django-email-gateway Documentation

Release 0.1

django-email-gateway

Jan 17, 2018

Contents

Introduction:

django-email-gateway is used for sending mails from your verified domains and verifed emails with less cost. We can also use django-email-gateway for receive messages and deliver them to an Amazon S3 bucket in an enctypted format, call your custom code via an AWS Lambda function, or publish notifications to Amazon SNS to process the response.

Source Code is available in Micropyramid Repository(https://github.com/MicroPyramid/django-email-gateway).

Modules used:

- lxml

CHAPTER 2

Requirements

Python	>= 2.6 (or Python 3.4)
lxml	>=3.6.1

Installation Procedure

1. Install django-email-gateway using the following command:

```
pip install django-email-gateway

            (or)

git clone git://github.com/micropyramid/django-email-gateway.git

cd django-email-gateway

python setup.py install
```

2. After installing/cloning this, add the following details in settings file to send/receive emails notifications

```
# AWS details

AWS_ACCESS_KEY_ID = "Your AWS Access Key"

AWS_SECRET_ACCESS_KEY = "Your AWS Secret Key"
```

Verifying Domains

We need to verify email address and domains to confirm that we are using those domains or to prevent from others using it.

Login into AWS console if you have already AWS account or signup at https://portal.aws.amazon.com/gp/aws/developer/registration/index.html

1. For each region, we need to verify the domain seperately.

2. In aws ses/domains panel, you can view all the domains with their verification status, bounce, compaint sns topic, dkim, mail from domain details

3. By clicking on Verify a New Domain, a popup will be appeared to create new domain along with dkim settings

4. After creation of new domain, a popup will be appeared by displaying a txt, cname, mx records of a repsective domain. You need to add these reocords your domains dns server

5. After adding these details, you can check your domain verification status

6. If your domain successfully verifed, then all your emails with respective domain, sub domains will also be verified.

Verifying Email Addresses

1. To send emails from aws ses, you need to verify your from email addresses in aws ses console/email address panel

2. By clicking on Verify a New Email Address, a popup will be appeared to create new email address

3. Confirmation email has been sent to you after creating a new email. By clicking on the Arrow Icon in the table, details of Email Feedback, Delivery Notifications SNS, Bounce Notifications, Complaint Notifications SNS for the respective email will be displayed.

CHAPTER 6

Receiving Email

1. In aws ses Email Receiving, choose Rule Sets. then in content panel choose Create a Receipt Rule.

2. In the add action menu, choose s3, then you need to select/create a bucket to store all encypted receiving emails content.

3. To recieve sns notifications, add another action of sns. Here you need to create/select a topic to process the receiving mail message content.

4. Then in Rule Details page, give the rule name, create the rule.

5. In AWS-SNS services, for the topic which you have created/selected in active receipt rule set, you need to add your subscription request which may either email, AWS SQS, HTTP or HTTPS.

6. For the http or https, you need to specify the endpoint(Your application URL to receive email content), SNS will send the receiving email content to specified url.

7. For email, email-json, it will send an Amazon SES Email Receipt Notification with receiving email content.

8. For AWS Lamda, SQS, you need to specify the endpoint(lamda function name, sqs queue name).

CHAPTER 7

Usage:

1. Sending an email:

```
sending_mail(subject, email_template_name, context, from_email, to_email,␣
→verified)
```

2. Receiving an email:

```
from django_email_gateway.receiving_mail import sns_notification
subject, from_mail, to_mail, hash_code, mail_content = sns_notification(request.
→body)
```

It will process the your message content, will return the email subject, from mail, to email(abc@yourdomain.com), hashcode(abc), mail content.