

Programme de Formation

**Formation PostgreSQL Avancé**

**Auteur :** Jean-Paul ARGUDO <[jean-paul.argudo@dalibo.com](mailto:jean-paul.argudo@dalibo.com)>  
**Type :** Programme de Formation  
**Classification :** Public  
**Relatif à :** Formation PostgreSQL Avancé

## Table des matières

<b>1</b>	<b>Description</b>	<b>4</b>
1.1	Objectif . . . . .	4
1.2	Durée . . . . .	4
1.3	Public concerné . . . . .	4
1.4	Pré-requis . . . . .	5
1.5	Travaux Pratiques . . . . .	5
<b>2</b>	<b>Plan de formation</b>	<b>6</b>
2.1	Présentation . . . . .	6
2.2	Paramétrage . . . . .	6
2.3	Administration Système . . . . .	7
2.4	Tâches courantes . . . . .	7
2.5	PostgreSQL avancé . . . . .	7
2.6	Sauvegarde et Restauration . . . . .	8
2.7	Optimisations . . . . .	8
2.8	Monitoring . . . . .	8

# 1

## Description

### 1.1 Objectif

Cette formation a pour objectif de transmettre les connaissances et le savoir-faire technique pour assurer les tâches suivantes :

- Maintenir un parc de serveurs de bases de données en environnement de production ;
- Sauvegarder et Restaurer des données ;
- Optimiser le système ;
- Assurer la haute disponibilité des services ;

### 1.2 Durée

La formation se déroule sur 4 jours.

La durée totale de la formation est de 27 heures.

### 1.3 Public concerné

Cette formation s'adresse aux utilisateurs confirmés et aux administrateurs de bases de données (DBA) ayant déjà une bonne connaissance de [PostgreSQL](#).

## 1.4 Pré-requis

- Savoir utiliser un système `Unix` ou, idéalement, `Linux` ;
- Avoir des connaissances minimales en système d'exploitation et informatique ;
- Avoir une bonne connaissance du langage `SQL` ;
- Avoir une bonne connaissance de [PostgreSQL](#).

## 1.5 Travaux Pratiques

Lors de chaque module de formation, l'enseignement sera accompagné de travaux pratiques. Ces exercices techniques permettront de valider l'acquisition des connaissances par les stagiaires.

# 2

## Plan de formation

### 2.1 Présentation

Objectif : **Acquérir une vision claire et précise des avantages et des évolutions du projet PostgreSQL.**

La formation débute avec un tour d'horizon de PostgreSQL et de sa communauté. Cette introduction revient sur les origines du projet, détaille les fonctionnalités des versions courantes et met un coup de projecteur sur les développements en cours au sein du projet.

Les outils de la communauté (serveur principal du projet, listes de diffusion, forums) sont décrits de manière exhaustive et les principaux projets « satellites » (Slony, pgpool, etc.) font l'objet d'une attention particulière.

À l'issue de ce module, le stagiaire possède une vision claire et précise des avantages et des futures évolutions du projet PostgreSQL.

### 2.2 Paramétrage

Objectif : **Comprendre l'ensemble des paramètres de configuration du serveur PostgreSQL.**

Les options du fichiers de configuration `postgresql.conf` sont détaillées de manière exhaustive. Une attention particulière est également apportée à la configuration de l'accès aux bases de données (fichier `pg_hba.conf`) et aux différentes méthodes d'authentification, notamment : PAM, Kerberos, MD5 et SSL.

## 2.3 Administration Système

Objectif : **Adapter le système d'exploitation pour PostgreSQL**

Un serveur hébergeant un cluster PostgreSQL doit être configuré de façon adéquate. Ce module propose de lister les systèmes de fichiers ( Ext3, ReiserFS, etc. ) les plus performants pour PostgreSQL ainsi que les « stratégies RAID » les plus intéressantes. Il dresse une cartographie complète du *cluster* PostgreSQL, en détaillant un à un ses fichiers et sous-répertoires.

Ce module apporte un éclairage sur les différents critères à prendre en compte afin de définir un profil matériel optimal pour accueillir une ou plusieurs base de données PostgreSQL.

## 2.4 Tâches courantes

Objectif : **Maîtriser les opérations de maintenance de PostgreSQL**

Ce module débute avec une présentation des commandes essentielles de création de base de données, avec un éclairage particulier sur la notion de *templates*. La gestion des utilisateurs est ensuite expliquée, notamment avec le concept de *rôles*.

La seconde partie du module s'attache à transmettre les techniques de maintenance et d'entretien de la base, en détaillant les commandes à exécuter périodiquement (*VACUUM*, *ANALYZE*, etc.) ainsi que les outils d'indexation et de ré-indexation.

## 2.5 PostgreSQL avancé

Objectif : **Exploiter toute la richesse de PostgreSQL**

Ce module est un véritable tour d'horizon des aspects les plus techniques et pointus de PostgreSQL :

- Une étude exhaustive des **tables systèmes** permet de comprendre leur utilisation et d'extraire les informations qui s'y trouvent afin de mieux contrôler le serveur de base de données.
- Le système de **tablespaces** est expliqué. Le stagiaire maîtrise alors leur utilisation et comprend leurs apports.
- Le système de **rules** permet de spécifier des opérations très évoluées sur les données de la base.

Ce module contient également une liste des **contribs** (projets connexes). Les plus pertinentes sont détaillées (champs d'application, avantages, etc).

## 2.6 Sauvegarde et Restauration

Objectif : **Assurer la sécurité des données**

Ce module propose au stagiaire d'apprendre à définir une politique de sauvegarde de ses bases de données. En analysant ses besoins en terme, de sécurité, de qualité de service (QoS) et d'archivage, le stagiaire pourra déterminer quels outils et quelles méthodes sont les plus adaptés à son serveur PostgreSQL.

PostgreSQL dispose de commandes internes de sauvegarde (le couple `pg_dump` / `pg_restore`) mais aussi d'un mécanisme évolué de restauration à un instant (Point In Time Recovery ou PITR). Ces différentes approches sont expliquées précisément.

## 2.7 Optimisations

Objectif : **Obtenir des performances optimales**

L'optimisation de bases de données PostgreSQL est traitée selon 5 axes :

- Le *tuning* matériel.  
Comment définir un profil matériel optimal en tenant compte des caractéristiques du système d'information ?
- La configuration du serveur.  
Quels paramètres du fichier de configuration faut-il modifier pour obtenir de meilleures performances ?
- Le design et la maintenance des bases.  
Quelles opérations permettent d'entretenir la base de donnée ?
- L'analyse de requêtes.  
Comment utiliser `EXPLAIN` et `ANALYZE` pour optimiser les requêtes les plus coûteuses ?
- Les tests aux limites et *benchmarks*.  
Quels outils utiliser pour mesurer les gains de performance obtenus ?

## 2.8 Monitoring

Objectif : **Définir une politique de surveillance de PostgreSQL**

Le thème de la surveillance de bases de données PostgreSQL est abordé sous trois angles distincts et complémentaires : l'analyse de fichiers de trace (*logs*), la visualisation d'activité et la gestion des alertes.

À travers l'utilisation du logiciel *pgFouine*, l'analyse des *logs* est expliquée. Le rôle prépondérant d'un tel outil lors des phases de mise au point et d'optimisation est démontré par des exemples simples et didactiques.

L'outil de métrologie *munin* et ses *plug-ins* PostgreSQL sont présentés. Le stagiaire acquiert la compétence lui permettant de développer des *plug-ins* spécifiques et adaptés aux applications qu'il développe et maintient.

Ce module comprend également la description détaillée de deux logiciels de gestion d'alertes : *Zabbix* et *Nagios*.

Muni de ces différents outils, le stagiaire dispose des compétences nécessaires afin de définir et mettre en place une politique de *monitoring* sur mesure permettant de surveiller les systèmes dont il est responsable.