Pacalet Hugo

Gantelet Mathis

Robin Ducret

Hodes Thomas

Bevliacqua Romain

Administration BDD – Projet Niloc

Sommaire

[Introduction 3](#_Toc139981795)

[Stockage 3](#_Toc139981796)

[Rôle et user 3](#_Toc139981797)

[Sauvegarde 4](#_Toc139981798)

[Service 5](#_Toc139981799)

[Duplication 5](#_Toc139981800)

# Introduction

À la suite du projet Niloc et la mise en place de la base OLTP, nous avons décidé de mettre en place des solutions pour gérer l’administration de la base de données NILOC.

Ce document se divisera en plusieurs parties expliquant et justifiant les différentes solutions qui ont été mise en place.

Il est important de noter que la grande majorité des éléments cités pourront être modifiés plus tard selon les besoins. Des questions devront se poser sur les raisons et l’utilité de certains choix effectué au début du projet.

# Stockage



Notre PDB possèdera 2 nouveaux *tablespaces* permettant la sauvegarde différée de nos données : nous aurons le tablespace « data » destiné aux données (table, contenu de celle-ci, etc…), ainsi qu’un autre tablespace pour la gestion des index. Ce premier découpage permet de mieux isoler les différentes entités de la base afin de leurs attribuer des fonctionnement différent (espaces disques et autres).

Une image contenant texte, capture d’écran, Police

Description générée automatiquement

Dans notre cas nous avons :

* « ts\_data » : contient toutes les tables
  + Taille 1Go
  + S’étend automatiquement de 100Mo à chaque étape avec une capacité max de 10Go (à modifier une fois que le *tablespace* approchera cette valeur, il est possible que l’accumulation de donnée posera des problèmes de stockage).
* « ts\_index »
  + Taille 100Mo S’étend automatiquement de 100Mo à chaque étape avec une capacité max de 2Go. Ce tablespace est plus limité car il ne contient que des index : très économe en stockage comparé aux données.

# Une image contenant texte, capture d’écran, Police, logiciel Description générée automatiquementRôle et user

Pour nos cas d’utilisation concernant la base de données nous avons opté pour la création de 2 utilisateurs :

* 1 Utilisateur admin, permettant la gestion de l’administration de la PDB. Dans notre cas celui-ci possède tous les droits. Ces accès pourront être modifié si nécessaire à l’avenir.
* 1 Utilisateur application. Cet utilisateur sera utilisé par la majorité des utilisateurs accédant à la base OLTP. L’objectif est de lui accorder des droits différents d’ « admin ». Nous lui avons donné un accès à toutes les tables (lecture / écriture). *Ces accès devront être modifiés car ils permettent à l’utilisateur « application » d’accéder aux tables d’administration de la PDB, ce qui n’est pas envisagé dans notre cas. Nous n’avions pas pris en compte ce cas lors de la création de l’utilisateur.*

Ces utilisateurs possèdent des profils différents, cela nous impose de leurs donner une « configuration » différente. Cette configuration de profile permet de gérer : la quantité d’utilisateur connectée en même temps, la gestion de leurs mot de passes, etc…

Voici les profils que nous avons réalisés :

* Profile type « Administrateur », il s’agit d’un profil avec peu de connexion simultanée en même temps (2 dans notre cas). Celui-ci possède un mot de passe dont la durée de validité est de 180 jours avec 5 essais de connexion avant blocage. A cela nous avons attribué un « Grace time » de 14 jours, équivalent à la durée autorisée à l’utilisateur afin de modifier son mot de passe.
* Profile type « application », nous avons à peu près les mêmes paramètre que précédemment :
  + Utilisateurs simultanés (nombre de sessions autorisé) : 100, il s’agit d’une application ayant de nombreux utilisateurs simultanés.
  + Durée de mot de passe : illimité, de même pour la quantité d’essais. Ce mot de passe pouvant être TRES grand (>100 caractères par exemple) il sera quasiment impossible à casser en *brut-force.* De plus la gestion de mot de passe application peut être très difficile à maintenir si celui-ci vient à être changé. Nous avons donc décidé que la mise en place d’un mot de passe à durée illimitée était acceptable car l’application serait utilisée uniquement en interne : limitant grandement le risque d’attaque.

# Sauvegarde

Une image contenant texte, capture d’écran, Police

Description générée automatiquement

La sauvegarde a automatique à été réalisée de manière à être 100% automatisée. Pour cela nous avons utilisé l’utilitaire R-MAN disponible pour Oracle.

Pour les sauvegardes nous avons choisi une durée de rétention de 7 jours. Cela signifie que le système de Backup permet un retour de la base à un état J-7 à tout moment. Nous aurons donc 7 sauvegardes stockées à tout moment. Cette fenêtre nous permet de limiter la quantité de donnée à stocker pour les backups et donc de ne pas avoir de saturation au cous du temps. De plus nous avons tout de même une « recovery » (remise à jour du système) complète possible pour les données des 7 jours précédents.

Il s’agit d’un compromis entre temps et consommation de stockage, il pourra être modifié au cours du temps selon les besoins.

# Une image contenant texte, Police, capture d’écran, noir Description générée automatiquementService

Une image contenant texte, Police, capture d’écran, noir

Description générée automatiquement

Pour la création de la sauvegarde automatisée nous avons utilisé un script permettant de lancer le Backup à l’aide d’un « crontab ». Ce système est natif à Linux et permet la réalisation de tâche programmé. Nous avons donc créé un fichier de script RMAN qui sera exécuté tous les jours à 1h du matin.

La configuration du stockage a été réalisée auparavant et développé dans la section précédente.

# Duplication

Pour l’instant aucun système de duplication à été mis en place durant la création de la base. Cependant pour des raisons de sécurité il est préférable de le réaliser. Voici ce que nous proposons afin de combler ce problème :

* **Stockage sur base OLTP (Disque dur classique) :** Système mis en place actuellement, aucune copie n’est réalisée, seulement la « base originel » qui devra être dupliqué ailleurs.
* **Stockage sur bandes**: Des stockages à bandes sont de très bons systèmes d’archivage pour de longue durée. Ils sont majoritairement utilisés pour la lecture une fois ceux-ci édités.
* **Stockages situé dans des zones géographique différentes** : Il s’agit de la méthode la moins technique d’un point de vue informatique mais est très importante. Il est important de limiter au maximum les potentiels dégâts liés à une catastrophe naturelle, inondation voire incendie (tout ce qui peux toucher plusieurs système informatique simultanément). Pour cela nous conseillons d’installer les duplication sur des disques et lieux géographique différents. *On ne met pas tous nos œufs dans le même panier, ici c’est le même principe.*
* **Duplication des redo-logs** : Une technique permettant de dupliquer en cours d’utilisation une base de données et de transférer une copie de chaque redo-logs (fichier contenant les modifications lié à la base) à une autre base de données. Il sera alors possible d’en avoir une copie parfaite à tout moment.