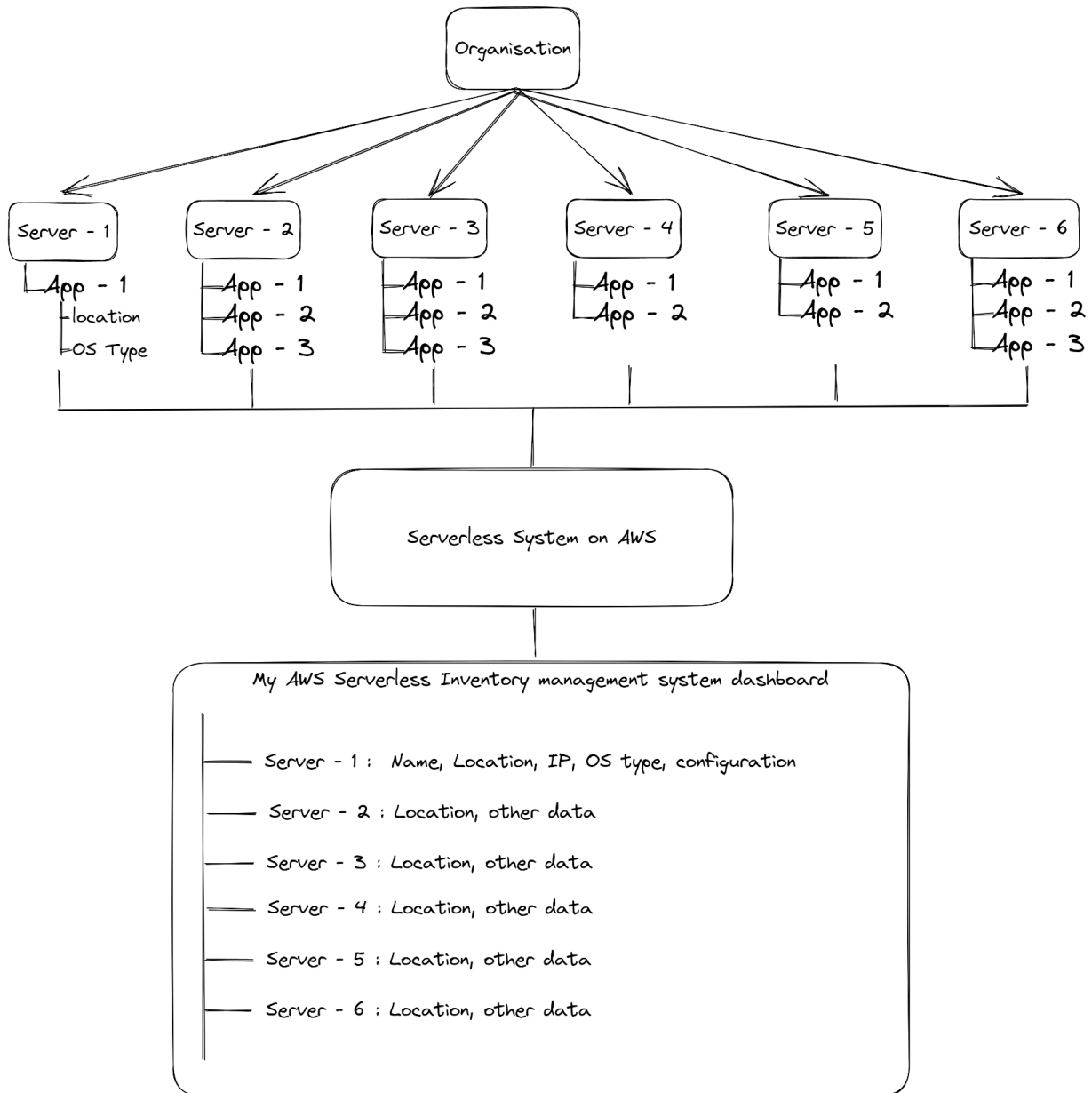# ServerGenie

Dashboard for your servers and application

Introduction

ServerGenie - Dashboard for servers and application is a powerful dashboard tool designed to monitor and manage servers and applications with ease. Built using Flask, AWS API Gateway, AWS Lambda, and DynamoDB, it offers a seamless and efficient experience for administrators, developers, and IT professionals alike.

With its intuitive interface, your webapp provides users with comprehensive visibility into the performance, health, and status of their servers and applications. Whether it's monitoring resource usage, checking for errors, or analyzing system metrics, your webapp offers a range of powerful features that make server and application management a breeze.

Application Flow

```
                          ┌──────────────┐
                          │ Organisation │
                          └──────────────┘

┌────────────┐ ┌────────────┐ ┌────────────┐ ┌────────────┐ ┌────────────┐ ┌────────────┐
│ Server - 1 │ │ Server - 2 │ │ Server - 3 │ │ Server - 4 │ │ Server - 5 │ │ Server - 6 │
└────────────┘ └────────────┘ └────────────┘ └────────────┘ └────────────┘ └────────────┘
  ├App - 1       ├App - 1       ├App - 1       ├App - 1       ├App - 1       ├App - 1
  ├location      ├App - 2       ├App - 2       └App - 2       └App - 2       ├App - 2
  └OS Type       └App - 3       └App - 3                                     └App - 3
```

Serverless System on AWS

My AWS Serverless Inventory management system dashboard

Server - 1 : Name, Location, IP, OS type, configuration

Server - 2 : Location, other data

Server - 3 : Location, other data

Server - 4 : Location, other data

Server - 5 : Location, other data

Server - 6 : Location, other data

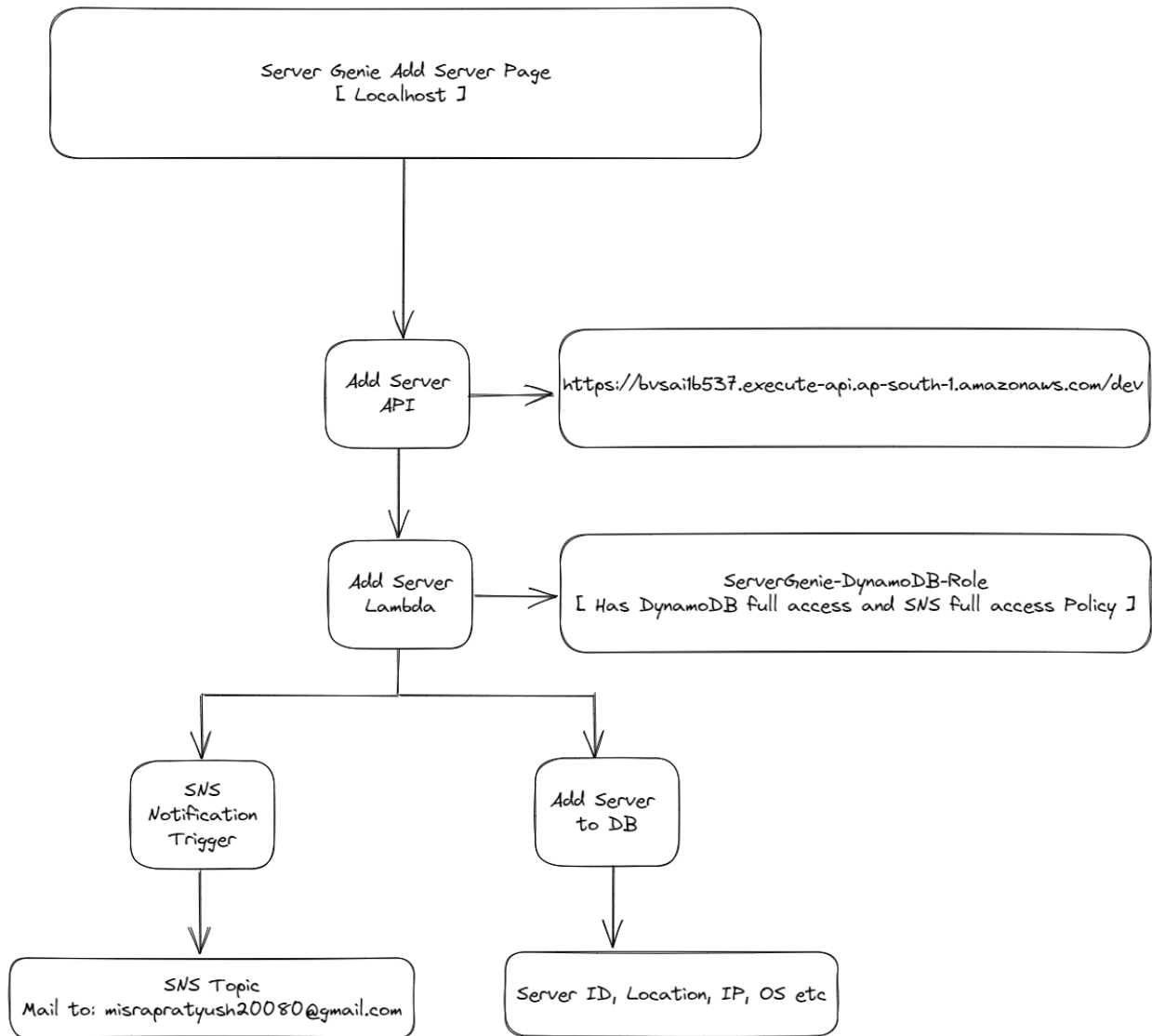⟶ New server add => CW Eventbridge event, SNS notification

Tools and technologies used

- Flask [2.2.3]: Flask is a lightweight web framework for Python that provides developers with tools and libraries to build web applications quickly and efficiently.
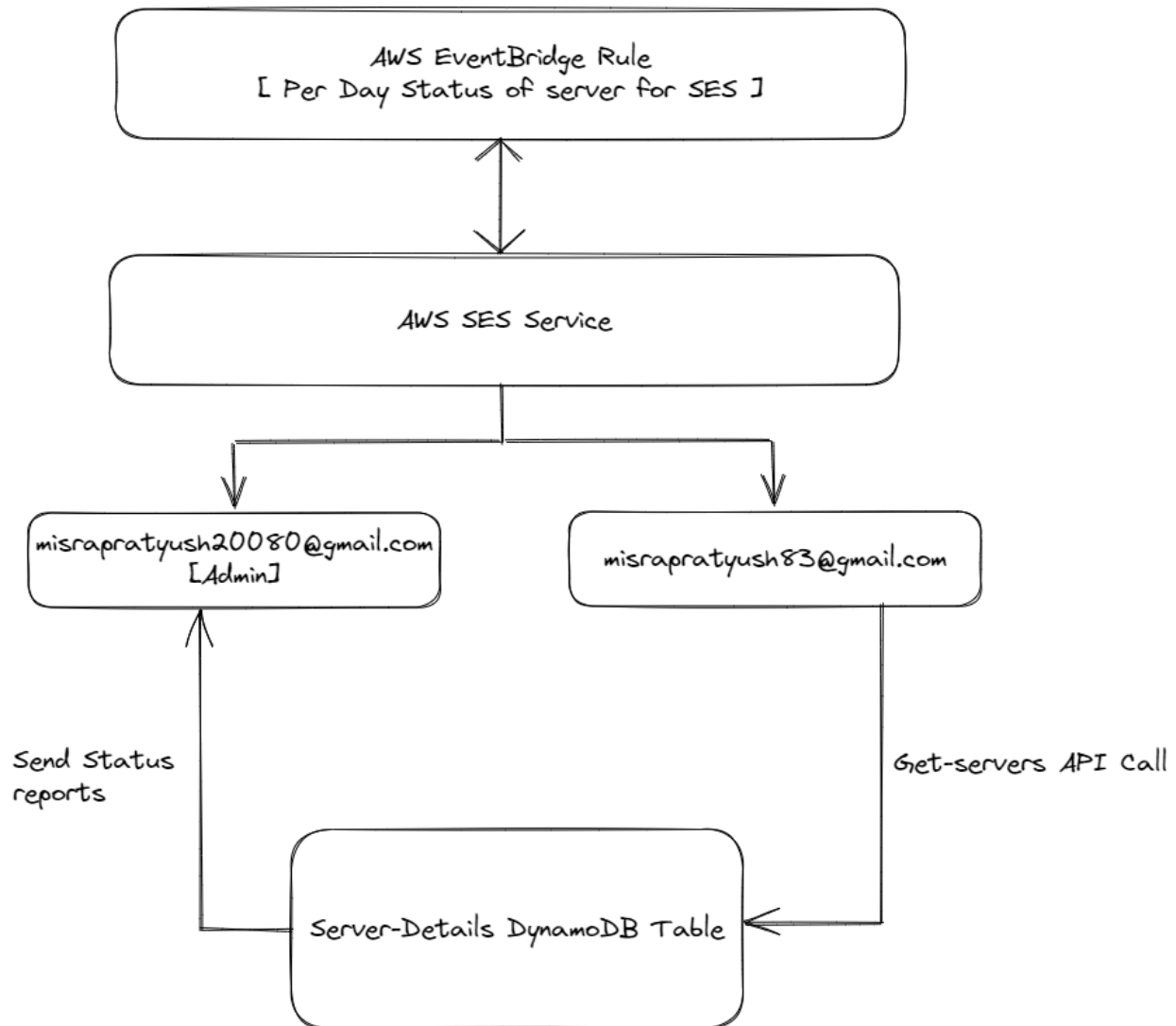
- AWS API Gateway: Amazon API Gateway is a fully managed service that makes it easy for developers to create, publish, and manage APIs at any scale.
- AWS Lambda: AWS Lambda is a serverless compute service that allows you to run code without provisioning or managing servers.
- DynamoDB: DynamoDB is a fully managed NoSQL database service that provides fast and predictable performance with seamless scalability.
- Jinja2: Jinja2 is a powerful and flexible templating engine for Python that allows developers to create dynamic HTML templates for web applications.
- Boto3 [3.1.2]: Boto3 is the Amazon Web Services (AWS) SDK for Python, which allows Python developers to write software that makes use of services like Amazon S3 and Amazon EC2.

Workflow
1) Add Server Workflow

```
┌─────────────────────────────────────────────┐
│          Server Genie Add Server Page         │
│               [ Localhost ]                   │
└─────────────────────────────────────────────┘
                      │
                      ▼
          ┌──────────────┐      ┌──────────────────────────────────────────────────────┐
          │  Add Server  │─────▶│ https://bvsai1b537.execute-api.ap-south-1.amazonaws.com/dev │
          │     API      │      └──────────────────────────────────────────────────────┘
          └──────────────┘
                      │
                      ▼
          ┌──────────────┐      ┌──────────────────────────────────────────────────────┐
          │  Add Server  │─────▶│              ServerGenie-DynamoDB-Role                 │
          │    Lambda    │      │ [ Has DynamoDB full access and SNS full access Policy ] │
          └──────────────┘      └──────────────────────────────────────────────────────┘
                │
        ┌───────┴────────┐
        ▼                ▼
  ┌──────────┐    ┌──────────┐
  │   SNS    │    │Add Server│
  │Notification│  │  to DB   │
  │ Trigger  │    └──────────┘
  └──────────┘         │
        │              ▼
        ▼      ┌──────────────────────────────┐
┌──────────────────────────┐  │ Server ID, Location, IP, OS etc │
│        SNS Topic          │  └──────────────────────────────┘
│ Mail to: misrapratyush20080@gmail.com │
└──────────────────────────┘
```

2) SES Service

```
┌─────────────────────────────────────────┐
│           AWS EventBridge Rule           │
│     [ Per Day Status of server for SES ] │
└─────────────────────────────────────────┘
                    ↕
┌─────────────────────────────────────────┐
│             AWS SES Service              │
└─────────────────────────────────────────┘
         ↓                      ↓
┌──────────────────────┐  ┌──────────────────────┐
│ misrapratyush20080   │  │ misrapratyush83      │
│ @gmail.com           │  │ @gmail.com           │
│ [Admin]              │  │                      │
└──────────────────────┘  └──────────────────────┘

Send Status                        Get-servers API Call
reports
            ┌────────────────────────────┐
            │ Server-Details DynamoDB Table │
            └────────────────────────────┘
```

3) API Keys

Api Keys have been used to provide authentication against Direct API Calls. This will provide a layer of protection against unwanted manipulations with the Database.

4) IAM Roles

Separate roles have been made for Lambda to provide them access with the AWS Service they are interacting with. Major services are:

- DynamoDB
- Simple Email Service (SES)
- SNS (Simple Notification Service)

5) Tables

There are 2 tables in the ServerGenie Application:

- Server Details: This contains details regarding the servers and their configurations. Primary Key [partition key]: server-id
- Application Details: This contains details regarding the applications and their configurations.
- Primary Key:
  - Partition Key: Application-id
  - Sort Key: server-id