[Document title]

[Document subtitle]

Antariksh Kalita

2023

Table of Contents

[Introduction 2](#_Toc142940773)

[Overview 2](#_Toc142940774)

[Azure Service Bus 2](#_Toc142940775)

[Key Concepts 2](#_Toc142940776)

[Topics and Subscriptions 2](#_Toc142940777)

[Queues 3](#_Toc142940778)

[Message Brokering 3](#_Toc142940779)

[Event Publishing System: Overview 3](#_Toc142940780)

[Project High Level Design: 3](#_Toc142940781)

[Getting Started 3](#_Toc142940782)

[Prerequisites: 3](#_Toc142940783)

[Cloud resource: 3](#_Toc142940784)

[Publisher Application: 4](#_Toc142940785)

[Subscriber application: 4](#_Toc142940786)

[Running the Applications locally: 4](#_Toc142940787)

[Dependency 4](#_Toc142940788)

[The publisher application: 4](#_Toc142940789)

[The Subscriber Application: 4](#_Toc142940790)

# Introduction

Welcome to the User Manual for the Event Publishing System using Azure Service bus.

This comprehensive guide is designed to help you understand and utilize the functionalities of our innovative system for efficiently publishing and subscribing of the event.

# Overview

## Azure Service Bus

Azure Service Bus is a cloud-based messaging service that provides reliable communication between applications and services. This document provides an overview of Azure Service Bus and how it can enhance your application's messaging capabilities.

# Key Concepts

## Topics and Subscriptions

Topics and subscriptions are fundamental components of Azure Service Bus. Understand how they work and how they enable effective communication between publishers and subscribers.

## Queues

Queues offer point-to-point communication and help decouple sender and receiver applications.

## Message Brokering

Learn about message brokers and how they facilitate asynchronous communication.

# Event Publishing System: Overview

Welcome to the documentation for the Flask application, a platform that allows users to subscribe to topics, publish articles, and receive email notifications. This document provides an overview of the application's functionality and guides you through the setup and usage.

This project contains two services independently interacting with Azure Service Bus.

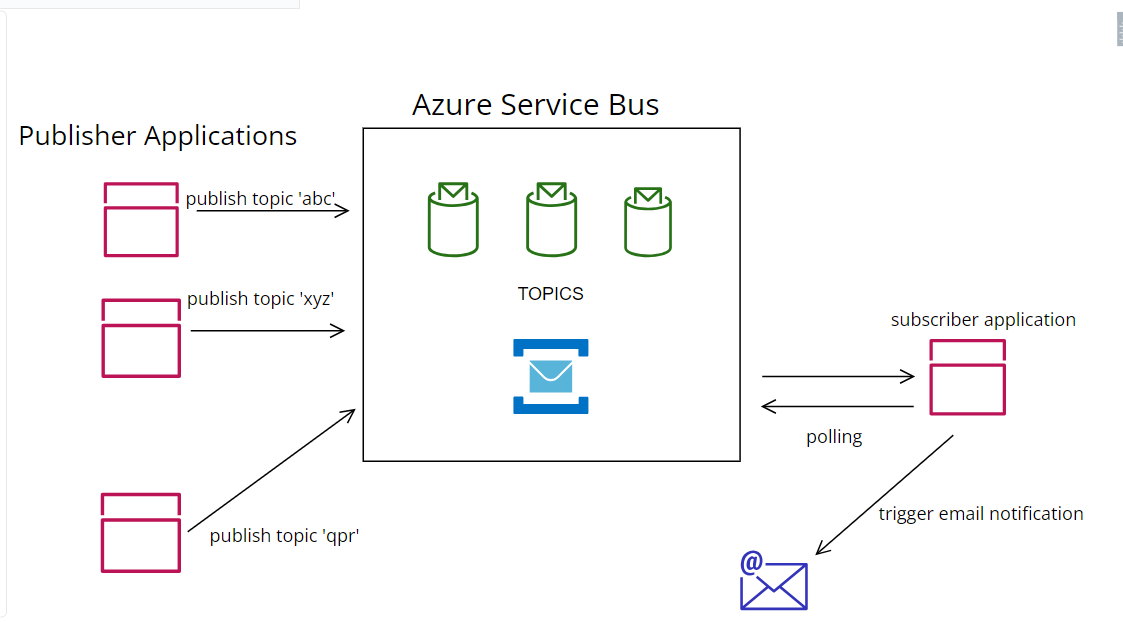
In this project Azure Service Bus’s Topic’s publisher and subscriber feature is being used.

1. Publisher Application: A flask application that allows users to:
   1. User registration and login
   2. Subscription to multiple topics
   3. Publishing articles to topics

TOPICS: Esports, Event, Marketing

1. Subscriber Application: a python script that triggers email notification whenever an event is published not a topic.

# Project High Level Design:



# Getting Started

Guide users through the steps to set up and run the Flask application locally:

# Prerequisites:

Python 3.8 and above.

## Cloud resource:

* Azure Service Bus (standard tier and above)

## Publisher Application:

* + alembic==1.11.2
  + azure-core==1.29.1
  + azure-servicebus==7.11.1
  + blinker==1.6.2
  + certifi==2023.7.22
  + charset-normalizer==3.2.0
  + click==8.1.6
  + colorama==0.4.6
  + Flask==2.3.2
  + Flask-Login==0.6.2
  + Flask-Migrate==4.0.4
  + Flask-SQLAlchemy==3.0.5
  + Flask-WTF==1.1.1
  + greenlet==2.0.2
  + idna==3.4
  + isodate==0.6.1
  + itsdangerous==2.1.2
  + Jinja2==3.1.2
  + Mako==1.2.4
  + MarkupSafe==2.1.3
  + requests==2.31.0
  + six==1.16.0
  + SQLAlchemy==2.0.19
  + typing\_extensions==4.7.1
  + urllib3==2.0.4
  + Werkzeug==2.3.6
  + WTForms==3.0.1

## Subscriber application:

* azure-core==1.29.1
* azure-servicebus==7.11.1
* certifi==2023.7.22

# Project Structure

+---publisher

| | app.py

| | Dockerfile

| | extention.py

| | requirements.txt

| | settings.py

| | topics.json

| |

| +---forms

| | | login\_form.py

| | | publish\_form.py

| | | registration\_form.py

| | | \_\_init\_\_.py

| |

| +---instance

| | db.sqlite3

| |

| +---log

| | | log.py

| | | \_\_init\_\_.py

| +---models

| | | TopicSubscriptionModel.py

| | | UserModel.py

| | | \_\_init\_\_.py

| |

| +---routes

| | | dashboard\_route.py

| | | login\_route.py

| | | publisher\_route.py

| | | registration\_route.py

| | | root.py

| | | \_\_init\_\_.py

| |

| +---services

| | | publish\_utils.py

| | | utils.py

| | | \_\_init\_\_.py

| | |

| +---static

| | \---styles

| | main.css

| |

| +---templates

| | base.html

| | dashboard.html

| | login.html

| | navbar.html

| | publish.html

| | register\_user.html

|

+---subscriber\_stand\_alone

| | Dockerfile

| | log.py

| | requirements.txt

| | send\_email.py

| | setting.py

| | subscriber.py

| | \_\_init\_\_.py

## Running the Applications locally:

### Dependency

* Create the Azure service Bus resource.

### The publisher application:

* Navigate to the publisher application source code directory and open command prompt.
* Install the dependencies by running the command ‘pip install -r requirements.txt” in the command prompt.
* Start the application by executing command “python app.py” in the command prompt.

### The Subscriber Application:

* Navigate to the publisher application source code directory and open command prompt.
* Install the dependencies by running the command ‘pip install -r requirements.txt” in the command prompt.
* Start the application by executing command “python subscriber.py” in the command prompt.

User Registration and Login

Explain how users can register and log in to the application:

Registration process

Login process

Authentication and session management

5. Subscriptions and Topics

Describe how users can subscribe to various topics:

Listing available topics

Subscribing to topics of interest

6. Publish to Topics

Explain how users can publish articles to topics:

Accessing the publishing page

Creating and submitting articles

7. Email Notification

Detail how email notifications are sent to subscribed users when new articles are published:

Email sending mechanism

User-specific email content