

## Amazon SES for CiviCRM

*Pre-requisites: DNS access*

- Have client sign up for Amazon AWS (at <https://aws.amazon.com>), which involves credit cards, identity verification, etc.
- Ideally, have them click on "SES" at the AWS console, to do the phone verification.
- Log onto AWS, select SES. Note that you only have sandbox access at this time.
- Click on "SMTP Settings" on the left, record the server info, generate SMTP credentials and record those too.
- Click on "Verified Senders: Domains". Click "Verify a new Domain". Enter the client's domain name; also click "Generate DKIM settings".
- Download these credentials, don't just copy/paste from the screen, because they tend to get cut off.
- Update the DNS with the verification credentials.
  - **NOTE: This will take several hours (3-4?), even after DNS has propagated.**
- Also add your own e-mail address to the verified sender - email address list. This just requires receiving a verification email. You need this because while you're still in sandbox mode, you can only send to verified addresses. Note that if your normal mail account has greylisting, you'll probably want to use an alternate account here, like Gmail.

You now have two options for SMTP setup:

### SMTP direct from CiviCRM

- Set up the SMTP Outbound settings in CiviCRM to use the Amazon SMTP server.
  - Without changing the core Net package, prefix the SMTP server with `ssl://` and select a port of 465.
  - If however the host doesn't allow mail to be sent over port 465, [patch the core file package/Net/SMTP.php](#) and don't use the `ssl://` prefix. For more information, [see this post and the posted links](#).
- Click "Save and Test".
- You may not see an error on the page - so check the ConfigAndLog log for error messages if necessary.

### SMTP relayed through Postfix (recommended)

See also here: <http://docs.aws.amazon.com/ses/latest/DeveloperGuide/postfix.html>

Postfix will give higher performance on large mailings, and also does a much better job of handling resends, greylisting, etc. This should be implemented wherever possible.

- Install Postfix.
- Paste this at the bottom of `/etc/postfix/main.cf`:

```
#jon@palantetech.coop SES setup
relayhost = email-smtp.us-west-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
#This last line isn't needed, I'm pretty sure
#smtp_tls_CAfile = /etc/ssl/certs/ca-bundle.crt

#jon@palantetech.coop rate limiting to 4 messages/second for now
smtp_destination_rate_delay = 1s
smtp_destination_concurrency_limit = 4
```

- Create a file with credentials at `/etc/postfix/sasl_passwd` in the format:

```
email-smtp.us-east-1.amazonaws.com:25 username:password
```

- Run `postmap /etc/postfix/sasl_passwd`.
- Configure CiviCRM's SMTP settings to either a) point to localhost, or b) use `mail()`.

## Bounce processing setup

There's code here to use a Amazon SNS to a bounce processing address instead of the typical bounce processing [here](#). I have NOT tested this, but I don't think this provides an advantage over typical bounce processing. An SNS endpoint in CiviCRM COULD provide higher performance than IMAP/POP-based bounce processing, I don't think this solution is it.

So instead, just see [Set up local bounce processing](#).

## Testing

- To test bounce processing, set up a test group made up of at least one verified address and a test contact with the e-mail address "[bounce@simulator.amazonses.com](mailto:bounce@simulator.amazonses.com)".
- Send a mailing to that group. You can speed up testing by manually running the "Send Scheduled Mailings" and "Fetch Bounces" scheduled jobs.
- Don't forget to request production access when you're done!

## Troubleshooting

In testing if CiviCRM reports that everything's been sent correctly, but then no emails are received, the problem could be with Postfix authenticating to SES. `/var/log/mail.log` might contain entries like this.

```
Aug 9 11:37:48 XXXXXXXX postfix/smtpd[24665]: connect from localhost[127.0.0.1]
Aug 9 11:37:48 XXXXXXXX postfix/smtpd[24665]: A30A4624F1: client=localhost[127.0.0.1]
Aug 9 11:37:48 XXXXXXXX postfix/cleanup[24669]: A30A4624F1: message-id=<20160809163748.A30A4624F1@XXXXXXXX.example.org>
Aug 9 11:37:48 XXXXXXXX postfix/qmgr[24664]: A30A4624F1: from=<XXXXXXXX@example.org>, size=453, nrcpt=1 (queue active)
Aug 9 11:37:48 XXXXXXXX postfix/smtpd[24665]: disconnect from localhost[127.0.0.1]
Aug 9 11:37:48 XXXXXXXX postfix/smtp[24670]: warning: SASL authentication failure: No worthy mechanisms found
Aug 9 11:37:48 XXXXXXXX postfix/smtp[24670]: A30A4624F1: SASL authentication failed; cannot authenticate to server email-smtp.us-west-2.amazonaws.com[54.68.106.242]: no mechanism available
Aug 9 11:37:48 XXXXXXXX postfix/smtp[24670]: warning: SASL authentication failure: No worthy mechanisms found
Aug 9 11:37:48 XXXXXXXX postfix/smtp[24670]: A30A4624F1: SASL authentication failed; cannot authenticate to server email-smtp.us-west-2.amazonaws.com[52.35.58.187]: no mechanism available
Aug 9 11:37:49 XXXXXXXX postfix/smtp[24670]: warning: SASL authentication failure: No worthy mechanisms found
Aug 9 11:37:49 XXXXXXXX postfix/smtp[24670]: A30A4624F1: to=<XXXXXXXX@example.com>, relay=email-smtp.us-west-2.amazonaws.com[52.35.228.26]:25, delay=0.39, delays=0.04/0.01/0.34/0, dsn=4.7.0, status=deferred (SASL authentication failed; cannot authenticate to server email-smtp.us-west-2.amazonaws.com[52.35.228.26]: no mechanism available)
```

If that's the case, you might need to install some additional libraries, namely the `libsasl2-modules` package in Debian/Ubuntu systems.

## SPECIAL NOTE FOR EC2 INSTANCES

EC2 severely throttles outgoing mail on ports 25/465/587. You need to submit a request to Amazon to have this throttle lifted. This is separate from any user-configurable firewalling! You can do that (and set up PTR) here:

<https://portal.aws.amazon.com/gp/aws/html-forms-controller/contactus/ec2-email-limit-rdns-request>