

# Projet RabbitMQ

**Équipe :** Florent Ielion, Florian Germain, Redjane Rajabaly, Thibault Delattre  
**Date :** 21/05/2025

## ***Table des matières***

- 1. Contexte & Objectifs*
- 2. Prérequis*
- 3. Installation & Lancement*
- 4. Architecture du système*
- 5. Configuration RabbitMQ*
- 6. Composants & Code*
- 7. Exemples de Messages*
- 8. Monitoring*
- 9. Améliorations & Perspectives*
- 10. Annexes*

## 1. Contexte & Objectifs

Cet exercice met en place un système de calcul distribué avec RabbitMQ :

- Un **producteur** envoie des requêtes mathématiques
- Plusieurs **workers** (*add, sub, mul, div, all*) traitent ces requêtes
- Un **consommateur** collecte et affiche les résultats
- Une **interface Web** permet d'interagir et de monitorer en temps réel

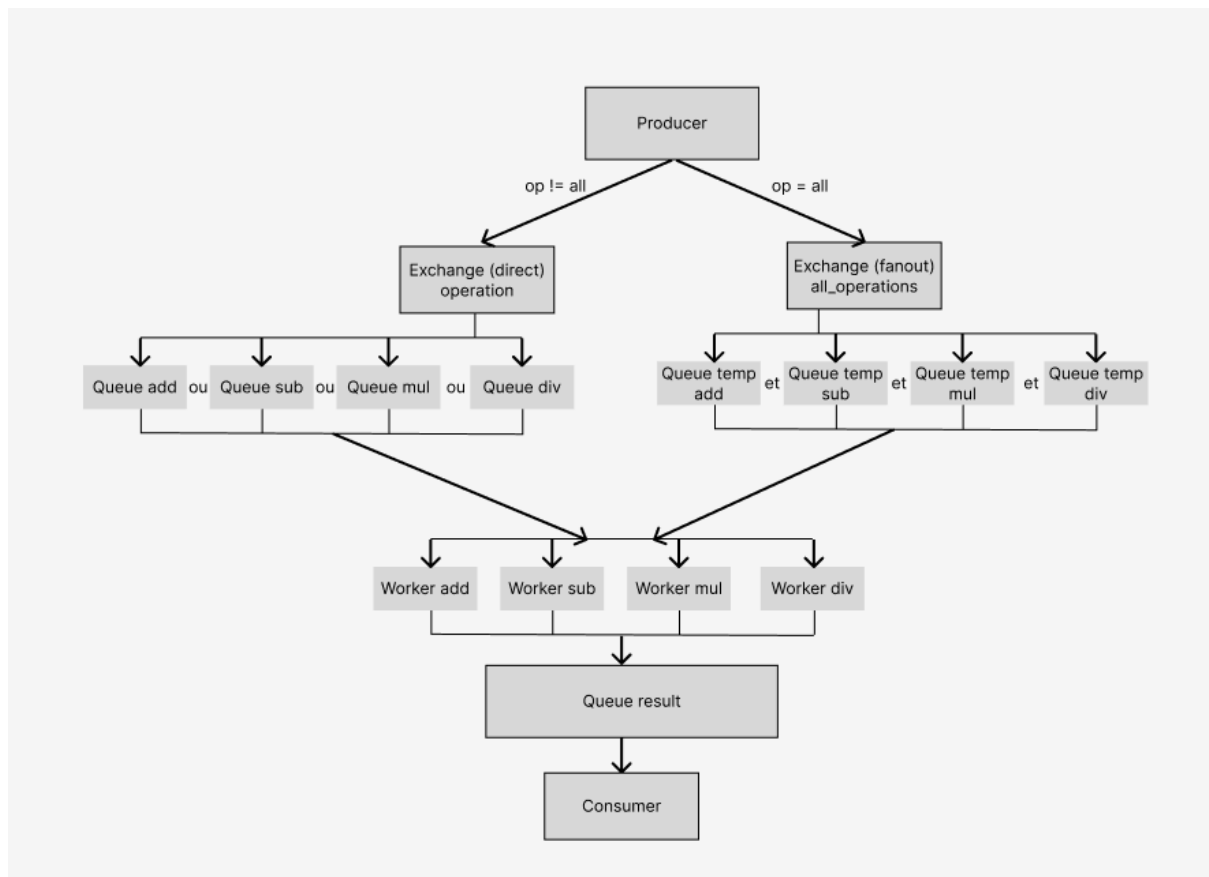
## 2. Prérequis

- Docker ≥ 19.03 & Docker Compose ≥ 1.25
- Node.js ≥ 14 + npm

## 3. Installation & Lancement

1. Cloner le dépôt : **git clone** <https://github.com/Tibo1310/efreirabbitmq.git>
2. `cd efreirabbitmq`
3. `npm install`
4. `docker-compose up --build`
5. Vérifier
  - RabbitMQ Management → <http://localhost:15672> (guest/guest)
  - Interface Web → <http://localhost:3000>

## 4. Architecture du système



## 5. Configuration RabbitMQ

Dans `config/rabbitmq.js`, on paramètre l'URI, les queues et l'échange `all_operations` :

**Fig. 2.** Extrait de `rabbitmq.js`

```
JS javascript

const config = {
  rabbitmq: {
    url: process.env.RABBITMQ_URL || 'amqp://localhost',
    queues: {
      calculations: process.env.QUEUE_NAME || 'calculations',
      results: process.env.RESULT_QUEUE || 'results'
    },
    exchanges: {
      all_operations: 'all_operations'
    }
  }
};
```

Fig. 3 – Flux direct : **calculations** → exchange → workers → **results**

## 7. Composants & Code

## 7.1 Producteur (**producer.js**)

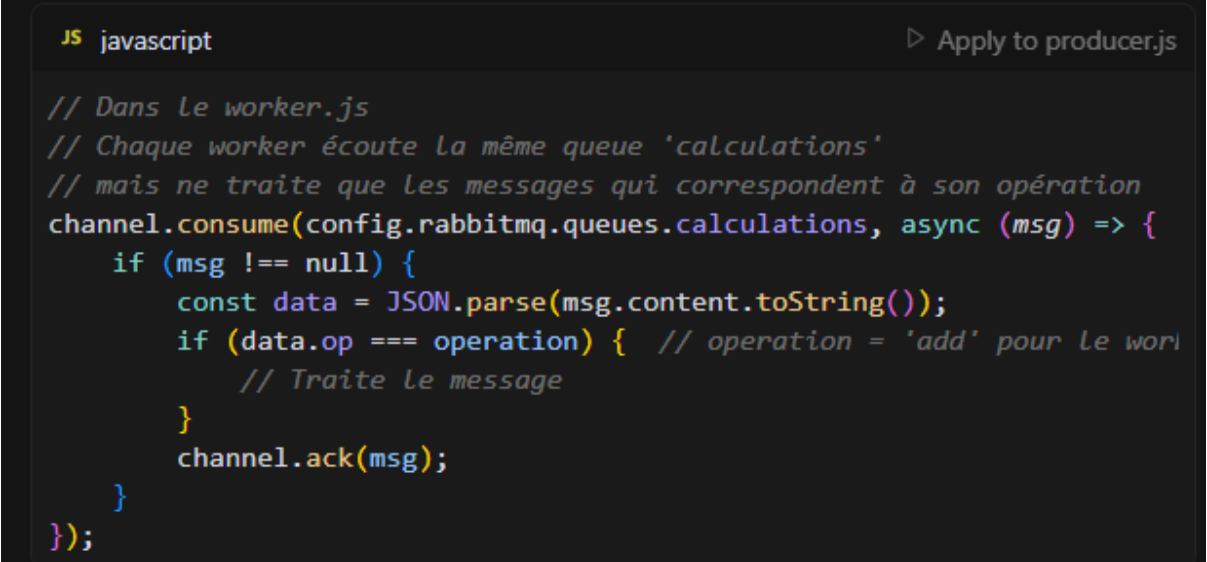
- Publie toutes les 5 s un message { n1, n2, op } sur `calculations` ou `all_operations`.

## 7.2 Workers (**worker.js**)

- Démarrage :

node worker.js --op add

- Bind sur `calculations` avec `routingKey = op`, et sur `all_operations`.
- Simule un **sleep 5–15 s**, calcule, puis envoie { n1, n2, op, result } sur `results`.

- 

## 7.3 Consommateur (**consumer.js**)

- Lit `results` et affiche en console ou émet vers l'UI Web.

## 7.4 Interface Web

**Fig. 5.** Exemple de capture d'écran UI

Mettre ici votre screenshot de l'interface Web (ex : `/mnt/data/...`).

## 8. Exemples de Messages

### Requête

```
{n1": 5, "n2": 3, "op": "add" }
```

```
{ "n1": 5, "n2": 3, "op": "add", "result": 8 }
```

## 9. Monitoring

- **RabbitMQ Management** : <http://localhost:15672>
- **Logs Docker** :

```
docker-compose logs -f worker-add
```

## 10. Améliorations & Perspectives

- Tests unitaires et intégration
- Documentation API (Swagger/OpenAPI)
- Monitoring avancé (Prometheus/Grafana)
- Déploiement Kubernetes
- Logging centralisé (ELK Stack)

## **11. Annexes**

### **11.1 docker-compose.yml**

```
1  version: '3.8'
2
3  services:
4    # Service RabbitMQ
5    rabbitmq:
6      image: rabbitmq:3-management
7      hostname: rabbitmq
8      ports:
9        - "5672:5672" # Port AMQP
10       - "15672:15672" # Port interface de gestion
11      environment:
12        - RABBITMQ_DEFAULT_USER=guest
13        - RABBITMQ_DEFAULT_PASS=guest
14      healthcheck:
15        test: ["CMD", "rabbitmqctl", "status"]
16        interval: 30s
17        timeout: 10s
18        retries: 5
19
20    # Producteur
21    producer:
22      build: .
23      environment:
24        - MODE=producer
25        - RABBITMQ_URL=amqp://guest:guest@rabbitmq:5672
26        - NODE_ENV=development
27      volumes:
28        - ./src:/usr/src/app/src
29        - ./package.json:/usr/src/app/package.json
30      command: npm run dev:producer
31      depends_on:
32        rabbitmq:
33          condition: service_healthy
```



```
34
35     # Consommateur
36     consumer:
37         build: .
38         environment:
39             - MODE=consumer
40             - RABBITMQ_URL=amqp://guest:guest@rabbitmq:5672
41             - NODE_ENV=development
42         volumes:
43             - ./src:/usr/src/app/src
44             - ./package.json:/usr/src/app/package.json
45         command: npm run dev:consumer
46         depends_on:
47             rabbitmq:
48                 condition: service_healthy
49
50     # Interface Web
51     web:
52         build: .
53         environment:
54             - MODE=web
55             - PORT=3000
56             - RABBITMQ_URL=amqp://guest:guest@rabbitmq:5672
57             - NODE_ENV=development
58         ports:
59             - "3000:3000"
60         volumes:
61             - ./src:/usr/src/app/src
62             - ./package.json:/usr/src/app/package.json
63         command: npm run dev:web
64         depends_on:
65             rabbitmq:
66                 condition: service_healthy
67
```

```
68     # Workers pour chaque type d'opération
69     worker-add:
70         build: .
71         environment:
72             - MODE=worker
73             - OPERATION=add
74             - RABBITMQ_URL=amqp://guest:guest@rabbitmq:5672
75             - NODE_ENV=development
76         volumes:
77             - ./src:/usr/src/app/src
78             - ./package.json:/usr/src/app/package.json
79         command: npm run dev:worker
80         depends_on:
81             rabbitmq:
82                 condition: service_healthy
83
84     worker-sub:
85         build: .
86         environment:
87             - MODE=worker
88             - OPERATION=sub
89             - RABBITMQ_URL=amqp://guest:guest@rabbitmq:5672
90             - NODE_ENV=development
91         volumes:
92             - ./src:/usr/src/app/src
93             - ./package.json:/usr/src/app/package.json
94         command: npm run dev:worker
95         depends_on:
96             rabbitmq:
97                 condition: service_healthy
98
99     worker-mul:
100         build: .
101         environment:
102             - MODE=worker
103             - OPERATION=mul
```

```
103         - OPERATION=mul
104         - RABBITMQ_URL=amqp://guest:guest@rabbitmq:5672
105         - NODE_ENV=development
106     volumes:
107         - ./src:/usr/src/app/src
108         - ./package.json:/usr/src/app/package.json
109     command: npm run dev:worker
110     depends_on:
111         rabbitmq:
112             condition: service_healthy
113
114     worker-div:
115         build: .
116         environment:
117             - MODE=worker
118             - OPERATION=div
119             - RABBITMQ_URL=amqp://guest:guest@rabbitmq:5672
120             - NODE_ENV=development
121         volumes:
122             - ./src:/usr/src/app/src
123             - ./package.json:/usr/src/app/package.json
124         command: npm run dev:worker
125         depends_on:
126             rabbitmq:
127                 condition: service_healthy
```