



IBM MQ Native HA

Une production : Demey Consulting

Ce document a été préparé à partir de différentes sources d'information publiquement disponibles.

Tous les efforts ont été faits pour rendre ce document complet et exact. Demey Consulting décline par avance toute responsabilité quant aux erreurs et omissions présentes dans ce support. Seuls les documents de référence IBM font foi quant aux capacités et au paramétrage des produits cités.

Version 1.20 – Octobre 2025



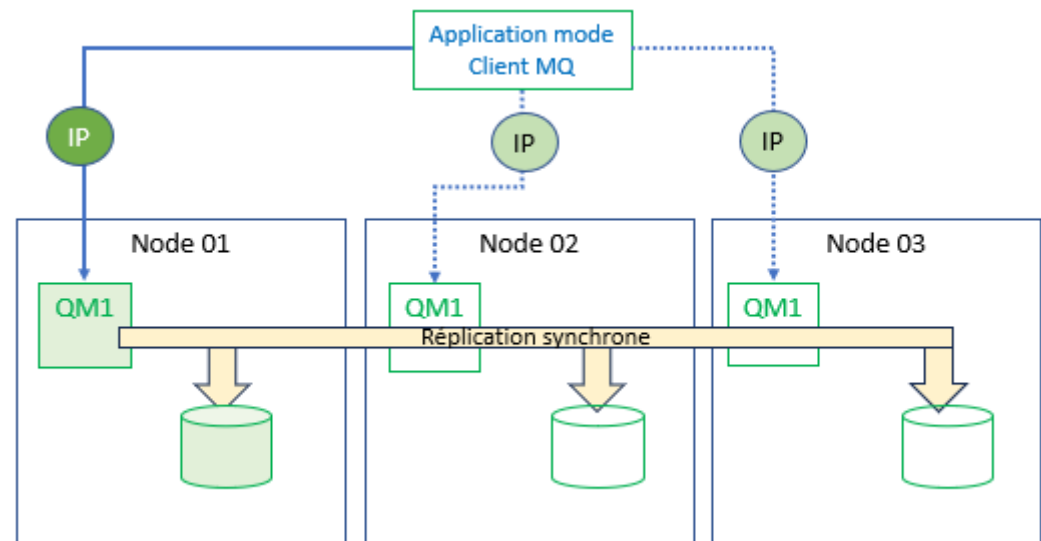
Spécificités de Native HA – version MQ 9.4.4

- Avant : containers uniquement
- MQ 944 : Disponible aussi sur Linux
 - pour l'instant
(pour info : les binaires NHA sont dans le package Linux depuis MQ 942 ...)
- Fonctionnement identique à celle de l'offre OpenShift/Kubernetes sur containers RHEL.
- Simple à installer et à exploiter
 - pas version de Linux spécifique comme RHEL
 - pas de DRBD ou PaceMaker
 - pas d'adhérence sur les versions de kernel



Native HA Linux - Schéma de principe

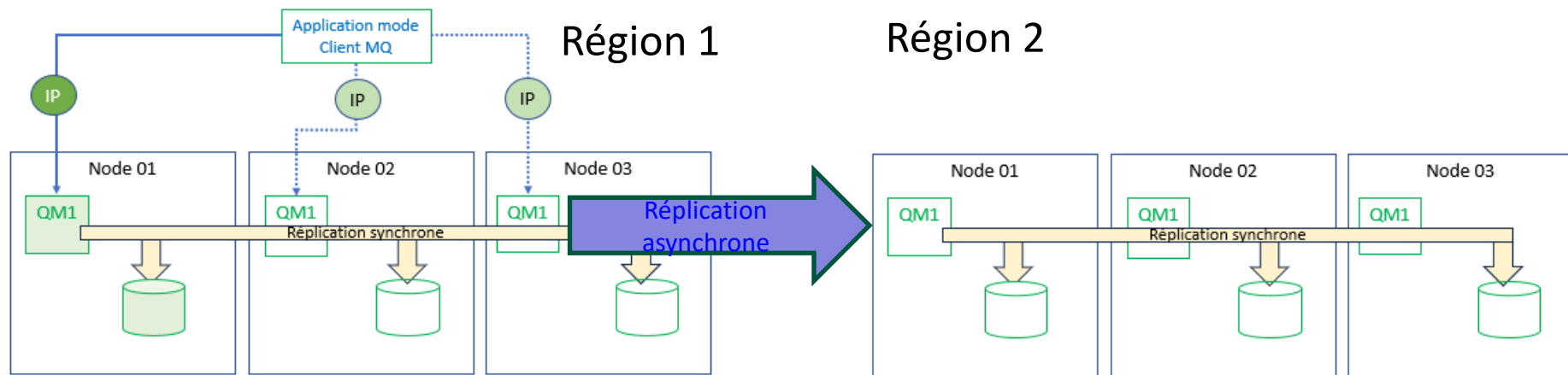
- Configuration à 3 nœuds, chacun dans son AZ
- La réplication des logs entre les nœuds est de type synchrone, gérée directement par les QM
- Elle utilise la technologie RAFT
- Aucune dépendance externe





Native HA - Cross Region Replication

VM Linux



- Configuration à 2 x 3 nœuds sur 2 régions
- Réplication asynchrone, RPO = 0 sur une bascule contrôlée, faible sur une bascule à chaud

<http://ibm.biz/mq-crr-materials>



Différences avec la version container

- Pas de MQ Operator donc :
 - Création manuelle des QM
 - Mise à jour manuelle de la configuration (.ini, .mqsc)
 - Pas de déploiement automatique
 - Pas de surveillance ni monitoring
 - Pas d'IP flottante



Mise en œuvre d'une configuration Native HA

- Objectif
 - Décrire rapidement les étapes d'installation et de configuration d'un environnement Native HA
 - Environnement VM Linux
 - Version 3 nœuds



Installation

- Approvisionner 3 VM Linux identiques
 - Distribution compatible avec IBM MQ 9.4.x
 - Noter les IP, ouvrir les flux
- Installer IBM MQ 944+ avec les valeurs par défaut
 - Minimum : runtime, server, gskit, samples
- Sur chaque serveur, créer le même Queue Manager
 - avec le paramètre **-lr nom_instance**
 - avec les mêmes valeurs pour les logs
- Exemple :
`crtmqm -lr mqha-1 -lf 8192 -lp 10 -ls 10 DCNHA01`



Configuration

- Créer un ou plusieurs scripts MQSC pour configurer le Queue Manager
 - Exemple : DCNHA01_base.mqsc, setup_events.mqsc, setup_mon_stats.mqsc
- Injecter ces scripts dans chaque Queue Manager
- Mettre à jour le qm.ini avec les IP des 3 noeuds NHA
- Evaluer :
 - La possibilité de faire un crtmqm -ic -ii
`crtmqm -lr mqha-1 -ic /var/mqm/autoconf/ -ii /var/mqm/autoconf/ DCNHA01`
 - L'ajout d'un bloc AutoConfig dans le qm.ini
 - Le clonage du premier serveur pour créer les deux autres



Exemple de qm.ini

NativeHALocalInstance:
Name=mqha-1

Ajouté par le crtmqm avec le nom d'instance spécifié au paramètre -lr

NativeHAInstance:
Name=mqha-1
ReplicationAddress=192.168.88.129(9414)

NativeHAInstance:
Name=mqha-2
ReplicationAddress=192.168.88.130(9414)

NativeHAInstance:
Name=mqha-3
ReplicationAddress=192.168.88.131(9414)

A ajouter dans le qm.ini de chaque instance

Adresse + port de réplication



Redémarrage auto des QM

- Script mqmonitor.py fournit dans /opt/mqm/samp
 - installation comme service Linux
- Démarre le QM après le boot du serveur
- Re-démarre le QM au bout de 10 secondes après un incident

MQ Monitor Service

positional arguments:

qmgr

Name of queue manager to monitor

optional arguments:

-h, --help

show this help message and exit

```
--start
```

Start monitoring queue manager

```
--stop
```

Stop queue manager and end monitoring

```
--mqm-path MQM_PATH
```

MQ installation path (default /opt/mqm)

```
--health-interval HEALTH_INTERVAL  Healthcheck interval in seconds (default 10)
```



Statut du QM Native HA

dspmq

QMNAME(DCNHA01) STATUS(En cours de fonctionnement)

dspmq -x

QMNAME(DCNHA01) STATUS(En cours de fonctionnement) INSTANCE(Rocky01) MODE(Actif)

dspmq -o nativeha

– sur actif

QMNAME(DCNHA01) ROLE(Actif) INSTANCE(mqha-1) INSYNC(oui) QUORUM(3/3) GRPLSN(<0:0:2918:2269>)
GRPNAME(DCMQHA01) GRPROLE(Groupe actif)

– sur replica 2

QMNAME(DCNHA01) ROLE(Réplique) INSTANCE(mqha-2) INSYNC(oui) QUORUM(3/3) GRPLSN(<0:0:2918:2269>)
GRPNAME(DCMQHA01) GRPROLE(Groupe actif)

– sur replica 3

QMNAME(DCNHA01) ROLE(Réplique) INSTANCE(mqha-3) INSYNC(oui) QUORUM(3/3) GRPLSN(<0:0:2918:2269>)
GRPNAME(DCMQHA01) GRPROLE(Groupe actif)



Conclusion

- Native HA : disponible aujourd'hui sous Linux
- HA à 3 nœuds
- Cross-Region Replication à 2 x 3 nœuds
- Pas de dépendances avec du code externe
- Beaucoup moins de contraintes que RDQM



Exemple Native HA avec 3 x Raspberry Py Zero 2 W





Merci de votre attention



La formation avec Demey Consulting



- Un organisme de formation déclaré et « Datadocké »
- Un catalogue de modules sur WebSphere Application Server et IBM MQ
- Plus de 50 modules MQ disponibles (1500 slides)
- Des supports de cours totalement francisés et au dernier niveau technique (MQ version 9.4.4)
- Des travaux pratiques sur Windows, Linux et IBM i avec les corrigés
- Des filières prédéfinies de 2 à 5 jours, ou à la carte.

<https://demey-consulting.fr/formationMQ>