### **Mandrill:**

Mandrill is a recipient-based system. You can’t store recipient information or contact lists. You’ll need to manage this data in your own application or database. You can then pass this data onto Mandrill with a SMTP message or an API request when you need a transactional email to be triggered.

MailChimp is a subscriber-based system. People subscribe to your email marketing list when they’ve give you express permission to have their address added to your marketing lists.

Creating and sending emails is pretty easy with both platforms. With Mandrill, you will be required to have API and HTML knowledge in order to send out messages.

Sender Policy Framework (SPF) is an email authentication method designed to detect forging sender addresses during the delivery of the email.SPF allows the receiving mail server to check during mail delivery that a mail claiming to come from a specific domain is submitted by an IP address authorized by that domain's administrators.

DomainKeys Identified Mail (DKIM) lets an organization take responsibility for a message that is in transit. The organization is a handler of the message, either as its originator or as an intermediary. Their reputation is the basis for evaluating whether to trust the message for further handling, such as delivery. Technically DKIM provides a method for validating a domain name identity that is associated with a message through cryptographic authentication.

DKIM attaches a new domain name identifier to a message and uses cryptographic techniques to validate authorization for its presence.

In cryptography, a message authentication code (MAC), sometimes known as a tag, is a short piece of information used to authenticate a message—in other words, to confirm that the message came from the stated sender (its authenticity) and has not been changed. The MAC value protects both a message's data integrity as well as its authenticity by allowing verifiers (who also possess the secret key) to detect any changes to the message content.

Postfix installation on EC2 Instance:

Switch to root user

sudo su

Install postfix

yum install postfix -y

Stop sendmail (installed on Amazon Linux by default)

cd /etc/init.d

sendmail stop

service sendmail stop

Start postfix

cd /etc/init.d

postfix start

service postfix status

Switch MTA from sendmail to postfix

alternatives --set <name> <path>

alternatives --set mta /usr/sbin/sendmail.postfix

Configure postfix

nano /etc/postfix/main.cf

then in the file make sure all the lines that start with mydestination are commented out, like this:

#mydestination etc.

Then add your own:

mydestination =

myhostname = [insertyourhostname].[insertyourdomainname].com

myorigin = $mydomain

relayhost = $mydomain

inet\_interfaces = loopback-only

Reload postfix

cd /etc/init.d

service postfix reload

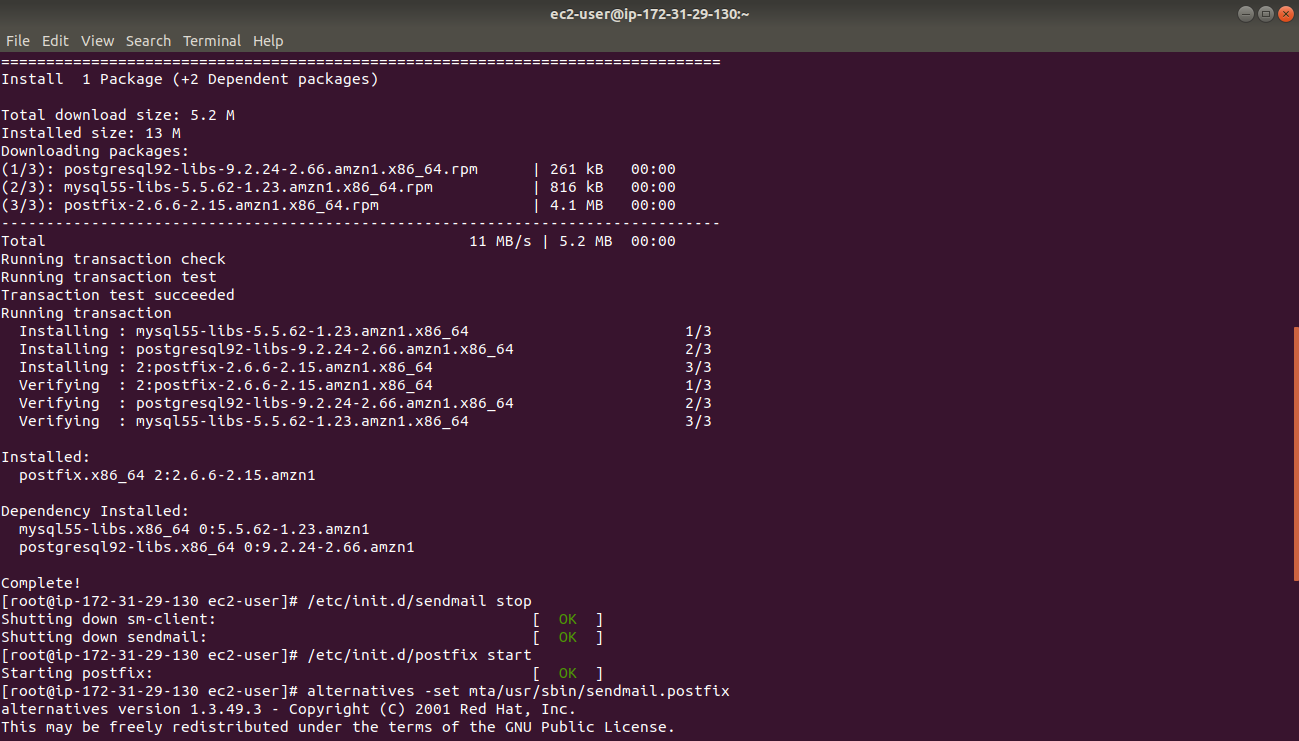
sudo yum install mailx

$ mail you@yourExistingEmailProvider

Subject: Test

This is a test email from my brand new email server!

.

~~~~

Integrating Amazon SES with Postfix:

relayhost = [email-smtp.us-east-1.amazonaws.com]:25

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

open the file /etc/postfix/master.cf. Search for -o smtp\_fallback\_relay=

If you find any line like that place # infront of that line.

In a text editor, open the goto /etc/postfix

nano sasl\_passwd

And type

email-smtp.us-east-1.amazonaws.com:587 SMTPUSERNAME:SMTPPASSWORD

Us-east-1 for North Virginia.

SMTP Username:

AKIA5JTG72JA2UKFRUVD

SMTP Password:

BNss3yZoxcqvsiHo4a1UZ//NABxR8KvugaqA1TdTBqou

For generating SMTP credentials go to ses smtp settings and create a IAM role and download credentials.

To create a hashmap database file containing your SMTP credentials:

sudo postmap hash:/etc/postfix/sasl\_passwd

change the permissions of the files so that only the root user can read or write to them:

sudo chown root:root /etc/postfix/sasl\_passwd /etc/postfix/sasl\_passwd.db

sudo chmod 0600 /etc/postfix/sasl\_passwd /etc/postfix/sasl\_passwd.db

Reload the postfix:

service postfix start

service postfix reload

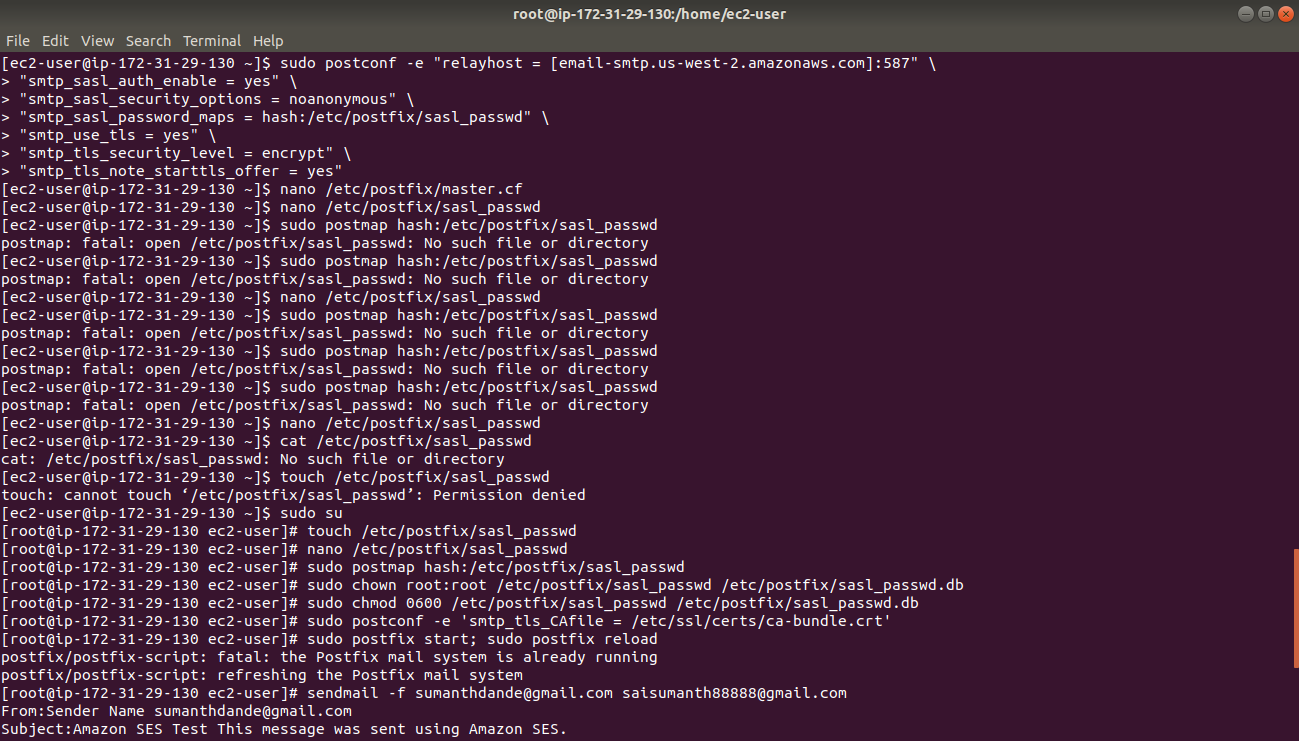
Send a test file by using following command:

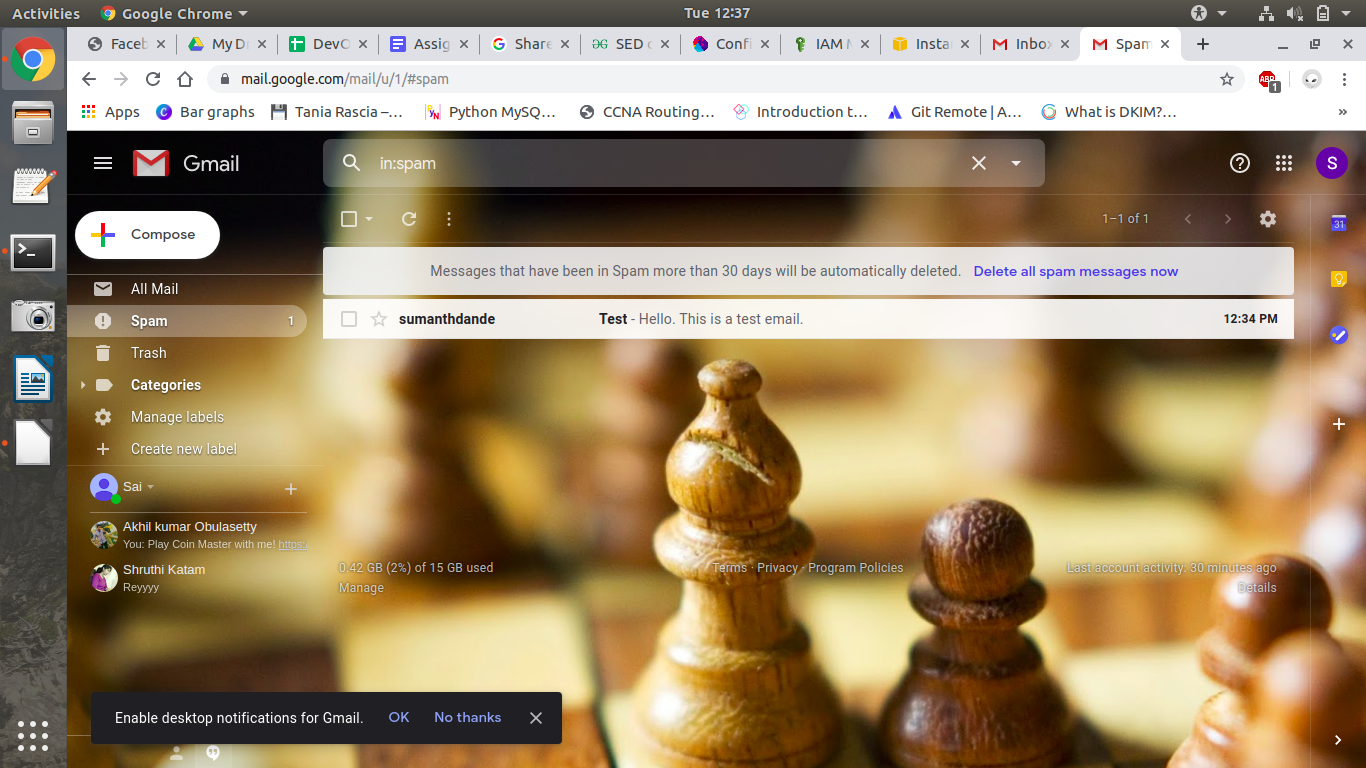
sendmail -f sender@example.com recipient@example.com

From: Sender Name <sender@example.com>

Subject: Amazon SES Test

Press enter and press dot to send email.

~~~~

~~~~

Create your own SPF:

Step 1: Gather IP addresses that are used to send email

Step 2: Make a list of your sending domains

Step 3: Create your SPF record

Example: v=spf1 mx include:\\_spf.example.com -all

v=spf1: Sets the SPF version that is used.

mx: Allows the domain’s MX details to send email.

include:\_spf.example.com: Includes example mail servers as authorized servers.

-all: Indicates that servers that are not listed previously are not authorized to send email

Step 4: Publish your SPF to DNS

Step 5: Test

How to check mx record :

MX record example:

owner-name ttl class rr pref name

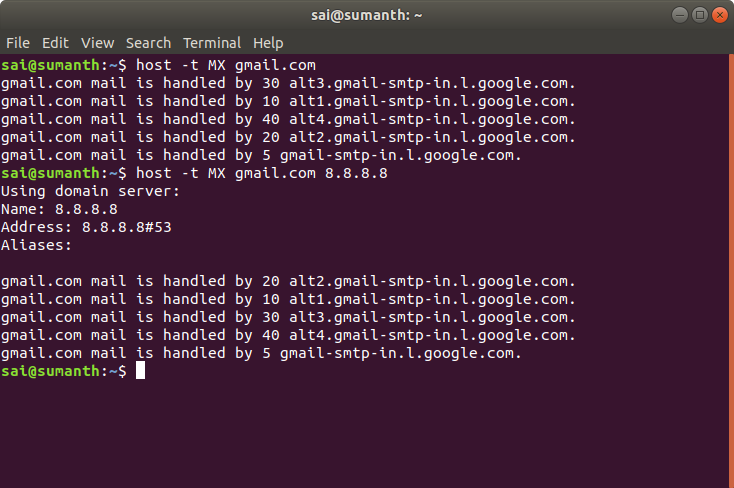
example.com. 3w IN MX 10 mail.example.com.

Back-up mail servers (higher pref values) are usually configured to simply forward mail over a prolonged period (multiple days or even weeks) to the primary mail server.

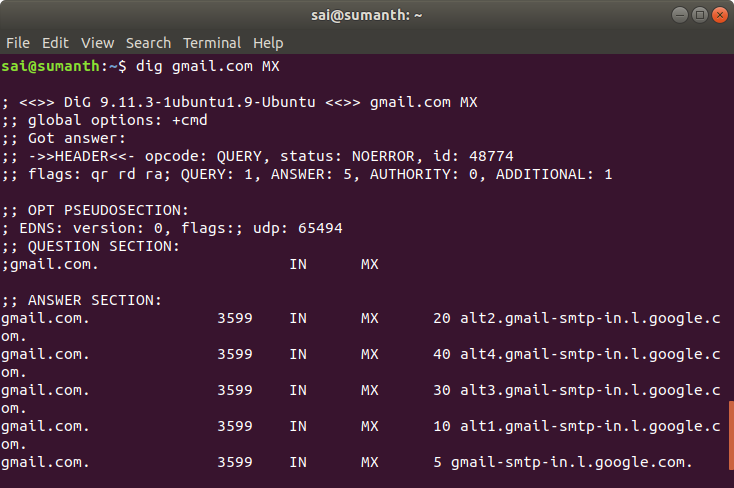
To check MX record:

host -t MX example.com

For example if we want to check the mx records of gmail   
 host -t MX gmail.com

~~~~

TO check mx record using dig command:

~~~~

DKIM:

1.Choose a DKIM selector :A DKIM selector is a string used to to point to a specific DKIM public key record in your DNS

2.Generate a public-private key pair by using ssh-keygen

3.Publish the selector and public key by creating a DKIM TXT record.

DKIM key example:

dk1024.\_domainkey.returnpath.com. 600 IN TXT "v=DKIM1\; p=MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQC1TaNgLlSyQMNWVLNLvyY/neDgaL2oqQE8T5illKqCgDtFHc8eHVAU+nlcaGmrKmDMw9dbgiGk1ocgZ56NR4ycfUHwQhvQPMUZw0cveel/8EAGoi/UyPmqfcPibytH81NFtTMAxUeM4Op8A6iHkvAMj5qLf4YRNsTkKAV

DKIM selector (s=): dk1024-2012

The domain (d=): returnpath.com

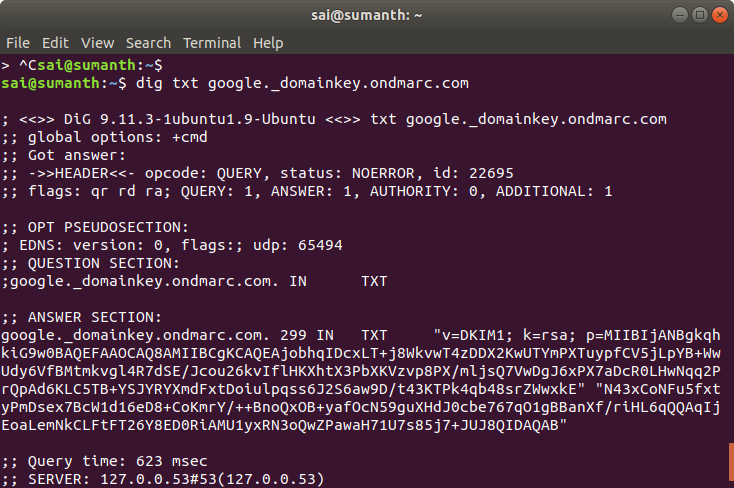
The version (v=): DKIM1

The public key (p=): MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQC1TaNgLlSyQMNWVLNLvyY/neDgaL2oqQE8T5illKqCgDtFHc8eHVAU+nlcaGmrKmDMw9dbgiGk1ocgZ56NR4ycfUHwQhvQPMUZw0cveel/8EAGoi/UyPmqfcPibytH81NFtTMAxUeM4Op8A6iHkvAMj5qLf4YRNsTkKAV

To check DKIM record:  
 dig txt selector.\_domainkey.domain

Here i checked the DKIM key of a domain using dig command:

dig txt google.\_domiankey.ondmarc.com

~~~~

In the answer section it returned DKIM key and we can see the selector and version and public key.