances when they are not being used due to the AWS Free Tier EC2 limit of 750 hours per month.

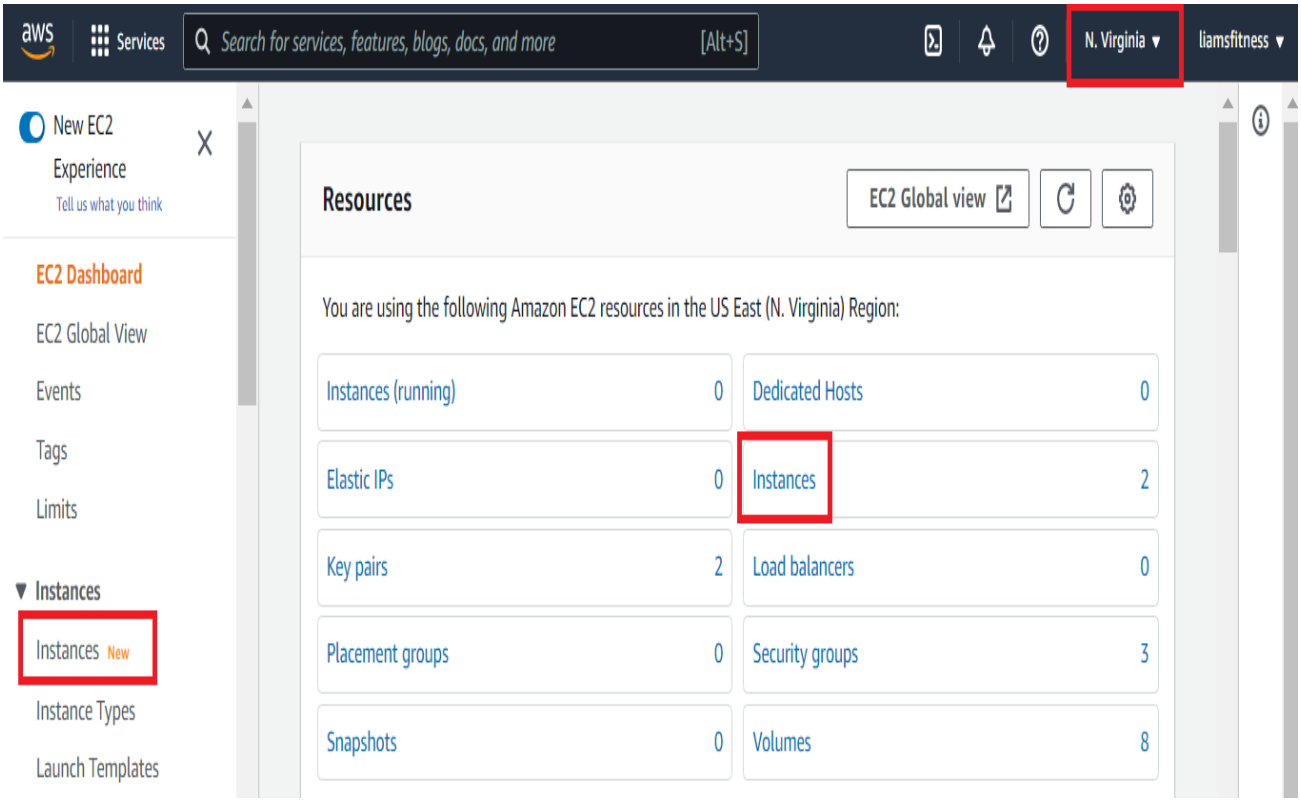
If you already know that your Ubuntu 20 EC2 instance is running, you can skip this step and go directly to **Connect to Ubuntu 20**.

Otherwise, we need to ensure that our instance is running. Once on the **Console Home** screen, enter **EC2** in the search bar and select the 1st EC2 listing.

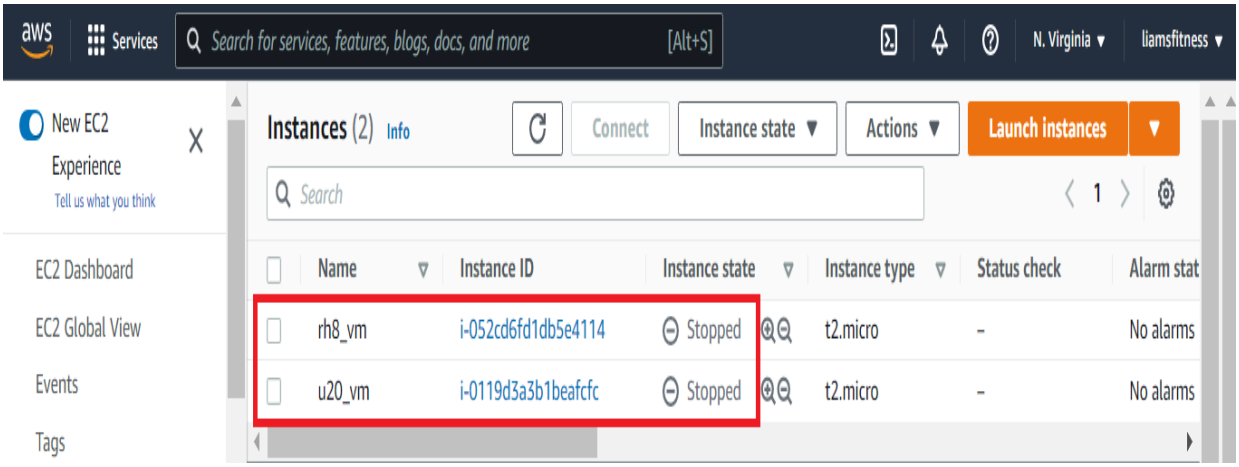


You will be brought to the **EC2** Dashboard. It contains links to the resources being used in the selected AWS region. In my case it's US East (N. Virginia).

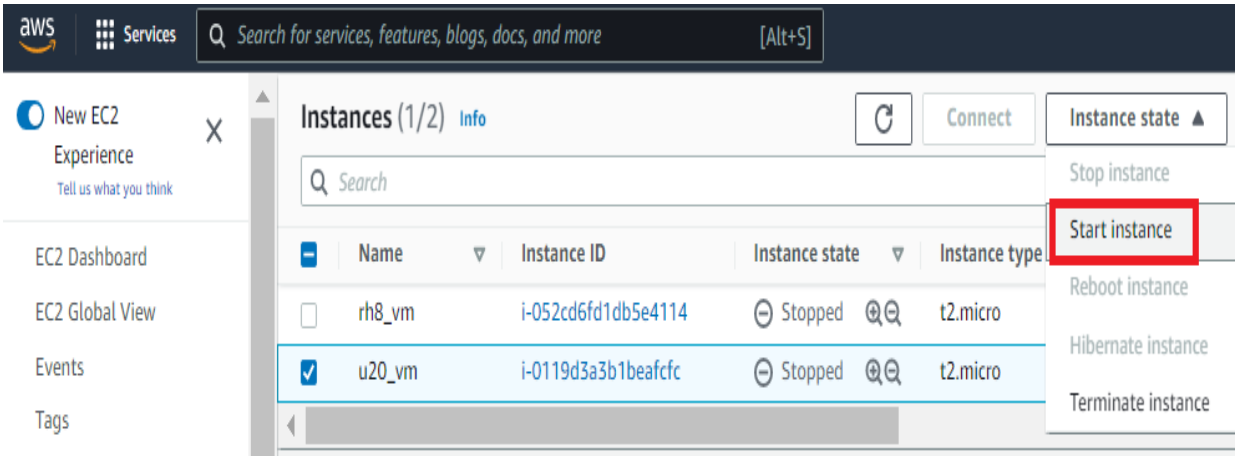
From the EC2 dashboard, click **Instances** (either link will work, your choice).



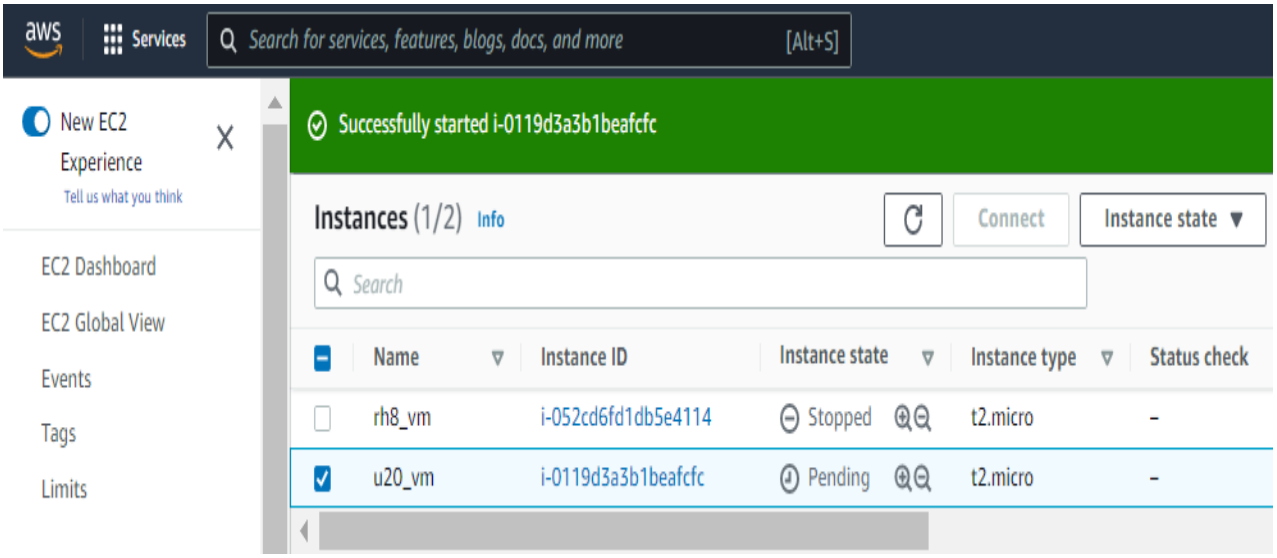
You will notice that both of my instances are in the stopped stated.



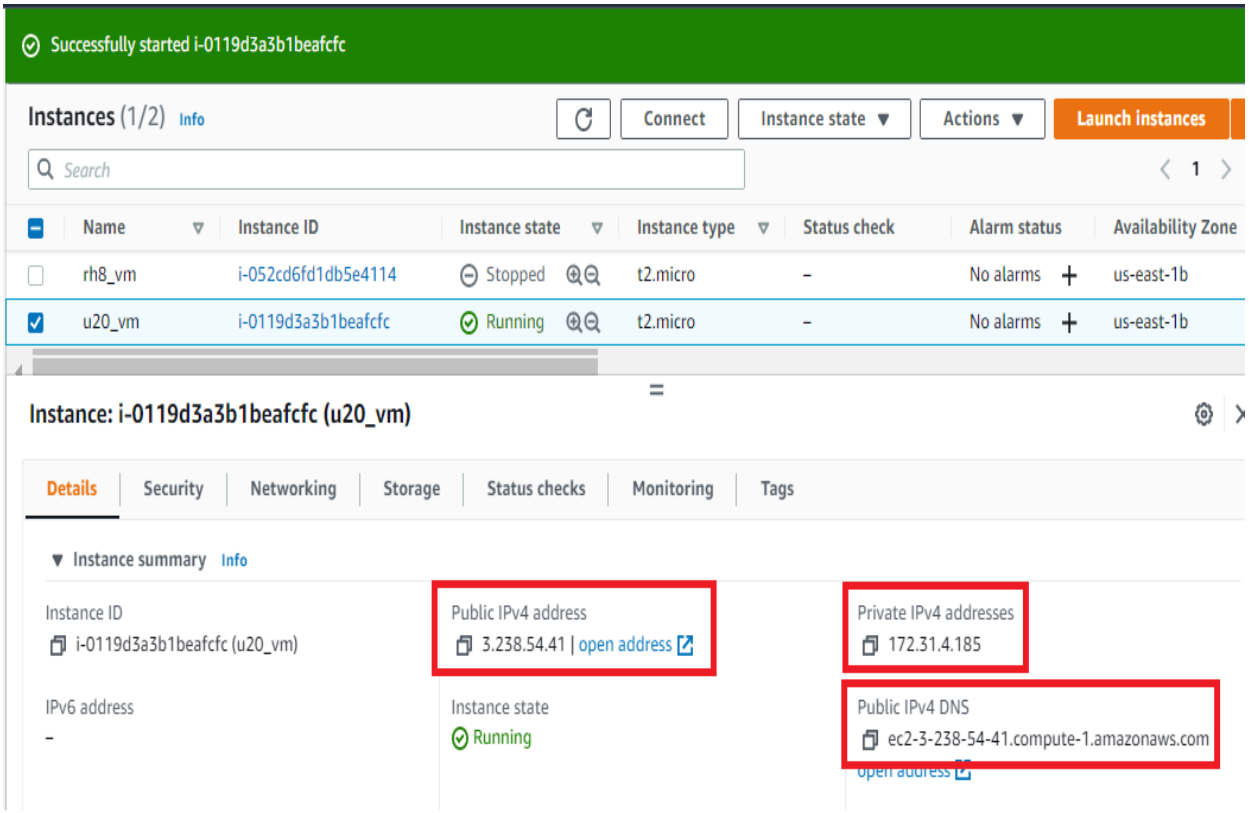
To start my Ubuntu 20 instance, I will ensure that my **u20_vm** is selected (checkbox), then I will click the **Instance state** drop-down menu and click **Start instance**.



The instance will start and be given new connection details.



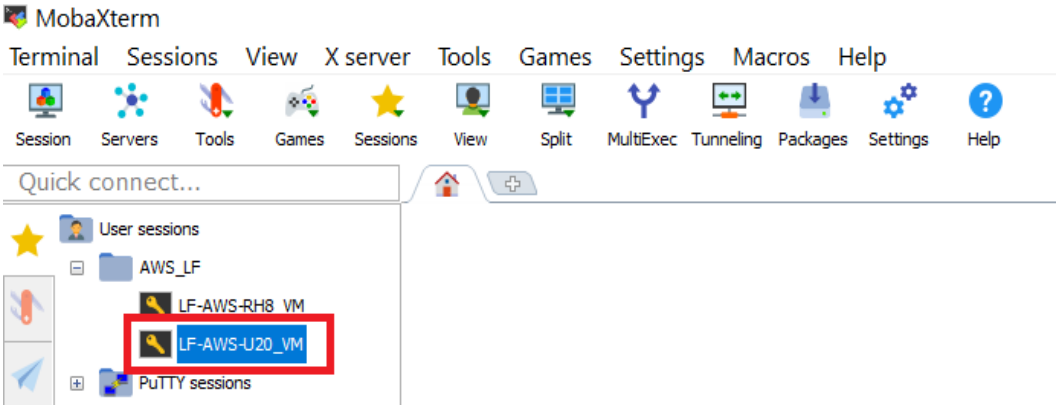
Ensure your new instance is selected (**u20_vm**), then on the **Details** tab, note the value for **Public IPv4 DNS**. I usually keep the instance's name, public IP, private IP and public IPv4 DNS stored for easy access. We will need this to connect to the instance.



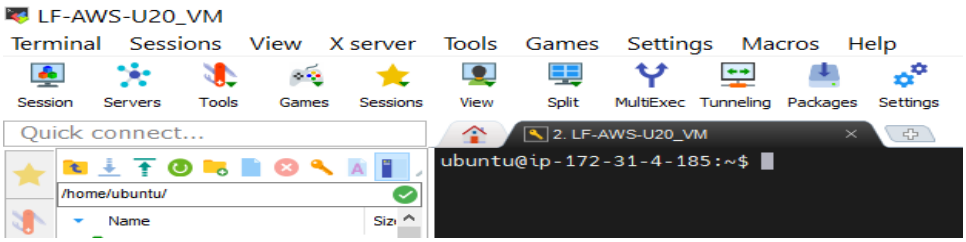
Connect to Ubuntu 20

As I mentioned in my **Create AWS Ubuntu 20 EC2 Instance** tutorial, accessible **here,** I am using **MobaXterm Portable** as my SSH client. Since I restarted my Ubuntu 20 EC2 instance, the connection details changed and I had to update my saved SSH session by changing the **Remote Host:** to the updated **Public IPv4 DNS** of my Ubuntu 20 instance.

After updating my saved SSH connection, from the MobaXterm main interface, I double clicked my SSH session to connect to my Ubuntu 20 EC2 instance.



After the session opened, I executed the **`clear`** command to clear the terminal.

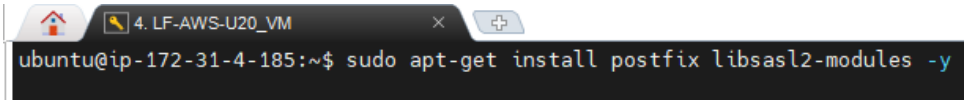


We are now ready to install the Postfix email server.

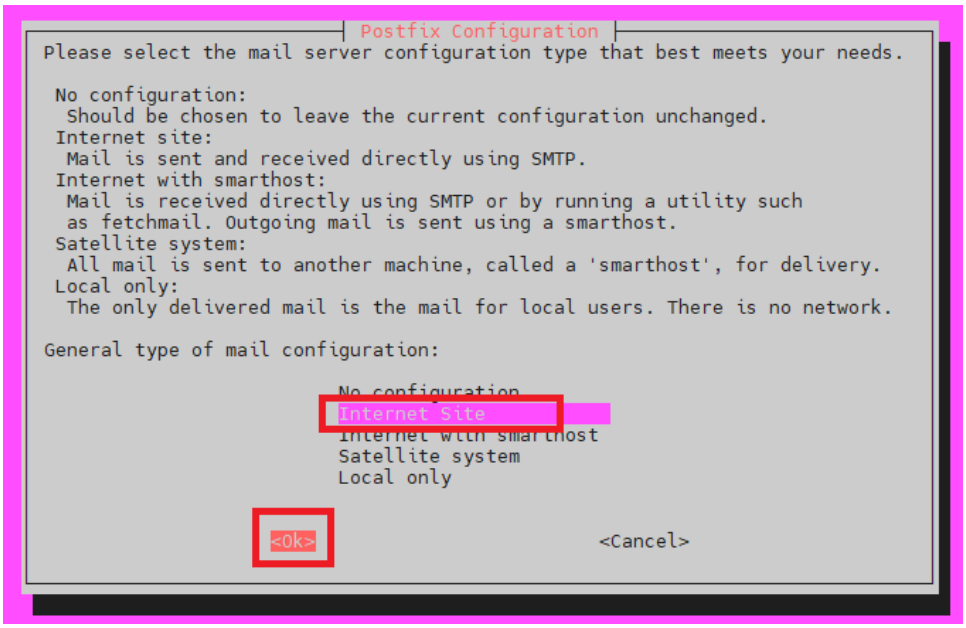
Install Postfix Email Server

We first install the postfix package, as well as, the authentication and security package for Ubuntu.

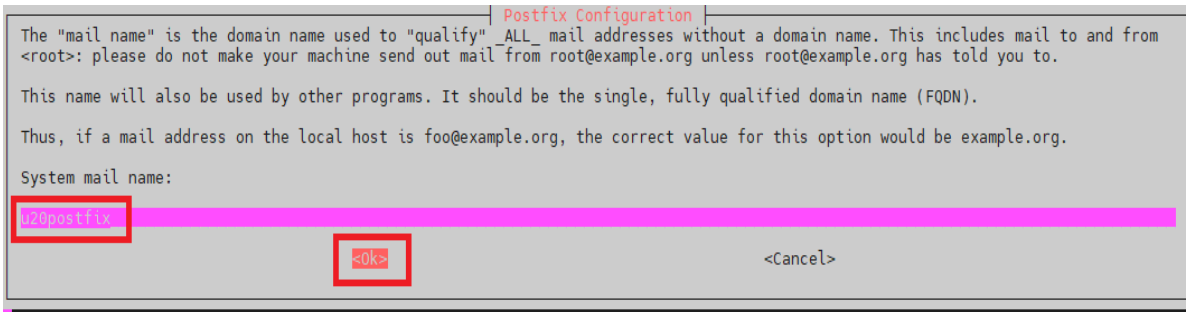
```
$ sudo apt-get install postfix libsasl2-modules -y
```



During Postfix installation, you will first be prompted for **type**. Select **Internet Site** and click **Ok**



Next, you will be prompted for **System mail name**. I entered **u20postfix** and clicked **Ok**



After Postfix is installed, we will check its status and see whether the service is enabled.

```
$ sudo systemctl is-enabled postfix
```

```
$ sudo systemctl status postfix
```

```
ubuntu@ip-172-31-4-185:~$ sudo systemctl is-enabled postfix
enabled
ubuntu@ip-172-31-4-185:~$ sudo systemctl status postfix
● postfix.service - Postfix Mail Transport Agent
     Loaded: loaded (/lib/systemd/system/postfix.service; enabled; vendor preset: enabled)
     Active: active (exited) since Mon 2022-07-18 16:12:00 UTC; 34s ago
   Main PID: 1863 (code=exited, status=0/SUCCESS)
      Tasks: 0 (limit: 1145)
     Memory: 0B
     CGroup: /system.slice/postfix.service

Jul 18 16:12:00 ip-172-31-4-185.ec2.internal systemd[1]: Starting Postfix Mail Transport Agent...
Jul 18 16:12:00 ip-172-31-4-185.ec2.internal systemd[1]: Finished Postfix Mail Transport Agent.
```

Postfix is running and is enabled to start during system startup.

Now, I will configure Postfix to use the **Amazon SES SMTP Service Endpoint** in my region (**us-east-1 US East (N. Virginia)**) along with the security group's inbound rule for port 587 (which I opened in my **Amazon SES Configuration** tutorial).

```
$ sudo postconf -e "relayhost = [email-smtp.us-east-1.amazonaws.com]:587" \
"smtp_sasl_auth_enable = yes" \
"smtp_sasl_security_options = noanonymous" \
"smtp_sasl_password_maps = hash:/etc/postfix/sasl_passwd" \
"smtp_use_tls = yes" \
"smtp_tls_security_level = encrypt" \
"smtp_tls_note_starttls_offer = yes"
```

```
ubuntu@ip-172-31-4-185:~$
ubuntu@ip-172-31-4-185:~$ sudo postconf -e "relayhost = [email-smtp.us-east-1.amazonaws.com]:587" \
> "smtp_sasl_auth_enable = yes" \
> "smtp_sasl_security_options = noanonymous" \
> "smtp_sasl_password_maps = hash:/etc/postfix/sasl_passwd" \
> "smtp_use_tls = yes" \
> "smtp_tls_security_level = encrypt" \
> "smtp_tls_note_starttls_offer = yes"
ubuntu@ip-172-31-4-185:~$
```

Using the Postfix configuration utility, **postconf**, the above command makes changes to `/etc/postfix/main.cf`

In the next step, we will create a password file that contains the SMTP credentials (**Amazon SES Configuration** tutorial). This will allow Postfix to connect to the **Amazon SES SMTP interface** which is acting as the **relayhost**.

"**relayhost = [email-smtp.us-east-1.amazonaws.com]:587**"

Open **your** SMTP credentials file (**credentials.csv**) downloaded while completing the **Amazon SES Configuration** tutorial and following the layout below, add to **/etc/postfix/sasl_passwd**

```
$ sudo vi /etc/postfix/sasl_passwd
```

```
# using STMP Credentials
```

```
#        SMTP Endpoint          SMTP Username  :   SMTP Password
```

```
[email-smtp.us-east-1.amazonaws.com]:587 AKIAYKM3A4YYIRWYC7JL:BFMyGjlh3Kmd+YYCqwMV6WPSlQyXb6X1eD6pGrakSNB6
```

```
ubuntu@ip-172-31-4-185:~$
ubuntu@ip-172-31-4-185:~$ sudo cat /etc/postfix/sasl_passwd
# using STMP Credentials
#             SMTP Endpoint              SMTP Username    :    SMTP Password
[email-smtp.us-east-1.amazonaws.com]:587 AKIAYKM3A4YYIRWYC7JL:BFMyGjlh3Kmd+YYCqwMV6WPSlQyXb6X1eD6pGrakSNB6
ubuntu@ip-172-31-4-185:~$
```

Next, type the following command to create a hashmap database file containing your SMTP credentials

```
$ sudo postmap hash:/etc/postfix/sasl_passwd
```

Now secure the **sasl_passwd** & **sasl_passwd.db** files by applying restrictive priviliges.

```
$ ls -l /etc/postfix | grep sasl_passwd*
```

```
$ sudo chmod 600 /etc/postfix/sasl_passwd*
```

```
$ ls -l /etc/postfix | grep sasl_passwd*
```

```
ubuntu@ip-172-31-4-185:~$
ubuntu@ip-172-31-4-185:~$ sudo postmap hash:/etc/postfix/sasl_passwd
ubuntu@ip-172-31-4-185:~$ ls -l /etc/postfix | grep sasl_passwd*
-rw-r--r-- 1 root root   190 Jul 18 16:41 sasl_passwd
-rw-r--r-- 1 root root 12288 Jul 18 16:44 sasl_passwd.db
ubuntu@ip-172-31-4-185:~$
ubuntu@ip-172-31-4-185:~$ sudo chmod 600 /etc/postfix/sasl_passwd*
ubuntu@ip-172-31-4-185:~$
ubuntu@ip-172-31-4-185:~$ ls -l /etc/postfix | grep sasl_passwd*
-rw------- 1 root root   190 Jul 18 16:41 sasl_passwd
-rw------- 1 root root 12288 Jul 18 16:44 sasl_passwd.db
```

Tell Postfix where to find the CA certificate (needed to verify the Amazon SES server certificate).

```
$ sudo postconf -e 'smtp_tls_CAfile = /etc/ssl/certs/ca-certificates.crt'
```

Since we are in the SES (Simple Email Service) sandbox, we can only use verified identities as the sender and recipient of emails. As I only have one verified identity, I will use the same email address for both sender and recipient.

Please note, for all of the following tests and configuration changes, use your verified identity (email address.)

Test Postfix Email Server using sendmail

First, create the contents of the email by creating a file called **input.txt** that contains the following:

Note: remember to use **your** SES verified identity (email address) as the sender address and place a period on the last line.

```
$ vi input.txt
```

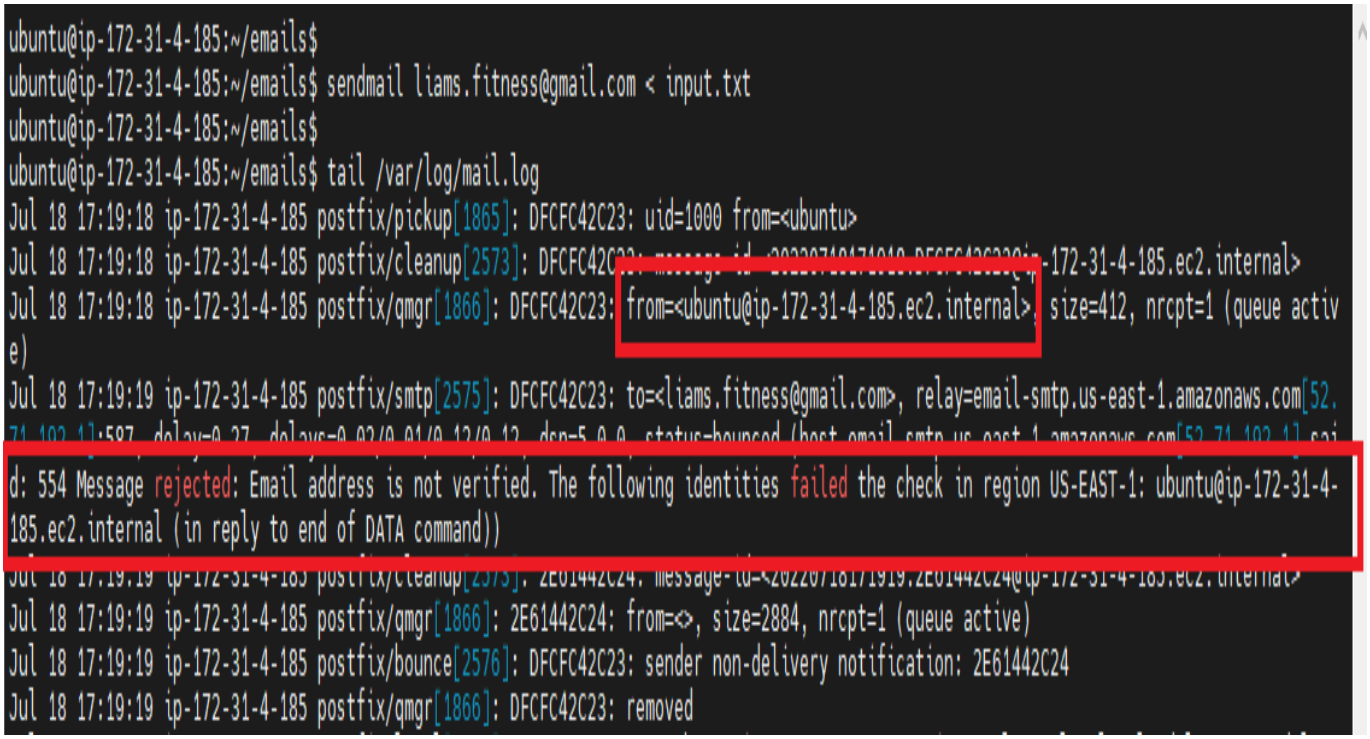
```
From: Sender Name <liams.fitness@gmail.com>
Subject: Amazon SES Test from Ubuntu 20 EC2
This message was sent using Amazon SES.
.
```

Then, execute the following command

```
$ sendmail liams.fitness@gmail.com < input.txt
```

If you did not receive the test email, an error occured, check the log file.

```
$ tail /var/log/mail.log
```



You will notice that the unverified identity was prevented from sending an email.

ubuntu@ip-172-31-4-185.ec2.internal

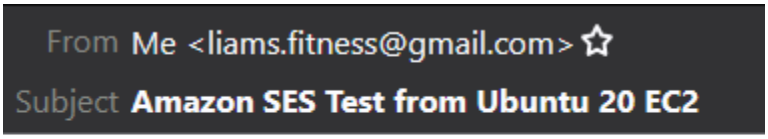
We will now change the command to the following (-f specifies sender):

```
$ sendmail -f liams.fitness@gmail.com liams.fitness@gmail.com < input.txt
```

```
$ tail /var/log/mail.log
```

```
ubuntu@ip-172-31-4-185:~/emails$
ubuntu@ip-172-31-4-185:~/emails$ sendmail -f liams.fitness@gmail.com liams.fitness@gmail.com < input.txt
ubuntu@ip-172-31-4-185:~/emails$
ubuntu@ip-172-31-4-185:~/emails$ tail /var/log/mail.log
Jul 18 17:19:19 ip-172-31-4-185 postfix/qmgr[1866]: 2E61442C24: from=<>, size=2884, nrcpt=1 (queue active)
Jul 18 17:19:19 ip-172-31-4-185 postfix/bounce[2576]: DFCFC42C23: sender non-delivery notification: 2E61442C24
Jul 18 17:19:19 ip-172-31-4-185 postfix/qmgr[1866]: DFCFC42C23: removed
Jul 18 17:19:19 ip-172-31-4-185 postfix/local[2578]: 2E61442C24: to=<ubuntu@ip-172-31-4-185.ec2.internal>, relay=local, delay=0.01, delays
=0/0.01/0/0, dsn=2.0.0, status=sent (delivered to mailbox)
Jul 18 17:19:19 ip-172-31-4-185 postfix/qmgr[1866]: 2E61442C24: removed
Jul 18 17:27:10 ip-172-31-4-185 postfix/pickup[1865]: 07BEA42C23: uid=1000 from=<liams.fitness@gmail.com>
Jul 18 17:27:10 ip-172-31-4-185 postfix/qmgr[1866]: 07BEA42C23: from=<liams.fitness@gmail.com>, size=412, nrcpt=1 (queue active)
Jul 18 17:27:10 ip-172-31-4-185 postfix/smtp[2597]: 07BEA42C23: to=<liams.fitness@gmail.com>, relay=email-smtp.us-east-1.amazonaws.com[54.
152.245.115]:587, delay=0.42, delays=0.02/0.01/0.19/0.2, dsn=2.0.0, status=sent (250 Ok 0100018212597aa8-5f308add-9990-4fdd-8e88-cb521776b
5b3-000000)
Jul 18 17:27:10 ip-172-31-4-185 postfix/qmgr[1866]: 07BEA42C23: removed
ubuntu@ip-172-31-4-185:~/emails$
```

I can confirm receipt by checking my email client:




```
This message was sent using Amazon SES.
.
```

Set Sender/Recipient Email Address

The next test emails will be transmitted as the result of successfully completed cron jobs. One for the **root** user and the other for the **ubuntu** user. Completed cron job notifications (emails) are sent locally, from the local user, to the local user. Since I am in the Amazon SES sandbox, only verified identities are capable of sending and receiving emails. I want these notifications sent from, and to, my Amazon SES verified identity (email address).

In order to do this, 2 changes must be made. The first, will be to configure Postfix address rewriting, for the sender, and the second will be to specify who should automatically be sent emails when no recipient is specified.

As we saw during the first test email sent via **sendmail**, the sender email address defaults to **username@hostname** (**ubuntu@ip-172-31-4-185.ec2.internal**). We can override this behaviour with Postfix address rewriting using the **smtp_generic_maps** parameter. The **smtp_generic_maps** parameter specifies generic lookup tables that replace local mail addresses with valid internet email addresses

when mail leaves the machine via SMTP. As it pertains to the Amazon SES sandbox, this will ensure that sent emails use a verified identity as their sender to prevent rejects.

To begin, open /etc/postfix/main.cf and add the following at the bottom of the file.

```
$ sudo vi /etc/postfix/main.cf
```

```
smtp_generic_maps = hash:/etc/postfix/generic
```

```
ubuntu@ip-172-31-4-185:~$
ubuntu@ip-172-31-4-185:~$ tail /etc/postfix/main.cf
smtp_tls_note_starttls_offer = yes
smtp_sasl_password_maps = hash:/etc/postfix/sasl_passwd
smtp_sasl_security_options = noanonymous
smtp_sasl_auth_enable = yes
smtp_use_tls = yes
smtp_tls_CAfile = /etc/ssl/certs/ca-certificates.crt
smtp_generic_maps = hash:/etc/postfix/generic
```

Open /etc/postfix/generic. If it doesn't already exist, create it and use **your** verified identity:

```
$ sudo vi /etc/postfix/generic
```

```
root     liams.fitness@gmail.com
```

```
ubuntu   liams.fitness@gmail.com
```

```
ubuntu@ip-172-31-4-185:~$
ubuntu@ip-172-31-4-185:~$ sudo cat /etc/postfix/generic
root     liams.fitness@gmail.com
ubuntu     liams.fitness@gmail.com
ubuntu@ip-172-31-4-185:~$
ubuntu@ip-172-31-4-185:~$
```

Create or update the generic postfix table using the **postmap** command and reload the Postfix service.

```
$ sudo postmap /etc/postfix/generic
```

```
$ sudo systemctl postfix reload
```

Then, execute the following command as the **root** user without specifying a sender email address.

```
$ sudo sendmail liams.fitness@gmail.com < input.txt
```

Followed by executing the same command as the **ubuntu** user.

```
$ sendmail liams.fitness@gmail.com < input.txt
```

```
ubuntu@ip-172-31-4-185:~$
ubuntu@ip-172-31-4-185:~$ sudo postmap /etc/postfix/generic
ubuntu@ip-172-31-4-185:~$
ubuntu@ip-172-31-4-185:~$ sudo systemctl reload postfix
ubuntu@ip-172-31-4-185:~$
ubuntu@ip-172-31-4-185:~$ sudo sendmail liams.fitness@gmail.com < input.txt
ubuntu@ip-172-31-4-185:~$
ubuntu@ip-172-31-4-185:~$ sendmail liams.fitness@gmail.com < input.txt
ubuntu@ip-172-31-4-185:~$
```

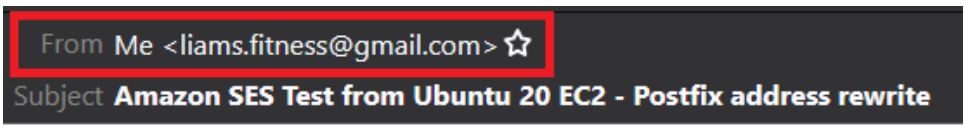
Check the log file at **/var/log/mail.log** to confirm that both the emails were sent.

`$ tail /var/log/mail.log`

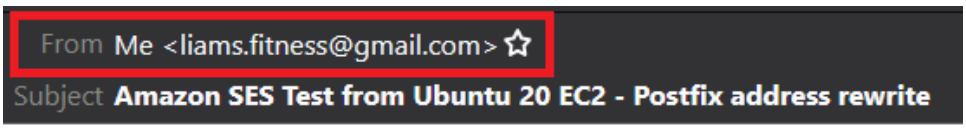
```
ubuntu@ip-172-31-4-185:~$
ubuntu@ip-172-31-4-185:~$ tail /var/log/mail.log
Jul 29 15:17:38 ip-172-31-4-185 postfix/pickup[6393]: C0ADB42B19: uid=0 from=<root>
Jul 29 15:17:38 ip-172-31-4-185 postfix/cleanup[6398]: C0ADB42B19: message-id=<20220729151738.C0ADB42B19@ip-
Jul 29 15:17:38 ip-172-31-4-185 postfix/qmgr[6392]: C0ADB42B19: from=<root@ip-172-31-4-185.ec2.internal>, si
Jul 29 15:17:39 ip-172-31-4-185 postfix/smtp[6400]: C0ADB42B19: to=<liams.fitness@gmail.com>, relay=email sm
.31.4.27]:587, delay=0.39, delays=0.02/0.02/0.04/0.31, dsn=2.0.0, status=sent (250 Ok 010001824a88d9ba-90 lc3
3-000000)
Jul 29 15:17:39 ip-172-31-4-185 postfix/qmgr[6392]: C0ADB42B19: removed
Jul 29 15:17:56 ip-172-31-4-185 postfix/pickup[6393]: 2C37C42B19: uid=1000 from=<ubuntu>
Jul 29 15:17:56 ip-172-31-4-185 postfix/cleanup[6398]: 2C37C42B19: message-id=<20220729151756.2C37C42B19@ip-
Jul 29 15:17:56 ip-172-31-4-185 postfix/qmgr[6392]: 2C37C42B19: from=<ubuntu@ip-172-31-4-185.ec2.internal>,
e)
Jul 29 15:17:56 ip-172-31-4-185 postfix/smtp[6400]: 2C37C42B19: to=<liams.fitness@gmail.com>, relay=email-s
.31.4.27]:587, delay=0.38, delays=0.01/0/0.04/0.33, dsn=2.0.0, status=sent (250 Ok 010001824a891d8d-422a62f
00000)
Jul 29 15:17:56 ip-172-31-4-185 postfix/qmgr[6392]: 2C37C42B19: removed
ubuntu@ip-172-31-4-185:~$
```

Notice that the **from** email addresses in the log are **root@ip-172-31-4-185.ec2.internal** and **ubuntu@ip-172-31-4-185.ec2.internal**, but the emails were sent.

Finally, from my email client, I can confirm that I've received two new emails from my verified identity.




```
This message was sent using Amazon SES.
.
```




```
This message was sent using Amazon SES.
.
```

This confirms that address rewriting was performed by Postfix.

Now we will specify who should automatically be sent emails when no recipient is specified. As it pertains to the Amazon SES sandbox, this will ensure that sent emails use a verified identity as their recipient to prevent rejects.

To begin, in the file **/etc/aliases** change the line that begins with postmaster:

```
postmaster:    root, ubuntu
```

Next, add the following entries to **/etc/aliases** (use **your** verified identity)

```
root: liams.fitness@gmail.com
```

```
ubuntu: liams.fitness@gmail.com
```

Now, to ensure that the aliases database file is updated by executing the following:

```
$ sudo newaliases
```

```
ubuntu@ip-172-31-4-185:~$
ubuntu@ip-172-31-4-185:~$ cat /etc/aliases
# See man 5 aliases for format
postmaster:    root,ubuntu
root: liams.fitness@gmail.com
ubuntu: liams.fitness@gmail.com

ubuntu@ip-172-31-4-185:~$ sudo newaliases
ubuntu@ip-172-31-4-185:~$
ubuntu@ip-172-31-4-185:~$
```

We can now configure a simple cron job, one for **root** and one for **ubuntu**, to test notification transmission, via Amazon SES, for successfully completed cron jobs.

Test Postfix Email Server using cronjob

We will create a simple cron job for each local user, **root** & **ubuntu**. Then, we will verify notification transmission via Amazon SES.

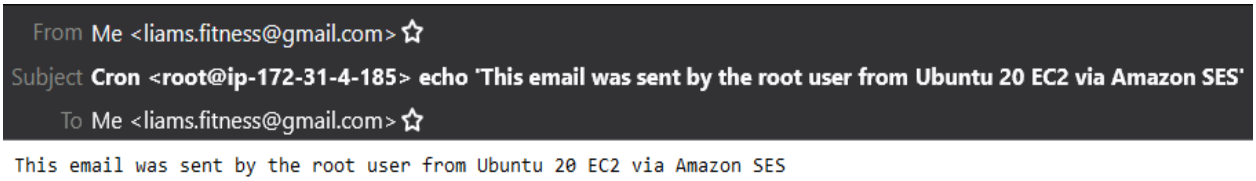
First, we will add a simple cron job to the root user's crontab file. For testing purposes, I will have the job execute every minute.

```
$ sudo crontab -e
```

```
# m h dom mon dow     (m==minute, h==hour, dom==day of month, mon==month, dow==day of week)
*/1 * * * * echo 'This email was sent by the root user from Ubuntu 20 EC2 via Amazon SES'
```

```
# m h dom mon dow      (m==minute, h==hour, dom==day of month, mon==month, dow==day of week)
*/1 * * * * echo 'This email was sent by the root user from Ubuntu 20 EC2 via Amazon SES'
```

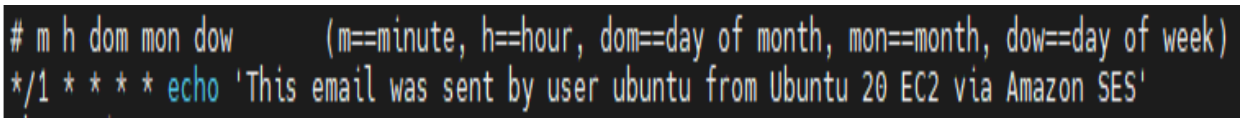
I will now confirm that the cron notification email was sent using my email client.



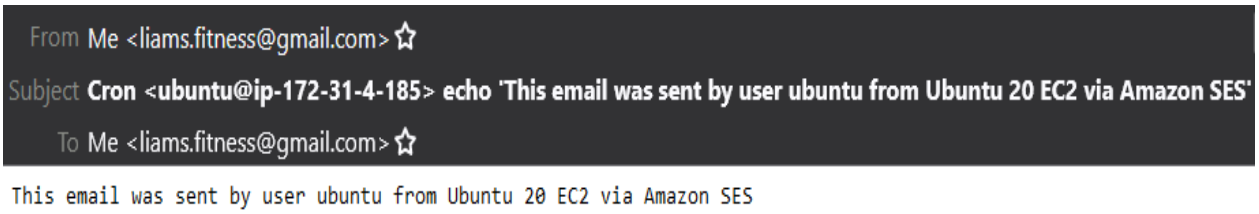
Now I will add a simple cron job to the **ubuntu** user's crontab file. Again, for testing purposes, I will have the job execute every minute.

```
$ crontab -e
```

```
# m h dom mon dow     (m==minute, h==hour, dom==day of month, mon==month, dow==day of week)
*/1 * * * * echo 'This email was sent by user ubuntu from Ubuntu 20 EC2 via Amazon SES'
```



I will now confirm that the cron notification email was sent using my email client.



Both cron job notifications were successfully sent and received. Now would be a good time to disable each cron job (root & ubuntu) to prevent unnecessary notification transmissions. Do this by commenting out, or deleting, each job via **`sudo crontab -e`** for root & **`crontab -e`** for ubuntu.

I hope you've enjoyed this tutorial.

We've successfully configured Postfix as an outbound send-only email server on an Ubuntu 20 EC2.

Now we can send emails from our Ubuntu 20 EC2 instance either manually using **sendmail**, or automatically, via **cron jobs**.

I have another tutorial where I configure Postfix as an outbound send-only email server on my **RHEL 8** EC2 instance, accessible **here**. Although the steps are practically identical, I wanted to have two different instances for testing purposes. This will enable me to test my shell scripts that monitor services locally, as well as, remotely. If you're interested, the scripts can be accessed via my Linux tutorials page, **here**, while my main tutorials page is accessible **here**.

Please note that the free tier allows for 750 hours per month of Amazon EC2. You can create many EC2 instances but beware of the limit. If you go over that limit, you will pay the cost. My advice to you is to shutdown your instance/s after you've done your work.

Back to Top