RabbitMQ

Durée 3 jour(s) (RABBITMQ-03)

Mettre en place simplement une architecture orientée messages fiable et performante avec RabbitMQ

Formation officielle

Description

Ce cours intensif de 3 jours couvre l'installation, la configuration et le développement d'applications orientées message avec RabbitMQ. Le cours commence par l'installation et la configuration de RabbitMQ. Il se poursuit en traitant du développement d'applications avec l'API Java. Le cours traite aussi des sujets avancés comme le clustering pour la montée en charge et la haute-disponibilité, ainsi que le monitoring d'un cluster RabbitMQ. Des travaux pratiques accompagnent chacun des modules.

Objectifs

- Installer et configurer RabbitMQ
- Activer et utiliser des plugins comme la console de management web
- Implémenter des applications de messaging en Java
- Monter un cluster RabbitMQ
- Choisir une stratégie de haute disponibilité et la mettre en pratique
- Paramétrer et optimiser RabbitMQ pour obtenir de meilleures performances
- Sécuriser RabbitMQ
- Monitorer RabbitMQ

Public

- Développeurs
- Architectes
- Administrateurs

Prérequis

- Connaissance de Java ou de tout autre langage de programmation généraliste.

Répartition

50% Théorie, 50% Pratique

Programme

Introduction au messaging et à AMQP

- Avantages du messaging et des systèmes asynchrones
- Pourquoi Java Message Service (JMS) n'est pas suffisant
- Le modèle Advanced Message Queuing Protocol (AMQP)
- Différences entre AMQP et JMS

Présentation de RabbitMQ

- Description et principales fonctionnalités
- Installation, structure des répertoires, configuration
- Persistance avec la base de données Mnesia
- Console de management web
- Architecture multi-tenant avec les hôtes virtuelles
- Journalisation avec le firehose tracer

Développement et intégration

- Bindings clients (Java, C#, Python, Ruby, etc)
- Focus sur le binding Java
- Abstractions de plus haut niveau (Spring AMQP, Pika)
- Routage AMQP avec exchanges et queues
- Patterns de messaging

Fiabilisation des applications de messaging

- Durabilité niveau queue, exchange et message
- Transactions avec AMQP et JMS
- Transaction et acquittement
- Dead lettering
- Bonnes pratiques pour fiabiliser les flots de messages

Clustering

- Architecture orientée message scalable avec le clustering
- Configuration d'un cluster
- Nœuds de types Disk et RAM
- Administration d'un cluster
- Load balancing

Plugins

- Authentification avec LDAP
- Exposition d'AMQP via le protocol STOMP
- Intégration de brokers différents avec Shovel
- Fédération d'instances RabbitMQ à travers un WAN

Haute disponibilité

- Gestion du crash d'un nœud
- « Mirrored queues »
- Synchronisation entre nœuds esclaves
- Failover coté client

Performances

- Impacts de la configuration et du code client sur les performances
- Contrôle du débit par la mémoire
- Bonnes pratiques

SpringAMQP

- AMQP template
- Configuration des ressources AMQP avec Spring
- Envoi et réception de messages

Sécurité

- Pourquoi sécuriser la communication AMQP
- Gestion des utilisateurs
- Sécuriser RabbitMQ au niveau protocole (SSL/SASL)
- Mise en place des permissions avec des hôtes virtuelles

Monitoring

- API de management
- Intégration avec les outils de supervision
- Métriques à monitorer