INTERNSHIP REPORT

1. INTRODUCTION

1.1 Problem Statement

The main objective is to create a REST API application with GoLang which will do CRUD operations around any cloud resource

1.2 Motivation

With the lightning speed of technological evolution, the cloud computing revolution is hitting full gear. There has been a drastic shift from Desktop to cloud in the last few decades. It has changed every enterprise and industry. Cloud Computing is the on-demand delivery of host services such as database, storage, applications or other IT resources over Internet.

2. PROJECT DETAILS

2.1 Resource:

- **Amazon Simple Notification Service** (Amazon SNS)- It is a web service provided by Amazon that harmonizes and governs the delivery of messages to the subscribing endpoints.

- **Swagger-** Swagger is the largest framework for designing APIs using a common language and enabling the development across the whole API lifecycle, including documentation, design, testing, and deployment. I have used swagger for API documentation.

- **Advanced Rest Client-** ARC is the API testing tool. API testing is done using the API endpoints and validation is performed based on the return value. It supports all the methods of HTTP, like- [GET, PUT, POST, DELETE].

2.2 Language:

Go Language

2.3 Description:

One2All is an application which uses Amazon SNS web service to send messages to subscribing clients. There are two types of clients – Publishers and Subscribers. Publishers act as producers i.e., they asynchronous communicate with subscribers by publishing/sending messages to the topic. Topic is a logical access point and communication channel. On the other hand, Subscribers are referred to Consumers i.e., they receive the messages over some standard supported protocols when they subscribe to the topic.

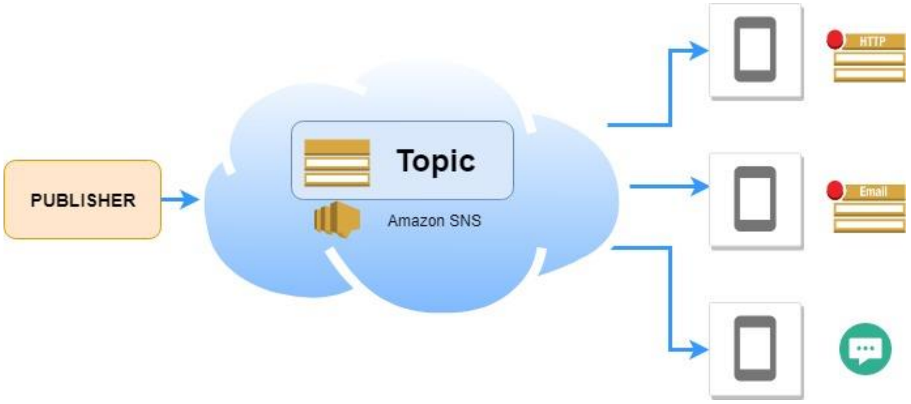


Figure 2.1 Basic explanation of supported protocols via message can be sent

There are three methods in which publisher can communicate with the subscriber i.e., HTTP/s, Email or SMS. The publisher sends message to a topic they have created or have access to publish to. Then, SNS matches the topic with the set of subscribes who have subscribed to that specific topic and delivers the message respectively.

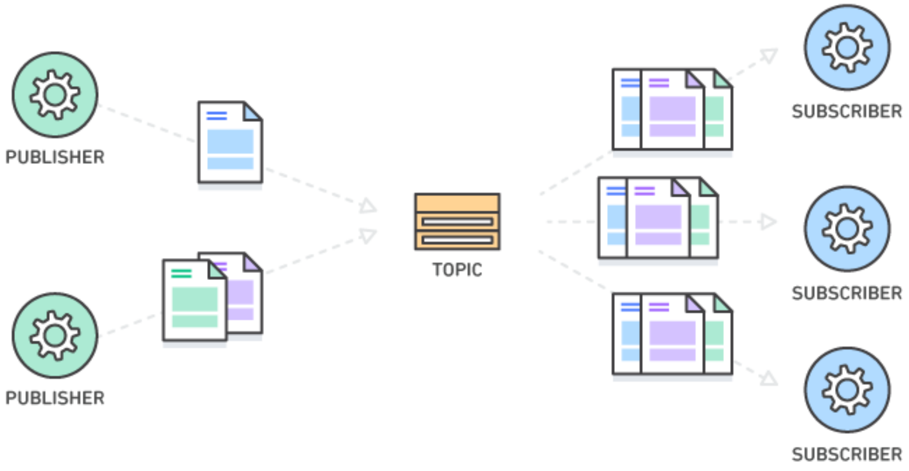
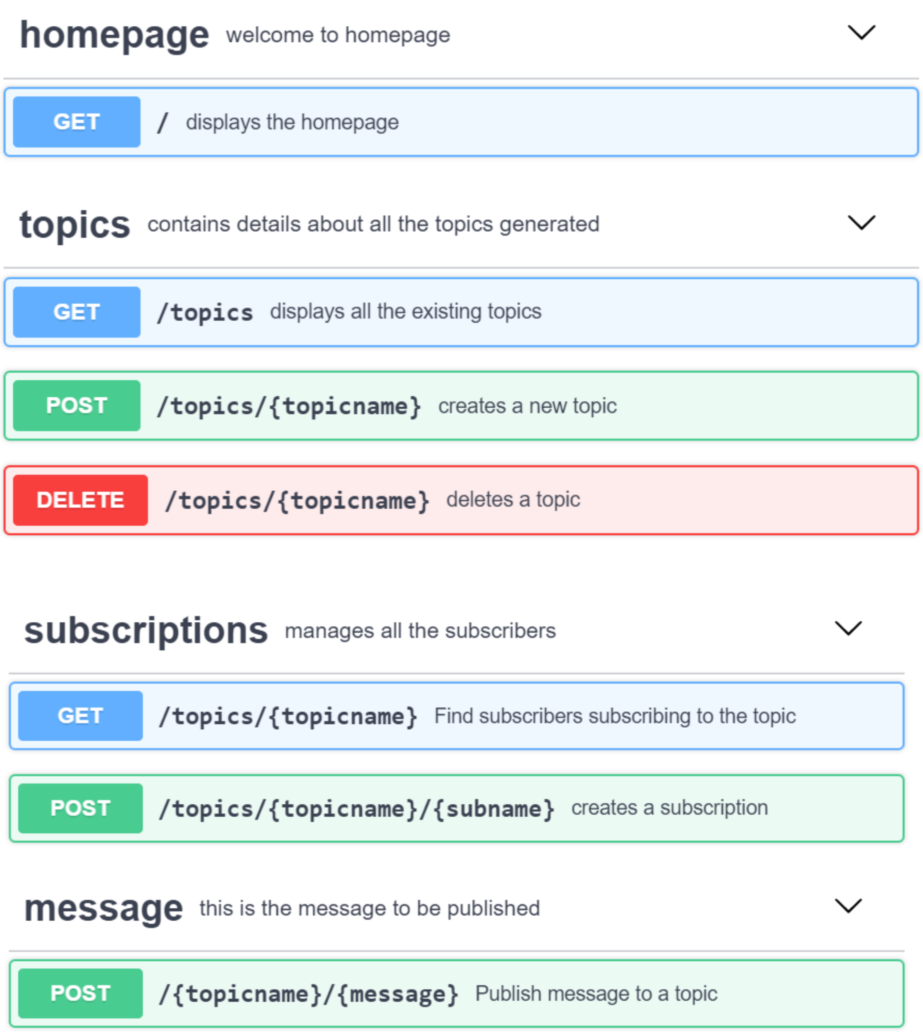


Figure 2.2 Pictorial Implementation of the application

2.4 API Documentation



homepage welcome to homepage

GET / displays the homepage

topics contains details about all the topics generated

GET /topics displays all the existing topics

POST /topics/{topicname} creates a new topic

DELETE /topics/{topicname} deletes a topic

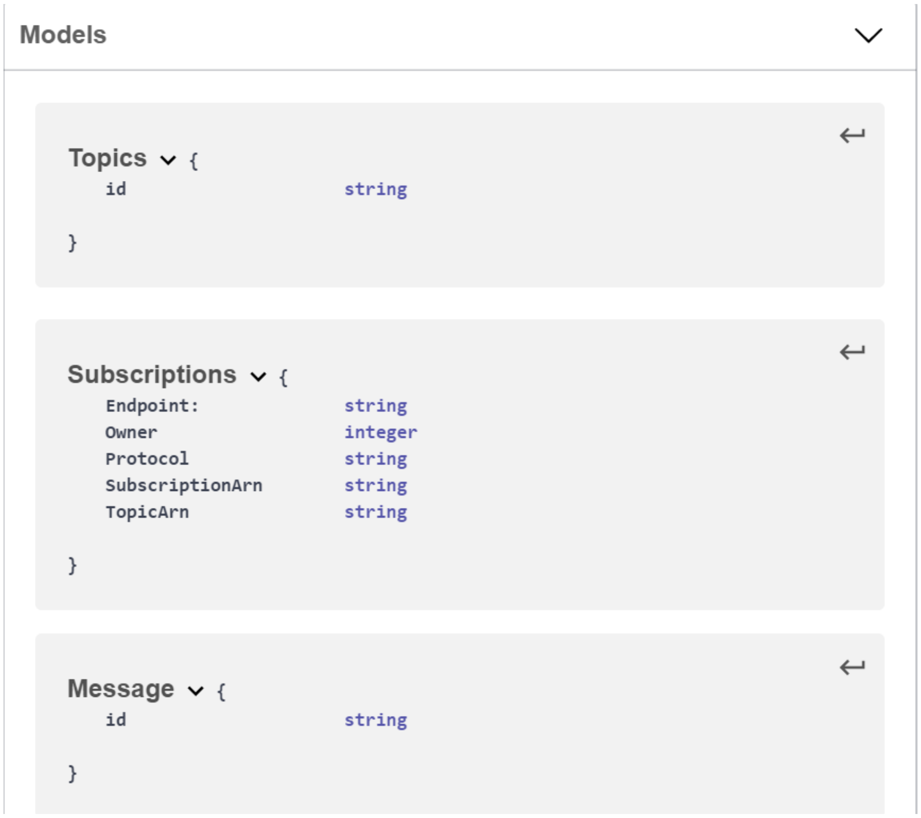
subscriptions manages all the subscribers

GET /topics/{topicname} Find subscribers subscribing to the topic

POST /topics/{topicname}/{subname} creates a subscription

message this is the message to be published

POST /{topicname}/{message} Publish message to a topic



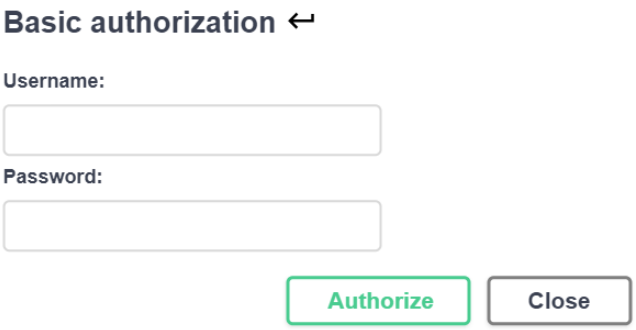
```
Topics {
    id                  string
}

Subscriptions {
    Endpoint:           string
    Owner               integer
    Protocol            string
    SubscriptionArn     string
    TopicArn            string
}

Message {
    id                  string
}
```

2.5 Authentication

Basic authentication is a simple authentication scheme built into the HTTP protocol. The client sends HTTP requests with the Authorization header that contains the word Basic word followed by a space and a base64-encoded string username:password.



Basic authorization
Username:
Password:
Authorize
Close