

## AWS RHEL 8 EC2 Postfix Install

In this tutorial, we will be installing Postfix as an outbound send-only email server on an RHEL 8 EC2.

### Prerequisites

- an AWS Free Tier account
- AWS RHEL 8 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 RHEL 8 EC2 instance, my tutorial **Create AWS RHEL 8 EC2 Instance** is [here](#).

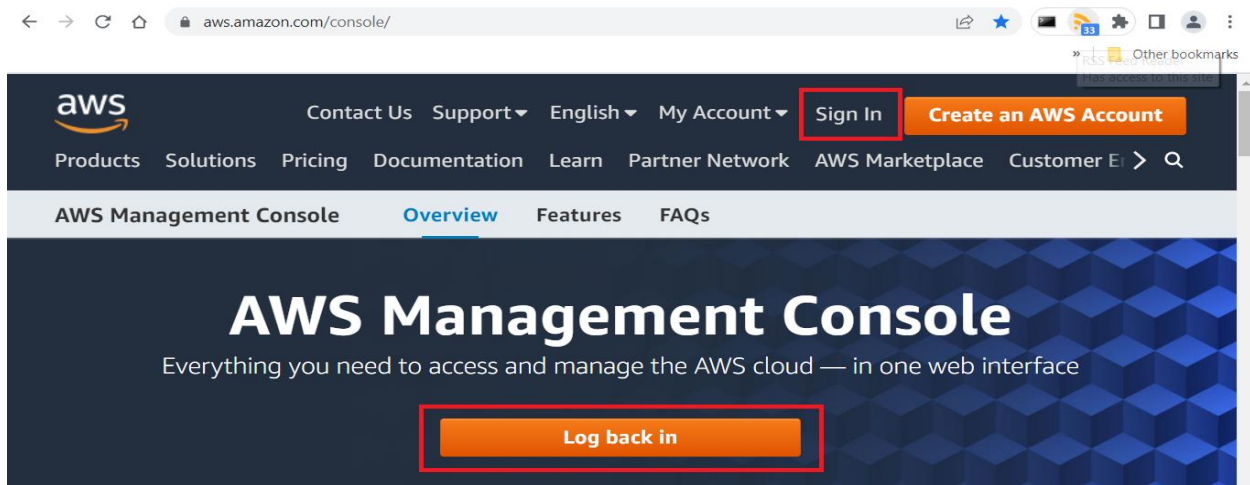
Finally, if you have not yet configured the required components to use Amazon SES, complete my other tutorial **Amazon SES Configuration**, accessible [here](#), then return to this tutorial.

Steps to complete tutorial:

- [Start RHEL 8 EC2 Instance](#)
- [Connect to RHEL 8](#)
- [Install Postfix Email Server](#)
- [Test Postfix Email Server using sendmail](#)
- [Set Sender/Recipient Email Address](#)
- [Test Postfix Email Server using cronjob](#)

### Start RHEL 8 EC2 Instance

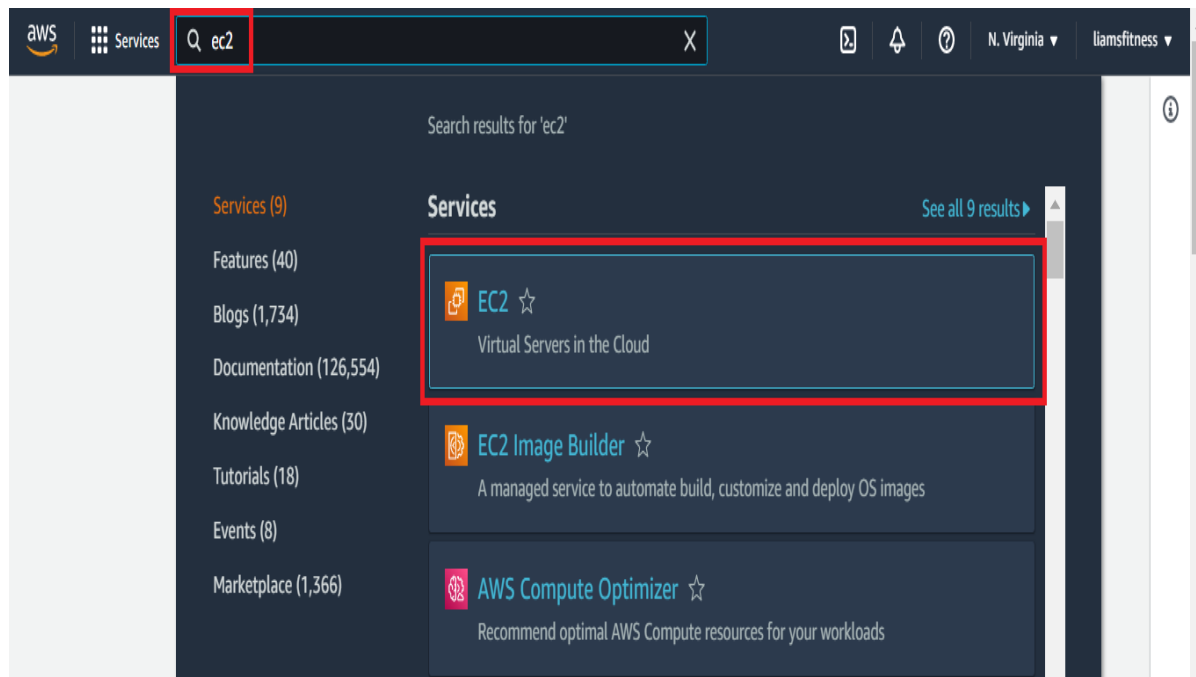
To begin, go to the following website, <https://aws.amazon.com/console/> and log in to the console.



At the end of my **Create AWS RHEL 8 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.

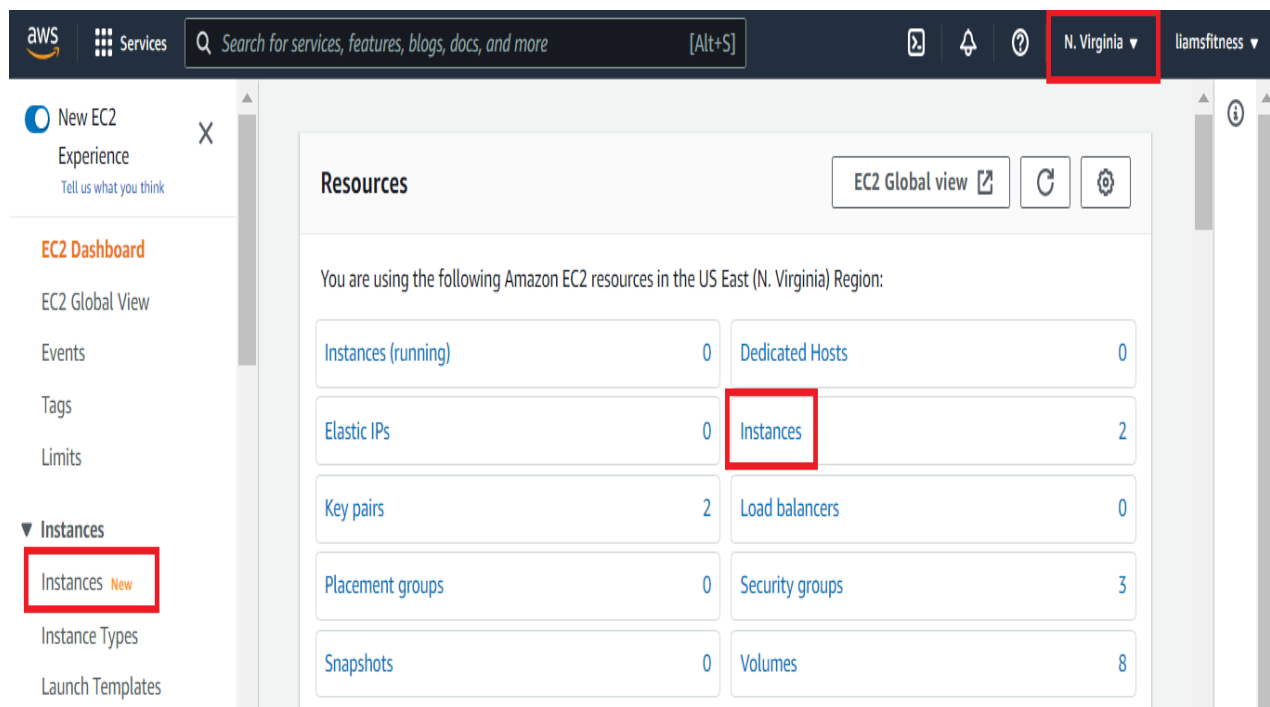
If you already know that your RHEL 8 EC2 instance is running, you can skip this step and go directly to [Connect to RHEL 8](#).

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.

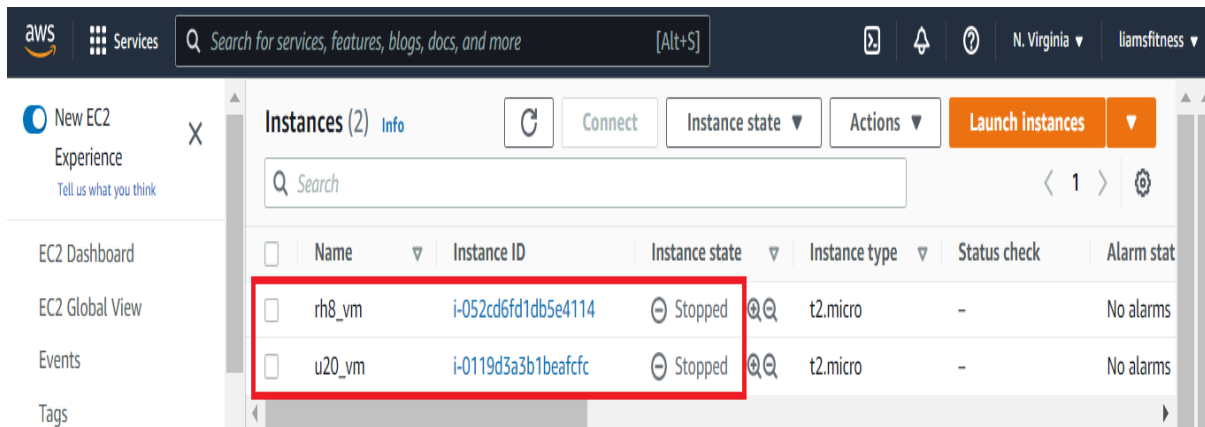


You will be brought to the **EC2 Dashboard**. It contains links to the resources being used in the selected AWS region. In my case it's US East (N. Virginia).

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



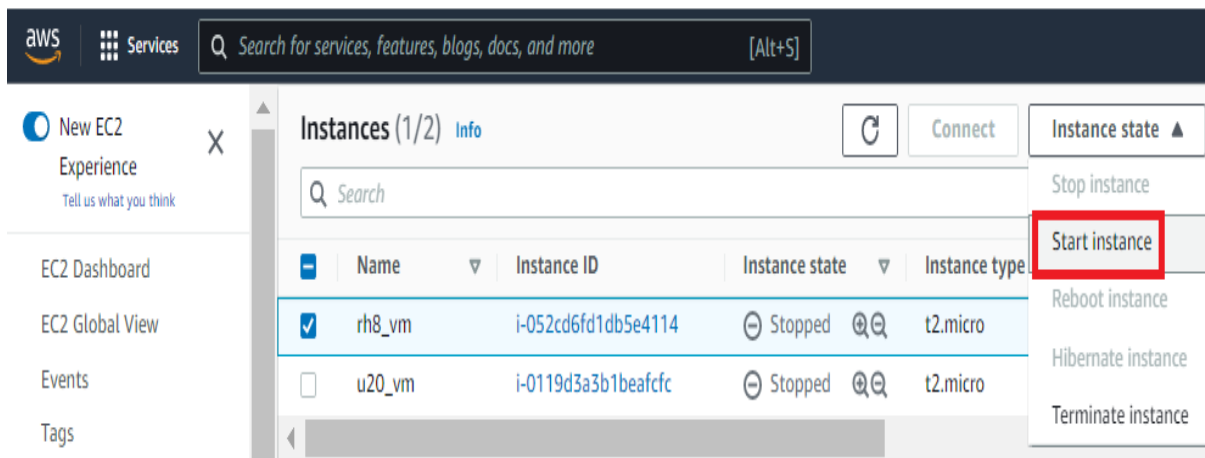
You will notice that both of my instances are in the stopped stated.



The screenshot shows the AWS Management Console with the 'Instances' page. Two instances are listed: 'rh8\_vm' and 'u20\_vm'. Both are in the 'Stopped' state. A red box highlights the rows for both instances.

Name	Instance ID	Instance state	Instance type	Status check	Alarm stat
rh8_vm	i-052cd6fd1db5e4114	Stopped	t2.micro	-	No alarms
u20_vm	i-0119d3a3b1beafcf	Stopped	t2.micro	-	No alarms

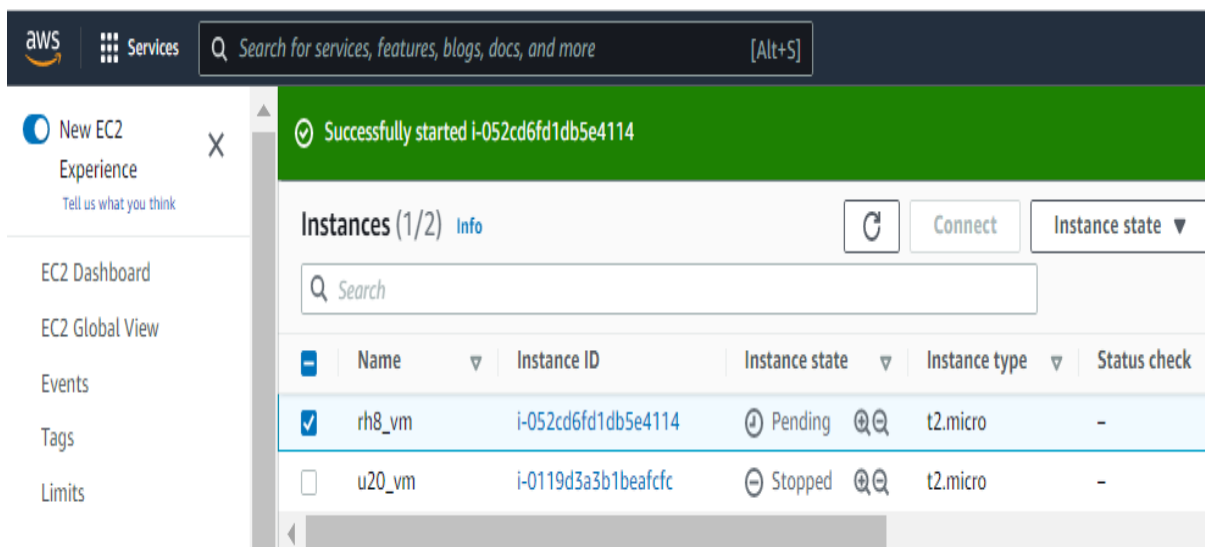
To start my RHEL 8 instance, I will ensure that my **rh8\_vm** is selected (checkbox), then I will click the **Instance state** drop-down menu and click **Start instance**.



The screenshot shows the AWS Management Console with the 'Instances' page. The 'rh8\_vm' instance is selected (checkbox checked). The 'Instance state' dropdown menu is open, and the 'Start instance' option is highlighted with a red box.

Name	Instance ID	Instance state	Instance type
rh8_vm	i-052cd6fd1db5e4114	Stopped	t2.micro
u20_vm	i-0119d3a3b1beafcf	Stopped	t2.micro

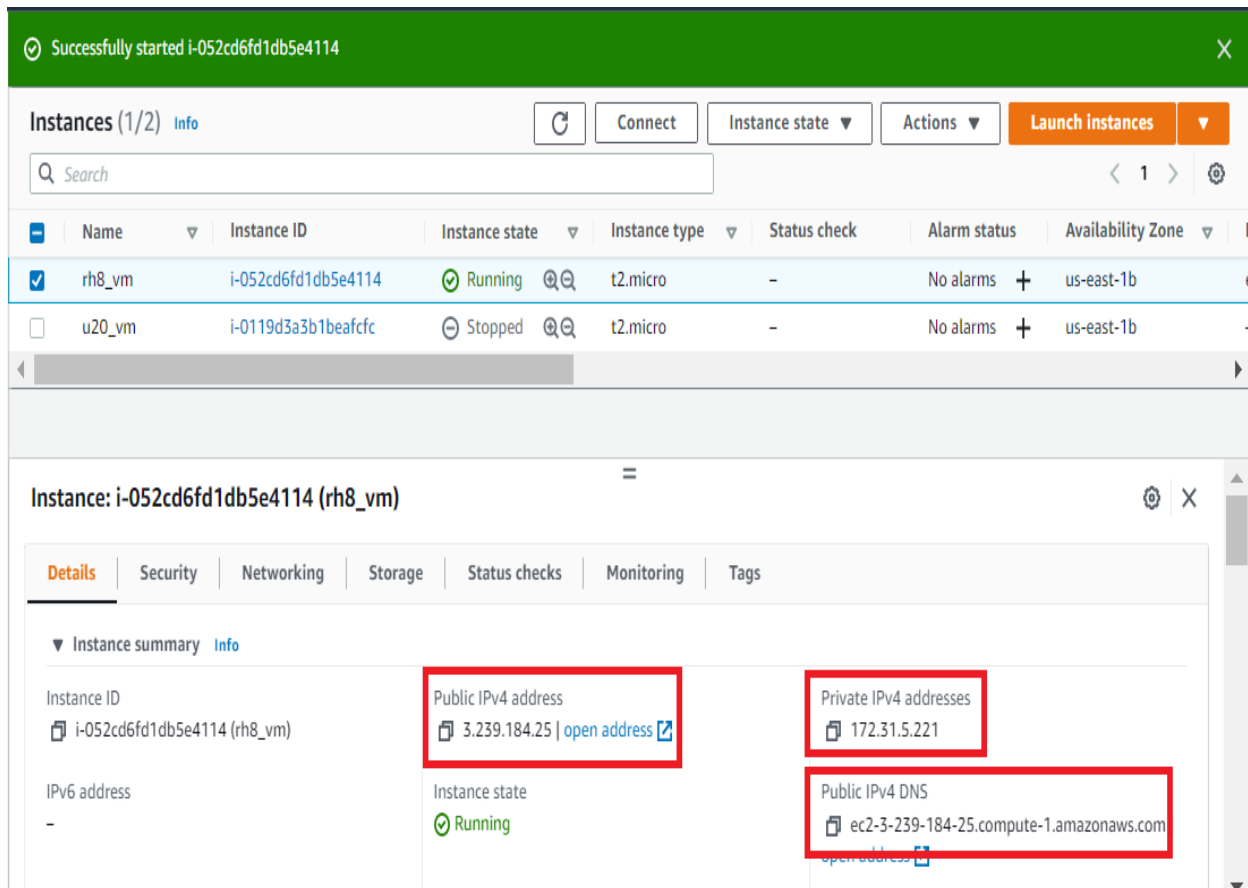
The instance will start and be given new connection details.



The screenshot shows the AWS Management Console with a green notification banner at the top stating 'Successfully started i-052cd6fd1db5e4114'. Below the banner, the 'rh8\_vm' instance is now in the 'Pending' state.

Name	Instance ID	Instance state	Instance type	Status check
rh8_vm	i-052cd6fd1db5e4114	Pending	t2.micro	-
u20_vm	i-0119d3a3b1beafcf	Stopped	t2.micro	-

Ensure your new instance is selected (**rh8\_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.



The screenshot shows the AWS Management Console interface. At the top, a green banner indicates "Successfully started i-052cd6fd1db5e4114". Below this, the "Instances (1/2)" page is displayed. A table lists two instances: "rh8\_vm" (ID: i-052cd6fd1db5e4114, state: Running) and "u20\_vm" (ID: i-0119d3a3b1beafcfc, state: Stopped). The "rh8\_vm" instance is selected. Below the table, the "Details" tab for instance "i-052cd6fd1db5e4114 (rh8\_vm)" is open. The "Instance summary" section shows the instance ID, IPv6 address, and instance state (Running). The "Networking" section shows the public IPv4 address (3.239.184.25) and private IPv4 addresses (172.31.5.221). The "Public IPv4 DNS" is listed as ec2-3-239-184-25.compute-1.amazonaws.com. Red boxes highlight the public IPv4 address, private IPv4 addresses, and public IPv4 DNS.

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
rh8_vm	i-052cd6fd1db5e4114	Running	t2.micro	-	No alarms	us-east-1b
u20_vm	i-0119d3a3b1beafcfc	Stopped	t2.micro	-	No alarms	us-east-1b

**Instance: i-052cd6fd1db5e4114 (rh8\_vm)**

**Details** | Security | Networking | Storage | Status checks | Monitoring | Tags

**Instance summary** Info

Instance ID: i-052cd6fd1db5e4114 (rh8\_vm)

IPv6 address: -

Instance state: Running

Public IPv4 address: 3.239.184.25 | open address

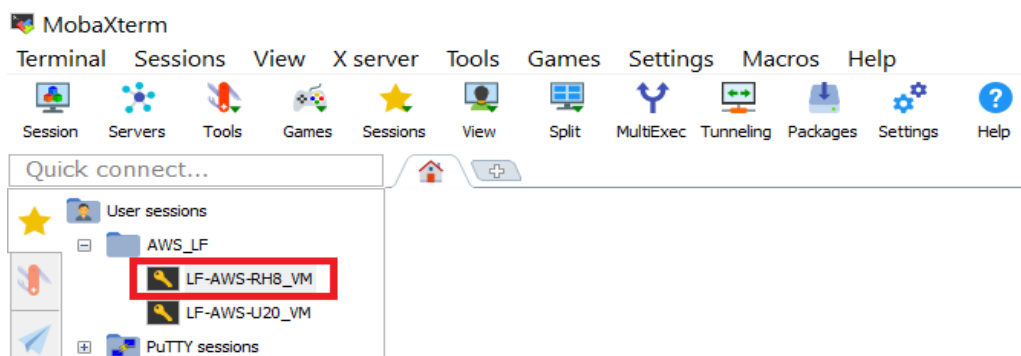
Private IPv4 addresses: 172.31.5.221

Public IPv4 DNS: ec2-3-239-184-25.compute-1.amazonaws.com | open address

## Connect to RHEL 8

As I mentioned in my **Create AWS RHEL 8 EC2 Instance** tutorial, accessible [here](#), I am using **MobaXterm Portable** as my SSH client. Since I restarted my RHEL 8 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 RHEL 8 instance.

After updating my saved SSH connection, from the MobaXterm main interface, I double clicked my SSH session to connect to my RHEL 8 EC2 instance.



The screenshot shows the MobaXterm main interface. The "Quick connect..." dialog is open, displaying a list of user sessions. The session "LF-AWS-RH8\_VM" is highlighted with a red box. The interface includes a menu bar (Terminal, Sessions, View, X server, Tools, Games, Settings, Macros, Help) and a toolbar with icons for Session, Servers, Tools, Games, Sessions, View, Split, MultiExec, Tunneling, Packages, Settings, and Help.

**MobaXterm**

Terminal | Sessions | View | X server | Tools | Games | Settings | Macros | Help

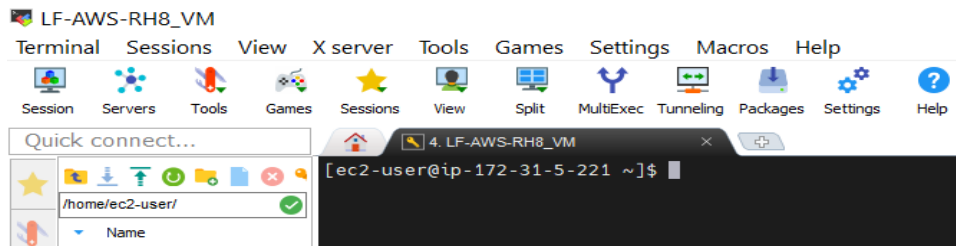
Session | Servers | Tools | Games | Sessions | View | Split | MultiExec | Tunneling | Packages | Settings | Help

Quick connect...

User sessions

- ★ AWS\_LF
  - LF-AWS-RH8\_VM
  - LF-AWS-U20\_VM
- Putty sessions

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

Please note that your RHEL 8 instance must be registered, and have access to a Red Hat subscription, in order to install packages. RHEL 8 registration was covered in my **Create AWS RHEL 8 EC2 Instance** tutorial, accessible [here](#). To verify system registration, issue the following:

```
$ sudo subscription-manager list
```

```
[ec2-user@ip-172-31-5-221 ~]$  
[ec2-user@ip-172-31-5-221 ~]$ sudo subscription-manager list  
+-----+  
Installed Product Status  
+-----+  
Product Name:   Red Hat Enterprise Linux for x86_64  
Product ID:     479  
Version:        8.4  
Arch:           x86_64  
Status:         Subscribed  
Status Details:  
Starts:         05/10/2022  
Ends:           05/10/2023  
[ec2-user@ip-172-31-5-221 ~]$
```

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

```
$ sudo yum install postfix cyrus-sasl-plain -y
```

```
[ec2-user@ip-172-31-5-221 ~]$  
[ec2-user@ip-172-31-5-221 ~]$ sudo yum install postfix cyrus-sasl-plain -y
```

Post-installation, we must start and enable the Postfix service.

```
$ sudo systemctl start postfix
```

```
$ sudo systemctl enable postfix
```

```
[ec2-user@ip-172-31-5-221 ~]$  
[ec2-user@ip-172-31-5-221 ~]$ sudo systemctl start postfix  
[ec2-user@ip-172-31-5-221 ~]$  
[ec2-user@ip-172-31-5-221 ~]$ sudo systemctl enable postfix  
Created symlink /etc/systemd/system/multi-user.target.wants/postfix.service → /usr/lib/systemd/system/postfix.service.  
[ec2-user@ip-172-31-5-221 ~]$
```

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"
```

```
[ec2-user@ip-172-31-5-221 ~]$  
[ec2-user@ip-172-31-5-221 ~]$ 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" \  
[ec2-user@ip-172-31-5-221 ~]$
```

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:BFMyGj1h3Kmd+YYCqwMV6WPSlQyXb6X1eD6pGrakSNB6
```

```
# using STMP Credentials  
# SMTP Endpoint SMTP Username : SMTP Password  
[email-smtp.us-east-1.amazonaws.com]:587 AKIAYKM3A4YYIRWYC7JL:BFMyGj1h3Kmd+YYCqwMV6WPSlQyXb6X1eD6pGrakSNB6
```

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 privileges.

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

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

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

```
[ec2-user@ip-172-31-5-221 ~]$  
[ec2-user@ip-172-31-5-221 ~]$ sudo postmap hash:/etc/postfix/sasl_passwd  
[ec2-user@ip-172-31-5-221 ~]$  
[ec2-user@ip-172-31-5-221 ~]$ ls -l /etc/postfix/sasl_passwd*  
-rw-r--r--. 1 root root 205 Jul 28 16:28 /etc/postfix/sasl_passwd  
-rw-r--r--. 1 root root 12288 Jul 28 16:29 /etc/postfix/sasl_passwd.db  
[ec2-user@ip-172-31-5-221 ~]$  
[ec2-user@ip-172-31-5-221 ~]$ sudo chmod 600 /etc/postfix/sasl_passwd*  
[ec2-user@ip-172-31-5-221 ~]$  
[ec2-user@ip-172-31-5-221 ~]$ ls -l /etc/postfix/sasl_passwd*  
-rw-----. 1 root root 205 Jul 28 16:28 /etc/postfix/sasl_passwd  
-rw-----. 1 root root 12288 Jul 28 16:29 /etc/postfix/sasl_passwd.db  
[ec2-user@ip-172-31-5-221 ~]$
```

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-bundle.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 RHEL 8 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 occurred, check the log file.

```
$ sudo tail /var/log/maillog
```

```
[ec2-user@ip-172-31-5-221 ~]$  
[ec2-user@ip-172-31-5-221 ~]$ sudo tail /var/log/maillog  
Jul 28 16:35:47 ip-172-31-5-221 postfix/pickup[14175]: 63C145E098: uid=1000 from=<ec2-user>  
Jul 28 16:35:47 ip-172-31-5-221 postfix/cleanup[14579]: 63C145E098: message-id=<20220728163547.63C145E098@ip-172-31-5-221.ec2.internal>  
Jul 28 16:35:47 ip-172-31-5-221 postfix/qmgr[14176]: 63C145E098: from=<ec2-user@ip-172-31-5-221.ec2.internal> size=412, nrcpt=1 (queue active)  
Jul 28 16:35:47 ip-172-31-5-221 postfix/smtp[14581]: 63C145E098: to=<liams.fitness@gmail.com>, relay=email-smtp.us-east-1.amazonaws.com[172.31.4.27]:587, delay=0.34, delays=0.03/0.04/0.12/0.16, dsn=5.0.0, status=bounced (host email-smtp.us-east-1.amazonaws.com[172.31.4.27] said: 554 Message rejected: Email address is not verified. The following identities failed the check in region US-EAST-1: ec2-user@ip-172-31-5-221.ec2.internal (in reply to end of DATA command))  
Jul 28 16:35:47 ip-172-31-5-221 postfix/cleanup[14579]: B77645E099: message-id=<20220728163547.B77645E099@ip-172-31-5-221.ec2.internal>  
Jul 28 16:35:47 ip-172-31-5-221 postfix/qmgr[14176]: B77645E099: from=<>, size=2894, nrcpt=1 (queue active)  
Jul 28 16:35:47 ip-172-31-5-221 postfix/bounce[14583]: 63C145E098: sender non-delivery notification: B77645E099  
Jul 28 16:35:47 ip-172-31-5-221 postfix/qmgr[14176]: 63C145E098: removed  
Jul 28 16:35:47 ip-172-31-5-221 postfix/local[14584]: B77645E099: to=<ec2-user@ip-172-31-5-221.ec2.internal>, relay=local, delay=0.01, delays=0/0.01/0/0, dsn=2.0.0, status=sent (delivered to mailbox)  
Jul 28 16:35:47 ip-172-31-5-221 postfix/qmgr[14176]: B77645E099: removed  
[ec2-user@ip-172-31-5-221 ~]$
```

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

```
ec2-user@ip-172-31-5-221.ec2.internal
```

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

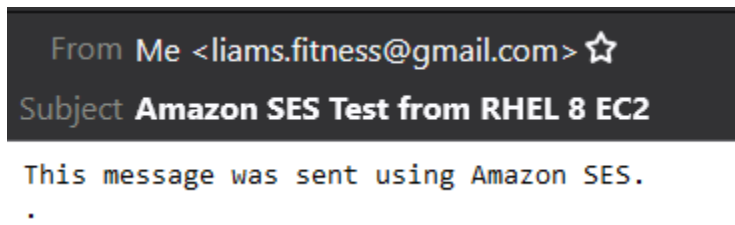
```
$ sendmail -f liams.fitness@gmail.com liams.fitness@gmail.com < input.txt
```

```
$ sudo tail /var/log/maillog
```

```
[ec2-user@ip-172-31-5-221 ~]$  
[ec2-user@ip-172-31-5-221 ~]$ sudo tail /var/log/maillog  
Jul 28 16:35:47 ip-172-31-5-221 postfix/qmgr[14176]: B77645E099: from=<>, size=2894, nrcpt=1 (queue active)  
Jul 28 16:35:47 ip-172-31-5-221 postfix/bounce[14583]: 63C145E098: sender non-delivery notification: B77645E099  
Jul 28 16:35:47 ip-172-31-5-221 postfix/qmgr[14176]: 63C145E098: removed  
Jul 28 16:35:47 ip-172-31-5-221 postfix/local[14584]: B77645E099: to=<ec2-user@ip-172-31-5-221.ec2.internal>, relay=local, delay=0.01, delays=0/0.01/0/0, dsn=2.0.0, status=sent (delivered to mailbox)  
Jul 28 16:35:47 ip-172-31-5-221 postfix/qmgr[14176]: B77645E099: removed  
Jul 28 16:41:15 ip-172-31-5-221 postfix/pickup[14175]: 0F0B65E098: uid=1000 from=<liams.fitness@gmail.com>  
Jul 28 16:41:15 ip-172-31-5-221 postfix/cleanup[14579]: 0F0B65E098: message-id=<20220728164115.0F0B65E098@ip-172-31-5-221.ec2.internal>  
Jul 28 16:41:15 ip-172-31-5-221 postfix/qmgr[14176]: 0F0B65E098: from=<liams.fitness@gmail.com> size=412, nrcpt=1 (queue active)  
Jul 28 16:41:15 ip-172-31-5-221 postfix/smtp[14714]: 0F0B65E098: to=<liams.fitness@gmail.com>, relay=email-smtp.us-east-1.amazonaws.com[172.31.4.27]:587, delay=0.34, delays=0.03/0.02/0.12/0.18, dsn=2.0.0, status=sent (250 Ok 010001825af08eb-44b3f995-d6a0-4773-a825-96b49b176e54-00000000)  
Jul 28 16:41:15 ip-172-31-5-221 postfix/qmgr[14176]: 0F0B65E098: removed  
[ec2-user@ip-172-31-5-221 ~]$
```



I can confirm receipt by checking my email client:



## 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 **ec2-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** (**ec2-user@ip-172-31-5-221.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
```

```
[ec2-user@ip-172-31-5-221 ~]$  
[ec2-user@ip-172-31-5-221 ~]$ sudo vi /etc/postfix/main.cf  
[ec2-user@ip-172-31-5-221 ~]$ sudo tail /etc/postfix/main.cf  
shlib_directory = /usr/lib64/postfix  
smtp_tls_note_starttls_offer = yes  
smtp_sasl_password_maps = hash:/etc/postfix/sasl_passwd  
smtp_sasl_security_options = noanonymous  
relayhost = [email-smtp.us-east-1.amazonaws.com]:587  
smtp_sasl_auth_enable = yes  
smtp_use_tls = yes  
  
smtp_generic_maps = hash:/etc/postfix/generic  
[ec2-user@ip-172-31-5-221 ~]$
```

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
```

```
ec2-user  liams.fitness@gmail.com
```

```
[ec2-user@ip-172-31-5-221 ~]$
[ec2-user@ip-172-31-5-221 ~]$ sudo tail -n 4 /etc/postfix/generic
#
#
root      liams.fitness@gmail.com
ec2-user  liams.fitness@gmail.com
[ec2-user@ip-172-31-5-221 ~]$
```

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

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

```
$ sudo systemctl reload postfix
```

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 **ec2-user** user.

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

```
[ec2-user@ip-172-31-5-221 ~]$
[ec2-user@ip-172-31-5-221 ~]$ sudo postmap /etc/postfix/generic
[ec2-user@ip-172-31-5-221 ~]$
[ec2-user@ip-172-31-5-221 ~]$ sudo systemctl reload postfix
[ec2-user@ip-172-31-5-221 ~]$
[ec2-user@ip-172-31-5-221 ~]$ sudo sendmail liams.fitness@gmail.com < input.txt
[ec2-user@ip-172-31-5-221 ~]$
[ec2-user@ip-172-31-5-221 ~]$ sendmail liams.fitness@gmail.com < input.txt
[ec2-user@ip-172-31-5-221 ~]$
```

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

```
$ tail -n 10 /var/log/maillog
```

```
[ec2-user@ip-172-31-5-221 ~]$
[ec2-user@ip-172-31-5-221 ~]$ sudo tail -n 10 /var/log/maillog
Jul 29 15:51:05 ip-172-31-5-221 postfix/pickup[36367]: 0AE585E098: uid=0 from=<root>
Jul 29 15:51:05 ip-172-31-5-221 postfix/cleanup[36474]: 0AE585E098: message_id=20220729155105.0AE585E098@ip-172-31-5-221.ec2.internal
Jul 29 15:51:05 ip-172-31-5-221 postfix/qmgr[36366]: 0AE585E098: from=<root@ip-172-31-5-221.ec2.internal>, size=0
Jul 29 15:51:05 ip-172-31-5-221 postfix/smtp[36476]: 0AE585E098: to=<liams.fitness@gmail.com>, relay=email-smtp-2.31.4.27:587, delay=0.38, delays=0.03/0.03/0.11/0.2, dsn=2.0.0, status=sent (250 Ok 010001824aa77705-dec53b84e-000000)
Jul 29 15:51:05 ip-172-31-5-221 postfix/qmgr[36366]: 0AE585E098: removed
Jul 29 15:51:12 ip-172-31-5-221 postfix/pickup[36367]: D9EE95E098: uid=1000 from=<ec2-user>
Jul 29 15:51:12 ip-172-31-5-221 postfix/cleanup[36474]: D9EE95E098: message_id=20220729155112.D9EE95E098@ip-172-31-5-221.ec2.internal
Jul 29 15:51:12 ip-172-31-5-221 postfix/qmgr[36366]: D9EE95E098: from=<ec2-user@ip-172-31-5-221.ec2.internal>, size=0
Jul 29 15:51:13 ip-172-31-5-221 postfix/smtp[36476]: D9EE95E098: to=<liams.fitness@gmail.com>, relay=email-smtp-2.31.4.27:587, delay=0.3, delays=0.01/0/0.12/0.18, dsn=2.0.0, status=sent (250 Ok 010001824aa79587-be724bce-48000000)
Jul 29 15:51:13 ip-172-31-5-221 postfix/qmgr[36366]: D9EE95E098: removed
[ec2-user@ip-172-31-5-221 ~]$
```

Notice that the **from** email addresses in the log are [root@ip-172-31-5-221.ec2.internal](mailto:root@ip-172-31-5-221.ec2.internal) and [ec2-user@ip-172-31-5-221.ec2.internal](mailto:ec2-user@ip-172-31-5-221.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.

From Me <liams.fitness@gmail.com> ☆  
Subject **Amazon SES Test from RHEL 8 EC2 - Postfix address rewrite**  
This message was sent using Amazon SES.  
.

From Me <liams.fitness@gmail.com> ☆  
Subject **Amazon SES Test from RHEL 8 EC2 - Postfix address rewrite**  
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, ec2-user
```

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

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

```
ec2-user: liams.fitness@gmail.com
```

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

```
$ sudo newaliases
```

```
[ec2-user@ip-172-31-5-221 ~]$  
[ec2-user@ip-172-31-5-221 ~]$ sudo head -15 /etc/aliases | tail -5  
mailer-daemon: postmaster  
postmaster: root,ec2-user  
root: liams.fitness@gmail.com  
ec2-user: liams.fitness@gmail.com  
  
[ec2-user@ip-172-31-5-221 ~]$ sudo newaliases  
[ec2-user@ip-172-31-5-221 ~]$
```

We can now configure a simple cron job, one for **root** and one for **ec2-user**, 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** & **ec2-user**. 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 RHEL 8 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 RHEL 8 EC2 via Amazon SES'
```

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

```
From Me <liams.fitness@gmail.com> ☆
Subject Cron <root@ip-172-31-5-221> echo 'This email was sent by the root user from RHEL 8 EC2 via Amazon SES'
To Me <liams.fitness@gmail.com> ☆
```

This email was sent by the root user from RHEL 8 EC2 via Amazon SES

Now I will add a simple cron job to the **ec2-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 ec2-user from RHEL 8 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 user ec2-user from RHEL 8 EC2 via Amazon SES'
```

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

```
From Me <liams.fitness@gmail.com> ☆
Subject Cron <ec2-user@ip-172-31-5-221> echo 'This email was sent by user ec2-user from RHEL 8 EC2 via Amazon SES'
To Me <liams.fitness@gmail.com> ☆
```

This email was sent by user ec2-user from RHEL 8 EC2 via Amazon SES

Both cron job notifications were successfully sent and received. Now would be a good time to disable each cron job (root & ec2-user) to prevent unnecessary notification transmissions. Do this by commenting out, or deleting, each job via **sudo crontab -e** for root & **crontab -e** for ec2-user.

I hope you've enjoyed this tutorial.

We've successfully configured Postfix as an outbound send-only email server on a RHEL 8 EC2.

Now we can send emails from our RHEL 8 EC2 instance either manually using **sendmail**, or automatically, via **cron jobs**.

I have another tutorial where I configure Postfix as a send-only email server on my **Ubuntu 20** 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](#)