

Chapitre 5 : Intégration des Données IoT

- 1 Concepts des processus ETL (Extraction, Transformation, Chargement)
- 2 Outils ETL pour l'IoT : Apache NiFi, Talend
- 3 Développement de pipelines ETL pour l'intégration des données IoT
- 4 Choix des bases de données selon les besoins applicatifs
- 5 Intégration des données IoT dans des bases relationnelles (MySQL, PostgreSQL)
- 6 Intégration des données IoT dans des bases NoSQL (MongoDB, Cassandra)
- 7 Introduction aux services cloud pour l'IoT : AWS IoT, Azure IoT, Google Cloud IoT
- 8 Intégration des données IoT avec les services cloud
- 9 Stockage, visualisation et analyse des données IoT sur le cloud
- 10 Exemples d'architectures cloud pour l'IoT

- Concepts des processus ETL (Extraction, Transformation, Chargement)
- Outils ETL pour l'IoT : Apache NiFi, Talend
- Développement de pipelines ETL pour l'intégration des données IoT
- Choix des bases de données selon les besoins applicatifs
- Intégration des données IoT dans des bases relationnelles (MySQL, PostgreSQL)
- Intégration des données IoT dans des bases NoSQL (MongoDB, Cassandra)

- Introduction aux services cloud pour l'IoT : AWS IoT, Azure IoT, Google Cloud IoT
- Intégration des données IoT avec les services cloud
- Stockage, visualisation et analyse des données IoT sur le cloud
- Exemples d'architectures cloud pour l'IoT

Définition du Processus ETL

Un capteur envoie "temp=22.5" » transformé en {temperature: 22.5} » stocké en base.

Le processus **ETL (Extract, Transform, Load)** est une méthodologie fondamentale en gestion de données. Il permet de **convertir des données brutes en informations structurées** prêtes pour l'analyse et la prise de décision.

PHASE 1



Extraction (Extract)

- ✓ Collecte des données brutes (JSON, CSV, etc.) de sources diverses.
- ✓ Gestion de sources hétérogènes et distribuées (IoT).
- ✓ Identification des métadonnées

PHASE 2



Transformation

- ✓ Nettoyage, filtrage (Bruit IoT) et normalisation (Unités, Formats)
- ✓ Enrichissement et agrégation pour la cohérence.

PHASE 3



Chargement (Load)

- ✓ Stockage dans un système de destination final.
- ✓ Entrepôts (Data Warehouse), Lacs de données (Data Lakes).
- ✓ Indexation pour requêtage rapide
- ✓ Disponibilité pour analyse BI/IA

- Concepts des processus ETL (Extraction, Transformation, Chargement)
- Outils ETL pour l'IoT : Apache NiFi, Talend
- Développement de pipelines ETL pour l'intégration des données IoT
- Choix des bases de données selon les besoins applicatifs
- Intégration des données IoT dans des bases relationnelles (MySQL, PostgreSQL)
- Intégration des données IoT dans des bases NoSQL (MongoDB, Cassandra)

- Introduction aux services cloud pour l'IoT : AWS IoT, Azure IoT, Google Cloud IoT
- Intégration des données IoT avec les services cloud
- Stockage, visualisation et analyse des données IoT sur le cloud
- Exemples d'architectures cloud pour l'IoT

ETL : IoT vs Systèmes Classiques



ETL Classique (utilisé dans les entreprises (ERP, CRM, bases SQL))

- ✓ **Sources :**
Bases de données relationnelles, fichiers structurés (ERP, CRM).
- 🕒 **Fréquence :**
Traitement par **Lots (Batch)**, souvent quotidien ou hebdomadaire. **! Pas en temps réel**
- 📁 **Variété :**
Données principalement **structurées** avec **schémas prédéfinis**.
- 🌐 **Volume :**
Volumes importants mais **prévisibles**, Stockées par des entrepôts de données (**Data Warehouse**).



ETL pour l'IoT

- ⚡ **Sources :**
Capteurs hétérogènes, objets connectés, protocoles variés (MQTT, CoAP), Edge gateways.
- ⇒ **Fréquence :**
Temps Réel (Streaming). Flux continu nécessitant une latence minimale.
- 🔗 **Variété :**
Données semi-structurées ou non structurées (JSON, binaire, séries temporelles).
- ∞ **Volume :**
Volumes massifs et imprévisibles (Big Data), vitesse extrême, Stockage dans un Data Lake (S3, Hadoop)



Conclusion : L'IoT exige une transition du modèle "Batch Statique" vers un modèle "Streaming Dynamique" et hautement scalable.

- Concepts des processus ETL (Extraction, Transformation, Chargement)
- Outils ETL pour l'IoT : Apache NiFi, Talend
- Développement de pipelines ETL pour l'intégration des données IoT
- Choix des bases de données selon les besoins applicatifs
- Intégration des données IoT dans des bases relationnelles (MySQL, PostgreSQL)
- Intégration des données IoT dans des bases NoSQL (MongoDB, Cassandra)

- Introduction aux services cloud pour l'IoT : AWS IoT, Azure IoT, Google Cloud IoT
- Intégration des données IoT avec les services cloud
- Stockage, visualisation et analyse des données IoT sur le cloud
- Exemples d'architectures cloud pour l'IoT

Phase Extract (sources IoT)

L'extraction est la **première étape critique** du pipeline. Elle consiste à collecter des données brutes depuis des sources massives, hétérogènes et géographiquement distribuées.

Objectif de la phase Extract

L'objectif est simple :

collecter les données telles qu'elles existent, sans les modifier.

- ✓ Pas de transformation ici
- ✓ Pas d'analyse
- ✓ Juste récupération fiable et complète

Modes de Lecture

Mode PUSH

L'appareil envoie les données dès qu'elles sont disponibles (Temps réel).

Avantage : temps réel

Inconvénient : besoin d'infrastructure réactive

Mode PULL

Le système ETL demande les données périodiquement (Batch/Requête).

Avantage : contrôle du flux

Inconvénient : pas temps réel

Sources Hétérogènes(formats, fréquences, technologies différentes).

- ✓ Capteurs environnementaux (Temp, Humidité)
- ✓ Compteurs intelligents (électricité, eau, gaz)
- ✓ Passerelles (Gateways) de bordure



Protocoles d'Acquisition

MQTT

HTTP/REST

CoAP

Nécessite une gestion de la latence et de la connectivité intermittente.

- Concepts des processus ETL (Extraction, Transformation, Chargement)
- Outils ETL pour l'IoT : Apache NiFi, Talend
- Développement de pipelines ETL pour l'intégration des données IoT
- Choix des bases de données selon les besoins applicatifs
- Intégration des données IoT dans des bases relationnelles (MySQL, PostgreSQL)
- Intégration des données IoT dans des bases NoSQL (MongoDB, Cassandra)

- Introduction aux services cloud pour l'IoT : AWS IoT, Azure IoT, Google Cloud IoT
- Intégration des données IoT avec les services cloud
- Stockage, visualisation et analyse des données IoT sur le cloud
- Exemples d'architectures cloud pour l'IoT

Phase Transformation

La phase de transformation est le "cerveau" de l'ETL. Elle convertit les données brutes hétérogènes en informations *cohérentes, fiables et prêtes pour l'analyse*.



Nettoyage

Suppression du bruit, gestion des valeurs manquantes et filtrage des données aberrantes (outliers).



Normalisation

Standardisation des unités (ex: Celsius vs Kelvin) et formats de dates pour une lecture uniforme.



Enrichissement

Ajout de métadonnées contextuelles (localisation GPS, ID machine, météo) aux mesures brutes.



Agrégation

Calculs statistiques (moyennes, max) sur des fenêtres temporelles pour réduire le volume de stockage.

- Concepts des processus ETL (Extraction, Transformation, Chargement)
- Outils ETL pour l'IoT : Apache NiFi, Talend
- Développement de pipelines ETL pour l'intégration des données IoT
- Choix des bases de données selon les besoins applicatifs
- Intégration des données IoT dans des bases relationnelles (MySQL, PostgreSQL)
- Intégration des données IoT dans des bases NoSQL (MongoDB, Cassandra)

- Introduction aux services cloud pour l'IoT : AWS IoT, Azure IoT, Google Cloud IoT
- Intégration des données IoT avec les services cloud
- Stockage, visualisation et analyse des données IoT sur le cloud
- Exemples d'architectures cloud pour l'IoT

Phase Load : Chargement des Données

Le **Chargement** est l'étape finale où les données transformées sont persistées dans un système de destination pour l'interrogation, la visualisation et l'analyse.



Data Warehouse

Stockage structuré optimisé pour les requêtes analytiques complexes et le reporting BI.

- Schéma rigide
- Données agrégées
- Historisation



Data Lake

Vaste réservoir pour les **données brutes** et **semi-structurées** à grande échelle.

- Schéma à la lecture
- Haute scalabilité
- Données IoT brutes



Bases Spécialisées

Bases de données **NoSQL** ou **Time-Series** pour des cas d'usage spécifiques.

- Séries temporelles
- Temps réel (Hot path)
- Haute vélocité

STRATÉGIES DE CHARGEMENT

Full Load

Remplacement complet

Incremental Load

Ajout des nouveaux enregistrements

CONTRAINTES TECHNIQUES

Intégrité référentielle

Indexation

Compression

- **Concepts des processus ETL (Extraction, Transformation, Chargement)**
- Outils ETL pour l'IoT : Apache NiFi, Talend
- Développement de pipelines ETL pour l'intégration des données IoT
- Choix des bases de données selon les besoins applicatifs
- Intégration des données IoT dans des bases relationnelles (MySQL, PostgreSQL)
- Intégration des données IoT dans des bases NoSQL (MongoDB, Cassandra)

- Introduction aux services cloud pour l'IoT : AWS IoT, Azure IoT, Google Cloud IoT
- Intégration des données IoT avec les services cloud
- Stockage, visualisation et analyse des données IoT sur le cloud
- Exemples d'architectures cloud pour l'IoT

Batch vs Streaming

Choisir le mode de traitement selon la vélocité des données IoT



Traitement par Lots

"Batch Processing"

- ✓ **Données:** Volumes massifs accumulés sur une période.
- ✓ **Latence:** Élevée (minutes, heures, jours).
- ✓ **Objectif:** Analyse historique, rapports complexes, Big Data.
- ✓ **Exemple IoT:** Calcul de la consommation mensuelle d'énergie.

OUTILS TYPES:

Hadoop MapReduce, Spark (Batch), Talend



Traitement en Flux

"Stream Processing"

- ✓ **Données:** Flux continu, message par message ou micro-batch.
- ✓ **Latence:** Très faible (millisecondes à secondes).
- ✓ **Objectif:** Réactivité immédiate, détection d'anomalies, alertes.
- ✓ **Exemple IoT:** Arrêt d'urgence d'une machine en surchauffe.

OUTILS TYPES:

Apache Kafka, Spark Streaming, Apache NiFi

Critère	Batch	Streaming
Fréquence	Ponctuelle / Planifiée	Continue / Temps réel
Complexité d'Analyse	Très élevée (Jointures massives)	Moyenne (Fenêtrage temporel)

Les Défis de l'Intégration IoT : Les 3 V

Maîtriser la complexité des données massives



VOLUME

Milliards d'appareils générant des pétaoctets de données. Nécessite des capacités de stockage massives et distribuées.

DÉFI : MISE À L'ÉCHELLE (SCALE)



VÉLOCITÉ

Flux de données en temps réel (streaming). Les décisions doivent souvent être prises en quelques millisecondes.

DÉFI : LATENCE & DÉBIT



VARIÉTÉ

Données structurées (SQL), semi-structurées (JSON/XML) et non structurées (vidéo, audio) issues de capteurs hétérogènes.

DÉFI : INTEROPÉRABILITÉ



Impacts sur l'infrastructure d'intégration :

Architecture Distribuée : Obligation d'utiliser des clusters (Spark, Flink, NoSQL) pour répartir la charge de calcul et de stockage.

Edge Computing : Prétraitement et filtrage des données à la périphérie pour réduire la charge sur le réseau et le cloud.

- Concepts des processus ETL (Extraction, Transformation, Chargement)
- Outils ETL pour l'IoT : Apache NiFi, Talend
- Développement de pipelines ETL pour l'intégration des données IoT
- Choix des bases de données selon les besoins applicatifs
- Intégration des données IoT dans des bases relationnelles (MySQL, PostgreSQL)
- Intégration des données IoT dans des bases NoSQL (MongoDB, Cassandra)

- Introduction aux services cloud pour l'IoT : AWS IoT, Azure IoT, Google Cloud IoT
- Intégration des données IoT avec les services cloud
- Stockage, visualisation et analyse des données IoT sur le cloud
- Exemples d'architectures cloud pour l'IoT

Introduction Apache NiFi

Plateforme open source conçue pour **automatiser et orchestrer visuellement les flux de données** entre différents systèmes. Cet outil d'ingestion et d'orchestration de données (ETL/ELT) permet de déplacer, transformer et charger les données en temps réel ou en mode batch, via une interface graphique intuitive.



Interface Web Intuitive

Conception visuelle des flux par **glisser-déposer** sans codage complexe.



Flow-Based Programming

Traitement des données via des "**FlowFiles**" transitant par des processeurs.



Provenance des Données

Visibilité complète et traçabilité de bout en bout du parcours de chaque donnée.



Gestion Back-pressure

Régulation automatique du débit pour éviter la surcharge des systèmes en aval.



Priorisation Dynamique

Possibilité de prioriser les données **critiques** en fonction de règles métier.



Haute Extensibilité

Plus de 188 processeurs intégrés et support de composants personnalisés.

Initialement développée par la NSA sous le nom de "*NiagaraFiles*".

- Concepts des processus ETL (Extraction, Transformation, Chargement)
- Outils ETL pour l'IoT : Apache NiFi, Talend
- Développement de pipelines ETL pour l'intégration des données IoT
- Choix des bases de données selon les besoins applicatifs
- Intégration des données IoT dans des bases relationnelles (MySQL, PostgreSQL)
- Intégration des données IoT dans des bases NoSQL (MongoDB, Cassandra)

- Introduction aux services cloud pour l'IoT : AWS IoT, Azure IoT, Google Cloud IoT
- Intégration des données IoT avec les services cloud
- Stockage, visualisation et analyse des données IoT sur le cloud
- Exemples d'architectures cloud pour l'IoT

Architecture de Apache NiFi

CŒUR DU SYSTÈME (ORCHESTRATION)

Web Server (porte d'entrée de NiFi)

Héberge l'interface graphique (UI) basée sur HTTP. Concevoir les flux (drag & drop), Démarrer / arrêter les processeurs, Surveiller les performances en temps réel, Gérer les erreurs.

Flow Controller (Le cerveau de NiFi)

Orchestre tout le système. Planifie l'exécution des processeurs (**scheduling**), Gère les threads et les ressources (**CPU, mémoire**), Contrôle le **flux** des **données** entre processeurs, Applique la logique du **workflow** (**routing, priorisation, back pressure**)

PERSISTANCE ET TRAÇABILITÉ (RÉPERTOIRES)

FlowFile Repository(Suivi des métadonnées)

Stocke les métadonnées des données en cours de traitement. Maintient l'état et les attributs (nom fichier, timestamp, type, etc.) des FlowFiles actifs. Assure la reprise après incident en mémorisant la position dans le flux.

Content Repository(Stockage des données)

Stocke les octets réels (**payload:(fichier, JSON, CSV, etc.)**) des données. Utilise un système de fichiers efficace pour **gérer de gros volumes de données**.

Provenance Repository (Historique NiFi complet)

Archive l'historique complet de chaque donnée. Crucial pour le débogage, l'audit (qui a fait quoi ?) et la conformité (Data Lineage).

 **Note technique :** Cette architecture modulaire permet à NiFi d'offrir une garantie de livraison et une gestion robuste de la contre-pression (back-pressure).

- Concepts des processus ETL (Extraction, Transformation, Chargement)
- **Outils ETL pour l'IoT : Apache NiFi, Talend**
- Développement de pipelines ETL pour l'intégration des données IoT
- Choix des bases de données selon les besoins applicatifs
- Intégration des données IoT dans des bases relationnelles (MySQL, PostgreSQL)
- Intégration des données IoT dans des bases NoSQL (MongoDB, Cassandra)

- Introduction aux services cloud pour l'IoT : AWS IoT, Azure IoT, Google Cloud IoT
- Intégration des données IoT avec les services cloud
- Stockage, visualisation et analyse des données IoT sur le cloud
- Exemples d'architectures cloud pour l'IoT

Apache NiFi et Apache MiNiFi

Apache NiFi

C'est le **moteur principal** de traitement de données.

- Fonctionne généralement sur des serveurs (ou cluster)
- Interface graphique complète (drag & drop)
- Permet :
 - ingestion de données
 - transformation
 - routage
 - monitoring avancé
- Gère des flux complexes et volumineux
- Utilisé dans les data centers ou le cloud

En gros : le **“cerveau central”**

Apache MiNiFi

C'est une **version légère et embarquée de NiFi**.

- Conçu pour fonctionner sur des appareils en périphérie (edge) :
 - IoT
 - capteurs
 - machines industrielles
- Pas d'interface graphique (configuration via fichiers)
- Consomme très peu de ressources
- Collecte et envoie les données vers NiFi

En gros : le **“collecteur terrain”**

- Concepts des processus ETL (Extraction, Transformation, Chargement)
- **Outils ETL pour l'IoT : Apache NiFi, Talend**
- Développement de pipelines ETL pour l'intégration des données IoT
- Choix des bases de données selon les besoins applicatifs
- Intégration des données IoT dans des bases relationnelles (MySQL, PostgreSQL)
- Intégration des données IoT dans des bases NoSQL (MongoDB, Cassandra)

- Introduction aux services cloud pour l'IoT : AWS IoT, Azure IoT, Google Cloud IoT
- Intégration des données IoT avec les services cloud
- Stockage, visualisation et analyse des données IoT sur le cloud
- Exemples d'architectures cloud pour l'IoT

Apache NiFi et Apache MiNiFi

Critère	NiFi	MiNiFi
Rôle	Traitement complet	Collecte / pré-traitement
Interface	Graphique (UI web)	Pas d'UI
Environnement	Serveurs / Cloud	Edge / IoT
Ressources	Plus lourd	Très léger
Usage typique	Pipeline data	Envoi de données vers NiFi

MiNiFi tourne sur des **machines proches des sources de données**, puis envoie ces données vers **NiFi** qui fait le traitement avancé.

- Concepts des processus ETL (Extraction, Transformation, Chargement)
- **Outils ETL pour l'IoT : Apache NiFi, Talend**
- Développement de pipelines ETL pour l'intégration des données IoT
- Choix des bases de données selon les besoins applicatifs
- Intégration des données IoT dans des bases relationnelles (MySQL, PostgreSQL)
- Intégration des données IoT dans des bases NoSQL (MongoDB, Cassandra)

- Introduction aux services cloud pour l'IoT : AWS IoT, Azure IoT, Google Cloud IoT
- Intégration des données IoT avec les services cloud
- Stockage, visualisation et analyse des données IoT sur le cloud
- Exemples d'architectures cloud pour l'IoT

Processors, Flow & Routing



Le Processor

Bloc de construction effectuant une **tâche spécifique sur les données.**

- ✓ ~188 processeurs intégrés
- ✓ Ex: GetFile, PutKafka, SQL
- ✓ Extensibilité personnalisée



Le FlowFile

L'unité d'information atomique circulant dans le système.

STRUCTURE :

Contenu	<i>Payload réel</i>
Attributs	<i>Métadonnées (Clé/Valeur)</i>



Routage & Queue

Orchestration dynamique du parcours des données.

- ✂ **Routage Dynamique** : Basé sur les attributs ou résultats (success/failure).
- 🔀 **Back-pressure** : Gestion automatique des files pour éviter la surcharge.

- Concepts des processus ETL (Extraction, Transformation, Chargement)
- **Outils ETL pour l'IoT : Apache NiFi, Talend**
- Développement de pipelines ETL pour l'intégration des données IoT
- Choix des bases de données selon les besoins applicatifs
- Intégration des données IoT dans des bases relationnelles (MySQL, PostgreSQL)
- Intégration des données IoT dans des bases NoSQL (MongoDB, Cassandra)

- Introduction aux services cloud pour l'IoT : AWS IoT, Azure IoT, Google Cloud IoT
- Intégration des données IoT avec les services cloud
- Stockage, visualisation et analyse des données IoT sur le cloud
- Exemples d'architectures cloud pour l'IoT

Introduction à Talend

Plateforme d'Intégration Unifiée

Talend est une solution d'intégration de données complète, offrant des outils pour la **préparation**, la **qualité** et la **gouvernance** des données.

Elle est mondialement reconnue pour ses capacités avancées en processus **ETL** et **ELT**.

Génération de Code

Génère du code natif (Java) pour des performances optimales sans moteur propriétaire lourd.

Modèle ETL / ELT

Supporte le traitement sur serveur dédié ou directement au sein de la base de données cible.

Qualité des Données

Outils intégrés de nettoyage, de dédoublonnage et de validation des flux IoT entrants.

Connectivité IoT

Vaste bibliothèque de connecteurs pour MQTT, AMQP, Kafka et les plateformes Cloud majeures.

 Approche basée sur des "jobs" visuels et une génération de code Java optimisée.

- Concepts des processus ETL (Extraction, Transformation, Chargement)
- **Outils ETL pour l'IoT : Apache NiFi, Talend**
- Développement de pipelines ETL pour l'intégration des données IoT
- Choix des bases de données selon les besoins applicatifs
- Intégration des données IoT dans des bases relationnelles (MySQL, PostgreSQL)
- Intégration des données IoT dans des bases NoSQL (MongoDB, Cassandra)

- Introduction aux services cloud pour l'IoT : AWS IoT, Azure IoT, Google Cloud IoT
- Intégration des données IoT avec les services cloud
- Stockage, visualisation et analyse des données IoT sur le cloud
- Exemples d'architectures cloud pour l'IoT

Jobs Talend : Orchestration & Exécution

Qu'est-ce qu'un Job Talend ?

Un **Job** est l'unité de base de l'intégration dans Talend. Il s'agit d'une conception graphique qui est compilée en **code Java natif** pour une exécution haute performance des transformations de données massives.

Composants d'Entrée

Lecture des sources IoT (MQTT, Kafka, SQL, fichiers JSON/CSV).

Transformation (tMap)

Mappage, filtrage, agrégation et enrichissement des flux de données.

Composants de Sortie

Chargement vers Data Lakes, Cloud Storage ou Bases de données.



Code Java Généré

Transformation visuelle en code robuste et scalable.



Connectivité Massive

+900 connecteurs pour l'écosystème IoT et Big Data.



Orchestration de Workflows

Gestion des dépendances entre tâches complexes (Triggering).



Réutilisabilité

Utilisation de "Joblets" pour standardiser les transformations.



SCALABILITÉ



FLEXIBILITÉ



PERFORMANCE

- Concepts des processus ETL (Extraction, Transformation, Chargement)
- **Outils ETL pour l'IoT : Apache NiFi, Talend**
- Développement de pipelines ETL pour l'intégration des données IoT
- Choix des bases de données selon les besoins applicatifs
- Intégration des données IoT dans des bases relationnelles (MySQL, PostgreSQL)
- Intégration des données IoT dans des bases NoSQL (MongoDB, Cassandra)

- Introduction aux services cloud pour l'IoT : AWS IoT, Azure IoT, Google Cloud IoT
- Intégration des données IoT avec les services cloud
- Stockage, visualisation et analyse des données IoT sur le cloud
- Exemples d'architectures cloud pour l'IoT

Comparaison NiFi vs Talend

Critère	Apache NiFi	Talend
Architecture	Flow-based Programmation basée sur les flux	Job-based Génération de code (Java)
Temps Réel / Streaming	★ Excellence Conçu pour le flux continu	Capable (ESB/Spark) Initialement orienté Batch
Facilité d'utilisation	Très Intuitive Interface Web / Glisser-déposer	Modérée Studio Eclipse / Courbe d'apprentissage
Transf. Complexes	Standard Nécessite souvent des scripts custom	★ Avancé 900+ composants dédiés
Traçabilité (Lineage)	Provenance Native Visibilité de bout en bout	Limitée / Metadata Via Talend Administration Center



Conseil d'expert : Choisissez **NiFi** pour l'ingestion de données IoT à haute vitesse et **Talend** pour les intégrations complexes de Data Warehousing et la qualité de données.

- Concepts des processus ETL (Extraction, Transformation, Chargement)
- **Outils ETL pour l'IoT : Apache NiFi, Talend**
- Développement de pipelines ETL pour l'intégration des données IoT
- Choix des bases de données selon les besoins applicatifs
- Intégration des données IoT dans des bases relationnelles (MySQL, PostgreSQL)
- Intégration des données IoT dans des bases NoSQL (MongoDB, Cassandra)

- Introduction aux services cloud pour l'IoT : AWS IoT, Azure IoT, Google Cloud IoT
- Intégration des données IoT avec les services cloud
- Stockage, visualisation et analyse des données IoT sur le cloud
- Exemples d'architectures cloud pour l'IoT

Choix d'un outil ETL

Critères de sélection stratégiques



Volumes de Données

Évaluer la **capacité de passage à l'échelle** (**scalability**). Gérer des **pétaoctets** vs **des flux modérés**.



Compétences Équipe

Préférence pour le **développement visuel** (**No-code/Low-code**) ou l'écriture de **code Java/Python**.



Exigences de Latence

Besoin de **traitement en temps réel strict** (**Streaming**) ou tolérance pour du **traitement par lots** (**Batch**).



Synthèse de décision

Privilégier **NiFi** pour l'orchestration de flux complexes en temps réel et **Talend** pour l'intégration de données massives structurées.

NiFi = Flexibilité

Talend = Performance Batch

- Concepts des processus ETL (Extraction, Transformation, Chargement)
- Outils ETL pour l'IoT : Apache NiFi, Talend
- Développement de pipelines ETL pour l'intégration des données IoT
- Choix des bases de données selon les besoins applicatifs
- Intégration des données IoT dans des bases relationnelles (MySQL, PostgreSQL)
- Intégration des données IoT dans des bases NoSQL (MongoDB, Cassandra)

- Introduction aux services cloud pour l'IoT : AWS IoT, Azure IoT, Google Cloud IoT
- Intégration des données IoT avec les services cloud
- Stockage, visualisation et analyse des données IoT sur le cloud
- Exemples d'architectures cloud pour l'IoT

Définition d'un Pipeline de Données

Le développement de pipelines ETL pour l'intégration des données IoT consiste à construire une chaîne capable de **collecter**, nettoyer, **transformer** et **acheminer** les données venues d'objets connectés.

- ✓ **Flux continu** : Automatisation du mouvement des données.
- ✓ **Raffinage** : Transformation des données brutes en "pépites" d'information.
- ✓ **Disponibilité** : Garantit que les données sont prêtes pour l'analyse ou l'IA.



SOURCE



PROCESSING



VALEUR

Le pipeline transforme le chaos des données brutes en valeur métier.



L'analogie de la chaîne de montage

Matières Premières

Données brutes issues de capteurs IoT (température, humidité).

Stations de Travail

Étapes de nettoyage, normalisation et enrichissement.

Produit Emballé

Données structurées prêtes pour les tableaux de bord ou l'IA.

- Concepts des processus ETL (Extraction, Transformation, Chargement)
- Outils ETL pour l'IoT : Apache NiFi, Talend
- Développement de pipelines ETL pour l'intégration des données IoT
- Choix des bases de données selon les besoins applicatifs
- Intégration des données IoT dans des bases relationnelles (MySQL, PostgreSQL)
- Intégration des données IoT dans des bases NoSQL (MongoDB, Cassandra)

- Introduction aux services cloud pour l'IoT : AWS IoT, Azure IoT, Google Cloud IoT
- Intégration des données IoT avec les services cloud
- Stockage, visualisation et analyse des données IoT sur le cloud
- Exemples d'architectures cloud pour l'IoT

Architecture Pipeline IoT

Couche Périphérique (Edge, Fog)

 **Capteurs IoT**
Collecte de données brutes.

 **Gateways IoT**
*Agrégation et pré-traitement
(filtrage, compression).*

Couche Plateforme (Cloud)

Ingestion : Kafka, AWS IoT Core, Azure IoT Hub.

Stockage Brut : Data Lakes (S3, Azure Data Lake).

Traitement : Spark, Flink, Stream Analytics.

Stockage Analytique : Redshift, BigQuery, Cassandra.

Couche Entreprise

 **Applications BI & IA**
Tableaux de bord, rapports, modèles ML.

 **API de Service**
Exposition des données aux systèmes tiers.

- Concepts des processus ETL (Extraction, Transformation, Chargement)
- Outils ETL pour l'IoT : Apache NiFi, Talend
- Développement de pipelines ETL pour l'intégration des données IoT
- Choix des bases de données selon les besoins applicatifs
- Intégration des données IoT dans des bases relationnelles (MySQL, PostgreSQL)
- Intégration des données IoT dans des bases NoSQL (MongoDB, Cassandra)

- Introduction aux services cloud pour l'IoT : AWS IoT, Azure IoT, Google Cloud IoT
- Intégration des données IoT avec les services cloud
- Stockage, visualisation et analyse des données IoT sur le cloud
- Exemples d'architectures cloud pour l'IoT

Étapes du pipeline de données

Un pipeline IoT suit une séquence logique pour transformer les données brutes des capteurs en **informations actionnables**.



1. Collecte

Acquisition directe depuis les capteurs et appareils.

EXEMPLE

Mesure de température toutes les 5s.



2. Ingestion

Transfert sécurisé vers la plateforme centrale.

EXEMPLE

Envoi via MQTT à un broker (Kafka, AWS IoT Core).

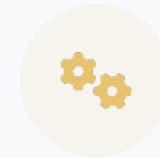


3. Data Lake

Conservation au format original pour flexibilité.

EXEMPLE

Stockage des (JSON/CSV) dans Amazon S3 ou Azure Blob.



4. Analyse

Nettoyage, normalisation et agrégation (Apache Spark/NiFi).

EXEMPLE

Calcul de moyenne horaire avec Spark.



5. Exposition

Visualisation et accès via APIs sécurisées.

EXEMPLE

Dashboard BI ou API REST pour app métier.

- Concepts des processus ETL (Extraction, Transformation, Chargement)
- Outils ETL pour l'IoT : Apache NiFi, Talend
- **Développement de pipelines ETL pour l'intégration des données IoT**
- Choix des bases de données selon les besoins applicatifs
- Intégration des données IoT dans des bases relationnelles (MySQL, PostgreSQL)
- Intégration des données IoT dans des bases NoSQL (MongoDB, Cassandra)

- Introduction aux services cloud pour l'IoT : AWS IoT, Azure IoT, Google Cloud IoT
- Intégration des données IoT avec les services cloud
- Stockage, visualisation et analyse des données IoT sur le cloud
- Exemples d'architectures cloud pour l'IoT

Orchestration & Monitoring

Orchestration

Coordination et automatisation des flux de données.

- ✓ Gestion des **dépendances** entre tâches
- ✓ Séquençage logique (Workflows)
- ✓ Reprise sur erreur automatique

Monitoring

Surveillance continue de la santé du système.

- ✓ Mesure du **débit** et de la **latence**
- ✓ Alertes en temps réel sur seuils
- ✓ **Journalisation (Logging) des événements**

Objectif Final : Fiabilité Industrielle

Garantir que les données parviennent à destination avec une qualité optimale et un retard minimal.

24/7
DISPONIBILITÉ

- Concepts des processus ETL (Extraction, Transformation, Chargement)
- Outils ETL pour l'IoT : Apache NiFi, Talend
- **Développement de pipelines ETL pour l'intégration des données IoT**
- Choix des bases de données selon les besoins applicatifs
- Intégration des données IoT dans des bases relationnelles (MySQL, PostgreSQL)
- Intégration des données IoT dans des bases NoSQL (MongoDB, Cassandra)

- Introduction aux services cloud pour l'IoT : AWS IoT, Azure IoT, Google Cloud IoT
- Intégration des données IoT avec les services cloud
- Stockage, visualisation et analyse des données IoT sur le cloud
- Exemples d'architectures cloud pour l'IoT

Sécurité du Pipeline IoT

Une préoccupation constante à chaque étape du flux

Mesures indispensables pour garantir l'intégrité et la confidentialité des données.



Authentification

Identification unique de chaque objet
Certificats X.509 ou clés d'API
Prévention des appareils non autorisés



Chiffrement

Données en transit (TLS/SSL)
Données au repos (AES-256)
Protection contre l'interception



Gestion des Accès

Modèle RBAC (Rôles)
Principe du moindre privilège
Gestion des identités (IAM)

DISPONIBILITÉ

Note : La sécurité doit être intégrée "by design" dès la conception du pipeline, et non ajoutée a posteriori.

- Concepts des processus ETL (Extraction, Transformation, Chargement)
- Outils ETL pour l'IoT : Apache NiFi, Talend
- **Développement de pipelines ETL pour l'intégration des données IoT**
- Choix des bases de données selon les besoins applicatifs
- Intégration des données IoT dans des bases relationnelles (MySQL, PostgreSQL)
- Intégration des données IoT dans des bases NoSQL (MongoDB, Cassandra)

- Introduction aux services cloud pour l'IoT : AWS IoT, Azure IoT, Google Cloud IoT
- Intégration des données IoT avec les services cloud
- Stockage, visualisation et analyse des données IoT sur le cloud
- Exemples d'architectures cloud pour l'IoT

Sécurité du Pipeline IoT

Une préoccupation constante à chaque étape du flux

Mesures indispensables pour garantir l'intégrité et la confidentialité des données.



Authentification

Identification unique de chaque objet
Certificats X.509 ou clés d'API
Prévention des appareils non autorisés



Chiffrement

Données en transit (TLS/SSL)
Données au repos (AES-256)
Protection contre l'interception



Gestion des Accès

Modèle RBAC (Rôles)
Principe du moindre privilège
Gestion des identités (IAM)

DISPONIBILITÉ

Note : La sécurité doit être intégrée "by design" dès la conception du pipeline, et non ajoutée a posteriori.

- Concepts des processus ETL (Extraction, Transformation, Chargement)
- Outils ETL pour l'IoT : Apache NiFi, Talend
- Développement de pipelines ETL pour l'intégration des données IoT
- **Choix des bases de données selon les besoins applicatifs**
- Intégration des données IoT dans des bases relationnelles (MySQL, PostgreSQL)
- Intégration des données IoT dans des bases NoSQL (MongoDB, Cassandra)

- Introduction aux services cloud pour l'IoT : AWS IoT, Azure IoT, Google Cloud IoT
- Intégration des données IoT avec les services cloud
- Stockage, visualisation et analyse des données IoT sur le cloud
- Exemples d'architectures cloud pour l'IoT

- Concepts des processus ETL (Extraction, Transformation, Chargement)
- Outils ETL pour l'IoT : Apache NiFi, Talend
- Développement de pipelines ETL pour l'intégration des données IoT
- Choix des bases de données selon les besoins applicatifs
- **Intégration des données IoT dans des bases relationnelles (MySQL, PostgreSQL)**
- Intégration des données IoT dans des bases NoSQL (MongoDB, Cassandra)

- Introduction aux services cloud pour l'IoT : AWS IoT, Azure IoT, Google Cloud IoT
- Intégration des données IoT avec les services cloud
- Stockage, visualisation et analyse des données IoT sur le cloud
- Exemples d'architectures cloud pour l'IoT

Intégration des données IoT dans des bases relationnelles (MySQL, PostgreSQL)

Un pipeline IoT suit une séquence logique pour transformer les données brutes des capteurs en **informations actionnables**.

Étapes principales

Collecte: Données reçues via MQTT, API, Kafka, etc.

Transformation: **Nettoyage** (suppression erreurs), **Normalisation** (formats, unités), **Ajout de métadonnées** (ID capteur, localisation),

Chargement (Load): Insertion dans des tables SQL (MySQL/PostgreSQL)

Exemple de structure:

- **capteurs** (id, type, localisation)
- **mesures** (id, capteur_id, valeur, timestamp)

Avantages: ✓ Données Structurées. ✓ Requêtes puissantes (JOIN, agrégations). ✓ Gestion des Métadonnées ✓ Idéal pour reporting et dashboards. ✓ Fiabilité (ACID).

```
1 SELECT
2     c.localisation,                -- Colonne provenant de la table capteurs : indique l'emplacement du capteur
3     AVG(m.valeur) AS moyenne_valeur -- Fonction d'agrégation : calcule la moyenne des valeurs mesurées
4 FROM mesures m                    -- Table principale contenant les mesures (alias m)
5
6 -- Jointure entre la table mesures et la table capteurs
7 -- Permet de récupérer les informations du capteur associé à chaque mesure
8 JOIN capteurs c
9     ON m.capteur_id = c.id         -- Condition de jointure : associe chaque mesure à son capteur
10
11 -- Filtrer uniquement les capteurs de type "temperature"
12 -- Cela permet de ne garder que les mesures liées à la température
13 WHERE c.type = 'temperature'
14
15 -- Regrouper les résultats par localisation
16 -- Nécessaire pour appliquer la fonction AVG sur chaque groupe de localisation
17 GROUP BY c.localisation;
18
19 -- Résultat final :
20 -- Pour chaque localisation, on obtient la moyenne des valeurs mesurées
21 -- uniquement pour les capteurs de type "temperature"
```

Exemple de requête

- Concepts des processus ETL (Extraction, Transformation, Chargement)
- Outils ETL pour l'IoT : Apache NiFi, Talend
- Développement de pipelines ETL pour l'intégration des données IoT
- Choix des bases de données selon les besoins applicatifs
- **Intégration des données IoT dans des bases relationnelles (MySQL, PostgreSQL)**
- Intégration des données IoT dans des bases NoSQL (MongoDB, Cassandra)

- Introduction aux services cloud pour l'IoT : AWS IoT, Azure IoT, Google Cloud IoT
- Intégration des données IoT avec les services cloud
- Stockage, visualisation et analyse des données IoT sur le cloud
- Exemples d'architectures cloud pour l'IoT

Intégration des données IoT dans des bases relationnelles (MySQL, PostgreSQL)

Avantages de SQL en IoT ?



Fiabilité et Conformité ACID

Garantit l'**Atomicité**, la **Consistance**, l'**Isolation** et la **Durabilité**.
Crucial pour les données critiques et l'intégrité des transactions.



Requêtes Puissantes

Le langage SQL permet des jointures puissantes entre les données de capteurs et les contextes métier (utilisateurs, bâtiments, zones).



Gestion des Métadonnées

Idéal pour stocker les informations sur les appareils (**ID**, **localisation**, **type**), les configurations utilisateurs et les inventaires.



Données Structurées

Convient parfaitement aux données qui ne changent pas fréquemment ou qui sont déjà agrégées pour le reporting.

i Le SQL reste indispensable pour le contexte et la cohérence des systèmes IoT.

- Concepts des processus ETL (Extraction, Transformation, Chargement)
- Outils ETL pour l'IoT : Apache NiFi, Talend
- Développement de pipelines ETL pour l'intégration des données IoT
- Choix des bases de données selon les besoins applicatifs
- **Intégration des données IoT dans des bases relationnelles (MySQL, PostgreSQL)**
- Intégration des données IoT dans des bases NoSQL (MongoDB, Cassandra)

- Introduction aux services cloud pour l'IoT : AWS IoT, Azure IoT, Google Cloud IoT
- Intégration des données IoT avec les services cloud
- Stockage, visualisation et analyse des données IoT sur le cloud
- Exemples d'architectures cloud pour l'IoT

PostgreSQL

Points Forts du SGBDR

Extensibilité & Standardisation

Conformité SQL stricte et support de types de données personnalisés.

JSONB (Semi-structuré)

Stockage et indexation performante de documents JSON pour les capteurs hétérogènes.

IoT ESSENTIAL

Robustesse & Fiabilité

Gestion avancée des transactions et des charges de travail complexes.

Focus : PostGIS & IoT

L'extension PostGIS

Incontournable pour les applications IoT nécessitant une gestion avancée des données géospatiales (GPS, zones, trajectoires).

- ✓ Suivi de flotte de véhicules ou drones
- ✓ Requêtes de proximité (Geofencing)
- ✓ Analyse spatiale complexe

Cas d'usage : Données Contextualisées

"Idéal pour croiser les flux de capteurs temps réel avec des données contextuelles fixes (localisation précise, type de bâtiment, zones géographiques)."

- Concepts des processus ETL (Extraction, Transformation, Chargement)
- Outils ETL pour l'IoT : Apache NiFi, Talend
- Développement de pipelines ETL pour l'intégration des données IoT
- Choix des bases de données selon les besoins applicatifs
- **Intégration des données IoT dans des bases relationnelles (MySQL, PostgreSQL)**
- Intégration des données IoT dans des bases NoSQL (MongoDB, Cassandra)

- Introduction aux services cloud pour l'IoT : AWS IoT, Azure IoT, Google Cloud IoT
- Intégration des données IoT avec les services cloud
- Stockage, visualisation et analyse des données IoT sur le cloud
- Exemples d'architectures cloud pour l'IoT

Modèle relationnel capteurs

Logique de Structuration

- ✓ **Normalisation** : Séparation des métadonnées (statiques) et des mesures (dynamiques).
- ✓ **Intégrité Référentielle** : Utilisation de clés étrangères pour garantir la cohérence entre capteurs et relevés.
- ✓ **Optimisation** : Indexation sur les IDs et les horodatages pour des requêtes temporelles rapides.

Note : Ce modèle est idéal pour MySQL ou PostgreSQL dans la gestion de parcs d'appareils et le suivi des configurations.

Types_Capteurs

type_id	INT
nom_mesure	VARCHAR
unite	VARCHAR

Appareils

device_id	VARCHAR
nom_device	VARCHAR
<i>type_id</i>	INT
localisation	POINT

Relevés (Séries Temporelles)

releve_id	BIGINT	valeur	DECIMAL
<i>device_id</i>	VARCHAR	horodatage	TIMESTAMP

- Concepts des processus ETL (Extraction, Transformation, Chargement)
- Outils ETL pour l'IoT : Apache NiFi, Talend
- Développement de pipelines ETL pour l'intégration des données IoT
- Choix des bases de données selon les besoins applicatifs
- **Intégration des données IoT dans des bases relationnelles (MySQL, PostgreSQL)**
- Intégration des données IoT dans des bases NoSQL (MongoDB, Cassandra)

- Introduction aux services cloud pour l'IoT : AWS IoT, Azure IoT, Google Cloud IoT
- Intégration des données IoT avec les services cloud
- Stockage, visualisation et analyse des données IoT sur le cloud
- Exemples d'architectures cloud pour l'IoT

Requêtes SQL exemples



Analyse des Moyennes

Calcul de la température moyenne par appareil sur les dernières 24h.

```
24 SELECT
25     device_id,                -- Identifier chaque appareil
26     AVG(value) AS avg_temp    -- Calculer la température moyenne
27 FROM sensor_readings        -- Table des mesures des capteurs
28
29 WHERE sensor_type = 'temperature' -- Garder uniquement les capteurs de température
30 AND timestamp > NOW() - INTERVAL '1 day' -- Prendre les données des dernières 24h
31
32 GROUP BY device_id          -- Regrouper les résultats par appareil
33
34 HAVING AVG(value) > 45;      -- Filtrer : moyenne > 45°C uniquement
```

Permet de détecter des surchauffes locales(température élevée
>45°C) par zone ou appareil.

✓ Jointures (Inner/Left)

✓ Agrégations (Group By)

✓ Filtres Temporels



Identification des Pannes

Jointure entre données de capteurs et inventaire pour localiser les batteries faibles.

```
37 SELECT
38     d.name,                    -- Nom de l'appareil
39     d.location,                -- Localisation de l'appareil
40     r.battery_level            -- Niveau de batterie mesuré
41 FROM devices d                -- Table des appareils
42
43 JOIN sensor_readings r        -- Associer chaque appareil à ses mesures
44     ON d.id = r.device_id
45
46 WHERE r.battery_level < 15     -- Garder les appareils avec batterie faible (<15%)
47 AND d.status = 'active'       -- Seulement les appareils actifs
48
49 ORDER BY r.battery_level ASC; -- Trier du plus faible au plus élevé
```

Actionable : Génère une liste d'intervention pour les techniciens de maintenance.
Afficher les appareils actifs avec batterie faible, triés du plus critique au moins critique.

- Concepts des processus ETL (Extraction, Transformation, Chargement)
- Outils ETL pour l'IoT : Apache NiFi, Talend
- Développement de pipelines ETL pour l'intégration des données IoT
- Choix des bases de données selon les besoins applicatifs
- **Intégration des données IoT dans des bases relationnelles (MySQL, PostgreSQL)**
- Intégration des données IoT dans des bases NoSQL (MongoDB, Cassandra)

- Introduction aux services cloud pour l'IoT : AWS IoT, Azure IoT, Google Cloud IoT
- Intégration des données IoT avec les services cloud
- Stockage, visualisation et analyse des données IoT sur le cloud
- Exemples d'architectures cloud pour l'IoT

Limites du SQL en IoT



Scalabilité Horizontale Limitée

Le SQL privilégie la **scalabilité verticale** (serveur plus puissant). En IoT, le volume massif nécessite une distribution sur plusieurs serveurs, complexe à gérer avec ACID.



Rigidité des Schémas

Les capteurs IoT sont hétérogènes. Un **schéma SQL fixe** impose des **migrations coûteuses** et des colonnes vides (NULL) dès que les types de données évoluent.



Performance en Écriture

Les contraintes d'intégrité et l'indexation ralentissent les insertions massives de séries temporelles (millions d'événements par seconde).



Jointures Coûteuses

Sur des milliards de lignes, les jointures complexes dégradent drastiquement les temps de réponse, rendant l'**analyse en temps réel difficile**.

"Le SQL reste excellent pour les **métadonnées structurées**, mais atteint ses limites pour les **métadonnées flexibles et variables les flux de données brutes haute fréquence**."

- Concepts des processus ETL (Extraction, Transformation, Chargement)
- Outils ETL pour l'IoT : Apache NiFi, Talend
- Développement de pipelines ETL pour l'intégration des données IoT
- Choix des bases de données selon les besoins applicatifs
- Intégration des données IoT dans des bases relationnelles (MySQL, PostgreSQL)
- Intégration des données IoT dans des bases NoSQL (MongoDB, Cassandra)

- Introduction aux services cloud pour l'IoT : AWS IoT, Azure IoT, Google Cloud IoT
- Intégration des données IoT avec les services cloud
- Stockage, visualisation et analyse des données IoT sur le cloud
- Exemples d'architectures cloud pour l'IoT

NoSQL

Le NoSQL répond aux défis de vitesse, de volume et de variété (3V) des données IoT là où le SQL traditionnel atteint ses limites.

Schémas Flexibles

Pas de structure rigide. Idéal pour des capteurs hétérogènes dont les formats de données évoluent rapidement.

Scalabilité Horizontale

Gestion de volumes massifs en ajoutant simplement des serveurs (nœuds) de manière distribuée et économique.

Haute Disponibilité

Conçu pour rester opérationnel même en cas de défaillance d'un nœud. Crucial pour les applications critiques.

Écritures Massives

Optimisé pour des taux d'insertion extrêmement élevés, indispensable pour ingérer le flux continu des capteurs.

Gestion des Séries Temporelles

Structure adaptée au stockage et à l'interrogation rapide de données horodatées, nature même des relevés IoT.

- Concepts des processus ETL (Extraction, Transformation, Chargement)
- Outils ETL pour l'IoT : Apache NiFi, Talend
- Développement de pipelines ETL pour l'intégration des données IoT
- Choix des bases de données selon les besoins applicatifs
- Intégration des données IoT dans des bases relationnelles (MySQL, PostgreSQL)
- Intégration des données IoT dans des bases NoSQL (MongoDB, Cassandra)

- Introduction aux services cloud pour l'IoT : AWS IoT, Azure IoT, Google Cloud IoT
- Intégration des données IoT avec les services cloud
- Stockage, visualisation et analyse des données IoT sur le cloud
- Exemples d'architectures cloud pour l'IoT

MongoDB

Pourquoi MongoDB pour l'IoT ?

Une approche orientée document (BSON) pour gérer l'hétérogénéité des capteurs.



Schéma Flexible

Schéma dynamique permettant d'ajouter des champs sans altérer la base. Chaque document peut avoir une structure différente. Idéal pour les attributs variés sans modifier le schéma global.



Évolutivité & Sharding

Supporte la réplication et le partitionnement des données (sharding) pour une scalabilité horizontale massive.



Optimisation BSON

Représentation binaire de JSON, permettant une traversée et un stockage plus rapides des données binaires.

Structure imbriquée facilitant l'accès aux données contextuelles.

EXEMPLE DE Structure Documentaire

JSON (IoT)

```
1  {
2    "deviceId": "sensor_temp_001",
3    "timestamp": "2025-10-27T10:30:00Z",
4
5    "location": {
6      "latitude": 48.8566,
7      "longitude": 2.3522
8    },
9
10   "temperature": {
11     "value": 22.5,
12     "unit": "Celsius"
13   },
14
15   "humidity": {
16     "value": 60,
17     "unit": "%"
18   },
19
20   "batteryLevel": 85
21 }
```

- Concepts des processus ETL (Extraction, Transformation, Chargement)
- Outils ETL pour l'IoT : Apache NiFi, Talend
- Développement de pipelines ETL pour l'intégration des données IoT
- Choix des bases de données selon les besoins applicatifs
- Intégration des données IoT dans des bases relationnelles (MySQL, PostgreSQL)
- Intégration des données IoT dans des bases NoSQL (MongoDB, Cassandra)

- Introduction aux services cloud pour l'IoT : AWS IoT, Azure IoT, Google Cloud IoT
- Intégration des données IoT avec les services cloud
- Stockage, visualisation et analyse des données IoT sur le cloud
- Exemples d'architectures cloud pour l'IoT

Apache Cassandra

Une Architecture Décentralisée

Cassandra est une base de données **NoSQL distribuée**, **orientée colonne**, conçue pour **gérer des volumes de données colossaux sur un grand nombre de serveurs**.

Stockage de Séries Temporelles Massives

Idéal pour l'IoT

Sa structure permet d'ingérer des **flux continus de données capteurs à une vitesse exceptionnelle** tout en garantissant une **lecture performante des séries temporelles**.

- ✓ Modèle de données orienté colonne
- ✓ Langage de requête **CQL** (proche du SQL)
- ✓ Réplication **multi-centres** de données



Pas de Point Unique de Défaillance

Architecture "**Masterless**" : chaque nœud du cluster est identique. Si un serveur tombe, le système continue de fonctionner sans interruption.



Scalabilité Linéaire

La capacité de stockage et la performance augmentent proportionnellement à l'ajout de nouveaux serveurs (**Scalabilité horizontale**).



Haute Disponibilité

Conçue pour garantir une disponibilité maximale (**Always-on**), cruciale pour les applications industrielles critiques et le monitoring 24/7.



Performance en Écriture

Optimisée pour des **taux d'écriture extrêmement élevés**, permettant l'**ingestion simultanée de millions de points de données IoT**.

- Concepts des processus ETL (Extraction, Transformation, Chargement)
- Outils ETL pour l'IoT : Apache NiFi, Talend
- Développement de pipelines ETL pour l'intégration des données IoT
- Choix des bases de données selon les besoins applicatifs
- Intégration des données IoT dans des bases relationnelles (MySQL, PostgreSQL)
- Intégration des données IoT dans des bases NoSQL (MongoDB, Cassandra)

- Introduction aux services cloud pour l'IoT : AWS IoT, Azure IoT, Google Cloud IoT
- Intégration des données IoT avec les services cloud
- Stockage, visualisation et analyse des données IoT sur le cloud
- Exemples d'architectures cloud pour l'IoT

Exemple de document MongoDB (format BSON/JSON) représentant des données d'un capteur IoT.

Modèle NoSQL IoT

Structure Documentaire

- ✓ **Flexibilité Totale** : Schéma dynamique permettant d'ajouter des champs sans altérer la base.
- ✓ **Modèle Orienté Objet** : Correspondance directe avec les objets de données applicatifs.
- ✓ **Encapsulation** : Métadonnées (ID, localisation) et mesures (valeurs) dans un seul bloc.

"Idéal pour les capteurs hétérogènes qui envoient des attributs variés (température, humidité, batterie) sans schéma fixe."

```
1 {
2   "deviceId": "sensor_temp_001",
3   "timestamp": "2025-10-27T10:30:00Z",
4   "location": {
5     "latitude": 48.8566,
6     "longitude": 2.3522
7   },
8   "measurements": {
9     "temperature": {
10      "value": 22.5,
11      "unit": "Celsius"
12    },
13    "humidity": {
14      "value": 60,
15      "unit": "%"
16    }
17  },
18  "batteryLevel": 85,
19  "firmware": "v2.1.4"
20 }
```

- Concepts des processus ETL (Extraction, Transformation, Chargement)
- Outils ETL pour l'IoT : Apache NiFi, Talend
- Développement de pipelines ETL pour l'intégration des données IoT
- Choix des bases de données selon les besoins applicatifs
- Intégration des données IoT dans des bases relationnelles (MySQL, PostgreSQL)
- Intégration des données IoT dans des bases NoSQL (MongoDB, Cassandra)

- Introduction aux services cloud pour l'IoT : AWS IoT, Azure IoT, Google Cloud IoT
- Intégration des données IoT avec les services cloud
- Stockage, visualisation et analyse des données IoT sur le cloud
- Exemples d'architectures cloud pour l'IoT

Comparaison SQL vs NoSQL

CRITÈRE	SQL (RELATIONNEL)	NOSQL (NON-RELATIONNEL)
Modèle de données	Tables (Lignes/Colonnes)	Documents, Colonnes, Clé-Valeur
Schéma	Fixe et prédéfini (Rigide)	Dynamique et flexible
Scalabilité	Verticale (Upgrade serveur)	Horizontale (Ajout de serveurs)
Intégrité	ACID (Forte cohérence)	BASE (Cohérence éventuelle)
Usage IoT Typique	Métadonnées, Config, Utilisateurs	Séries temporelles, Flux massifs

✓ Choisir SQL quand...

- Les données sont hautement structurées.
- L'intégrité transactionnelle est critique.
- On gère le parc d'appareils (Inventaire).

⚡ Choisir NoSQL quand...

- On reçoit des téraoctets de données brutes.
- Les capteurs ont des formats variés.
- Une haute disponibilité est requise.

- Concepts des processus ETL (Extraction, Transformation, Chargement)
- Outils ETL pour l'IoT : Apache NiFi, Talend
- Développement de pipelines ETL pour l'intégration des données IoT
- Choix des bases de données selon les besoins applicatifs
- Intégration des données IoT dans des bases relationnelles (MySQL, PostgreSQL)
- Intégration des données IoT dans des bases NoSQL (MongoDB, Cassandra)

- Introduction aux services cloud pour l'IoT : AWS IoT, Azure IoT, Google Cloud IoT
- Intégration des données IoT avec les services cloud
- Stockage, visualisation et analyse des données IoT sur le cloud
- Exemples d'architectures cloud pour l'IoT

Pourquoi utiliser le Cloud pour l'IoT ?

L'adoption du Cloud permet de surmonter les défis de gestion de données massives tout en optimisant la flexibilité et les coûts opérationnels.



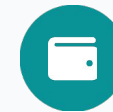
Scalabilité Élastique

- ✓ Adaptation automatique aux pics de charge.
- ✓ Gestion de millions d'appareils sans interruption.



Services Managés

- ✓ Outils pré-configurés pour l'ingestion et l'analyse.
- ✓ Réduction de la charge opérationnelle IT.



Modèle Pay-as-you-go

- ✓ Conversion des CapEx en OpEx.
- ✓ Paiement uniquement pour les ressources consommées.



Sécurité et Écosystème

Intégration native avec l'IA/ML et mécanismes de protection des données gérés par le fournisseur.

INFRASTRUCTURE MONDIALE
Disponibilité Multi-Régions

- Concepts des processus ETL (Extraction, Transformation, Chargement)
- Outils ETL pour l'IoT : Apache NiFi, Talend
- Développement de pipelines ETL pour l'intégration des données IoT
- Choix des bases de données selon les besoins applicatifs
- Intégration des données IoT dans des bases relationnelles (MySQL, PostgreSQL)
- Intégration des données IoT dans des bases NoSQL (MongoDB, Cassandra)

- Introduction aux services cloud pour l'IoT : AWS IoT, Azure IoT, Google Cloud IoT
- Intégration des données IoT avec les services cloud
- Stockage, visualisation et analyse des données IoT sur le cloud
- Exemples d'architectures cloud pour l'IoT

Écosystème AWS IoT

Amazon Web Services



AWS IoT Core — Le Service Central

✓ Connectivité massive d'appareils

✓ MQTT, HTTPS, LoRaWAN

✓ Authentification & Autorisation

AWS IoT Greengrass

Étend le cloud à la périphérie (Edge). Exécution locale de fonctions Lambda et synchronisation sécurisée même hors ligne.

IoT Device Management

Enregistrement, organisation et surveillance à distance de flottes d'appareils IoT de manière automatisée.

AWS IoT Analytics

Service managé pour collecter, traiter et stocker les données à grande échelle. Permet des requêtes complexes et statistiques.

IoT Device Defender

Sécurisation des flottes par détection de comportements anormaux et alertes proactives pour les opérateurs.

Intégration native : L'écosystème AWS IoT s'intègre parfaitement avec Amazon S3 (Data Lakes), AWS Lambda (Serverless) et Amazon SageMaker (IA/ML).

- Concepts des processus ETL (Extraction, Transformation, Chargement)
- Outils ETL pour l'IoT : Apache NiFi, Talend
- Développement de pipelines ETL pour l'intégration des données IoT
- Choix des bases de données selon les besoins applicatifs
- Intégration des données IoT dans des bases relationnelles (MySQL, PostgreSQL)
- Intégration des données IoT dans des bases NoSQL (MongoDB, Cassandra)

- Introduction aux services cloud pour l'IoT : AWS IoT, Azure IoT, Google Cloud IoT
- Intégration des données IoT avec les services cloud
- Stockage, visualisation et analyse des données IoT sur le cloud
- Exemples d'architectures cloud pour l'IoT

Azure IoT

Écosystème Microsoft

Une plateforme complète pour connecter, surveiller et gérer des milliards d'actifs IoT avec une sécurité intégrée de bout en bout.



Azure IoT Hub

Passerelle cloud centrale pour l'ingestion de messages bidirectionnelle et la gestion des identités.



Stream Analytics

Moteur d'analyse en temps réel pour traiter des millions d'événements par seconde via SQL.



Azure IoT Edge

Déploiement de l'IA et de la logique métier directement sur les appareils périphériques.



Cosmos DB / TSI

Stockage multi-modèle et analyse de séries temporelles pour une visibilité historique.

✓ Ingestion à grande échelle

Supporte les protocoles standard (MQTT, AMQP, HTTPS) avec une mise à l'échelle élastique automatique.

✓ Traitement Temps Réel

Complex Event Processing (CEP) permettant de détecter des anomalies ou de déclencher des alertes instantanément.

- Concepts des processus ETL (Extraction, Transformation, Chargement)
 - Outils ETL pour l’IoT : Apache NiFi, Talend
 - Développement de pipelines ETL pour l’intégration des données IoT
 - Choix des bases de données selon les besoins applicatifs
 - Intégration des données IoT dans des bases relationnelles (MySQL, PostgreSQL)
 - Intégration des données IoT dans des bases NoSQL (MongoDB, Cassandra)
- Introduction aux services cloud pour l’IoT : AWS IoT, Azure IoT, Google Cloud IoT
 - Intégration des données IoT avec les services cloud
 - Stockage, visualisation et analyse des données IoT sur le cloud
 - Exemples d’architectures cloud pour l’IoT

Comparaison cloud providers

 Analyse comparative des leaders du marché IoT

☰ Critère	 AWS IoT	 Azure IoT	 Google Cloud
 Ingestion	IoT Core: MQTT, HTTPS, LoRaWAN. Très mature.	IoT Hub: MQTT, AMQP, HTTPS. Support bi-directionnel fort.	Pub/Sub: Ingestion massive, scalabilité globale native.
 Edge	Greengrass: Lambda local, ML Inference, Offline sync.	IoT Edge: Modules Docker, support AI, intégration Windows.	Edge TPU: Optimisé pour l'IA locale (Coral).
 Analytique	IoT Analytics: Nettoyage, enrichissement, requêtes SQL.	Stream Analytics: Temps réel, Digital Twins pour simulation.	Dataflow / BigQuery: Analyse Big Data et ML intégrée.
 Sécurité	Device Defender: Audit continu, détection d'anomalies.	Defender for IoT: Sécurité complète (OT/IT), Threat intel.	IAM & VPC: Sécurité réseau robuste, gestion fine.
 Ecosystème	Intégration profonde (S3, Lambda, Kinesis, DynamoDB).	Power BI, Time Series Insights, services industriels.	Looker (Viz), Vertex AI, intégration Android/Firebase.

VERDICT AWS

Le plus complet et mature pour les flottes hétérogènes massives.

VERDICT AZURE

Idéal pour le secteur industriel et l'intégration Microsoft Enterprise.

VERDICT GOOGLE

Excellent pour l'analyse Big Data et l'IA de pointe.

- Concepts des processus ETL (Extraction, Transformation, Chargement)
- Outils ETL pour l'IoT : Apache NiFi, Talend
- Développement de pipelines ETL pour l'intégration des données IoT
- Choix des bases de données selon les besoins applicatifs
- Intégration des données IoT dans des bases relationnelles (MySQL, PostgreSQL)
- Intégration des données IoT dans des bases NoSQL (MongoDB, Cassandra)

- Introduction aux services cloud pour l'IoT : AWS IoT, Azure IoT, Google Cloud IoT
- **Intégration des données IoT avec les services cloud**
- Stockage, visualisation et analyse des données IoT sur le cloud
- Exemples d'architectures cloud pour l'IoT

- Concepts des processus ETL (Extraction, Transformation, Chargement)
- Outils ETL pour l'IoT : Apache NiFi, Talend
- Développement de pipelines ETL pour l'intégration des données IoT
- Choix des bases de données selon les besoins applicatifs
- Intégration des données IoT dans des bases relationnelles (MySQL, PostgreSQL)
- Intégration des données IoT dans des bases NoSQL (MongoDB, Cassandra)

- Introduction aux services cloud pour l'IoT : AWS IoT, Azure IoT, Google Cloud IoT
- Intégration des données IoT avec les services cloud
- **Stockage, visualisation et analyse des données IoT sur le cloud**
- Exemples d'architectures cloud pour l'IoT

Stockage + Analyse Cloud



Data Lake (Lac de Données)

- ✓ **Stockage Brut** : Conservation des données dans leur format original (JSON, CSV, binaire).
- ✓ **Archivage Long Terme** : Coût de stockage extrêmement bas (ex: Amazon S3 Glacier).
- ✓ **Flexibilité** : Permet une analyse rétrospective sur des données non encore traitées.



Analyse Avancée & IA/ML

- ✓ **Data Warehouse** : Données structurées et optimisées pour des requêtes SQL ultra-rapides.
- ✓ **Modèles ML** : Entraînement d'algorithmes de maintenance prédictive (ex: SageMaker, Azure ML).
- ✓ **Business Intelligence** : Connexion directe aux outils de visualisation (QuickSight, Power BI).



Scalabilité
Capacité illimitée



Sécurité Gérée
Chiffrement & IAM



Optimisation Coûts
Pay-as-you-go



Services Managés
Moins d'Ops

- Concepts des processus ETL (Extraction, Transformation, Chargement)
- Outils ETL pour l'IoT : Apache NiFi, Talend
- Développement de pipelines ETL pour l'intégration des données IoT
- Choix des bases de données selon les besoins applicatifs
- Intégration des données IoT dans des bases relationnelles (MySQL, PostgreSQL)
- Intégration des données IoT dans des bases NoSQL (MongoDB, Cassandra)

- Introduction aux services cloud pour l'IoT : AWS IoT, Azure IoT, Google Cloud IoT
- Intégration des données IoT avec les services cloud
- **Stockage, visualisation et analyse des données IoT sur le cloud**
- Exemples d'architectures cloud pour l'IoT

Tableaux de bord IoT (KPIs)



APPAREILS ACTIFS

1,240

VOLUME / HEURE

45 GB

ALERTES ACTIVES

12

DISPONIBILITÉ

99.9%

☰ Suivi et Monitoring

- **État de la Flotte :**
Visualisation en temps réel de la connectivité et du statut opérationnel des capteurs.
- **Gestion des Alertes :**
Seuils critiques, notifications push et historique des incidents par sévérité.
- **Métriques de Performance :**
Analyse de la latence, du débit de données et de la consommation d'énergie (Edge).

Répartition des Alertes (Dernières 24h)

