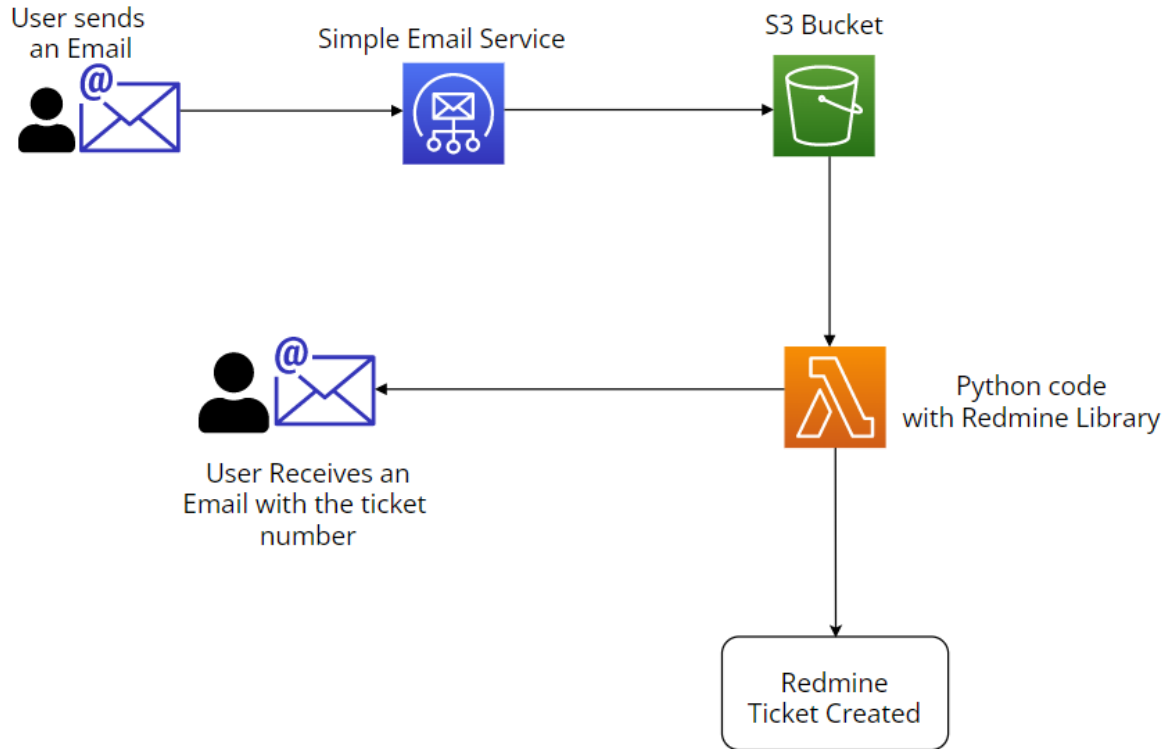


Automate Redmine Ticket creation from Email

Architecture:



In the SES console,

- Create an SES verified Domain. Make sure you added cname and MX records.
- In the Amazon SES console, in the navigation pane, Click Email Receiving.
- Choose Rule Sets and create a Receipt Rule.
- On the Recipients page, enter the email address and choose Next Step.
- For Add action, Choose to deliver to the s3 bucket and choose the *name of the bucket*.
- Click Next Step, on the review page, check the details, and click Create Rule.

Your new serverless project should contain a `serverless.yml` file. This file defines what will be deployed to AWS: functions, events, resources, and more.

In my `serverless.yml` file

- I am creating an IAM role to allow log groups to be created, sendmail permissions to the SES resource and s3 all permissions.
- Along with that I am creating an s3 bucket with no public access and attaching a bucket policy.
- Creating a lambda function and adding an event `s3:ObjectCreated` to trigger my function. Adding Redmine library as a layer to the lambda function.

The `handler.py` file contains my lambda script.

In my `handler.py` file,

- First we are taking the last modified Object in the Bucket.
- Next we are parsing the required data into variables.
- Using `redmine.issue.create` function we are creating an issue with the required fields.
- After this we are sending the created ticket details back to the sender as a response.

Commands for deploying the project using AWS-google-auth:

- `pip install aws-google-auth`
- `aws-google-auth -u <BIZ EMAIL> -I C03k51459 -S 736275908251 -R <AWS_REGION> -d 3600 -p <ANY_PROFILE_NAME> --bg-response js_enabled`
- `setx AWS_PROFILE sahith`
- `setx AWS_SDK_LOAD_CONFIG 1`
- Go the directory of the serverless project
- `Serverless deploy`

Github Link :

[SahithPalika/Redmine-email at redmine-email · SahithPalika/SahithPalika \(github.com\)](https://github.com/SahithPalika/Redmine-email)