

# Configure Applications to Send Email with SES

On-campus developers are used to having applications send email via `appsmtplibmail.cornell.edu`. Currently, there is no similar Cornell service in AWS for developers to use. This page shows how to setup the AWS Simple Email Service (SES) to send email from applications.

- [Step 1. Validate Your Sender Address with SES](#)
- [Step 2. Create IAM Resources for Using SES](#)
- [Step 3. Generate the SMTP Password](#)
- [Step 4. Configure SMTP Parameters in your Application](#)
- [Step 5. Sending Email from Additional Applications](#)
- [Step 6. Rotate Access Keys](#)

## Step 1. Validate Your Sender Address with SES

This solution uses SES in limited or sandbox mode, where you validate sender (i.e., "from") email addresses one at a time. This convinces SES that you are the owner of the email address so it will send email from the address on your behalf. For this example, we assume that you own or control the [myapp-support@cornell.edu](mailto:myapp-support@cornell.edu) email address.

Verify the [myapp-support@cornell.edu](mailto:myapp-support@cornell.edu) email address following instructions in AWS documentation: <http://docs.aws.amazon.com/ses/latest/DeveloperGuide/verify-email-addresses.html>.

## Step 2. Create IAM Resources for Using SES



If you want fine-grained control of the IAM resources used for using SES, then follow the directions below. If you are impatient and just want to create SES credentials then use AWS documentation <https://docs.aws.amazon.com/ses/latest/DeveloperGuide/smtp-credentials.html> and you can skip the rest of the steps below.

In this step, you create an IAM role, group, and user to facilitate sending email using Simple Email Service using a CloudFormation Template.

1. If you are unfamiliar with using CloudFormation, take a look at AWS documentation to orient yourself: <http://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/GettingStarted.Walkthrough.html>
2. Use the provided CloudFormation template to create the CloudFormation stack that creates the required AWS resources: [iam-ses.yaml](#)
  - a. Recommended name for your stack: "myapp-iam-ses-stack"
  - b. You can override the default names for each of the resources created when providing parameter values for the template to CloudFormation.
3. The output of the stack will include the credentials (access key and secret access key) and they will be shown in the CloudFormation Console UI. While this isn't ideal, those credentials allow only the newly created IAM user to send email using SES.

## Step 3. Generate the SMTP Password

This step generates the SMTP password from the secret access key generated in the previous step. Use [iam-smtp-credentials.rb](#) Ruby script to convert the secret access key into an SES SMTP password, and optionally send a test email via SES.

You will need the following Ruby gems installed to run the script:

- openssl
- base64
- net/smtp
- optparse

```
$ ./iam-smtp-credentials.rb --help
Convert an IAM secret access key to an SES SMTP password. Optionally send a test email, if access_key and
sender_email are provided.
    --access_key ACCESS_KEY          This is the AWS API access key. Required only if
you want to send a test email after the password conversion.
    --secret_access_key SECRET_ACCESS_KEY    REQUIRED. This is the AWS API secret access key to be
converted to an SES SMTP password.
    --sender_email EMAIL_ADDRESS          A sender email address that is already validated
with SES. Required only if you want to send a test email after the password conversion.
    -h, --help                          Prints this help

$ ./iam-smtp-credentials.rb --access_key AKIQWESAAIYKVNSHZUFAKE --secret_access_key
vwj0cAD7OmDldgPki9P6#DSFAFF$%1K4vK2FZZy+6DOgrFAKE --sender_email peal@cornell.edu
SES_SMTP_PASSWORD Ag36CNaodsfdsd24QHNC4qG20/cCtUbMsdfdsfsPZdY0yYSCpkPB+N9fake
$
```

Other resources you might find helpful for this:

- <http://docs.aws.amazon.com/ses/latest/DeveloperGuide/smtp-credentials.html>
- <https://www.lisenet.com/2014/convert-iam-secret-access-key-to-ses-smtp-password-in-bash/>
- <https://gist.github.com/sysadmiral/c85c09e6534428a8bb578ab420a40207>
- <https://github.com/CU-CommunityApps/aws-examples/blob/master/aws-ruby-sdk/ses/iam-smtp-credentials.rb>

## Step 4. Configure SMTP Parameters in your Application

<b>From Address</b>	In sandbox or limited mode, all of the emails sent via SES must use one of the sender (i.e. "from") addresses previously validated with SES (in step 1).
<b>To Address</b>	any desired
<b>Host</b>	email-smtp.us-east-1.amazonaws.com (i.e., email-smtp.<AWS_REGION>.amazonaws.com)
<b>Port</b>	587
<b>Use TLS?</b>	yes
<b>User name</b>	This is the <b>access key</b> produced by the <a href="#">iam-ses.yaml</a> CloudFormation template. It will look something like "AKIAI42SV3U3D465DCGA"
<b>Password</b>	This is the SES_SMTP_PASSWORD output from the script in step 3.

## Step 5. Sending Email from Additional Applications

We recommend creating an IAM user for each application (or project) that you wish to send email via SES. The second and further IAM users can use the same IAM role and group created by the CloudFormation template we use here. It may be easiest to manually create these additional users:

- Create a new IAM user.
- Add the user to the group created by the CloudFormation template above. E.g., ses-email-senders.
- Don't configure a password for this IAM user.
- Do create access key credentials for the user.

## Step 6. Rotate Access Keys

Ideally, you want to rotate the access keys for all your IAM users periodically. This includes the keys used by these IAM users created for SES access. Here's some AWS documentation on how to do that: <https://aws.amazon.com/blogs/security/how-to-rotate-access-keys-for-iam-users/>. You will have to recompute the SMTP password from the new secret access key.