**Project into manageable steps and tasks:**

***(Daily motivational quotes***

***serverless Email Automation)***

**## Steps to Create and Deploy the Serverless Project**

**### 1. Save `quotes.json` to the S3 Bucket**

1. Navigate to your S3 bucket in the AWS Management Console.

2. Upload the `quotes.json` file to the desired bucket.

**### 2. Create a Serverless Project**

1. \*\*Install Serverless Framework\*\* (if not already installed):

```sh

npm install -g serverless

```

2. \*\*Create a new Serverless project\*\*:

```sh

serverless

```

- Choose `python starter api` or `python flask with dynamo`.

- Select `Python 3.12`.

- Open the project in VS Code:

```sh

code .

```

**### 3. Set Up Handlers**

1. \*\*Create a `handler` folder\*\* inside the project.

2. \*\*Create the following Python files in the `handler` folder\*\*:

**#### `getQuotes.py`**

- Fetches the JSON (quotes) from the `quotes.json` file stored in S3 and renders it via API Gateway.

**#### `subscribeUser.py`**

- API for adding subscribers/members to the Daily Quotes service.

- Saves the data to a DynamoDB table (`user-table`).

**#### `getSubscribers.py`**

- API to get the member details added to the service by `subscribeUser` API.

**### 4. Modify `serverless.yml`**

- Define the functions for each of the above handlers.

- Example configuration for `getQuotes` function:

```yaml

functions:

getQuotes:

handler: handler/getQuotes.handler

events:

- http:

path: quotes

method: get

```

**### 5. Deploy the Project**

1. \*\*Run the deployment command\*\*:

```sh

serverless deploy --verbose

```

2. \*\*Verify the deployment\*\*:

- Check the endpoints:

- GET: `https://5euy86sr2l.execute-api.us-east-1.amazonaws.com/dev/quotes`

- POST: `https://5euy86sr2l.execute-api.us-east-1.amazonaws.com/dev/subscribe`

- GET: `https://5euy86sr2l.execute-api.us-east-1.amazonaws.com/dev/subscribers`

**### 6. Create `staticMailer.py`**

- \*\*Add a new handler for sending email notifications using AWS SNS\*\*.

- Modify `serverless.yml` to include the new function.

- Deploy and test the new function:

```sh

serverless deploy --verbose

```

**### 7. Configure SNS**

- Confirm the owner's email to receive notifications.

- Update `serverless.yml` with the ARN.

- Redeploy the project:

```sh

serverless deploy --verbose

```

**### 8. Create `sendEmail.py`**

- \*\*Send emails via SendGrid platform\*\*.

- Create a SendGrid account and generate an API key.

- Store the API key in `serverless.yml` as an environment variable.

- Add layers to Lambda:

- Create `lambda-layer` folder.

- Install dependencies:

```sh

pip install requests -t python/lib/python3.12/site-packages/

pip install sendgrid -t python/lib/python3.12/site-packages/

```

- Zip and upload the layer to AWS Lambda.

- Update `serverless.yml` with the layer ARN.

- Deploy the project:

```sh

serverless deploy --verbose

```

**### 9. Schedule Email Notifications**

- Add a cron job operation to `serverless.yml` for scheduled emails.

- Deploy the project and verify scheduled tasks in AWS CloudWatch.

**### 10. Create Frontend Using Next.js**

1. \*\*Set up the project\*\*:

```sh

mkdir quotes-site

cd quotes-site

npm install axios

npm install next@14.2.5 react@18.2.0 react-dom@18.2.0 --force

npx create-next-app quotes-site --use-npm --example "https://github.com/vercel/next-learn/tree/1011f2f19fc821f09d2ec685e42229d0826fca71/basics/learn-starter"

```

2. \*\*Run the development server\*\*:

```sh

npm run dev

```

- The frontend should be available at [http://localhost:3000](http://localhost:3000).