

# Database

---

## Relational Database Service (RDS)

### Configuration

#### Database Engines

- MySQL
- MariaDB (fork MySQL)
- PostgreSQL
- Oracle
- SQL Server (Microsoft)
- Amazon Aurora (Amazon fork de MySQL)

#### DB Instance types

#### Compute scaling

- Vertical scaling : on augmente les capacités au niveau instance
- Horizontal scaling : on augmente le nombre d'instances

#### Haute disponibilité

#### RDS multi AZ

- En cas de défaillance

#### Read replicas

- En cas d'accès important en lecture fourni un accès en lecture seule via un réplica
- Uniquement pour MySQL, MariaDB et PostgreSQL

#### Backup

#### Automatic

- Automatiquement supprimé après la période de rétention (35 jours max)

## Manual snapshot

- Pas de période de rétention et pas de suppression automatique

## AWS backup

- Utilise le snapshot

## RDS Performance Insights

- AAS : Average Active Sessions
- 

[Haut de page](#)

## DynamoDB

- Bdd relationnelle non SQL
- RCU : Read Capacity Units
- WCU : Write Capacity Units

## Avantages

- Entièrement géré par AWS (Sauvegarde, redondance, mise à jour)
- Pas de datamodel
- Redondant (3 AZ au sein d'un région)
- Rapide

## Inconvénients

- Avec la redondance on peut ne pas obtenir la dernière version d'une donnée dans certains cas
- Pas de requête avancée comme en SQL (Join, ...) pour une question de performance
- 400 ko max par enregistrement
- 20 index primaires max et 5 secondaires par table
- Débit provisionné à l'avance ... à contrôler

## Backup

### Automatic

- Automatiquement supprimé après la période de rétention (35 jours max)

## On demand (manuel)

- Pas de suppression automatique

## DAX DynamoDB Accelerator

- Cache mémoire permettant de passer de temps de réponse en millisecondes à des microsecondes
- De 3 à 9 noeuds (1 primaire et 9 répliques).
- Entité séparé au sein d'un VPC avec un DAX cluster par sous réseau
- DAX client installé sur instance EC2
- Communication via port 8111 vers le DAX cluster (Security Group à configurer)

---

[Haut de page](#)

## Amazon ElastiCache

Cache mémoire pour des données lues fréquemment

### Engines

#### Memcached

key-value store service utilisé comme cache aussi bien qu'en plus d'un data store. Plus de fonctionnalités.

#### Redis

Data store en mémoire. Plus simple et plus performant.

### Composants

#### Node

- network-attached RAM

#### Redis shards

- Groupe de noeuds (jusqu'à 6 noeuds)

## Redis Cluster

- Groupe de 1 à 90 Redis shards

## Memcached cluster

- collection d'un ou plusieurs cache node

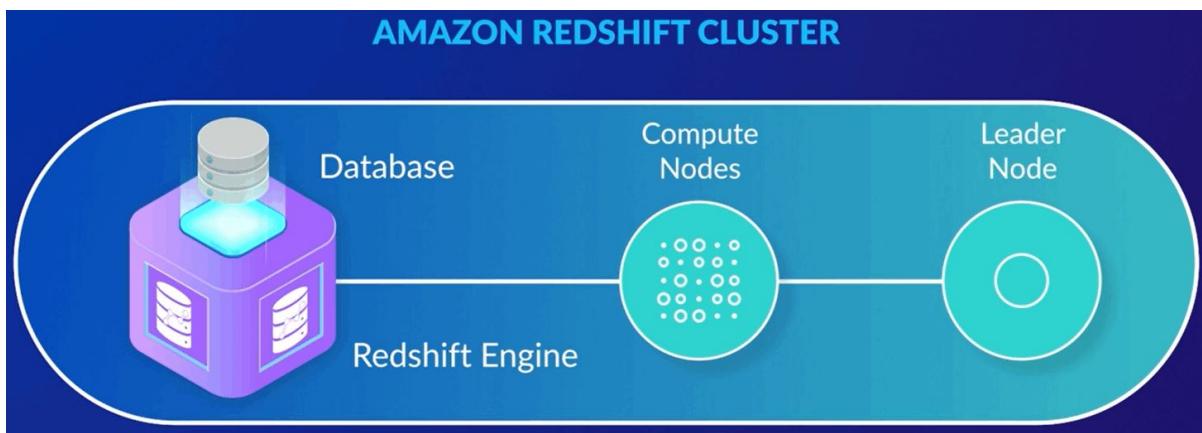
## Utilisation

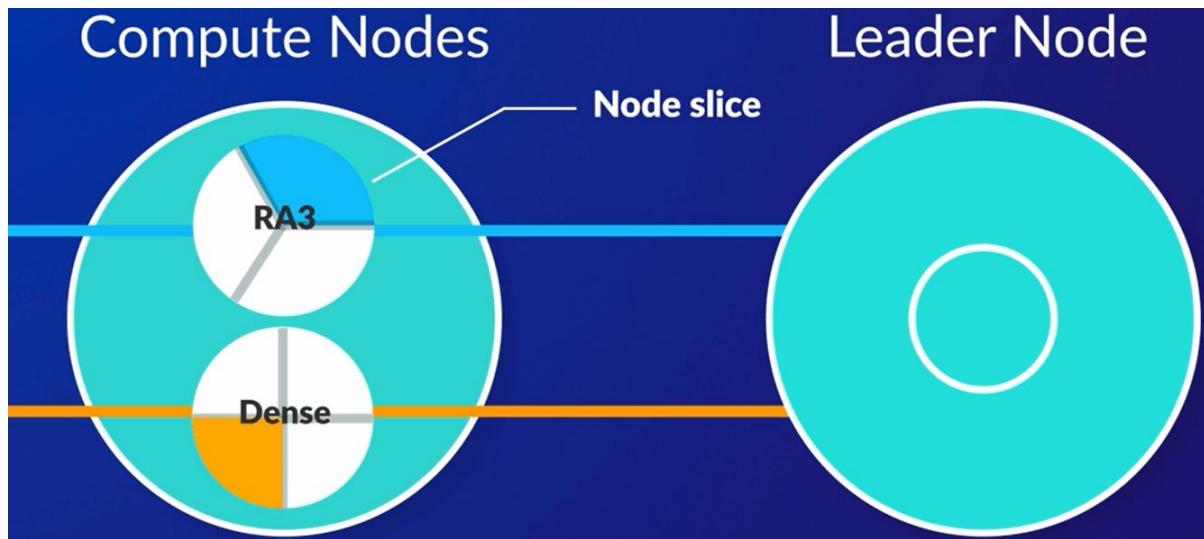
- réseaux sociaux
- jeux en ligne
- analyses temps réel

[Haut de page](#)

## Amazon Redshift

- SQL compatible avec d'autres applications RDBMS
- Basé sur PostgreSQL 8.0.2





[Haut de page](#)

From:

<https://wiki.iot-ac.s.fr/> - **Wiki**



Permanent link:

<https://wiki.iot-ac.s.fr/doku.php?id=all:bibles:aws:presentation:5-database>

Last update: **2024/06/14 11:10**