DAWAN

Plan de la formation

www.dawan.fr

Formation Apache Spark

Durée :	4 jours (28 heures)
Tarifs inter-entreprise :	2 760,00 CHF HT (standard) 2 208,00 CHF HT (remisé)
Public :	Développeurs, architectes système et responsables techniques qui veulent déployer des solutions Spark dans leur entreprise
Pré-requis :	Maîtrise de la programmation orientée objet en Java ou en C#
Objectifs :	- Développer des applications avec Spark - Utiliser les bibliothèques pour SQL, les flux de données et l'apprentissage automatique - Retranscrire des difficultés rencontrées sur le terrain dans des algorithmes parallèles - Développer des applications métier qui s'intègrent à Spark
Modalités pédagogiques, techniques et d'encadrement :	• Formation synchrone en présentiel et distanciel. • Méthodologie basée sur l'Active Learning : 75 % de pratique minimum. • Un PC par participant en présentiel, possibilité de mettre à disposition en bureau à distance un PC et l'environnement adéquat. • Un formateur expert.
Modalités d'évaluation :	• Définition des besoins et attentes des apprenants en amont de la formation. • Auto-positionnement à l'entrée et la sortie de la formation. • Suivi continu par les formateurs durant les ateliers pratiques. • Évaluation à chaud de l'adéquation au besoin professionnel des apprenants le dernier jour de formation.
Sanction :	Attestation de fin de formation mentionnant le résultat des acquis
Référence :	BUS100299-F
Note de satisfaction des participants:	Pas de données disponibles

Contacts :	commercial@dawan.fr - 09 72 37 73 73
Modalités d'accès :	Possibilité de faire un devis en ligne (www.dawan.fr, moncompteformation.gouv.fr, maformation.fr, etc.) ou en appelant au standard.
Délais d'accès :	Variable selon le type de financement.
Accessibilité :	Si vous êtes en situation de handicap, nous sommes en mesure de vous accueillir, n'hésitez pas à nous contacter à referenthandicap@dawan.fr, nous étudierons ensemble vos besoins

Introduction

Définition du Big Data et des calculs
À quoi sert Spark
Quels sont les avantages de Spark

Applications évolutives

Identifier les limites de performances des CPU modernes
Développer les modèles de traitement en parallèle traditionnels

Créer des algorithmes parallèles

Utiliser la programmation fonctionnelle pour l'exécution des programmes en parallèles
Retranscrire des difficultés rencontrées sur le terrain dans des algorithmes parallèles

Structures de données parallèles

Répartir les données dans le cluster avec les RDD (Resilient Distributed Datasets) et les DataFrames
Répartir l'exécution des tâches entre plusieurs nœuds
Lancer les applications avec le modèle d'exécution de Spark

Structure des clusters Spark

Créer des clusters résilients et résistants aux pannes
Mettre en place un système de stockage distribué évolutif

Gestion du cluster

Surveillance et administration des applications Spark
Afficher les plans d'exécution et les résultats

Choisir l'environnement de développement

Réaliser une analyse exploratoire avec le shell Spark
Créer des applications Spark autonomes

Utiliser les API Spark

Programmation avec Scala et d'autres langages compatibles
Créer des applications avec les API de base
Enrichir les applications avec les bibliothèques intégrées

Interroger des données structurées

Traiter les requêtes avec les DataFrames et le code SQL embarqué
Développer SQL avec les fonctions définies par l'utilisateur (UDF)
Utiliser les ensembles de données aux formats JSON et Parquet

Intégration à des systèmes externes

Connexion aux bases de données avec JDBC
Lancer des requêtes Hive sur des applications externes

Qu'appelle-t-on flux de données ?

Utiliser des fenêtres glissantes
Déterminer l'état d'un flux de données continu
Traiter des flux de données simultanés
Améliorer les performances et la fiabilité

Traiter les flux des sources de données

Traiter les flux des sources intégrées (fichiers journaux, sockets Twitter, Kinesis, Kafka)
Développer des récepteurs personnalisés
Traiter les données avec l'API Streaming et Spark SQL

Classer les observations

Prévoir les résultats avec l'apprentissage supervisé
Créer un élément de classification pour l'arbre de décision

Identifier les schémas récurrents

Regrouper les données avec l'apprentissage non supervisé
Créer un cluster avec la méthode k-means

Développer des applications métier avec Spark

Mise à disposition de Spark via un service Web RESTful
Générer des tableaux de bord avec Spark

Utiliser Spark sous forme de service

Service cloud vs. sur site
Choisir un fournisseur de services (AWS, Azure, Databricks, etc.)

Développer Spark pour les clusters de grande taille
Améliorer la sécurité des clusters multifournisseurs
Suivi du développement continu de produits Spark sur le marché
Projet Tungsten : repousser les performances à la limite des capacités des équipements modernes
Utiliser les projets développés avec Spark
Revoir l'architecture de Spark pour les plateformes mobiles