

# DSCD Assignment 1 Part 3

## README

### Implementing YouTube using RabbitMQ

#### Setting Up Rabbit MQ

To set up RabbitMQ, Erlang is required, both on the local host and Google Cloud.

#### Run YouTubeServer.py

A VM instance has to be created on Google Cloud to run the YouTube server. The external IP address of the VM would be utilized in the user and the YouTuber code to ensure communication between the three components.

<input type="checkbox"/>	Status	Name ↑	Zone	Recommendations	In use by	Internal IP	External IP	Connect
<input type="checkbox"/>	✓	<a href="#">youtubeserver</a>	us-central1-a			10.128.0.2 ( <a href="#">nic0</a> )	<a href="#">34.41.221.73</a> ( <a href="#">nic0</a> )	SSH ▾ ⋮

Run the following command to run the code for YouTube Server.

```
python YouTubeServer.py
```

This would start the server and now it could accept request from both User and YouTuber. To stop the server from running, press Ctrl + C.

#### Run User.py and YouTuber.py

Once the server is running, you can run the User and YouTuber. The command line argument for the two is as follows -

YouTuber - It allows YouTubers to publish videos. A user can publish multiple videos and there can be multiple YouTubers. The data of YouTubers along with their videos are stored on the YouTube server.

```
python Youtuber.py <YoutuberName> <VideoName>
```

User - Users can log in to the system and can also subscribe/unsubscribe to various YouTubers. They receive notifications whenever a YouTuber they have subscribed to publishes a new video.

```
python User.py <UserName>
```

```
python User.py <UserName> <Action: s/u> [<YouTuberName>]
```

Both the above commands are acceptable for Users.