

Plunk로 Amazon SES

✨ ✨ 누구보다 쉽게 사용하기

배진수

알아볼 내용들

- AWS SES에 대한 간단한 소개
- Plunk에 대한 간단한 소개와 설정 방법
- 이메일 발송에 대한 몇몇 개념 이해

Amazon SES (Simple Email Service)

- 별도의 서버 없이도 이메일을 송/수신
- API 또는 SMTP를 통해 이메일 전송
- 발송된 이메일 추적 (전송, 오픈, 클릭, 수신 거부 등)
- 사용한 만큼 요금 지불



Amazon SES (Simple Email Service)

- 샌드박스 제한 => 초당 최대 1개, 24시간동안 최대 200개의 이메일만 전송 가능
- 이메일 발신 1000건당 0.1달러, 수신 1000건당 0.1달러.

AWS SES / API로 이메일 보내기

```
import boto3

def send_email():
    client = boto3.client('ses', region_name='us-east-1')
    response = client.send_email(
        Destination={
            'ToAddresses': [
                'recipient@example.com',
            ],
        },
        Message={
            'Body': {
                'Html': {
                    'Charset': 'UTF-8',
                    'Data': '<html><body><h1>This is a test email</h1></body>
</html>',
                },
                'Text': {
                    'Charset': 'UTF-8',
                    'Data': 'This is a test email',
                },
            },
            'Subject': {
                'Charset': 'UTF-8',
                'Data': 'Test Email',
            },
        },
        Source='example@example.com',
    )
    print("Email sent! Message ID:", response['MessageId'])

send_email()
```

AWS SES / API로 이메일 보내기

AWS SDK / AWS CLI가 없다면...?

```
#!/bin/bash

# AWS 자격 증명 및 설정
AWS_ACCESS_KEY_ID="YOUR_ACCESS_KEY"
AWS_SECRET_ACCESS_KEY="YOUR_SECRET_KEY"
AWS_REGION="us-east-1"
SERVICE="ses"
HOST="email.${AWS_REGION}.amazonaws.com"
ENDPOINT="https://${HOST}"
REQUEST_PARAMETERS="Action=SendEmail&Source=your-verified-email@example.com&Destination.ToAddresses.member.1=recipient@example.com&Message.Subject.Data=Test&Message.Body.Text.Data=Hello"

# 날짜 및 시간 설정
TODAY=$(date +%Y%m%d)
NOW=$(date -u +%Y%m%dT%H%M%S)
ALGO="AWS4-HMAC-SHA256"
REQUEST_TYPE="aws4_request"
SIGNED_HEADERS="host;x-amz-date"
CANONICAL_URI="/"

# 서명 키 생성
function hmac_sha256() {
    printf "%2" | openssl dgst -binary -sha256 -hmac "$1"
}

function aws_signature_v4() {
    KEY_DATE=$(hmac_sha256 "AWS4$AWS_SECRET_ACCESS_KEY $TODAY")
    KEY_REGION=$(hmac_sha256 $KEY_DATE $AWS_REGION)
    KEY_SERVICE=$(hmac_sha256 $KEY_REGION $SERVICE)
    KEY_SIGNING=$(hmac_sha256 $KEY_SERVICE $REQUEST_TYPE)

    SIGNATURE=$(hmac_sha256 $KEY_SIGNING $STRING_TO_SIGN | xxd -p -c 256)

    echo $SIGNATURE
}

# Canonical Request 생성
CANONICAL_QUERY_STRING=$(echo -n "$REQUEST_PARAMETERS" | sed -e 's/&/\n/g' | LC_ALL=C sort | sed -e ':a;N;$!ba;s/\n/ /g')
CANONICAL_REQUEST="GET\n$CANONICAL_URI\n$CANONICAL_QUERY_STRING\nhost:$HOST\nx-amz-date:$NOW\n\n$SIGNED_HEADERS\n\n$(echo -n '' | openssl dgst -sha256 | sed 's/^.* //')")

# String to Sign 생성
STRING_TO_SIGN="$ALGO\n$NOW\n$TODAY/$AWS_REGION/$SERVICE/$REQUEST_TYPE\n$(echo -n "$CANONICAL_REQUEST" | openssl dgst -sha256 | sed 's/^.* //')")

# 서명 생성
SIGNATURE=$(aws_signature_v4)

# Signed request 전송
curl -s -X GET "${ENDPOINT}/${CANONICAL_QUERY_STRING}" \
-H "Host: $HOST" \
-H "x-amz-date: $NOW" \
-H "Authorization: $ALGO Credential=$AWS_ACCESS_KEY_ID/$TODAY/$AWS_REGION/$SERVICE/$REQUEST_TYPE, SignedHeaders=$SIGNED_HEADERS, Signature=$SIGNATURE"
```

Plunk

- AWS SES 위에서 동작하는 이메일 플랫폼
- 오픈 소스 SaaS (오픈소스와 SaaS간 기능 차이 없음)
- 워크플로(자동화) 기능 제공
- 이메일 템플릿 에디터 제공



<https://www.useplunk.com>

Plunk / 설정

1. SNS 토픽 생성
2. SES 구성 세트(Configuration Set) 생성
3. IAM User 및 Credential 생성
4. 배포
5. SNS 토픽 구독

<https://docs.useplunk.com/getting-started/self-hosting>

Plunk / 설정 / SNS 토픽 생성

[Amazon SNS](#) > [Topics](#) > Create topic

Create topic

Details

Type [Info](#)
Topic type cannot be modified after topic is created

☐ FIFO (first-in, first-out)

- Strictly-preserved message ordering
- Exactly-once message delivery
- High throughput, up to 300 publishes/second
- Subscription protocols: SQS

☒ Standard

- Best-effort message ordering
- At-least once message delivery
- Highest throughput in publishes/second
- Subscription protocols: SQS, Lambda, HTTP, SMS, email, mobile application endpoints

Name

plunk-topic

Maximum 256 characters. Can include alphanumeric characters, hyphens (-) and underscores (_).

Display name - *optional* [Info](#)
To use this topic with SMS subscriptions, enter a display name. Only the first 10 characters are displayed in an SMS message.

My Topic

Maximum 100 characters.

Plunk / 설정 / SES 구성 세트 생성

[Amazon SES](#) > [Configuration: Configuration sets](#) > Create set

Create set

A *configuration set* is a group of rules you can apply to the messages you send through Amazon SES.

General details

Configuration set name

64 character limit. Only letters, numbers, underscores, and dashes are permitted.

Sending IP pool [Info](#)

When you send email using this configuration (config) set, messages will be sent from the dedicated IPs in the assigned pool.

default ▼

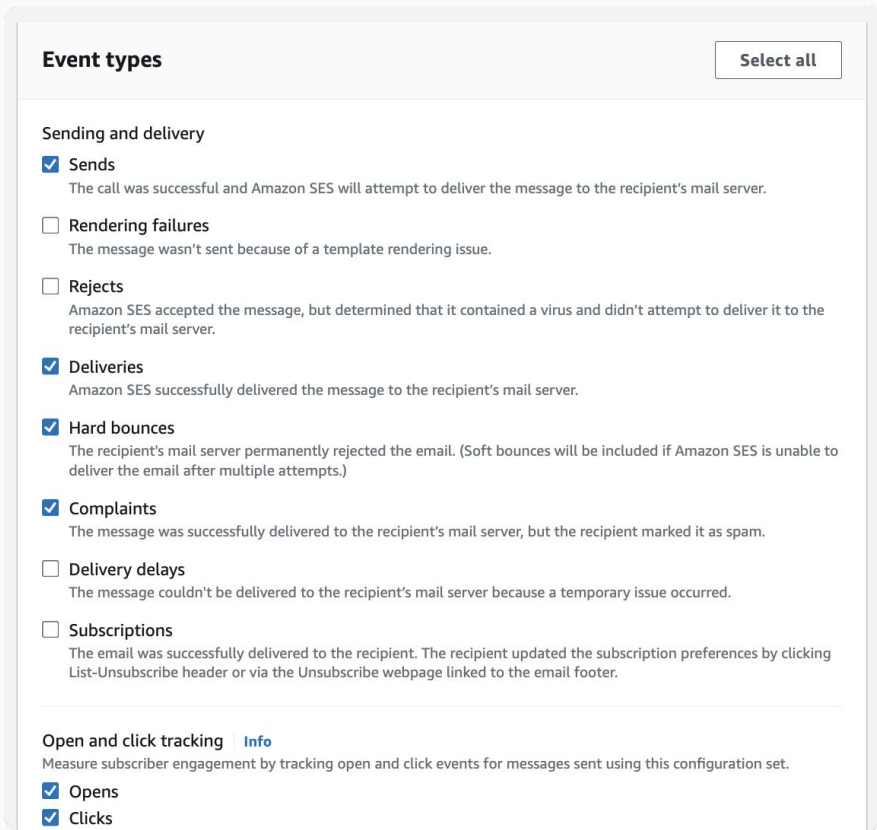
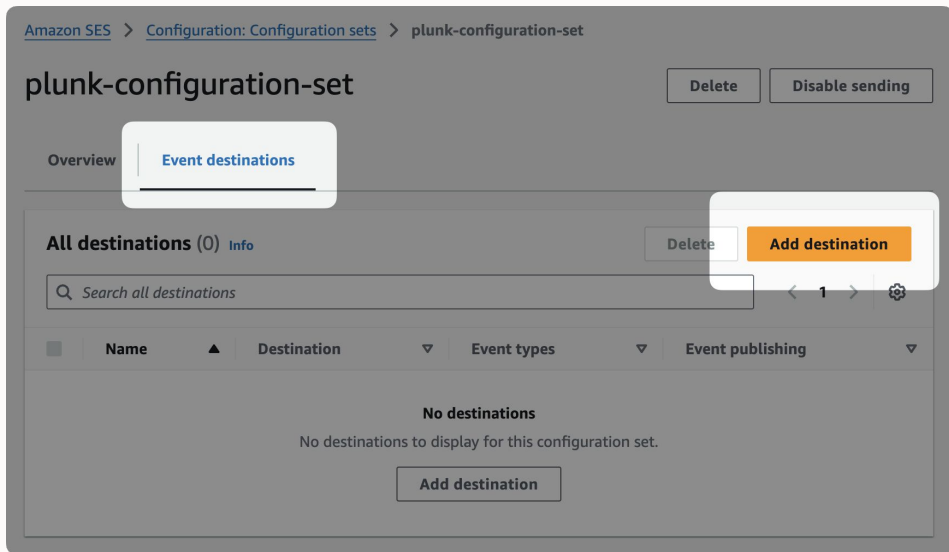
Tracking options [Info](#)

Use your own redirect domain to handle open and click tracking for this configuration set. To enable tracking, you must add an event destination.

☐ Use a custom redirect domain

▶ Advanced delivery options

Plunk / 설정 / SES 구성 세트 생성



Plunk / 설정 / SES 구성 세트 생성

Specify destination

An event destination is an AWS service to which email sending events can be published. Choosing the appropriate destination depends on the level of detail you want to capture and how you want to receive the data.

Destination options

Destination type [Info](#)

☐ Amazon CloudWatch

☐ Amazon Kinesis Data Firehose

☐ Amazon Pinpoint

☒ Amazon SNS

☐ Amazon EventBridge

Name

Name can include letters (A-Z), numbers (0-9), dashes (-), and hyphens (—). No spaces.

Event publishing

Amazon SES will publish the selected events to this event destination.

☒ Enabled

Amazon Simple Notification Service (SNS) topic

In Amazon SNS, a topic is a logical access point that acts as a communication channel.

SNS topic

Amazon SES will notify the selected topic when a message produces any of the chosen event types.

[Cancel](#) [Previous](#) [Next](#)

Plunk / 설정 / IAM User 및 Credentials 생성

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "VisualEditor0",
      "Effect": "Allow",
      "Action": [
        "ses:SetIdentityMailFromDomain",
        "ses:GetIdentityDkimAttributes",
        "ses:SendRawEmail",
        "ses:GetIdentityVerificationAttributes",
        "ses:VerifyDomainDkim",
        "ses:ListIdentities",
        "ses:SetIdentityFeedbackForwardingEnabled"
      ],
      "Resource": "*"
    }
  ]
}
```

Plunk / 설정 / 배포

Docker Image)

<https://hub.docker.com/r/driaug/plunk>

- EC2
- ECS
- Docker Image + ElastiCache + RDS
- ...

```
version: '3'
services:
  plunk:
    image: "driaug/plunk"
    ports:
      - "3000:3000"
    depends_on:
      db:
        condition: service_healthy
      redis:
        condition: service_started
    environment:
      REDIS_URL: '${REDIS_URL:-redis://redis:6379}'
      DATABASE_URL: '${DATABASE_URL:-postgresql://postgres:postgres@db:5432/postgres}'
      JWT_SECRET: '${JWT_SECRET}'
      AWS_REGION: '${AWS_REGION}'
      AWS_ACCESS_KEY_ID: '${AWS_ACCESS_KEY_ID}'
      AWS_SECRET_ACCESS_KEY: '${AWS_SECRET_ACCESS_KEY}'
      AWS_SES_CONFIGURATION_SET: '${AWS_SES_CONFIGURATION_SET}'
      NEXT_PUBLIC_API_URI: '${API_URI}'
      API_URI: '${API_URI}'
      APP_URI: '${APP_URI}'
      DISABLE_SIGNUPS: 'False'
    entrypoint: [ "/app/entry.sh" ]
  db:
    image: postgres
    environment:
      POSTGRES_PASSWORD: postgres
      POSTGRES_USER: postgres
      POSTGRES_DB: postgres
    volumes:
      - postgres_data:/var/lib/postgresql/data
    healthcheck:
      test: [ "CMD-SHELL", "pg_isready -U postgres -d postgres" ]
      interval: 10s
      retries: 5
      timeout: 10s
  redis:
    image: redis

volumes:
  postgres_data:
```

Plunk / 설정 / SNS 토픽 구독

[Amazon SNS](#) > [Subscriptions](#) > Create subscription

Create subscription

Details

Topic ARN

arn:aws:sns:ap-northeast-2:358100353008:plunk-topic

Protocol

The type of endpoint to subscribe


HTTPS

Endpoint

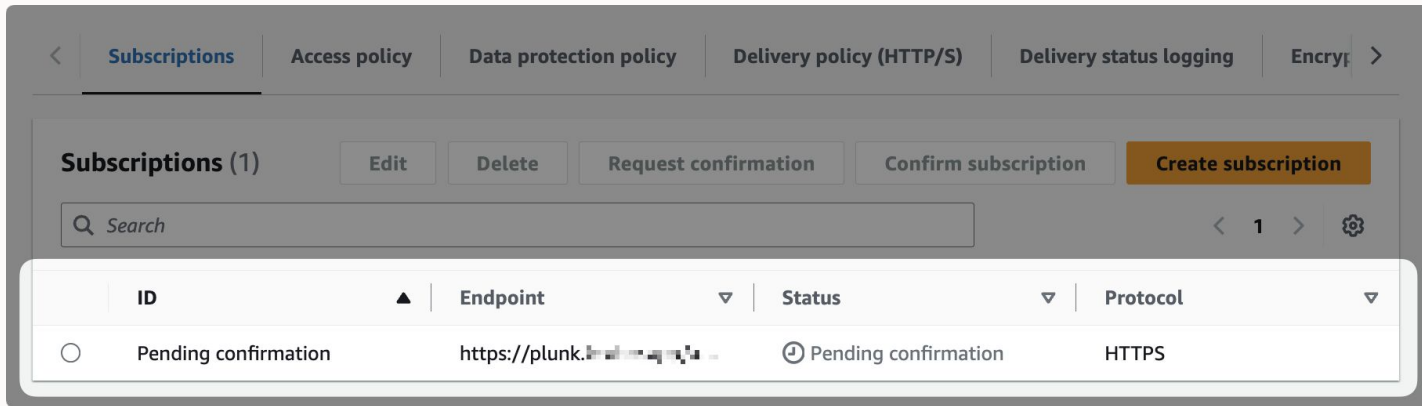
A web server that can receive notifications from Amazon SNS.

https://plunk.example.com/api/webhooks/incoming/sns

☐ Enable raw message delivery

 After your subscription is created, you must confirm it. [Info](#)

Plunk / 설정 / SNS 토픽 구독



```
i info Running scheduled tasks
i info Updating verified identities
::ffff:127.0.0.1 - POST /tasks HTTP/1.1 200 - - 4.412 ms
::ffff:127.0.0.1 - POST /identities/update HTTP/1.1 200 - - 52.573 ms
i info -----
i info SNS Topic Confirmation URL:
i info https://sns.ap-northeast-2.amazonaws.com/?Action=ConfirmSubscription&TopicArn=arn:aws:sns:ap-northeast-2:35810035300
8:plunk-topic&Token=2336412f37fb687f5d51e6e2425ba1f30b17252899b2be880ec79e5648f046f1334d065943ec3af9008321de1d57fb11166d86e5c0d49
b05e7c9e5242e55c6a9eb6bb08c80b7bf1521ca2cae39f4ae56151eb9973c0bf8f355113a103d91803d46d672450f6ab62143fe358006160f78912cd991353c95
15ac3a8424738a48f7
i info -----
::ffff:127.0.0.1 - POST /webhooks/incoming/sns HTTP/1.1 200 - - 0.560 ms
i info Running scheduled tasks
i info Updating verified identities
::ffff:127.0.0.1 - POST /tasks HTTP/1.1 200 - - 2.648 ms
```


Plunk / 기능



```
curl --request POST \  
  --url https://plunk.example.app/api/v1/send \  
  \ --header 'Authorization: Bearer <TOKEN>' \  
  --header 'Content-Type: application/json' \  
  --data '{  
    "to": "to@example.com",  
    "subject": "Hello!",  
    "body": "Your first Email.",  
    "from": "from@example.com"  
  }'
```

Plunk / 기능

Create a new template

Reusable blueprints of your emails

Subject

안녕하세요!

Type ⓘ



Marketing <>

Plunk Editor

HTML

Email Body

≡ ≡ ≡



안녕하세요!

테스트 이메일입니다.

Cancel

+ Create

Create a new action

Name

Onboarding

Run on triggers

1 selected <>

Exclude contacts with triggers

0 selected <>

Template

Onboarding email <>

Delay before sending

1

Days <>

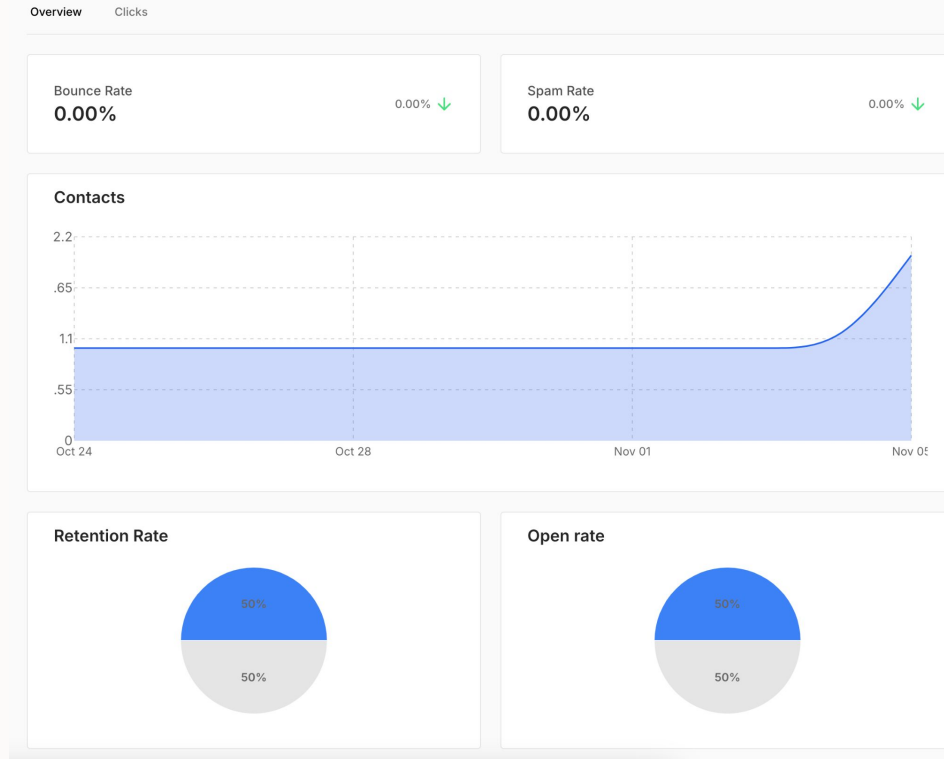
Run once

Toggle this on if you want to run this action only once per contact. ☐

Cancel

+ Create

Plunk / 기능



Plunk / 기능 / 도메인 검증

[Project Settings](#)[API Keys](#)[Verified Domain](#)[Members](#)











Domain

By sending emails from your own domain you build up domain authority and trust.

[Unlink domain](#)

Waiting for DNS verification

Please add the following records to naru200.com to verify hello@naru200.com, this may take up to 15 minutes to register. In the meantime you can already start sending emails, we will automatically switch to your domain once it is verified.

Type	Key	Value
TXT	plunk 	v=spf1 include:amazonses.com ~all 
MX	plunk 	10 feedback-smtp..amazonses.com 
CNAME	qihpzqm3licfzjqnwocx7mt53q6znz4h._domainkey 	qihpzqm3licfzjqnwocx7mt53q6znz4h.dkim.amazonses.com 
CNAME	cbmcjvkmp7zu4zm3tdp3fi3eqv5pfi4k._domainkey 	cbmcjvkmp7zu4zm3tdp3fi3eqv5pfi4k.dkim.amazonses.com 
CNAME	ys24smuwytouzdt2ca2hf2uc24d3dcy5._domainkey 	ys24smuwytouzdt2ca2hf2uc24d3dcy5.dkim.amazonses.com 

Plunk / 기능 / 도메인 검증

- DKIM ⇒ 이메일이 중간에 변조되지 않았는지 확인하는 것.
- SPF ⇒ 특정 서버가 특정 도메인을 사용하여 이메일을 보낼 수 있는지 확인하는 것.
- DMARC => 이메일이 DKIM 또는 SPF 검증에 실패했을 시의 정책을 정의하는 것.
(none: 허용 / reject: 차단 / quarantine: 스팸 처리)

DMARC(Domain-based Message Authentication, Reporting and Conformance)

Route53에 DNS 레코드 게시

DMARC는 이메일 서버가 인증 검사에 실패한 메시지를 처리하는 방법을 지정합니다.

▼ DNS 레코드 게시

아래 제공된 값(을)을 편집하여 도메인에 적용하려는 DMARC 정책을 수정할 수 있습니다. [자세히 알아보기](#)

유형	이름	값
TXT	_dmarc.brulee.app	"v=DMARC1; p=none;"

Plunk / 기능 / 도메인 검증

Welcome to the "Learn and Test DMARC" console! Here, you'll get a visual breakdown of how email servers communicate, giving you a better understanding of SPF, DKIM, and DMARC and how they work together.

Send an email to this address to get started:

ld-7278116799@learndmarc.com

Other options:

- > [Spoof my email](#)
- > [Load random example](#)
- > [Test your DMARC knowledge](#)
- > [Paste email headers](#)

Don't worry, we only use your email to check your DMARC setup. We don't store any of your data.

Waiting for incoming email... /█

<https://www.learndmarc.com>