



Образование для инженеров
и технических лидеров

RabbitMQ - Первые шаги

Спикер: Алексей Барабанов



Цель урока

Научиться запускать и пользоваться RabbitMQ

План урока

- Запуск RabbitMQ в **докере**
- **Обязательные** параметры
- **Стейты** - сохранение персистентных данных
- Не обязательные, но **важные** параметры
- Возможности **консоли**
- Возможности **веб интерфейса**
- **Практическое задание**

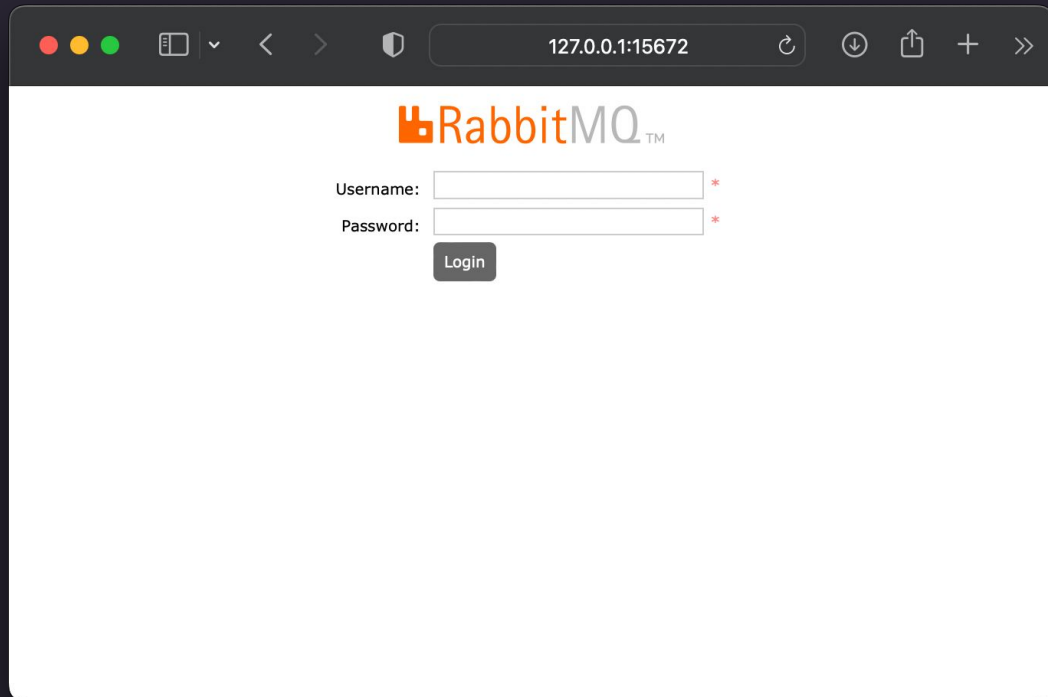
Запуск в докере

Самый простой запуск через голый докер

```
docker run -p 15672:15672 rabbitmq:3.10.7-management
```

Запуск в докере

Открываем в браузере <http://127.0.0.1:15672>



Запуск в докере через docker-compose

docker-compose.yml

```
MacBookPro:rabbitmq_slurm kilex$ cat docker-compose.yml
```

```
version: "2.1"
```

```
services:
```

```
  rabbitmq:
```

```
    image: rabbitmq:3.10.7-management
```

```
    ports:
```

```
      - 15672:15672
```

```
MacBookPro:rabbitmq_slurm kilex$ docker-compose up -d
```

```
Starting rabbitmq_slurm_rabbitmq_1 ... done
```

```
MacBookPro:rabbitmq_slurm kilex$ docker-compose down
```

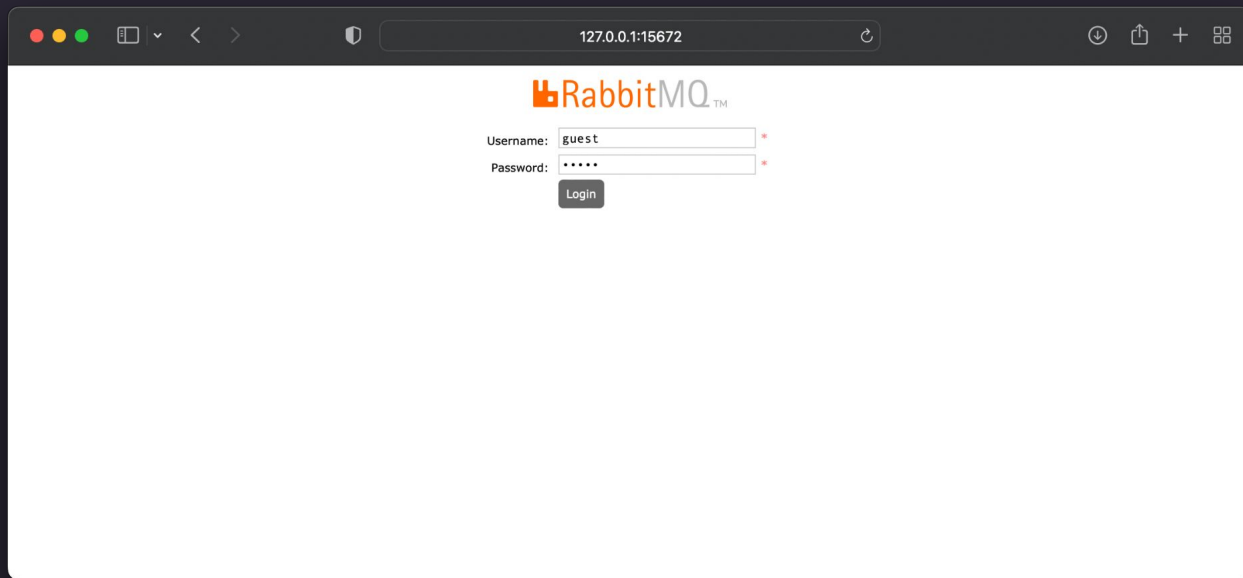
```
Stopping rabbitmq_slurm_rabbitmq_1 ... done
```

```
Removing rabbitmq_slurm_rabbitmq_1 ... done
```

```
Removing network rabbitmq_slurm_default
```

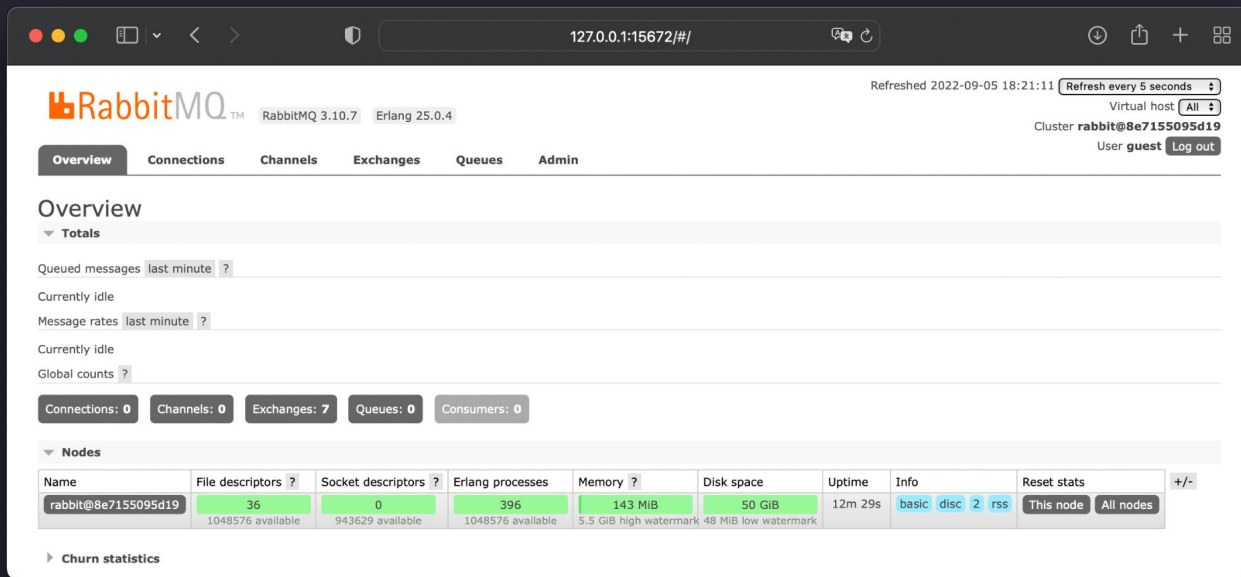
Запуск в докере через docker-compose

Открываем в браузере <http://127.0.0.1:15672>



Запуск в докере через docker-compose

Авторизуемся как **guest:guest**



The screenshot shows the RabbitMQ management interface. At the top, the RabbitMQ logo is visible along with the version information: RabbitMQ 3.10.7 and Erlang 25.0.4. The interface is refreshed every 5 seconds. The user is logged in as 'guest' and can log out. The 'Overview' tab is selected, showing various system statistics for the node 'rabbit@8e7155095d19'.

Overview

Totals

Queued messages last minute ?

Currently idle

Message rates last minute ?

Currently idle

Global counts ?

Connections: 0 Channels: 0 Exchanges: 7 Queues: 0 Consumers: 0

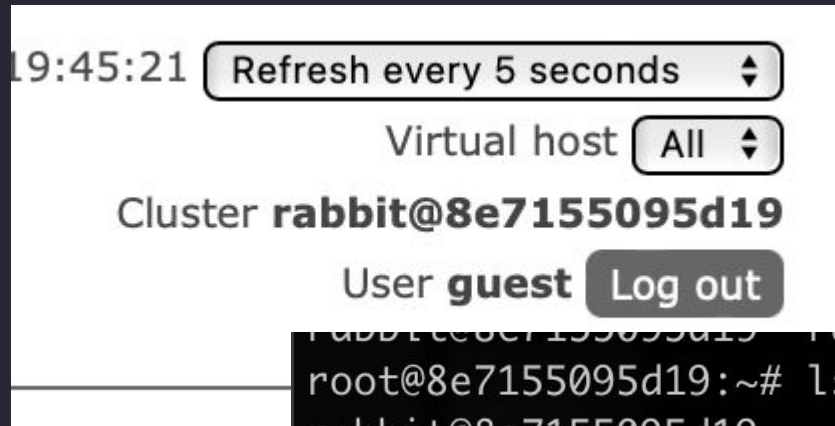
Nodes

Name	File descriptors <small>?</small>	Socket descriptors <small>?</small>	Erlang processes	Memory <small>?</small>	Disk space	Uptime	Info	Reset stats	+/-
rabbit@8e7155095d19	36 <small>1048576 available</small>	0 <small>943629 available</small>	396 <small>1048576 available</small>	143 MIB <small>5.5 GiB high watermark</small>	50 GiB <small>48 MiB low watermark</small>	12m 29s	basic disc 2 rss	This node All nodes	

Churn statistics

Запуск в докере через docker-compose

Правый верхний угол



```
rabbit@8e7155095d19:~# ls -l /var/lib/rabbitmq/mnesia/  
rabbit@8e7155095d19  
rabbit@8e7155095d19-feature_flags  
rabbit@8e7155095d19-plugins-expand  
rabbit@8e7155095d19.pid  
root@8e7155095d19:~#
```

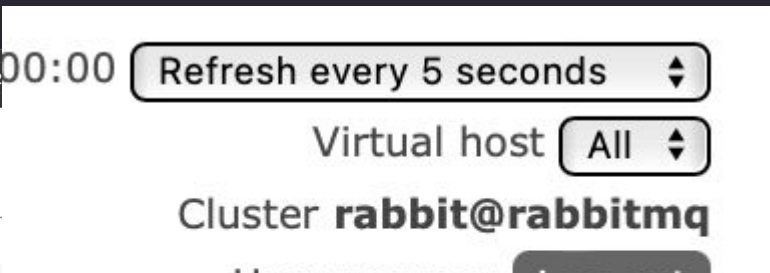
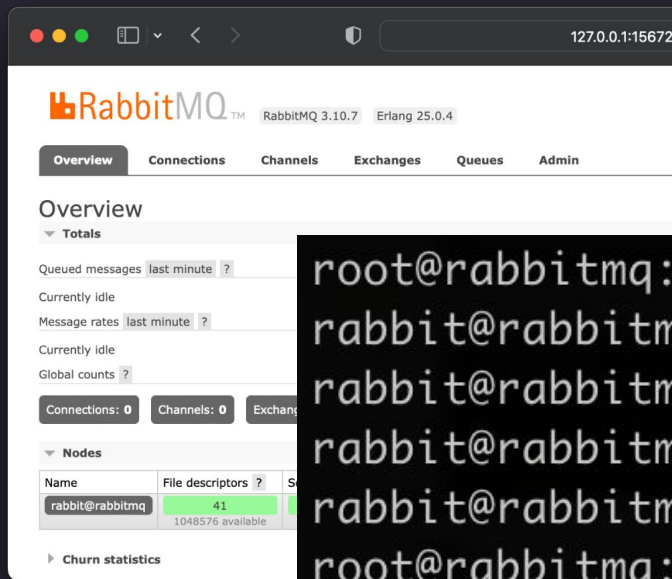
Запуск в docker-compose, hostname, auth

Добавляем **hostname** и **USER/PASS**

```
rabbitmq:  
  image: rabbitmq:3.10.7-management  
  hostname: rabbitmq  
  restart: always  
  environment:  
    - RABBITMQ_DEFAULT_USER=ruser  
    - RABBITMQ_DEFAULT_PASS=rpassword  
  ports:  
    - 15672:15672
```

Запуск в docker-compose, hostname, auth

Логинимся под новыми реквизитами



```
root@rabbitmq:/# ls -1 /var/lib/rabbitmq/mnesia/  
rabbit@rabbitmq  
rabbit@rabbitmq-feature_flags  
rabbit@rabbitmq-plugins-expand  
rabbit@rabbitmq.pid  
root@rabbitmq:/#
```

Запуск в docker-compose, стейт

Добавляем хранение стейта на диске

```
rabbitmq:
  image: rabbitmq:3.10.7-management
  hostname: rabbitmq
  restart: always
  environment:
    - RABBITMQ_DEFAULT_PASS=rabbitmq
    - RABBITMQ_DEFAULT_USER=rabbitmq
  volumes:
    - ./rabbitmq:/var/lib/rabbitmq
  ports:
    - 15672:15672
```

```
MacBookPro:rabbitmq_slERM kilex$ ls -l
total 12
-rw-r--r--  1 kilex  staff  4096  11  11  11:11 rabbitmq

MacBookPro:rabbitmq_slERM kilex$ ls -l rabbitmq/mnesia/
total 12
-rw-r--r--  1 rabbit  staff  4096  11  11  11:11 rabbitmq

MacBookPro:rabbitmq_slERM kilex$
```

Запуск в docker-compose, свободное место

Порог свободного места

The screenshot shows the RabbitMQ management interface in a browser window. The address bar displays the URL `127.0.0.1:15672/#/`. The page title is "RabbitMQ" and it indicates the version "RabbitMQ 3.10.7 Erlang 25.0.4". The interface is refreshed every 5 seconds. The user is logged in as "rmuser".

The "Overview" tab is selected, showing various statistics. A prominent green box highlights the "Disk space" usage, which is currently at "333 GiB". Below this, a warning message states "Warning: 48 MiB low watermark".

The "Nodes" section shows a table with the following data:

Name	File descriptors ?	Socket descriptors ?
rabbit@rabbitmq	35 1048576 available	0 943629 available

Other statistics shown include "Connections: 0", "Channels: 0", "Exchanges: 7", and "Queues: 0".

Запуск в docker-compose, порог свободного места

2147483648 бит это два гигабита

```
rabbitmq:  
  image: rabbitmq:3.10.7-management  
  hostname: rabbitmq  
  restart: always  
  environment:  
    - RABBITMQ_DEFAULT_USER=rmuser  
    - RABBITMQ_DEFAULT_PASS=rmuser  
    - RABBITMQ_SERVER_ADDITIONAL_DNS_SERVERS=127.0.0.1  
  volumes:  
    - ./rabbitmq:/var/lib/rabbitmq  
  ports:  
    - 15672:15672
```

Disk space		Up
333 GiB		0%
mark 2.0 GiB low watermark		

Запуск в докере

Самый полный запуск



```
rabbitmq:  
  image: rabbitmq:3.10.7-management  
  hostname: rabbitmq  
  restart: always  
  environment:  
    - RABBITMQ_DEFAULT_USER=rmuser  
    - RABBITMQ_DEFAULT_PASS=rmpassword  
    - RABBITMQ_SERVER_ADDITIONAL_ERL_ARGS=-rabbit log_levels [{connection,error},{default,error}]  
  disk_free_limit 2147483648  
  volumes:  
    - ./rabbitmq:/var/lib/rabbitmq  
  ports:  
    - 15672:15672  
    - 5672:5672
```

Открываем консоль

Будем запускать **RabbitMQ**

Практика

- Запустить RabbitMQ в **докере** с сохранением стейта
- Настроить порог алерта на **свободное место** и уровень логирования
- Создать **очереди, эксчейнджи, биндинги** между ними
- **Опубликовать сообщения** в разные эксчейнджи
- **Перезагрузить** RabbitMQ
- Проверить что **сообщения успешно сохранилось в очередях**

Конец

А кто слушал - молодец

