

Message broker et programmation asynchrone - RabbitMQ

Objectifs

L'objectif de ce TP est de vous initier à l'utilisation de RabbitMQ, un message broker open source très populaire. À la fin de ce TP, vous devriez être capable de :

- Installer et configurer RabbitMQ
- Utiliser la ligne de commande pour envoyer et recevoir des messages
- Utiliser un client RabbitMQ pour envoyer et recevoir des messages

Étape 1 : Installation et configuration de RabbitMQ

La première étape consiste à installer RabbitMQ. Vous pouvez trouver des instructions d'installation pour différentes plateformes sur le site web officiel de RabbitMQ :

<https://www.rabbitmq.com/download.html>.

Une fois que vous avez installé RabbitMQ, vous pouvez le démarrer en exécutant la commande suivante dans votre terminal :

```
rabbitmq-server
```

Cela démarrera le serveur RabbitMQ. Par défaut, RabbitMQ utilise le port 5672 pour la communication.

Vous pouvez également accéder à l'interface graphique de RabbitMQ en ouvrant un navigateur web et en accédant à l'URL suivante : <http://localhost:15672>. L'interface graphique permet de visualiser les différents échanges, les files d'attente, les connexions et les canaux.

Étape 2 : Utilisation de la ligne de commande pour envoyer et recevoir des messages

La deuxième étape consiste à utiliser la ligne de commande pour envoyer et recevoir des messages à partir de RabbitMQ. Nous allons utiliser la commande `rabbitmqadmin`, un outil en ligne de commande pour gérer RabbitMQ.

Envoyer un message avec un échange direct

Un échange direct est le type d'échange le plus simple. Il achemine les messages vers les files d'attente basées sur leur clé de routage. Pour envoyer un message avec un échange direct à partir de la ligne de commande, vous pouvez utiliser la commande suivante :

```
rabbitmqadmin publish exchange=<nom_de_l_echange>  
routing_key=<nom_de_la_file_d_attente> payload="Hello, world!"  
--vhost=<nom_de_vhost>
```

Remplacez `<nom_de_l_echange>` par le nom de l'échange direct, `<nom_de_la_file_d_attente>` par la clé de routage utilisée par la file d'attente de destination et `<nom_de_vhost>` par le nom du vhost.

Envoyer un message avec un échange fanout

Un échange fanout achemine les messages vers toutes les files d'attente auxquelles il est lié, quelles que soient les clés de routage. Pour envoyer un message avec un échange fanout à partir de la ligne de commande, vous pouvez utiliser la commande suivante :

```
rabbitmqadmin publish exchange=<nom_de_l_echange> routing_key=""  
payload="Hello, world!" --vhost=<nom_de_vhost>
```

Remplacez `<nom_de_l_echange>` par le nom de l'échange fanout et `<nom_de_vhost>` par le nom du vhost.

Envoyer un message avec un échange topic

Un échange topic achemine les messages vers les files d'attente basées sur des expressions de routage qui utilisent des caractères génériques (*) et des caractères multi-niveaux (#). Pour envoyer un message avec un échange topic à partir de la ligne de commande, vous pouvez utiliser la commande suivante :

```
rabbitmqadmin publish exchange=<nom_de_l_echange>  
routing_key=<routing_key> payload="Hello, world!"  
--vhost=<nom_de_vhost>
```

Remplacez `<nom_de_l_echange>` par le nom de l'échange topic, `<routing_key>` par l'expression de routage utilisée par la file d'attente de destination et `<nom_de_vhost>` par le nom du vhost.

Recevoir un message

Pour recevoir un message à partir de la ligne de commande, vous pouvez utiliser la commande suivante :

```
rabbitmqadmin get queue=<nom_de_la_queue> --vhost=<nom_de_vhost>
```

Remplacez `<nom_de_la_queue>` par le nom de la file d'attente à partir de laquelle vous voulez recevoir le message et `<nom_de_vhost>` par le nom du vhost.

Étape 3 : Utilisation d'un client RabbitMQ pour envoyer et recevoir des messages

La troisième étape consiste à utiliser un client RabbitMQ pour envoyer et recevoir des messages. Vous pouvez choisir le langage de votre choix.