CommunityConnect using On-Cloud Continuous Integration and Continuous Delivery

Final Project Report

1. Introduction

The Community Connect App is a cloud-based platform that utilizes AWS services to facilitate seamless communication and engagement between neighbors. The app provides a centralized hub for neighbors to come together and share information about local events, services, and anything else that they believe would be beneficial to their community.With the ability to set post priority and trigger email notifications for high-priority posts, residents stay informed.

Our AWS pipeline automates the deployment of new code releases from Github, ensuring a smooth and hassle-free user experience. This involves regularly merging code changes and testing them automatically to prevent errors and maintain consistency. This will ensure that our code is always up and running and will also ensure it passes all the required test cases whenever any new code is committed to the main pipeline. This approach keeps the platform up-to-date and reliable while reducing deployment efforts, time and costs.

2. Background and Related Work

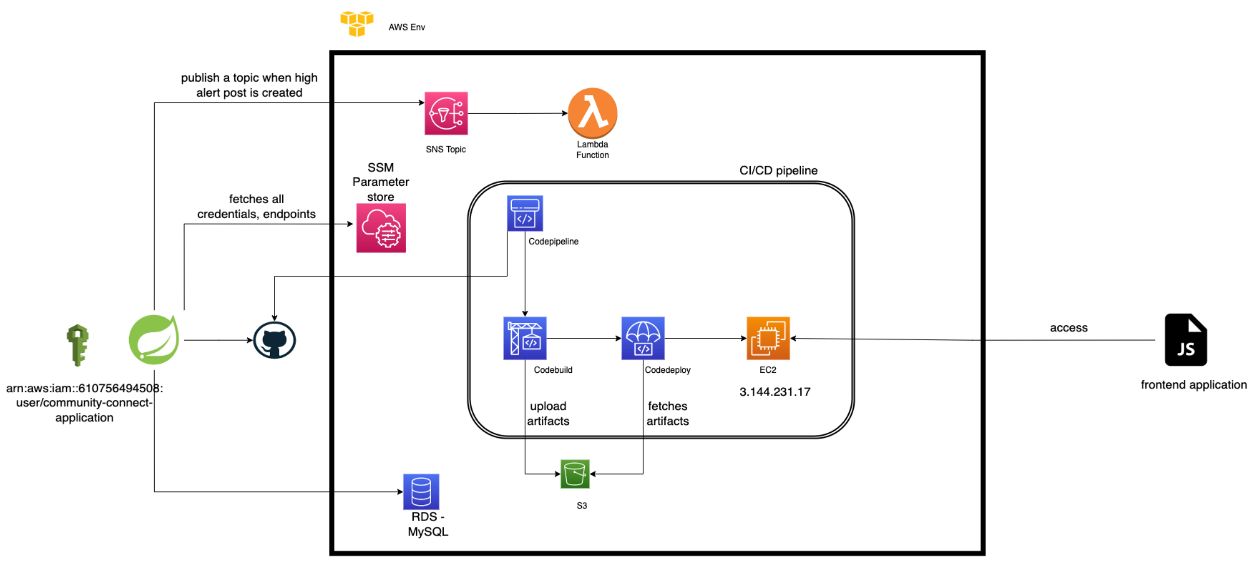
In the traditional software development process, there is manual code deployment which is highly prone to error and extremely time consuming. If not detected early, these errors can go to production and may be expensive to fix. Additionally, there is less automated testing as well as limited visibility about the status of development. Considering fast paced development methodologies like Agile, where multiple changes are committed in every release, it is important to have fast delivery and development as well in order to match the real-world. With manual deployment, it is hard to do it in less time, that too with no errors. Continuous Integration and Delivery/ Deployment helps us to achieve automated deployment. This reduces the time to market thus ensuring that the software stays relevant and up-to-date. Unlike on-premise CI/CD, the cloud based CI/CD offers a more scalable, flexible, reliable and cost efficient infrastructure.

The authors of "Continuous Integration, Delivery and Deployment: A Systematic Review on Approaches, Tools, Challenges and Practices” provide valuable insights into the current state of CI/CD practices and highlight the importance of CI/CD in modern software development processes. They suggest that implementing a set of best practices, such as a robust testing strategy, using version control, and promoting a DevOps culture can help overcome the challenges associated with CI/CD implementation like organizational and cultural issues, technical debt and test automation.

The author of "Practicing Continuous Integration and Continuous Delivery on AWS," discusses how to apply DevOps techniques on the Amazon Web Services (AWS) infrastructure to speed up software delivery. Amazon tools including AWS CodePipeline, AWS CodeBuild, and AWS CodeDeploy are used to construct a CI/CD pipeline in this. It also discusses the advantages of adopting DevOps and CI/CD processes, as well as the tools and services that are offered on AWS. The authors also discuss testing, monitoring and infrastructure as code.

3. Proposed Design/System/Architecture/Application

a. Architecture:



Above diagram shows the architecture of our backend system and how it connects and communicates through various AWS services. As we all know, the key factor is the security in the cloud environment. For this we created, specific IAM user for the application itself to access various AWS services. We attached only specific required policies to the user like SNS policy, RDS policy so that it can access only those services.

We have used AWS Developer tools like CodePipeline, CodeBuild and CodeDeploy. So, whenever we commit new changes to the github on the main branch, codepipeline will get triggered and start the codebuild phase. In this phase, we'll check if all required test cases are passed, no conflicts are there etc. After that, we'll create a jar and upload that to S3. Once, our codebuild phase has been completed, it will go on the codedeploy and deploy our application to EC2 instance. This overall implementation refers to CI/CD i.e Continuous Integration and Continuous Deployment to manage smooth development and deployment cycle.

Coming to the application point of view, we are using AWS RDS mysql to store our data. Also, as we know we have secret information like endpoints, access keys and we should not commit those to GitHub. Hence we are using AWS Parameter store to store those credentials and fetching them directly without the need of hard coding them.

Coming to the high priority post system, we are using AWS SNS and AWS Lambda to cater that requirement. Whenever a high priority post has been created, we are publishing to the SNS topic with the payload of message and email address of the community connect system. This SNS will trigger the lambda and fetch the message and email addresses from the payload and will send an email to every user that a high alert post has been created.

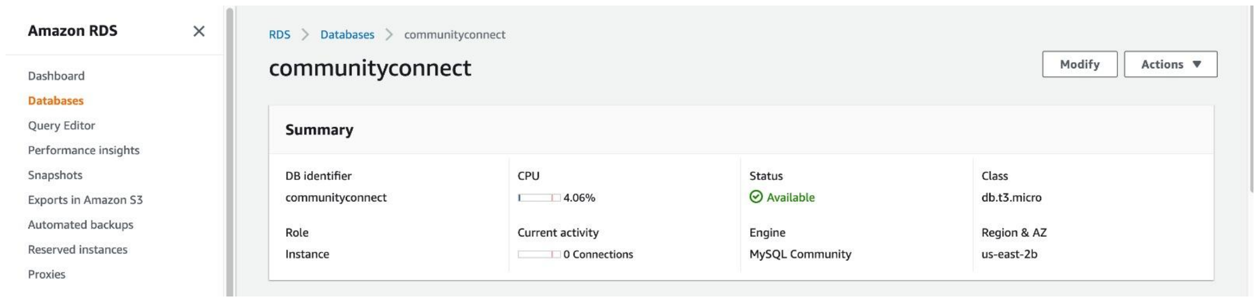
b. **Application:**
   CommunityConnect first needs the user to sign up. The user needs to provide personal and residential details like email, name, apartment number, etc. and needs to set up a password. The username is autogenerated. Once signed up, the user needs to login using the email and password provided at the time of sign-up. Then the user can view the bulletin board wherein all the posts are displayed. The posts with the high priority are highlighted in light red. All the users have the ability to create a post which would be visible to all other users of CommunityConnect. The user can set a priority for his/her post. If a high priority post is made by any user, an email, containing the post, is sent to all users using the CommunityConnect web-app. This helps to inform the users immediately whenever an emergency takes place.

4. Evaluation/Result Analysis:

The CI/CD pipeline successfully starts whenever a code commit is done to the github main branch. The pipeline executes the build and deploy phase, at the end of which the application gets deployed onto the EC2 instance. This new instance holds the latest version of the application. The CommunityConnect app data is successfully saved onto AWS RDS which is confirmed by unit testing the login and sign up functionality. The users can successfully create high and low priority posts and the users are immediately informed via email whenever a high priority post is made by any user. Thus, all of the planned features are working perfectly.

Screenshots:

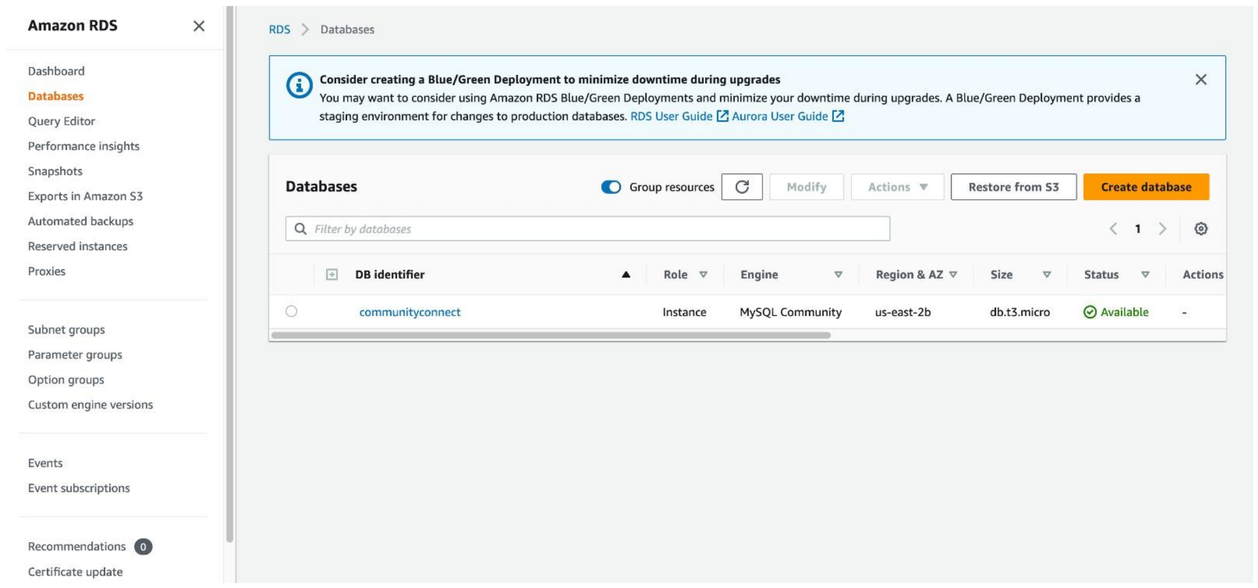
1. AWS RDS



communityconnect

Summary

DB identifier	CPU	Status	Class
communityconnect	4.06%	Available	db.t3.micro
Role	**Current activity**	**Engine**	**Region & AZ**
Instance	0 Connections	MySQL Community	us-east-2b

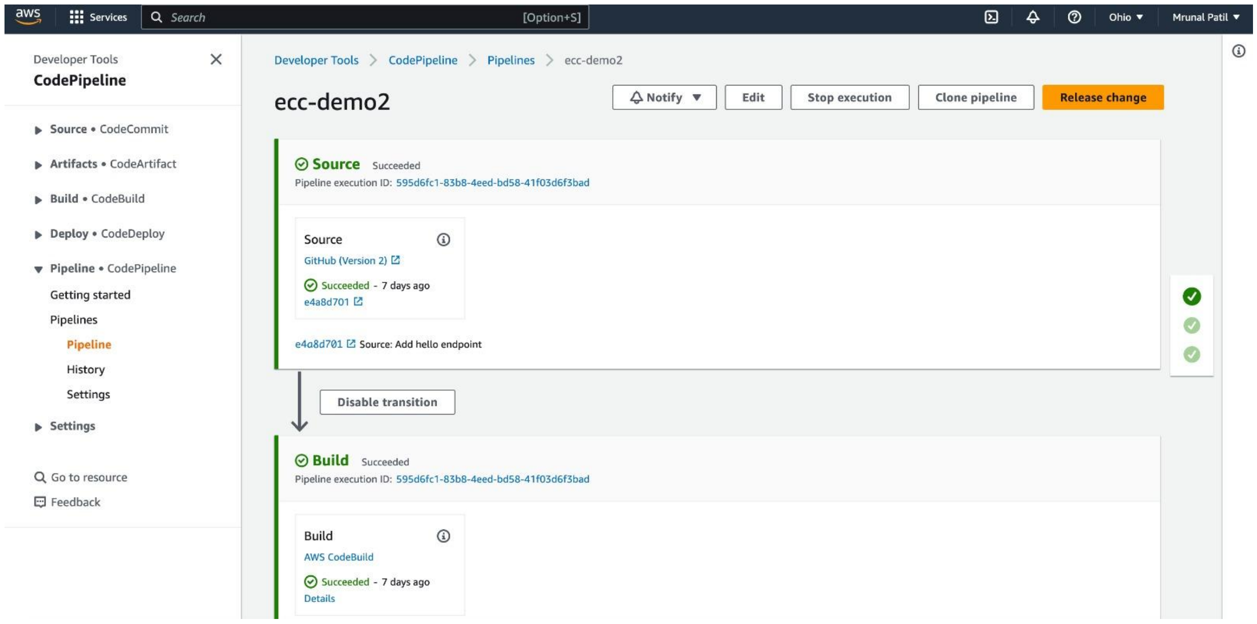


Consider creating a Blue/Green Deployment to minimize downtime during upgrades
You may want to consider using Amazon RDS Blue/Green Deployments and minimize your downtime during upgrades. A Blue/Green Deployment provides a staging environment for changes to production databases. RDS User Guide Aurora User Guide

Databases

DB identifier	Role	Engine	Region & AZ	Size	Status	Actions
communityconnect	Instance	MySQL Community	us-east-2b	db.t3.micro	Available	-

2. AWS Codepipeline with CodeBuild and CodeDeploy



ecc-demo2

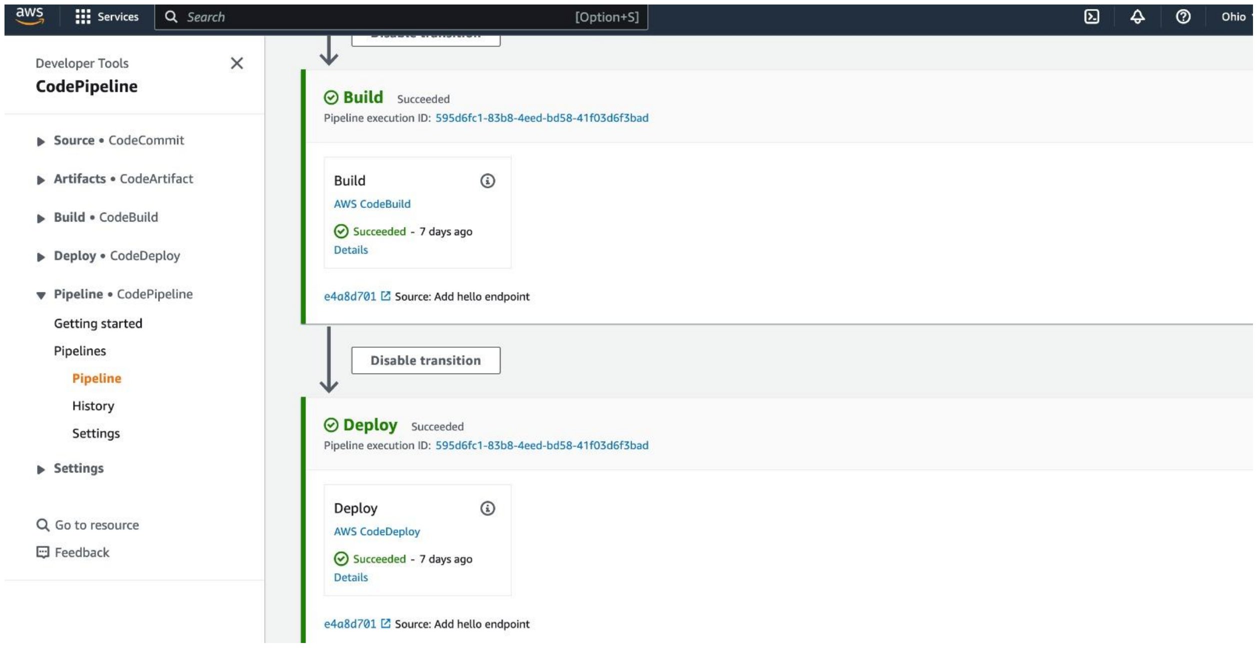
Source Succeeded
Pipeline execution ID: 595d6fc1-83b8-4eed-bd58-41f03d6f3bad

Source
GitHub (Version 2)
Succeeded - 7 days ago
e4a8d701

e4a8d701 Source: Add hello endpoint

Build Succeeded
Pipeline execution ID: 595d6fc1-83b8-4eed-bd58-41f03d6f3bad

Build
AWS CodeBuild
Succeeded - 7 days ago
Details



Build Succeeded
Pipeline execution ID: 595d6fc1-83b8-4eed-bd58-41f03d6f3bad

Build
AWS CodeBuild
Succeeded - 7 days ago
Details

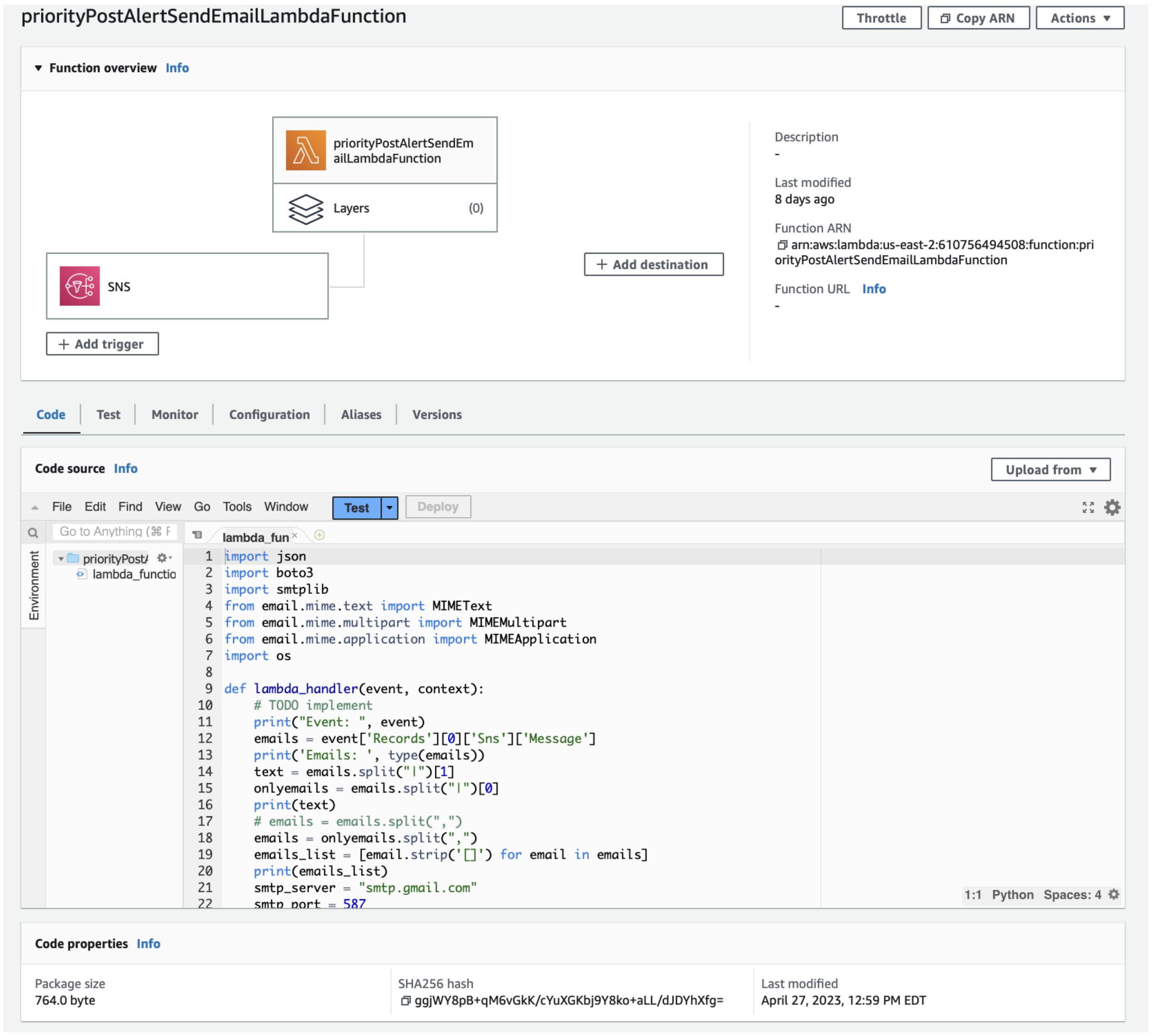
e4a8d701 Source: Add hello endpoint

Deploy Succeeded
Pipeline execution ID: 595d6fc1-83b8-4eed-bd58-41f03d6f3bad

Deploy
AWS CodeDeploy
Succeeded - 7 days ago
Details

e4a8d701 Source: Add hello endpoint

3. AWS Lambda triggered by AWS SNS



Description
-

Last modified
8 days ago

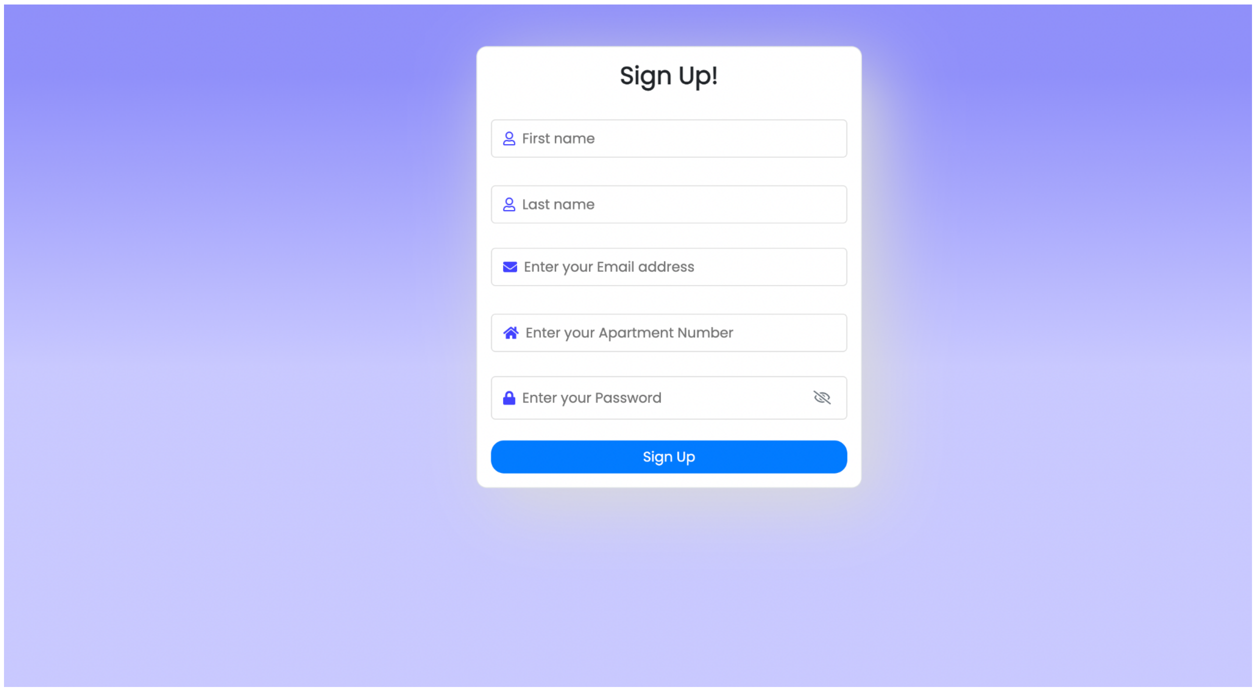
Function ARN
arn:aws:lambda:us-east-2:610756494508:function:priorityPostAlertSendEmailLambdaFunction

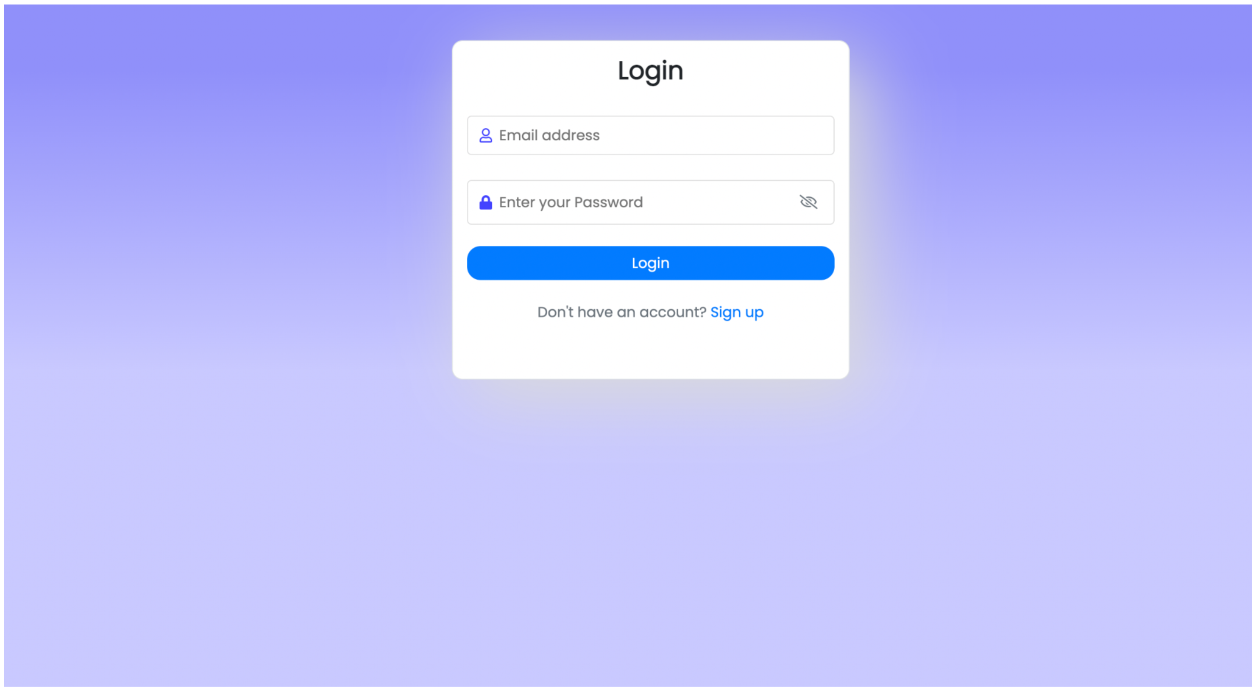
Function URL Info
-

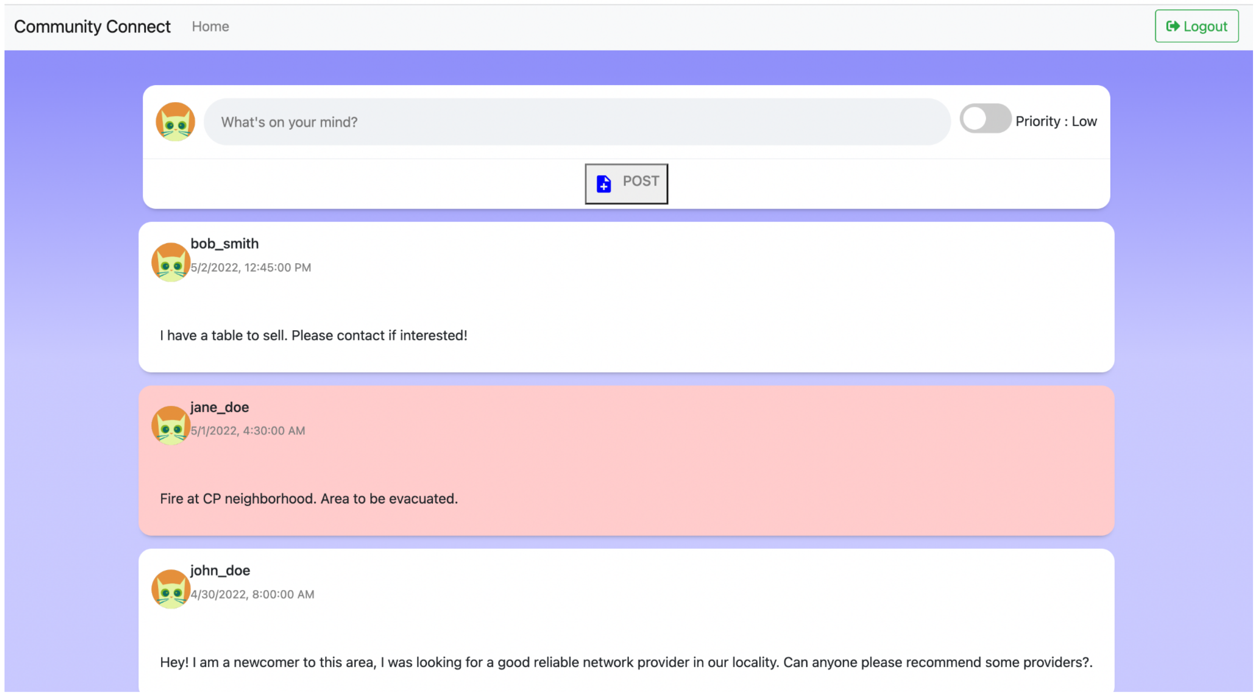
```python
import json
import boto3
import smtplib
from email.mime.text import MIMEText
from email.mime.multipart import MIMEMultipart
from email.mime.application import MIMEApplication
import os

def lambda_handler(event, context):
    # TODO implement
    print("Event: ", event)
    emails = event['Records'][0]['Sns']['Message']
    print('Emails: ', type(emails))
    text = emails.split("|")[1]
    onlyemails = emails.split("|")[0]
    print(text)
    # emails = emails.split(",")
    emails = onlyemails.split(",")
    emails_list = [email.strip('[]') for email in emails]
    print(emails_list)
    smtp_server = "smtp.gmail.com"
    smtp_port = 587
```

Package size	SHA256 hash	Last modified
764.0 byte	ggjWY8pB+qM6vGkK/cYuXGKbj9Y8ko+aLL/dJDYhXfg=	April 27, 2023, 12:59 PM EDT

4. The Application UI:







bob_smith
5/2/2022, 12:45:00 PM

I have a table to sell. Please contact if interested!

jane_doe
5/1/2022, 4:30:00 AM

Fire at CP neighborhood. Area to be evacuated.

john_doe
4/30/2022, 8:00:00 AM

Hey! I am a newcomer to this area, I was looking for a good reliable network provider in our locality. Can anyone please recommend some providers?.

5. Future Work:

a. Ideally, to make sure that new version of code is deployed only when it passes all the code changes and does not affect the current version, a separate pipeline should be developed whenever a pull request is raised and perform the code build functionality.
b. The developers can be informed by improving the current logging. AWS Lambda can be used to send alerts on slack whenever any server side error occurs.
c. In any development team, there are different environments like integration, development, production, etc. Separate pipelines can be built for each of the environments to have easy, bug and error free deployments.
d. Currently, we are configuring and managing all the services through the AWS console. To make this configuration and management easy, we can use IaaC like Terraform, etc.
e. From the application point of view, we can develop a mobile application so that the users receive instant notifications instead of email.