

# SQL Server 2017 sur Linux

Martin Turcotte,  
Architecte de solution infonuagique  
Plateforme de données

# SQL SERVER 2017

MEILLEURE PERFORMANCE ET SÉCURITÉ DE L'INDUSTRIE  
MAINTENANT SUR LINUX ET DOCKER

## Choix de plateforme et de langage



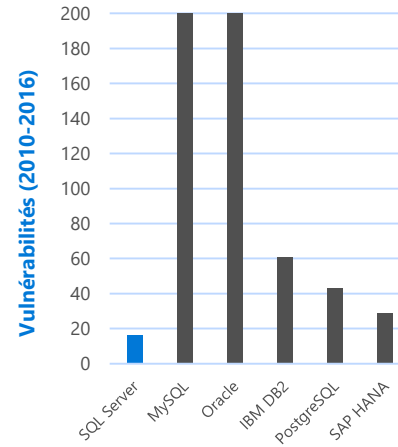
T-SQL	PHP
Java	Node.js
C/C++	Python
C#/VB.NET	Ruby

## Meilleure performance de l'industrie

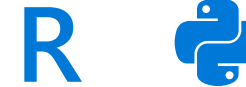


- #1 TPC-H performance 1TB, 10TB, 30TB
- #1 TPC-E performance
- #1 prix/performance

## Meilleur historique de sécurité 7 années consécutives

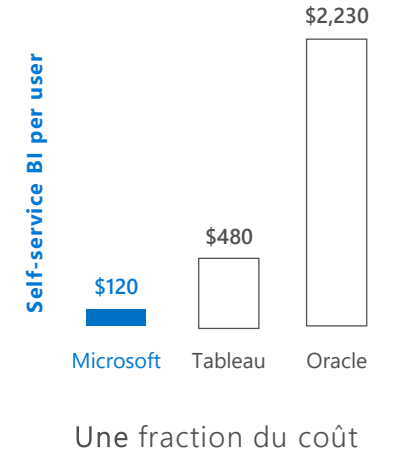


## Seul SGBD commercial offrant IA embarquée



R et Python + en-mémoire à une échelle massive  
Pointage natif en T-SQL

## BI mobile de bout-en-bout sur n'importe quel appareil



En-mémoire pour toutes les charges de travail



Infonuagique privé

Plateforme de données la plus uniforme



Infonuagique



# GESTION DES DONNÉES FLEXIBLE ET FIABLE

## SQL Server sur la plateforme de votre choix

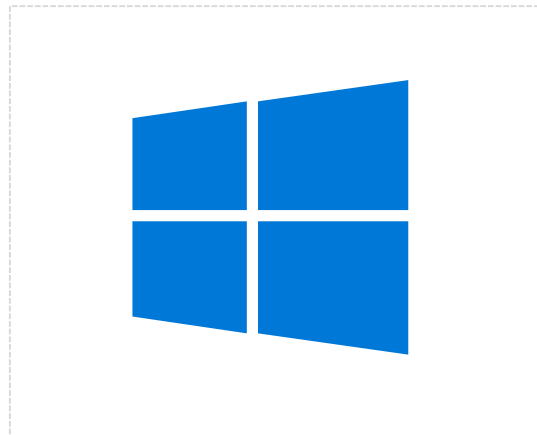
Support pour RedHat Enterprise Linux (RHEL),  
Ubuntu, et SUSE Enterprise Linux (SLES)

Conteneurs Dockers Linux et Windows

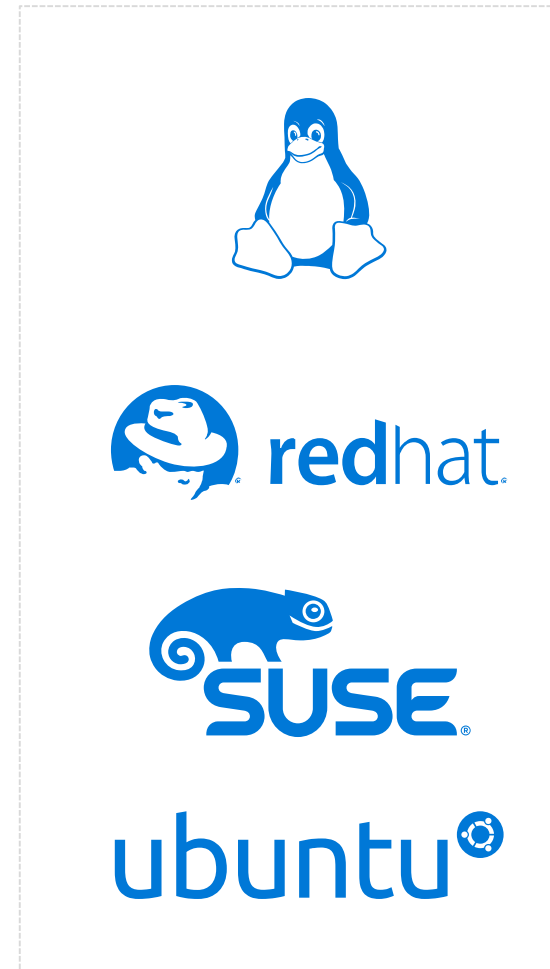
Windows Server / Windows 10

Installation par Package: Yum Install, Apt-Get, et  
Zypper

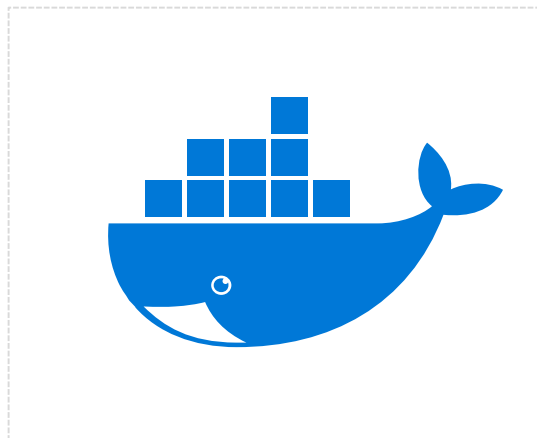
Windows



Linux



Conteneurs Linux/Windows

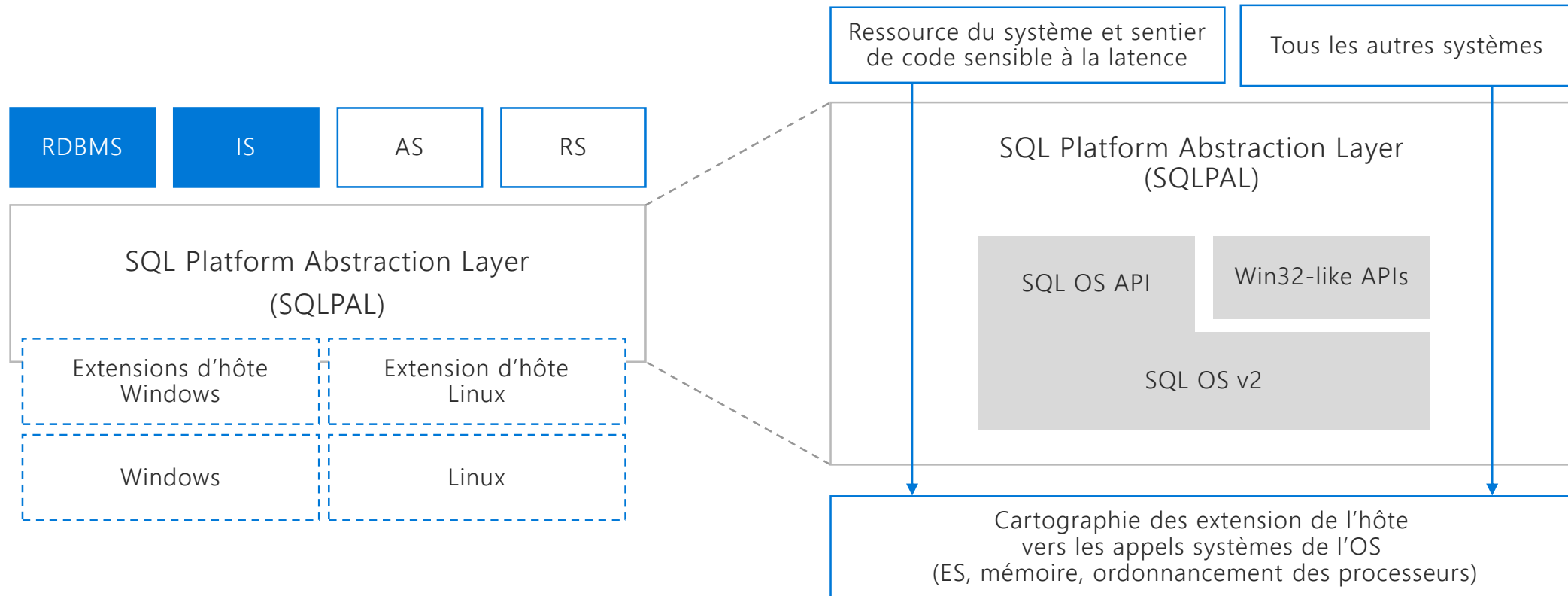


# CE QUI EST DANS SQL SERVER SOUS LINUX

		Windows	Linux
<b>Éditions</b>	Developer, Express, Web, Standard, Enterprise	●	●
<b>Services</b>	Database Engine, Integration Services	●	●
	ML Services, Analysis Services, Reporting Services, MDS, DQS	●	
<b>Performance critique de la mission</b>	Nombre maximum de coeurs de processeur	Unlimited	Unlimited
	Mémoire maximale utilisée par instance	12 TB	12 TB
	Taille maximale de la base de données	524 PB	524 PB
	OLTP de base (OLTP de base en mémoire, analyse opérationnelle de base)	●	●
	OLTP avancé (OLTP en mémoire avancée, analyse opérationnelle avancée, traitement des requêtes adaptatives)	●	●
	Haute disponibilité de base (basculement de base de données unique à 2 nœuds, secondaire non lisible)	●	●
	Haute disponibilité avancée (toujours sur multi-nœud, basculement Multi-DB, répliques secondaires lisibles)	●	●
<b>Sécurité</b>	Sécurité de base (Journalisation de base, sécurité des rangées, Masquage dynamique des données, Always Encrypted)	●	●
	Sécurité avancée (Transparent Data Encryption)	●	●
<b>Entreposage de données</b>	PolyBase	●	
	Entrepôt de données / comptoirs de données de base (ColumnStore en-mémoire de base, Partitionnement, Compression)	●	●
	Entrepôt de données avancé (ColumnStore en-mémoire avancé)	●	●
	Intégration avancée des données (Groupement par logique floue et table de "look ups")	●	●
<b>Outils</b>	Écosystème Windows : Outils de gestion et de développement complets (SSMS & SSDT), Outils en ligne de commande	●	●
	Écosystème Linux / OS X / Windows : Outils DEV (VS Code), Outils Admin de BDs, Outils en ligne de commande	●	●
<b>Développeur</b>	Programmabilité (T-SQL, CLR, Types de données, JSON, Graph)	●	●
	Intégration au système de fichiers de Windows - FileTable	●	
<b>Intelligence d'affaire et analytique avancée</b>	Intelligence d'affaire corporative (Modèles multi-dimensionnels OLAP, Modèles tabulaires de base)	●	
	Machine Learning Services (Intégration avec R et Python)	●	
<b>Infonuagique hybride</b>	Stretch Database	●	

# SQL SERVER SUR LINUX:

MÊME QUE SQL SERVER SUR WINDOWS GRÂCE À LA COUCHE D'ABSTRACTION



# EXPÉRIENCE UTILISATEUR NATIVE SUR LINUX

## Processus d'installation standard



Installation basée sur des packages à l'aide de yum pour les distributions basées sur Fedora, apt-get pour les distributions basées sur Debian et zypper pour SLES

Processus de mise à jour/mise à niveau des packages existants pour la mise à niveau SQL Server

---

## Outils multi-plateformes



Outils de ligne de commande SQL Server (sqlcmd, BCP) disponibles pour Linux (et bientôt sur MacOS)

Les outils Windows existants tels que SQL Server Management Studio (SSMS), les outils de données SQL Server (SSDT) et le module PowerShell (sqlps) pour gérer SQL Server sous Linux à partir de Windows

Outils multi-plateforme Open source tels SQL Server OPS Studio, MSSQL-CLI

Extension Visual Studio Code pour SQL Server sur OS X, Linux ou Windows

---

## Expérience familière



Service SQL Server s'exécute nativement grâce à "systemd"

Sentiers de fichiers Linux supportés dans les requêtes et scripts T-SQL (definition /changement de sentier, fichier de sauvegarde)

Solution populaire pour la haute disponibilité sous Linux comme Pacemaker et Corosync

# HAUTE DISPONIBILITÉ POUR LES MISSIONS CRITIQUES SUR N'IMPORTE QUELLE PLATEFORME

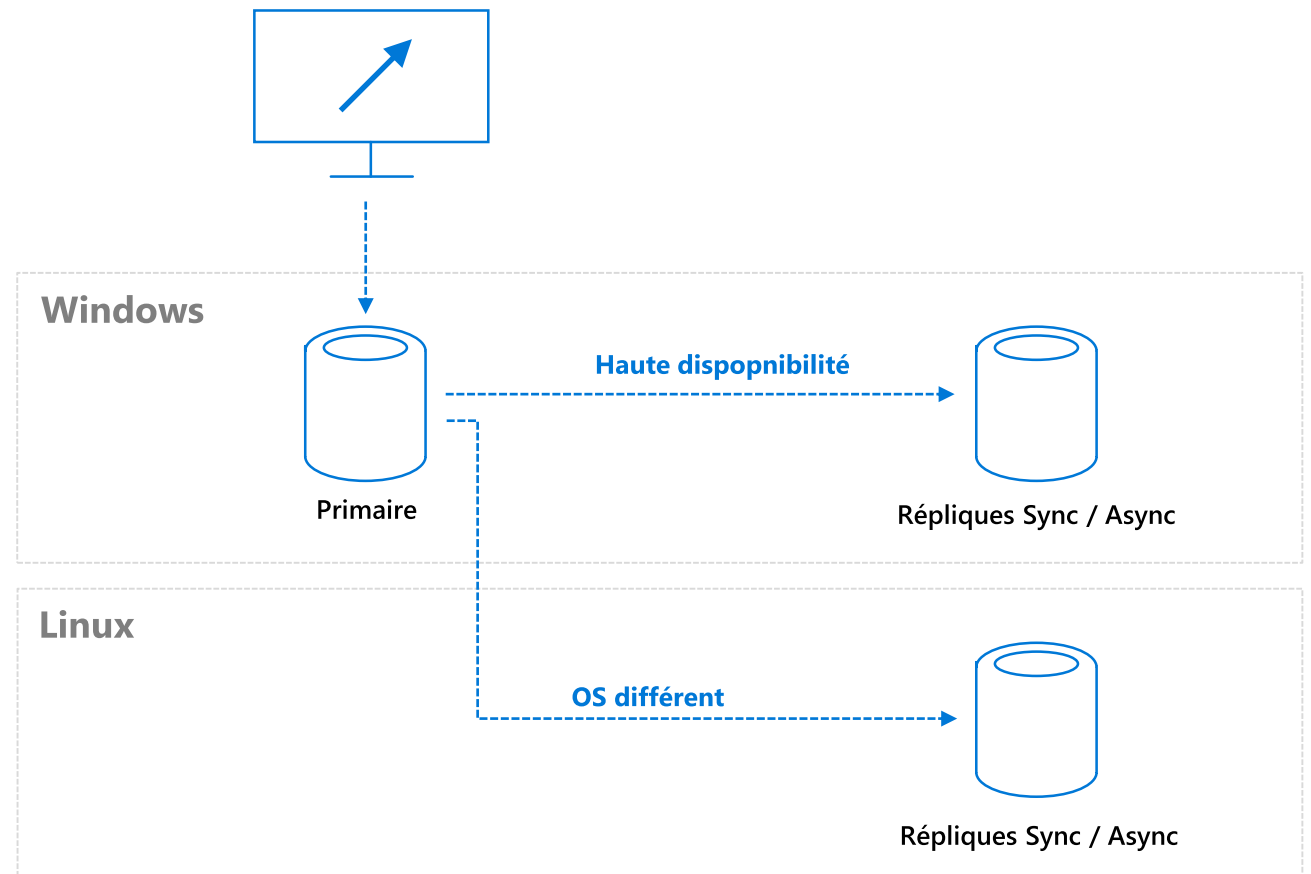
## Fonctionnalités Always On multiplateformes

HA et DR pour Linux et Windows

Support pour les Availability Groups sans WSFC

HA ultime avec redondance au niveau de l'OS  
et migration avec interruption minime

Balancement des charges des répliques  
secondaires lisibles

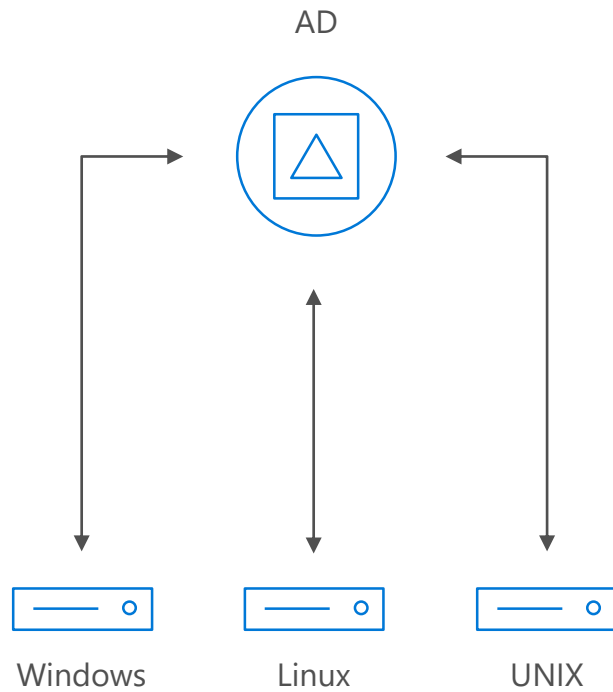




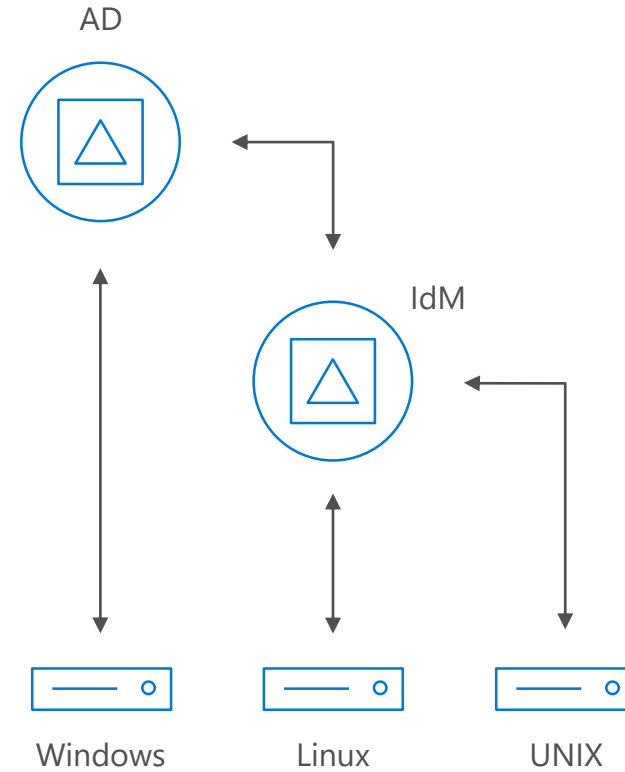
# INTÉGRER RED HAT AVEC MICROSOFT ACTIVE DIRECTORY

## Options d'intégration

### Intégration directe



### Intégration indirecte



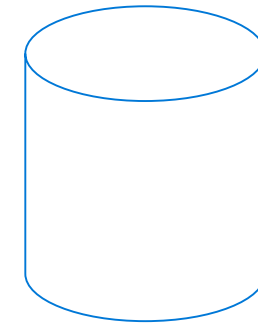
# DISPONIBILITÉ PARTOUT POUR SQL SERVER TOUTE PLATEFORME, TOUT INFONUAGIQUE

Exécutez la meilleure plateforme de données du marché sur n'importe quel environnement

Fonctionne sous Windows, Linux et docker

Fonctionne sur n'importe quel infonuagique, y compris Azure, tiers fournisseurs d'hébergement

Utiliser des licences sur site dans l'infonuagique avec la mobilité de licence, un avantage de l'assurance logicielle



SQL Server 2017



Azure



# AVANTAGES DES PLATEFORMES CONTENEURS POUR SQL SERVER

Temps de déploiement plus rapide

Encapsulation des dépendances avec SQL Server

Automatisation des processus CI/CD - Traiter SQL Server comme une dépendance

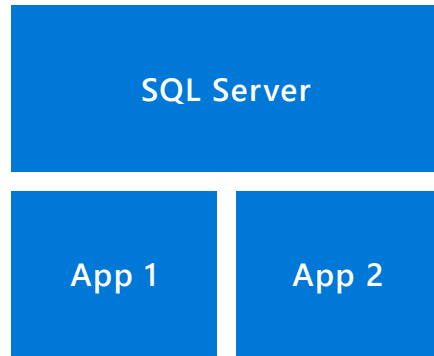
Déployer exactement ce qu'il y a dans DEV/TEST

Multiplateforme - Développer sur Windows / Linux / OS X; Déployer sur Linux

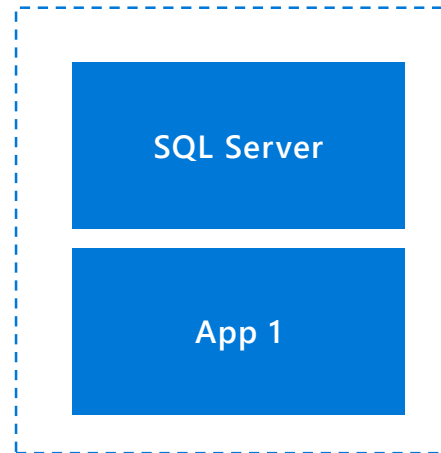
Architectures de micro-services

Densité plus élevée, moins d'espace disque requis

# MODÈLES DE DÉPLOIEMENT D'APPLICATION À L'AIDE DE CONTENEURS



SQL Server centralisé



Composition sur Docker



App monolithique ou  
microservice

# DÉMO #1

---

## 1) Installation de SQL Server 2017 sur RedHat Enterprise Linux (via SSH)

```
sudo yum install -y mssql-server
```

## 2) Configuration de SQL Server 2017 (via SSH)

```
sudo /opt/mssql/bin/mssql-conf setup
```

## 3) Restauration d'une copie de BD avec SQL OPS Studio (Windows 10 local)

```
RESTORE DATABASE WideWorldImporters FROM DISK = '/var/opt/mssql/WideWorldImporters-Full.bak' WITH  
MOVE 'WWI_Primary' TO '/var/opt/mssql/data/WideWorldImporters.mdf',  
MOVE 'WWI_UserData' TO '/var/opt/mssql/data/WideWorldImporters_UserData.ndf',  
MOVE 'WWI_Log' TO '/var/opt/mssql/data/WideWorldImporters.ldf',  
MOVE 'WWI_InMemory_Data_1' TO '/var/opt/mssql/data/WideWorldImporters_InMemory_Data_1',  
STATS=5;  
GO
```

## 4) Validation de la BD avec SQL OPS Studio (Windows 10 local)

```
SELECT COUNT(1) FROM Sales.Orderlines;  
SELECT TOP(10) * FROM Sales.Orderlines;  
SELECT DISTINCT(PackageTypeID) FROM Sales.Orderlines; SELECT * FROM Sales.OrderLines WHERE PackageTypeID = 7;
```

# DÉMO #1 – Installation de mssql-server

```
[root@sql2017rhel74 enap_scripts]# cat 11-Config_repo_and_install_mssql.sh
#!/bin/bash
set -x
# Add a repo file for mssql-server
curl -o /etc/yum.repos.d/mssql-server.repo https://packages.microsoft.com/config/rhel/7/mssql-server-2017.repo

# Confirm addition of new repo file
ls -l /etc/yum.repos.d/

# Show content of new repo file
cat /etc/yum.repos.d/mssql-server.repo

# Timestamp before launching installation of mssql-server
date

# Install mssql-server using yum
yum install -y mssql-server

# Timestamp after completing installation of mssql-server
date
[root@sql2017rhel74 enap_scripts]# ./11-Config_repo_and_install_mssql.sh
```

# DÉMO #1 – Configuration de mssql-server

```
=====
Install 1 Package
```

```
Total download size: 168 M
```

```
Installed size: 168 M
```

```
Downloading packages:
```

```
mssql-server-14.0.3026.27-2.x86_64.rpm
```

```
| 168 MB 00:00:03
```

```
Running transaction check
```

```
Running transaction test
```

```
Transaction test succeeded
```

```
Running transaction
```

```
Installing : mssql-server-14.0.3026.27-2.x86_64
```

```
1/1
```

```
+-----+
Please run 'sudo /opt/mssql/bin/mssql-conf setup'
to complete the setup of Microsoft SQL Server
+-----+
```

```
SQL Server needs to be restarted in order to apply this setting. Please run
'systemctl restart mssql-server.service'.
```

```
Verifying : mssql-server-14.0.3026.27-2.x86_64
```

```
1/1
```

```
Installed:
```

```
mssql-server.x86_64 0:14.0.3026.27-2
```

```
Complete!
```

```
+ date
```

```
Thu Jun 14 01:01:48 UTC 2018
```

```
[root@sql2017rhel74 enap_scripts]# c
```

# DÉMO #1 – Restauration d'un backup

```
Do you accept the license terms? [Yes/No]:yes
```

```
Configuring SQL Server...
```

```
ForceFlush is enabled for this instance.
```

```
ForceFlush feature is enabled for log durability.
```

```
Setup has completed successfully. SQL Server is now starting.
```

```
+ systemctl status mssql-server
```

```
• mssql-server.service - Microsoft SQL Server Database Engine
```

```
  Loaded: loaded (/usr/lib/systemd/system/mssql-server.service; enabled; vendor preset: disabled)
```

```
  Active: active (running) since Thu 2018-06-14 01:16:19 UTC; 131ms ago
```

```
  Docs: https://docs.microsoft.com/en-us/sql/linux
```

```
  Main PID: 118165 (sqlservr)
```

```
  CGroup: /system.slice/mssql-server.service
```

```
          └─118165 /opt/mssql/bin/sqlservr
```

```
Jun 14 01:16:19 sql2017rhel74 systemd[1]: Started Microsoft SQL Server Database Engine.
```

```
Jun 14 01:16:19 sql2017rhel74 systemd[1]: Starting Microsoft SQL Server Database Engine...
```

```
[root@sql2017rhel74 enap_scripts]# cat 13A-Restore_backup_and_confirm_db.sh
```

```
#!/bin/bash
```

```
set -x
```

```
# Call sqlcmd to execute sql restore script to restore the database from /var/opt/mssql/backup/WideWorldImporters-Full.bak
```

```
/opt/mssql-tools/bin/sqlcmd -U SA -P $SA_PASSWORD -i ./13B-Restore_backup.sql
```

```
# Call sqlcmd to execute sql script to confirm proper database state for next demo.
```

```
/opt/mssql-tools/bin/sqlcmd -U SA -P $SA_PASSWORD -i ./13C-Confirm_db.sql
```

```
[root@sql2017rhel74 enap_scripts]#
```



# DÉMO #1 – Ouverture du firewall

```
779.256125
```

```
(1 rows affected)
```

```
-----  
-----  
-----  
Microsoft SQL Server 2017 (RTM-CU7) (KB4229789) - 14.0.3026.27 (X64)
```

```
May 10 2018 12:38:11
```

```
Copyright (C) 2017 Microsoft Corporation
```

```
Developer Edition (64-bit) on Linux (Red Hat Enterprise Linux Server 7.5 (Maipo))
```

```
(1 rows affected)
```

```
[root@sql2017rhel74 enap_scripts]# cat 14-Open_firewall.sh
```

```
#!/bin/bash
```

```
set -x
```

```
# Open firewall on port 1433/tcp
```

```
firewall-cmd --zone=public --add-port=1433/tcp --permanent
```

```
# Reload firewall rules
```

```
firewall-cmd --reload
```

```
[root@sql2017rhel74 enap_scripts]# ./14-Open_firewall.sh
```

```
+ firewall-cmd --zone=public --add-port=1433/tcp --permanent
```

# MEILLEURE PERFORMANCE DE L'INDUSTRIE AVEC SQL SERVER 2017

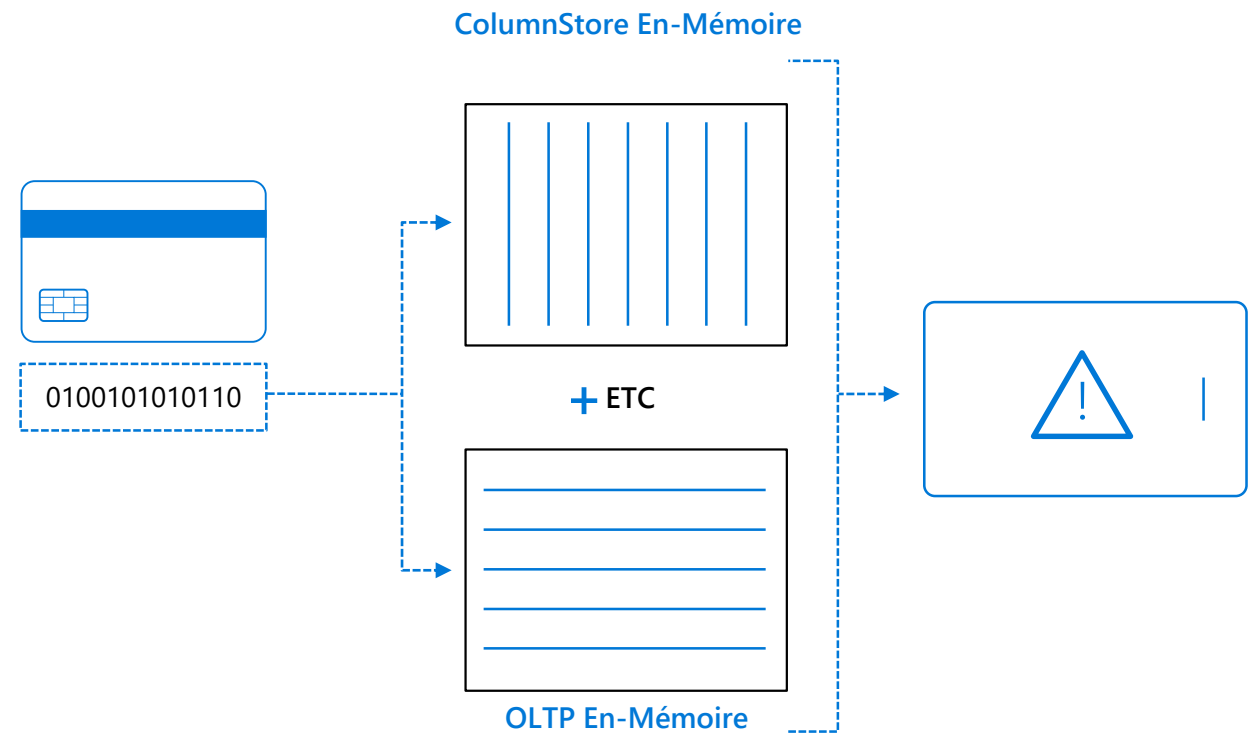
## Analytique opérationnelle en temps-réel

Rapprocher l'analytique des données  
opérationnelles

Améliorer la performance transactionnelle avec  
l'OLTP En-Mémoire

Accélérer l'analytique et réduire les besoins de  
stockage avec le ColumnStore En-Mémoire

Compression des LOB dans l'index ColumnStore



# ENTREPÔT DE DONNÉES À L'ÉCHELLE DES PÉTAOCTETS

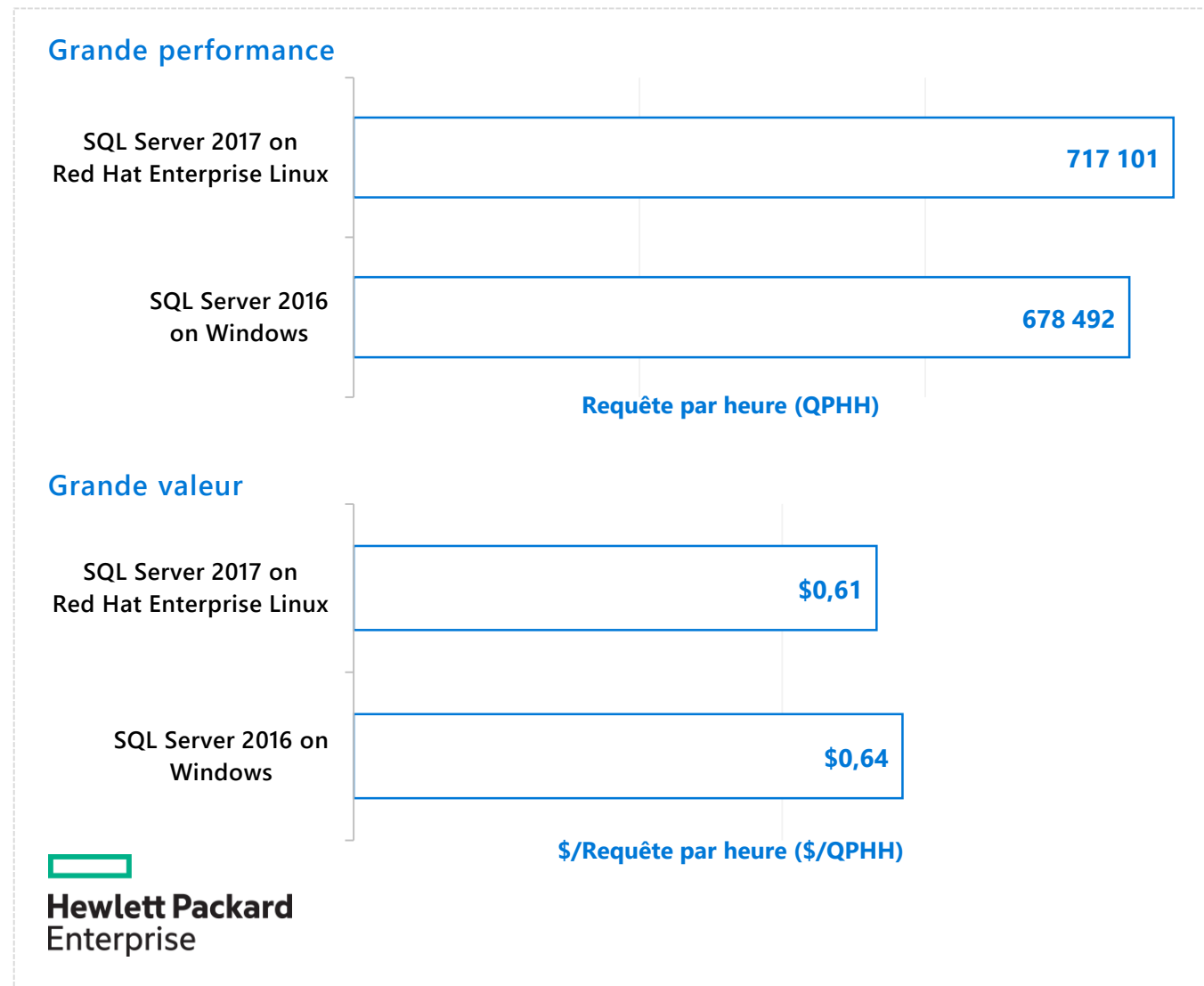
## Traitement efficace des requêtes

Disponible sous Windows et Linux

#1 prix/performance dans les bancs d'essais  
« non-clustered » TPC-H de 1TO, de 10TO et de  
30TO

Architectures de référence provenant des  
principaux partenaires fabricant de matériel

## Record Mondial TPC-H 1TO « non-clustered »



Read the performance brief at [hpe.com/servers/benchmarks](http://hpe.com/servers/benchmarks).

Microsoft and Windows are U.S. registered trademarks of Microsoft Corporation. Red Hat, Red Hat Enterprise Linux, and the Shadowman logo are registered trademarks of Red Hat, Inc. Linux is a registered trademark of Linus Torvalds. Intel and Xeon are trademarks of Intel Corporation in the U.S. and other countries. TPC and TPC-H are trademarks of the Transaction Processing Performance Council. TPC-H results show the HPE ProLiant DL380 Gen9 with a result of 717,101 QphH @ 1000GB and \$0.61/QphH USD with system availability as of 10-19-2017 (results published 04-19-2017; see <http://www.tpc.org/xxxx>); the HPE ProLiant DL380 Gen9 with a result of 678,492 QphH @1000GB and \$0.64/QphH @ 1000GB with system availability as of 07-31-2016 (results published 03-24-2016; see [tpc.org/3320](http://tpc.org/3320)). The TPC believes that comparisons of TPC-H results published with different scale factors are misleading and discourages such comparisons. Please see [tpc.org](http://tpc.org) for up-to-date information. Competitive claims valid as of 04-19-2017.

# TRAITEMENT OPTIMISÉ DES REQUÊTES

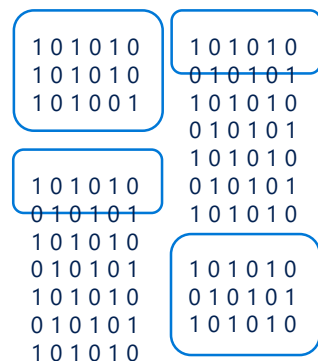
## Amélioration de l'efficacité avec le traitement adaptatif des requêtes

Allocation optimisée de la mémoire pour les requêtes répétables afin d'éviter la sur-allocation de même que la sous-allocation

Ajustement de la stratégie de jointure des données pour les petites et grandes tables afin d'accélérer les jointures

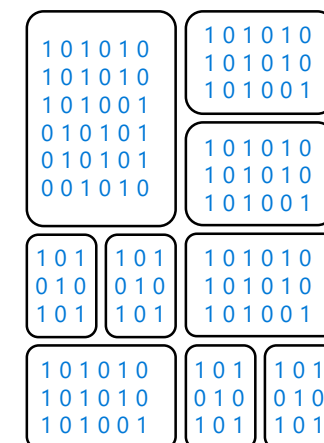
Allocation par lot pour le rétrocontrôle des allocations de mémoire et des jointures adaptatives

Avant

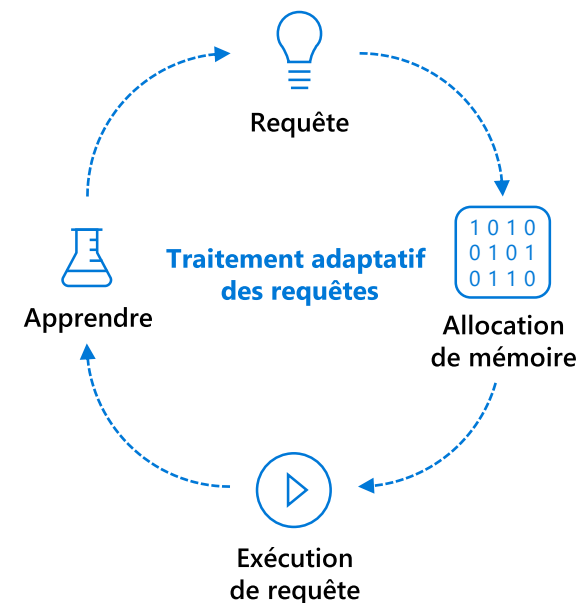


Débordement sur disque

Après



Tout en mémoire



# SÉCURITÉ ÉPROUVÉE SUR TOUTES LES PLATEFORMES

Plusieurs couches de sécurité au  
niveau des disques, des bases de  
données, et des applications

Plateforme ayant connu le moins de vulnérabilités  
dans la base de données de la NIST

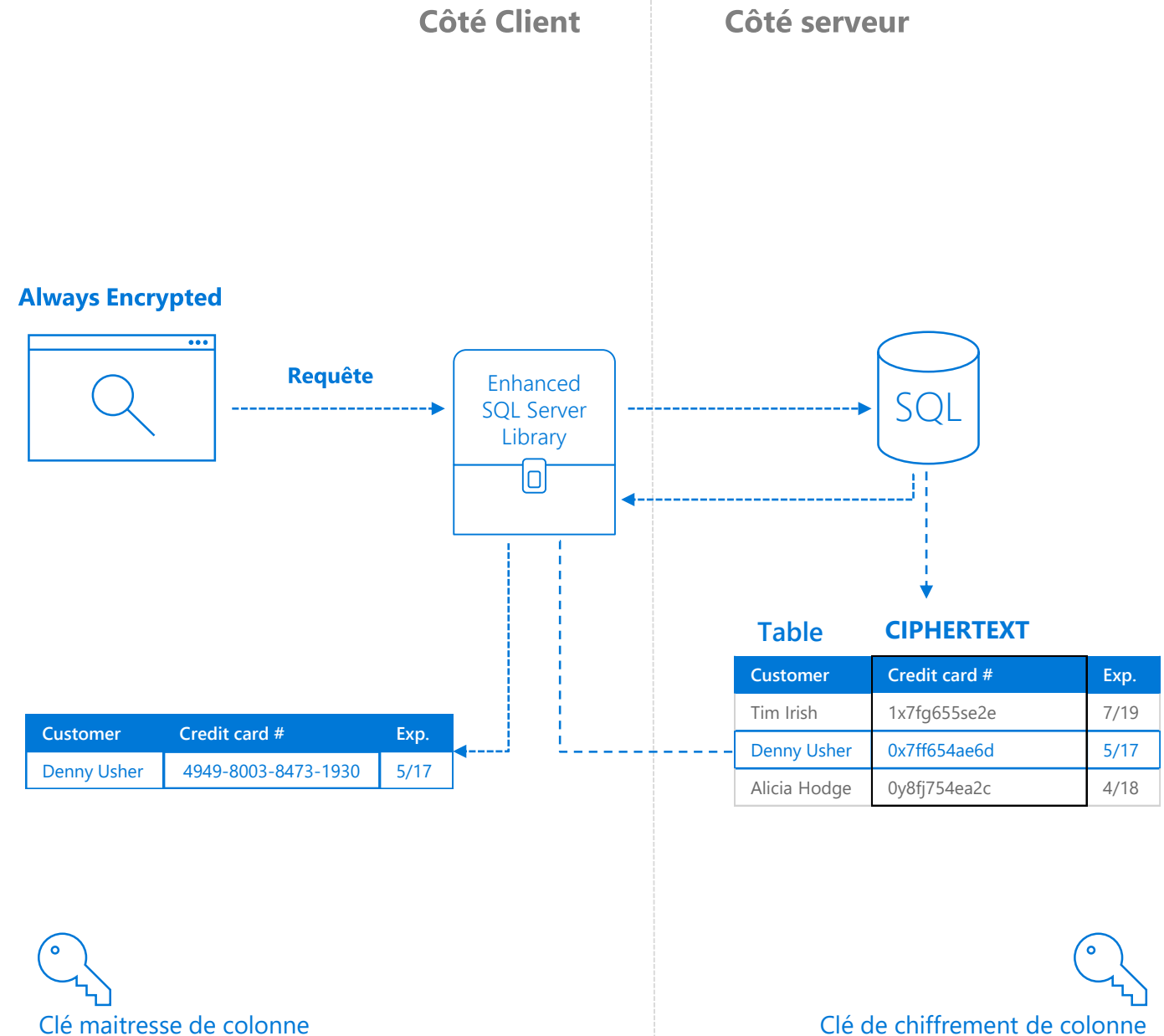
« Always Encrypted » pour le chiffrement des  
données en mouvement et au repos, à l'aide de  
bibliothèques de pilotes à jour

Dissimulation d'information sensible avec  
« Dynamic Data Masking »

Contrôle des accès aux rangées de BDs basé sur  
les caractéristiques de l'utilisateur avec « Row-  
Level Security »

Protection au niveau des fichiers avec  
« Transparent Data Encryption »

➤ Meilleur historique de sécurité au cours des 7 dernières années



# EXTRACTION DE CONNAISSANCE EN PROVENANCE DE DONNÉES DIVERSES

## Nouvelles relations découvertes avec le support pour les données de graphe

Apporter le support pour les données de graphe  
aux données relationnelles

Analyse des données interconnectées et  
génération de meilleures connaissances

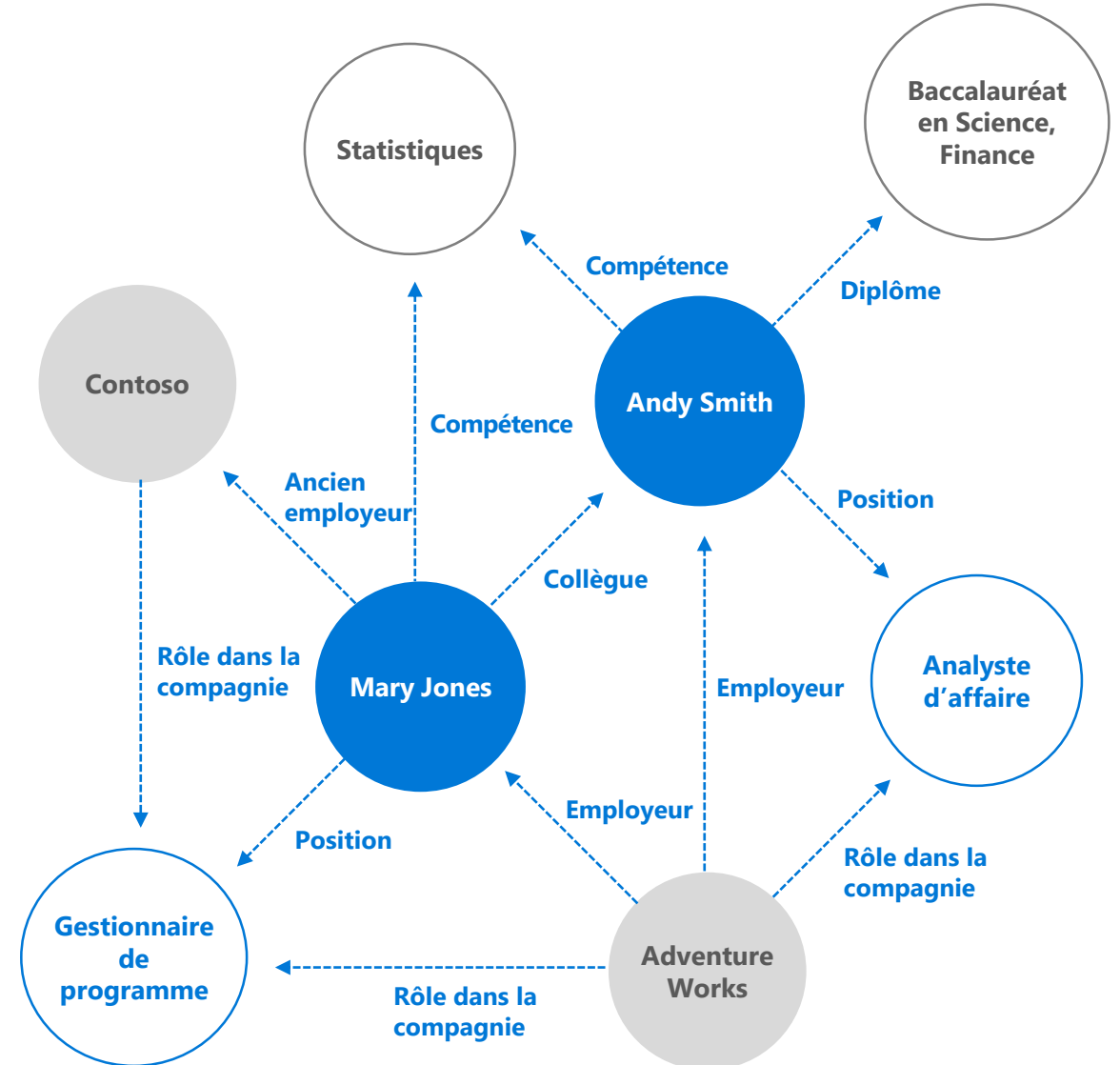
Extensions T-SQL pour supporter l'appariement de  
motif et la traverse de graphes

## Requête dans les mégadonnées

Requête dans les données stockées dans Hadoop  
avec PolyBase

Extraction de la valeur et des connaissances des  
lacs de données à l'aide de Hadoop combiné à  
SQL Server

## Application de réseautage professionnel



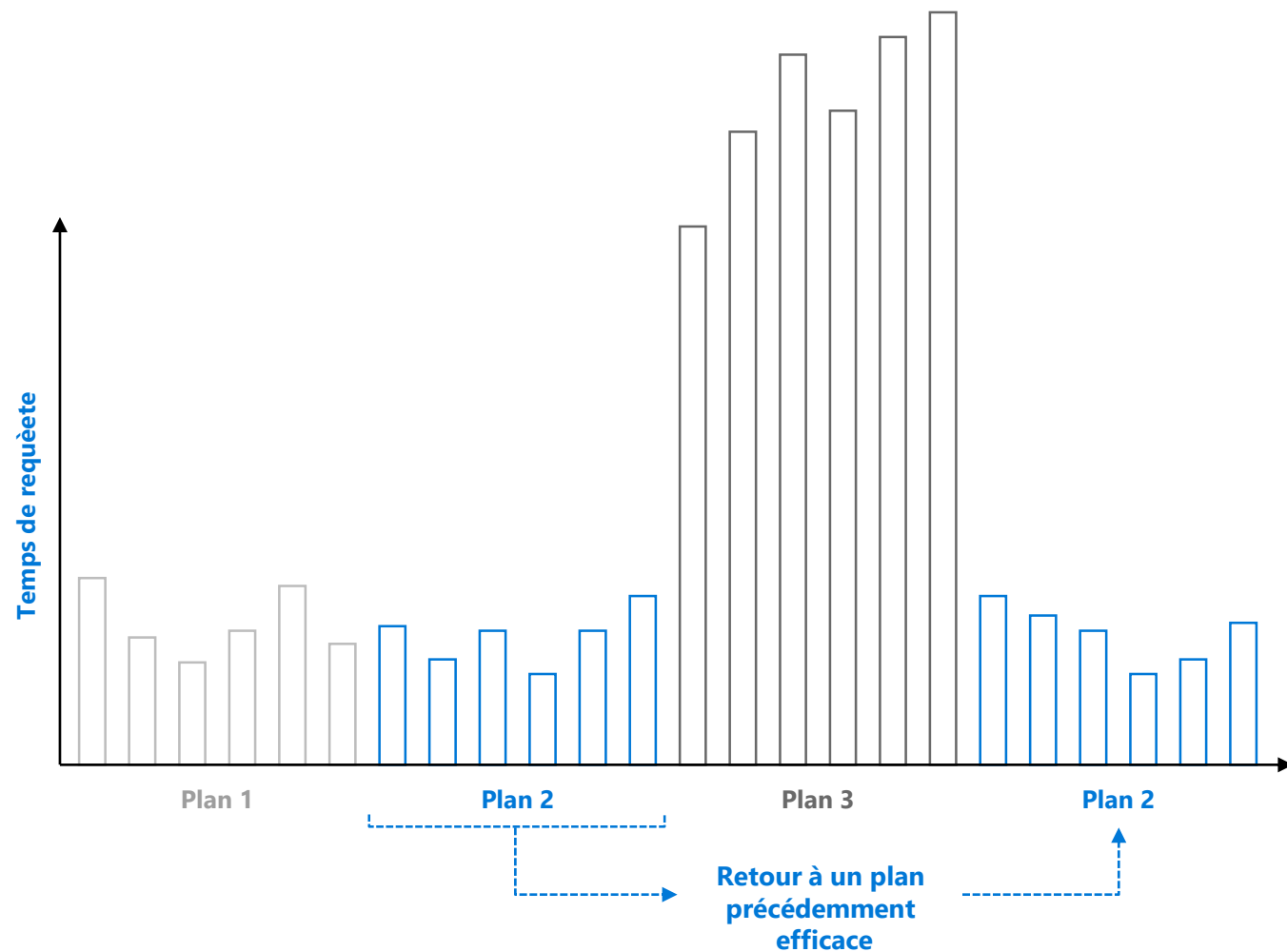
# RÉSOLUTION AUTOMATIQUE DES PROBLÈMES SANS AJUSTEMENT

Meilleure performance avec la  
correction automatique des plans

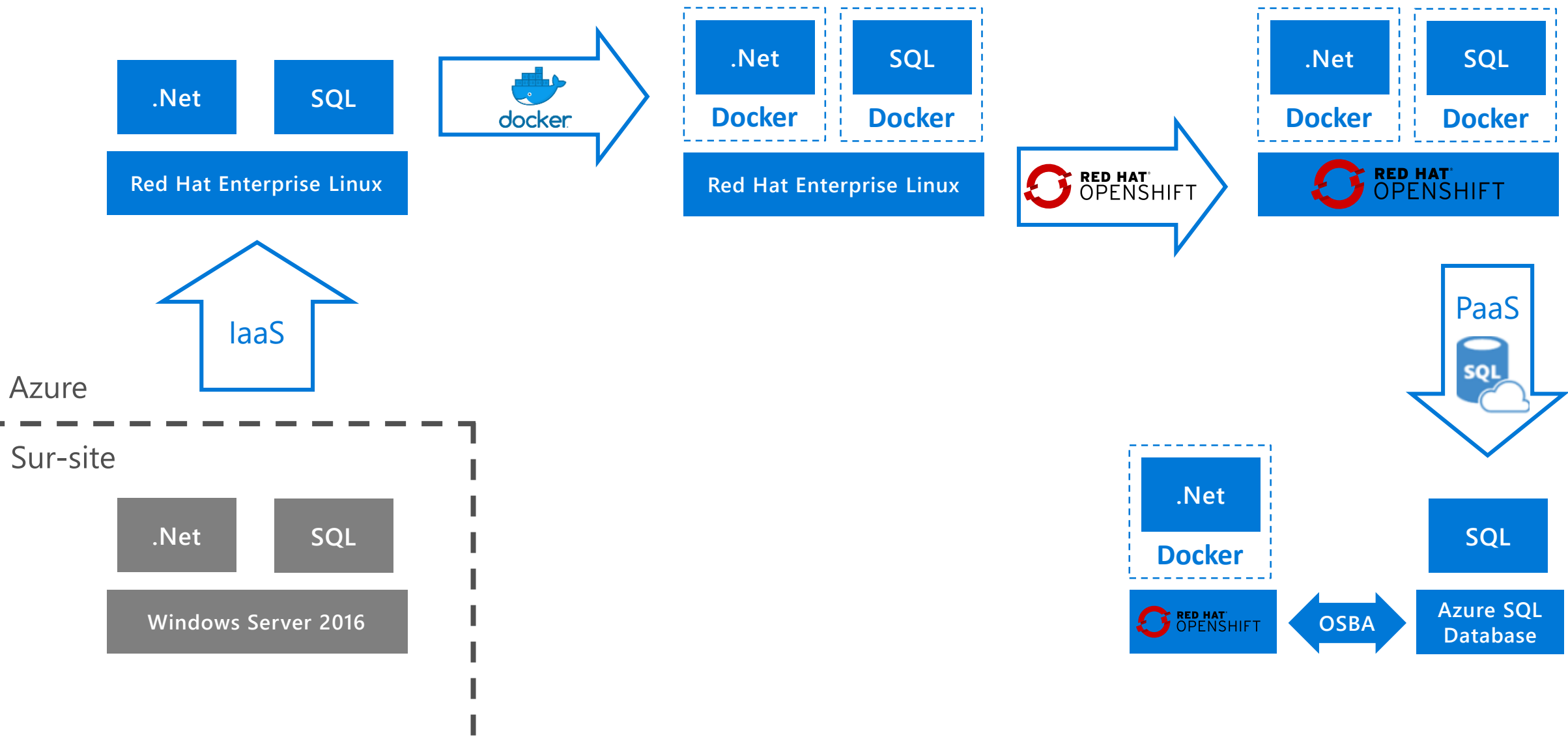
Surveillance et analyse continue de la  
performance des plans

Détection des plans problématiques

Résolution automatique des problèmes de  
performance causée par la régression dans la  
sélection de plan SQL



# Parcours d'un déploiement... vers **OpenShift** et **Azure**





# DÉMO #2

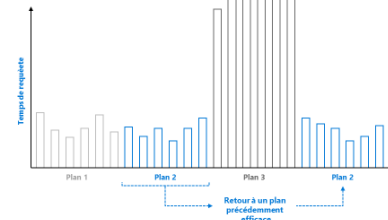
## Ajustement automatique de la performance (« Automatic Tuning »)

SQL Server 2017 sur Linux - "Automatic Tuning"  
Performance transactionnelle



Dégradation artificielle:  Dégradation  
"Automatic Tuning":  OFF  ON

Tableau de bord Web  
DotNet Core 2.0  
(Port 8888)



Database de données  
SQL Server 2017  
(Port 1435)

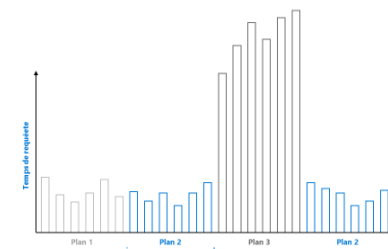
SQL Server 2017 sur Linux - "Automatic Tuning"  
Performance transactionnelle



Dégradation artificielle:  Dégradation  
"Automatic Tuning":  OFF  ON

Tableau de bord Web  
DotNet Core 2.0  
(Port 9999)

Conteneur Docker-Web



Database de données  
SQL Server 2017  
(Port 1436)

Conteneur Docker-SQL

Red Hat Enterprise Linux 7 dans Azure

Phase 1 – Exécution native sur  
Red Hat Enterprise Linux 7.4

Phase 2 – Exécution dans des  
conteneurs Dockers sur  
Red Hat Enterprise Linux 7.4

# DÉMO #2A – Démarrage Web App .Net

```
---[root@sql2017rhel74 enap_scripts]# cat 21-DotNet.sh
#!/bin/bash
set -x

# Change directory to the location of the DotNet Core Web Dashboard App
cd ~/RedHatOpenShiftAndMicrosoftAzureWorkshop/SqlServerAutoTuningDashboard/

# Build, restore and run the DotNet Core Web Dashboard App
dotnet build; dotnet restore; dotnet run
[root@sql2017rhel74 enap_scripts]# ./21-DotNet.sh
```

# DÉMO #2A – SQL 2017 Auto Tuning Dashboard



## Unable to connect



Firefox can't establish a connection to the server at sql2017rhel74:8888.

- The site could be temporarily unavailable or too busy. Try again in a few moments.
- If you are unable to load any pages, check your computer's network connection.
- If your computer or network is protected by a firewall or proxy, make sure that Firefox is permitted to access the Web.

Try Again

# DÉMO #2B – Docker Web App .Net

```
Database 'WideWorldImporters' running the upgrade step from version 867 to version 868.
Database 'WideWorldImporters' running the upgrade step from version 868 to version 869.
RESTORE DATABASE successfully processed 58455 pages in 35.710 seconds (12.788 MB/sec).
Changed database context to 'WideWorldImporters'.

(1 rows affected)

(1 rows affected)
+ firewall-cmd --zone=public --add-port=1436/tcp --permanent
success
+ firewall-cmd --reload
success
[root@sql2017rhel74 enap_scripts]# cat 32-Docker-web.sh
#!/bin/bash
set -x

# Docker run container for web
docker run \
  -e 'ConnectionString_Wwi=Server=10.1.1.4,1436;Database=WideWorldImporters;User Id=SA;Password=SQL2017R0ck5;' \
  -p 9999:80 \
  --name web \
  -d mabenoit/sql-autotune-dashboard:latest

# Open firewall on port 9999/tcp
firewall-cmd --zone=public --add-port=9999/tcp --permanent

# Reload firewall rules
firewall-cmd --reload
[root@sql2017rhel74 enap_scripts]# ./32-Docker-web.sh
```

# DÉMO #2B – Docker du SQL 2017 Auto Tuning



sql2017rhel74:9999



Search

100%



## Unable to connect



Firefox can't establish a connection to the server at localhost:9999.

- The site could be temporarily unavailable or too busy. Try again in a few moments.
- If you are unable to load any pages, check your computer's network connection.
- If your computer or network is protected by a firewall or proxy, make sure that Firefox is permitted to access the Web.

Try Again



Merci!