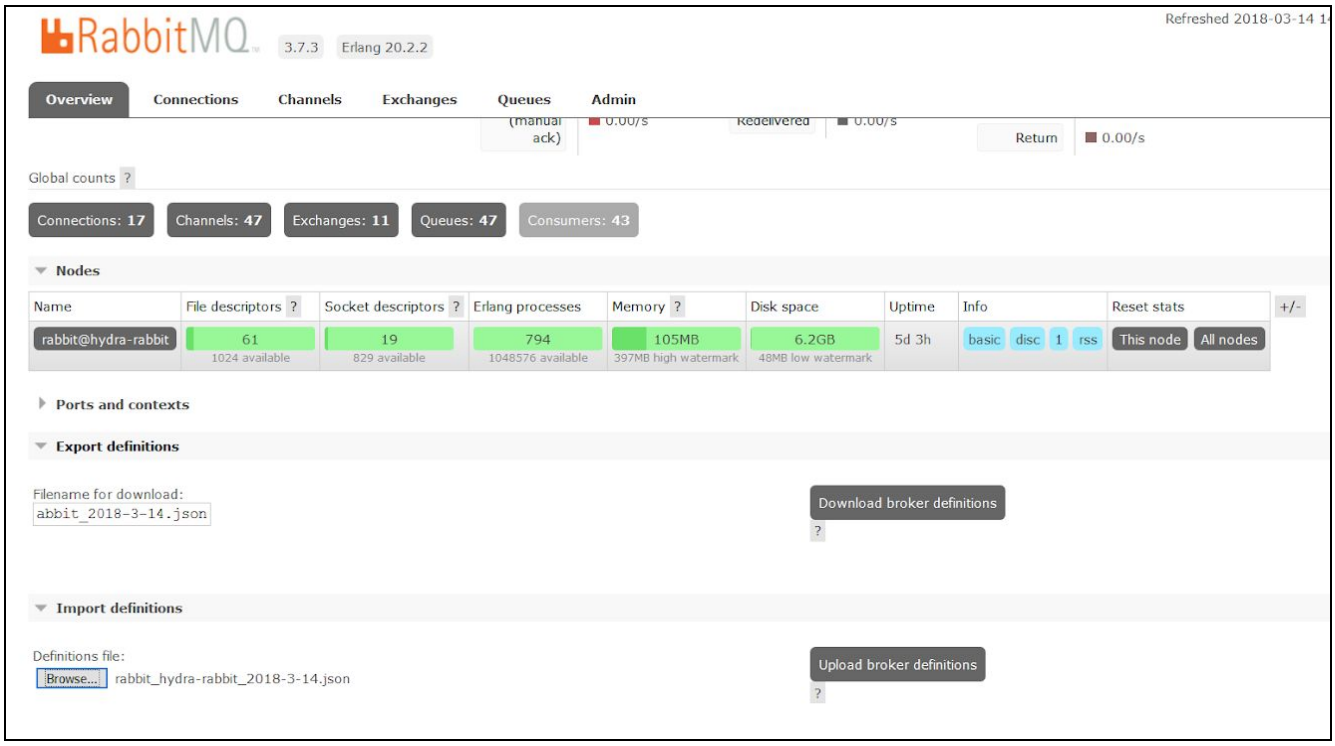
# Installation and Configuration of RabbitMQ on Amazon EC2

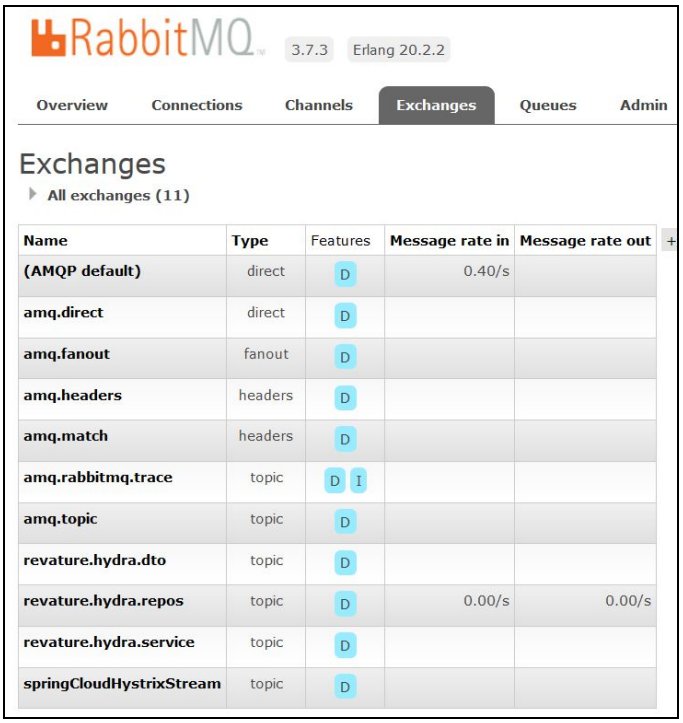
1. Link to install Docker on an EC2.
   https://docs.aws.amazon.com/AmazonECS/latest/developerguide/docker-basics.html#install_docker
2. Link to create a RabbitMQ server onto a Docker container.
   https://docs.docker.com/samples/library/rabbitmq/
   Main command for running container:
   **docker run -d --name <CONTAINER_NAME> -p 5672:5672 -p 15672:15672 rabbitmq:3-management**
   - "-d" is a flag to tell docker to run the container in the background
   - "--name <CONTAINER_NAME>" is command to assign a name to container. Without this flag, docker will assign a random name to the container
   - "-p {ec2Port:containerPort}"
     The most **important** flag of the run command. This is needed in order to tell the ec2 what request on its ports need to be mapped to the container. Without this the docker container simply will not receive any request. So it's good to keep them the same as defined for rabbitmq but it is possible to change these
     Ex: -p 8080:15672 will map all request to the ec2 on port 8080 to containers personal listening port of 15672. ec2ip:8080 => dockercontainer:15672.

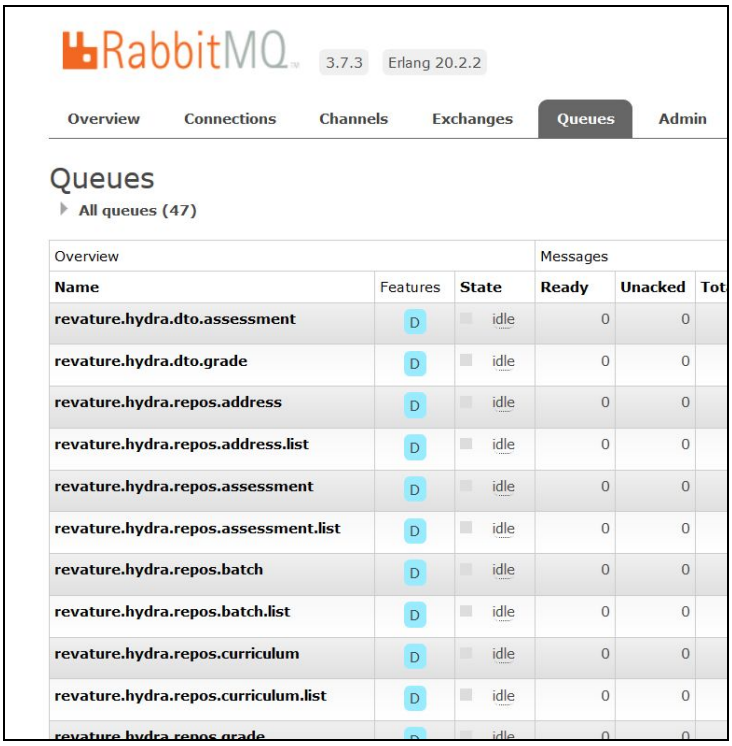
3. Accessing the RabbitMQ is possible through the enabled management plugin. This can be done by entering "http://ec2ip:port" into a web browser and using the default credentials.
   Username: "guest", Password: "guest"

4. Once logged into the configuration there is an option at the bottom that allows you to import the definitions which is included in the configuration service's src/main/resources called rabbit_hydra-rabbit_2018-3-14 of the first iteration of Hydra.




5. Importing of definitions is successful if exchanges and queues tab look similar to below screenshots



## Adding more exchanges/queues to RabbitMQ.

1. Click on the appropriate tab to open either an exchange or queue.
2. At the bottom there are options to fill out a new exchange/queue.
3. Make sure to follow proper naming conventions (revature. )
4. Arguments are not necessary but allow for customization of the exchange or queue.
5. After the exchange or queue fits specifications then click the add button.
6. Messaging is handled in each microservice within their respective service package that you can reference.
    a. Example for Batch Service: (located in com.revature.hydra.batch.service)
    "BatchCompositionMessageService" contains calls to RabbitMQ Server exchange and queues
    "BatchRepositoryMessageService" contains listeners to RabbitMQ Server queue

## Other useful Docker commands:

List Containers:

```
docker ps
```

Delete container

```
docker rm <CONTAINER_ID>
```

List all exited containers

```
docker ps -aq -f status=exited
```

Remove stopped containers

```
docker ps -aq --no-trunc | xargs docker rm
```

To bash into running container

```
docker exec -t -i <CONTAINER_NAME> /bin/bash
```

Start an existing container

```
docker start <CONTAINER_NAME>
```

Get information about the container

```
docker inspect <CONTAINER_ID>
```

Start Docker Daemon

```
sudo service docker start
```