Part3: Building a YouTube-like application with RabbitMQ

Components

1. Youtuber -
    - Can publish videos by taking arguments of youtuber name and video name
    - Video name can contains spaces but not youtuber's name

2. User -
    - Logs in by passing 1 or 3 arguments
    - Can subscribe/unsubscribe to youtubers
    - Receives notifications when the youtubers he/she is subscribed to uploads a video and he logs in later, or when he's logged in and the youtuber he/she is subscribed to uploads a video in real-time.

3. YoutubeServer -
    - Receives login requests from users
    - Receives subscription/unsubscription requests from users
    - Receives video uploads from youtubers
    - Sends notifications to users

Steps to run the program

1. Ensure you have Python installed in your system
2. Install RabbitMQ on your workstation (This might require additional requirements like pika, Docker Desktop, KVM virtualization support)
3. Run YoutubeServer.py (**python3 YoutubeServer.py**)- This sets up our server. Ensure it keeps running for the entire test duration.
4. Run Youtuber.py to create a youtuber with an appropriate video name - (For example **python3 Youtuber.py Alice hi guys)**. Ensure this is reflected on the Server as "Alice uploaded hi guys"

5. Run User.py to create a user (For example **python3 User.py newuser s Alice).** Ensure on doing so we get a message on the Server saying “newuser logged in” followed by “newuser subscribed to Alice”.
6. While newuser is logged in, let Alice upload another video (For example **python3 Youtuber.py Alice bye guys).** Ensure that on doing so we get a notification in real-time on User.py as **New Notification: Alice uploaded bye guys)**
7. Stop running newuser. Create another user (For example **python3 User.py anotheruser).** Let this user not be subscribed to Alice.
8. Let Alice upload few more videos. Then let newuser log in again to see those videos reflected once he does log in. Also ensure these videos are not reflected in anotheruser because he/she isn’t subscribed to Alice.
9. Feel free to run more clients and youtubers and vary the test runs.

Troubleshooting

1. Ensure that the required dependencies are installed correctly.
2. Make sure YoutubeServer.py is running at all times.
3. Make sure appropriate users and youtubers are running.
4. Ensure there are no connectivity issues.

Contributors

This project was developed by Deepanshu Dabas and Rudra Jyotirmay as part of Assignment 1.