

## Formation RabbitMQ

<b>Durée :</b>	3 jours
<b>Public :</b>	Développeurs Java EE
<b>Pré-requis :</b>	Maîtrise de la programmation objet en Java - Notions en Java EE
<b>Objectifs :</b>	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 ; paramétrer et optimiser la solution - Sécuriser et monitorer RabbitMQ
<b>Sanction :</b>	Attestation de fin de stage mentionnant le résultat des acquis
<b>Taux de retour à l'emploi:</b>	Aucune donnée disponible
<b>Référence:</b>	JAV100902-F
<b>Note de satisfaction des participants:</b>	Pas de données disponibles

### Découvrir RabbitMQ

MoM : présentation, apports  
Protocoles et APIs Java : AMQP, JMS  
RabbitMQ : description, fonctionnalités  
Procédure d'installation, configuration  
Gestion des données : Mnesia  
Console de management web  
Architecture multi-tenant avec les hôtes virtuelles  
Gestion des logs  
**Atelier : installation et configuration de RabbitMQ – Accès à la console web**

### Interagir avec RabbitMQ depuis Java

Bindings clients  
Bibliothèques pour simplifier l'accès (Spring AMQP, Pika)  
Routage AMQP  
Patterns de messaging  
**Atelier : Implémentation d'un envoi de messages et traitement**

### Fiabiliser les flots de messages

Durabilité  
Gestion des transactions  
Dead lettering  
**Atelier : fiabilisation de flots de messages.**

## **Mettre en place un Cluster**

Architecture et configuration d'un cluster

Nœuds de types Disk et RAM

Tâches d'administration

Répartition de charges

**Atelier : configuration d'un cluster RabbitMQ**

## **Utiliser des plugins**

Connecteur LDAP

Protocole STOMP

Shovel

Utilisation d'un WAN

**Atelier : utilisation de multiples plugins pour RabbitMQ**

## **Mettre en place une stratégie de Haute disponibilité**

Stratégies disponibles et critères de choix

Gestion du crash d'un nœud

Création de miroirs de files d'attente

Synchronisation entre nœuds esclaves

Reprise après incident

**Atelier : mise en place d'une stratégie de haute disponibilité**

## **Optimiser les performances de la solution**

Bonnes pratiques de configuration

Optimisation du code

## **Utiliser Spring AMQP**

Rappels sur l'apport de Spring Core

Apports de Spring AMQP

Configuration des ressources

Gestion de l'envoi et de la réception de messages

**Atelier : implémentation d'un envoi et de réception de messages avec Spring AMQP.**

## **Sécuriser la solution de messaging**

Gestion de la communication AMQP

Utilisateurs et rôles, gestion des permissions

Protocole de transport

**Atelier : configuration de la sécurité côté code/RabbitMQ.**

## **Surveiller la solution**

Choix des métriques à surveiller

API de gestion

Couplage avec des outils de supervision

**Atelier : monitoring de RabbitMQ et couplage avec des outils de supervision**