

Formation Databricks — Préparation à la certification Databricks Certified Data Engineer Associate

■ Durée :	5 jours (35 heures)
■ Tarif inter-entreprises :	3 475,00 CHF HT (Présentiel) 2 780,00 CHF HT (Distanciel)
■ Public :	Data engineers, développeurs, administrateurs de données souhaitant maîtriser la plateforme Databricks et préparer la certification Data Engineer Associate
■ Pré-requis :	Bases de SQL et notions de Python - Une première expérience avec des pipelines de données est un plus, mais n'est pas obligatoire
■ Objectifs :	Utiliser la plateforme Databricks Data + AI pour réaliser des tâches d'ingénierie des données de bout en bout : ingestion, transformation, modélisation, orchestration, CI/CD, gouvernance, sécurité, optimisation — et se préparer à l'examen de certification Databricks Certified Data Engineer Associate
■ 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.
■ Modalité 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 :	En cas de réussite aux épreuves certificatives et après validation des résultats, délivrance de la certification Databricks Certified Data Engineer Associate
■ Référence :	BUS103113-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

Découvrir la plateforme Databricks Intelligence Platform

Architecture Lakehouse : entrepôt de données vs data lake vs lakehouse

Structure du workspace : notebooks, clusters, jobs, repos

Unity Catalog : hiérarchie catalogue / schéma / table, metastore

Types de compute : clusters all-purpose, job clusters, SQL Warehouses, serverless

Langages disponibles dans les notebooks (SQL, Python, magic commands)

Atelier : Prise en main du workspace

Créer un workspace, configurer un cluster, naviguer dans Unity Catalog, créer un catalogue et un schéma, exécuter un premier notebook multi-langage.

Ingérer des données dans le Lakehouse

Formats de fichiers sources : CSV, JSON, Parquet, Delta

CREATE TABLE, read_files(), et lecture avec Apache Spark

Auto Loader : ingestion incrémentale de fichiers (cloudFiles)

COPY INTO : chargement idempotent par lots

Gestion des schémas : inférence, évolution, schema hints

Atelier : Ingérer des fichiers bruts

Mettre en place un pipeline d'ingestion incrémentale avec Auto Loader et un chargement par lots avec COPY INTO, sur des fichiers CSV/JSON déposés dans un volume.

Gérer et faire vivre les tables Delta Lake

Tables managées vs tables externes, volumes Unity Catalog
Delta Lake : transaction log, ACID, schema enforcement
Time travel, DESCRIBE HISTORY, RESTORE
Maintenance : OPTIMIZE, ZORDER, VACUUM
Chargement complet vs incrémental, gestion des doublons

Atelier : Créer et faire vivre des tables Delta

Créer des tables managées et externes, effectuer des mises à jour, consulter l'historique des versions, revenir en arrière avec time travel, optimiser une table.

Transformer les données avec PySpark et SQL

DataFrame API PySpark : sélection, filtres, agrégations, jointures
Fonctions SQL avancées : fenêtres analytiques, fonctions de date
User Defined Functions (UDF) Python et SQL
Nettoyage et déduplication de données (MERGE INTO)

Atelier : Pipeline de transformation ETL

Construire un pipeline PySpark/SQL qui nettoie, joint et agrège plusieurs jeux de données bruts, avec gestion des mises à jour via MERGE INTO.

Modéliser les données dans le Lakehouse

Architecture médaillon : couches Bronze / Silver / Gold
Vues, vues matérialisées, tables live vs tables statiques
Slowly Changing Dimensions (SCD type 1 et type 2)
CTAS (CREATE TABLE AS SELECT) et bonnes pratiques de modélisation

Atelier : Construire une architecture médaillon

Concevoir un pipeline Bronze → Silver → Gold complet à partir des données ingérées précédemment, avec gestion d'une dimension SCD type 2.

Construire des pipelines déclaratifs avec Lakeflow

Déclaration de tables streaming et de vues matérialisées
Modes d'exécution : triggered vs continuous
Qualité des données : expectations (DROP, FAIL, WARN)
Suivi d'exécution : graphe de lignage, logs d'événements

Atelier : Pipeline déclaratif avec Lakeflow

Transformer le pipeline médaillon de l'atelier précédent en pipeline déclaratif, avec règles de qualité de données et visualisation du graphe de lignage.

Orchestrer les traitements avec Lakeflow Jobs

Création de jobs multi-tâches, dépendances entre tâches

Types de tâches : notebook, pipeline, Python script, SQL

Planification : cron, déclencheurs de fichiers, jobs pilotés par événement

Paramétrage des jobs, gestion des échecs et des ré-essais, alerting

Atelier : Orchestrer un pipeline complet

Assembler l'ingestion, la transformation et le pipeline déclaratif en un job Lakeflow multi-tâches, avec planification et alertes en cas d'échec.

Industrialiser les déploiements avec le CI/CD

Intégration Git / Databricks Repos, branches et revues de code

Databricks Asset Bundles (DAB) : structure, databricks.yml

Séparation des environnements dev / staging / production

Déploiement automatisé d'un pipeline via bundle

Atelier : Déployer avec les Asset Bundles

Versionner le projet dans un dépôt Git, packager le pipeline avec un Databricks Asset Bundle et le déployer sur un environnement cible.

Sécuriser et gouverner les données avec Unity Catalog

Modèle de permissions Unity Catalog (GRANT/REVOKE), principe du moindre privilège

Gestion des propriétaires (ownership) et des groupes

Sécurité au niveau ligne et colonne, masquage de données

Lignage des données et audit des accès

Atelier : Sécuriser les données

Mettre en place des permissions différenciées par rôle sur les

catalogues/schémas/tables du projet, configurer un masquage de colonne sensible et consulter le lignage.

Diagnostiquer et optimiser les pipelines

Lecture de l'historique des requêtes et du Spark UI

Diagnostic des erreurs courantes (skew, out of memory, petits fichiers)

Optimisation : partitionnement, Z-order, caching, taille de cluster

Bonnes pratiques de coût et de performance

Atelier : Diagnostiquer et optimiser

Analyser un job lent, identifier le goulot d'étranglement via le Spark UI et l'historique

des requêtes, puis appliquer les optimisations adaptées.