

Mis à jour le 02/07/2025

S'inscrire

Formation FoundationDB

4 jours (28 heures)

Présentation

Découvrez la formation FoundationDB qui vous permettra de maîtriser les principes d'une base transactionnelle distribuée et de déployer des clusters robustes en production.

Vous stockerez les données, gèrerez les transactions multi clés et structurerez vos données à l'aide des couches Tuple et Record Layer.

Vous saurez concevoir un système de stockage transactionnel scalable, implémenter un catalogue structuré avec index, gérer la concurrence d'accès, et déployer un cluster FoundationDB résilient.

Vous serez capable d'exploiter FoundationDB dans un pipeline data ou backend microservices, pour modéliser des métadonnées, documents ou objets relationnels, au sein d'une architecture unifiée, cohérente et haute performance.

Comme pour toutes nos formations, elle se déroulera sur ma toute dernière version de l'outil [FoundationDB](#).

Objectifs

- Comprendre le fonctionnement et l'architecture de FoundationDB
- Développer des applications clientes (Python, Java) avec transactions ACID
- Structurer les données à l'aide des couches Tuple Layer et Record Layer
- Déployer, monitorer et tester un cluster FoundationDB tolérant aux pannes
- Appliquer FoundationDB à des cas d'usage concrets

Public visé

- Data engineers

- Architectes techniques
- Développeurs Back-end

Pré-requis

- Connaissances en gestion de bases de données
- Maîtrise de base d'un langage de programmation
- Notions de transactions, de structures de données et de systèmes distribués
- Familiarité avec le shell ou les environnements en ligne de commande

Programme de la formation FoundationDB

Introduction à FoundationDB et à son positionnement

- Qu'est-ce que FoundationDB ? Historique et cas d'usage
- NoSQL, NewSQL, clé-valeur ordonné : typologie technique
- Rôle de FoundationDB dans l'écosystème Apple, Snowflake, etc.
- Comparaison avec d'autres bases : Cassandra, MongoDB, etc.
- Vue d'ensemble du modèle transactionnel ACID distribué
- Vision multi modèle et architecture en couches

Comprendre l'architecture distribuée

- Rôles système : coordinator, proxies, master, resolvers
- Fragmentation de l'espace de clés et répartition dynamique
- Journalisation des transactions, réplication et haute disponibilité
- Stockage sur disque : moteur B-Tree et moteur Redwood
- Monitoring, failover et redémarrage sans perte de données
- Fonctionnement d'un cluster en lecture/écriture simultanée

Structure des données et espace clé-valeur

- Convention de nommage des clés (sous-espaces)
- Encodage binaire, sérialisation typée (Tuple Layer)
- Range queries et lexicographic ordering
- Pratiques de design clé/valeur pour la performance
- Schémas implicites et versioning applicatif
- Atelier : Exploration manuelle avec fdbcli, range queries, lecture/écriture de clés typées

Transactions distribuées et propriétés ACID

- Isolation forte : serializability, conflits de lecture/écriture
- Atomicité inter-clés et entre opérations
- Déroulement d'une transaction (begin ? commit ? retry)

- Conflits et gestion des erreurs (conflict range, retries)
- Tests de cohérence sur plusieurs clients concurrents
- Atelier : Écriture d'une application Python transactionnelle avec gestion de conflits

Programmation client avec l'API FoundationDB

- Installation du client (Python, Java ou autre langage)
- Fonctions principales : get, set, clear, range, atomic_ops
- Transactions contextuelles et fermeture automatique
- Intégration dans des pipelines ou microservices
- Gestion des erreurs, timeouts, retraits sécurisés
- Encodage structuré des clés avec Tuple()

Structurer ses données avec Tuple Layer

- Objectif de la Tuple Layer : sérialisation typée et ordonnée
- Conversion types natifs ? clés binaires
- Noms de champs, structuration hiérarchique des clés
- Bonnes pratiques de design de schéma clé-valeur
- Encapsulation de logique applicative dans les clés
- Lecture partielle ou parcours en profondeur

Record Layer : modéliser des données structurées

- Présentation de la Record Layer d'Apple (CloudKit)
- Définition de schémas d'enregistrement (records)
- Champs typés, index secondaires, pagination
- Comparaison avec des bases relationnelles (SQL sans JOIN)
- Déploiement d'une couche record sur le cluster
- Atelier : Créer un modèle de type "table utilisateur" avec index secondaire dans la Record Layer

Indexation avancée et requêtes par plage

- Index primaires et secondaires
- Index composés, triés, multi-clés
- Parcours de graphe via les relations indexées
- Scan efficace de plages triées (prefix matching)
- Optimisation des lectures via des schémas d'index

Conception d'une application multi modèle

- Combiner modèles documentaires et relationnels
- Gestion de métadonnées critiques (catalogues, historiques)
- Synchronisation inter-modèles et cohérence globale
- Centraliser plusieurs types de données dans une seule base
- Cas d'usage : moteur de recommandation ou historique d'événements

Déploiement d'un cluster FoundationDB

- Installation en local, en Docker ou Kubernetes
- Configuration des rôles, réplication, réplica sets
- Démarrage, arrêt, mise à l'échelle horizontale
- Stratégies de stockage (SSD, HDD, Redwood vs SQLite)
- Atelier : Déploiement d'un mini-cluster multi-nœuds avec simulation de panne et rebalancing

Surveillance, tests et fiabilité

- Logs, métriques, états système (status)
- Tests de charge, injection de fautes (simulation framework)
- Résilience : tolérance aux pannes, élection automatique
- Alerting et seuils critiques (latence, espace, quorum)
- Bonnes pratiques d'exploitation continue

Cas d'usage réels et choix d'architecture

- Étude de cas : Snowflake, iCloud, grands catalogues
- Stockage de métadonnées critiques et systèmes transactionnels
- Quand préférer FoundationDB à Cassandra, PostgreSQL ou etcd ?
- Intégration dans des architectures cloud natives (CI/CD, microservices)
- Limites connues, extensions possibles, roadmap open source

Sociétés concernées

Cette formation s'adresse à la fois aux particuliers ainsi qu'aux entreprises, petites ou grandes, souhaitant former ses équipes à une nouvelle technologie informatique avancée ou bien à acquérir des connaissances métiers spécifiques ou des méthodes modernes.

Positionnement à l'entrée en formation

Le positionnement à l'entrée en formation respecte les critères qualité Qualiopi. Dès son inscription définitive, l'apprenant reçoit un questionnaire d'auto-évaluation nous permettant d'apprécier son niveau estimé sur différents types de technologies, ses attentes et objectifs personnels quant à la formation à venir, dans les limites imposées par le format sélectionné. Ce questionnaire nous permet également d'anticiper certaines difficultés de connexion ou de sécurité interne en entreprise (intraentreprise ou classe virtuelle) qui pourraient être problématiques pour le suivi et le bon déroulement de la session de formation.

Méthodes pédagogiques

Stage Pratique : 60% Pratique, 40% Théorie. Support de la formation distribué au format numérique à tous les participants.

Organisation

Le cours alterne les apports théoriques du formateur soutenus par des exemples et des séances de réflexions, et de travail en groupe.

Validation

À la fin de la session, un questionnaire à choix multiples permet de vérifier l'acquisition correcte des compétences.

Sanction

Une attestation sera remise à chaque stagiaire qui aura suivi la totalité de la formation.