Tuto2 hdf

Version 1.0

2019-01-31

DATASOFT CONSULTING

Historique des modifications du document

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Auteur** |
| 2019-01-31 | 1.0 | Tuto2\_hdf | Abdelaziz HANI |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table des matières

1. INTRODUCTION 4

2. CONCEPTS 4

2.1 STREAM ANALYTICS MANAGER (SAM) 4

2.2 APACHE STORM 4

2.3 SCHEMA REGISTRY 4

2.4 KAFKA 4

2.5 VUE D'ENSEMBLE DE TRUCKING IOT REF APP 4

2.6 ÉTAPE 1: EXPLOREZ L'APPLICATION DATAFLOW 5

2.7 ÉTAPE 2: VIEW SCHEMA REGISTRY 7

2.8 ETAPE 3: ANALYSER L’ APPLICATION STREAM ANALYTICS 8

2.8.1 Vue d'ensemble de la toile SAM: 10

2.8.2 Vue d'ensemble de la topologie SAM: 10

2.9 ÉTAPE 4: VISUALISEZ LE MOTEUR STORM DANS SAM 10

3. RESUME 12

# INTRODUCTION

Dans ce didacticiel, vous apprendrez à déployer une application de diffusion en temps réel moderne. Cette application sert de cadre de référence pour le développement d'un pipeline de données volumineuses, comprenant une large gamme de cas d'utilisation et de puissants composants centraux réutilisables. Vous explorerez l'application NiFi DataFlow, les thèmes Kafka, les schémas et la SAM topology.

# CONCEPTS

## STREAM ANALYTICS MANAGER (SAM)

Stream Analytics Manager est un programme de glisser-déposer qui permet aux développeurs de traitement de flux de créer des topologies de données en quelques minutes par rapport à la pratique traditionnelle consistant à écrire plusieurs lignes de code. Une topologie est un graphe acyclique dirigé (DAG) de processeurs. Désormais, les utilisateurs peuvent configurer et optimiser la manière dont ils souhaitent que chaque composant ou processeur effectue des calculs sur les données. Ils peuvent effectuer des fenêtres, joindre plusieurs flux et manipuler d'autres données. SAM prend actuellement en charge le moteur de traitement de flux appelé Apache Storm, mais il prendra ultérieurement en charge d'autres moteurs tels que Spark et Flink. À ce moment-là, ce sera au choix de l'utilisateur sur le moteur de traitement de flux qu'il choisira.

## APACHE STORM

Apache Storm est le moteur de traitement informatique actuel pour Stream Analytics Manager. Une fois que l'utilisateur a créé sa topologie SAM, tout le traitement des données s'effectue dans une topologie Storm, qui est également un DAG, mais est composée de becs et de boulons avec des flux de n-uplets représentant les bords.

Un bec ingère généralement les données d'un sujet Kafka dans la topologie, tandis que les boulons effectuent tout le traitement. Ainsi, tous les mêmes composants de la topologie SAM sont représentés dans la topologie Storm, mais comme des goulottes et des boulons appropriés.

Storm est le système distribué, fiable et tolérant aux pannes Open Source, qui gère les analyses en temps réel, les modèles d’apprentissage par machine de scoring, les calculs statiques continus et l’application des paradigmes d’extraction, de transformation et de chargement (ETL).

## SCHEMA REGISTRY

Schema Registry (SR) stocke et récupère Avro Schemas via l'interface RESTful. SR stocke un historique de version contenant tous les schémas. Les sérialiseurs sont fournis pour se connecter aux clients Kafka responsables du stockage du schéma et récupérer les messages Kafka envoyés au format Avro.

## KAFKA

Apache Kafka est un système de messagerie open source basé sur la publication-abonnement, responsable du transfert des données d'une application à une autre.

## VUE D'ENSEMBLE DE TRUCKING IOT REF APP

L'application de référence IoT de camionnage est construite à l'aide de la plate-forme Hortonworks DataFlow.

Les données IoT de camionnage proviennent d'un simulateur d'événements de camion ingéré par Apache NiFi. NiFi envoie les données à des rubriques Kafka, qui sont ensuite ingérées par Stream Analytics Manager (SAM). Une explication plus détaillée du pipeline sera expliquée lorsque vous explorerez l'application NiFi DataFlow, Schema Registry et SAM.

## ÉTAPE 1: EXPLOREZ L'APPLICATION DATAFLOW

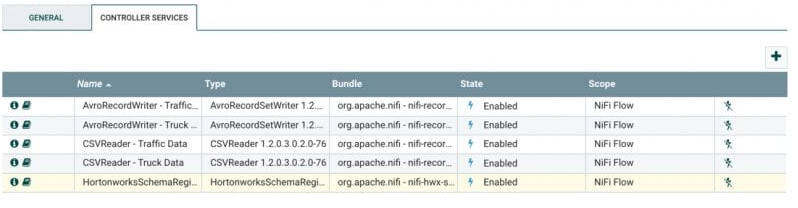
1. Ouvrez l'interface Utilisateur NiFi http://sandbox-hdf.hortonworks.com:9090/nifi/

2. Le modèle d’applications NiFi Dataflow: Une demo IoT de camionnage apparaîtra sur le canevas.

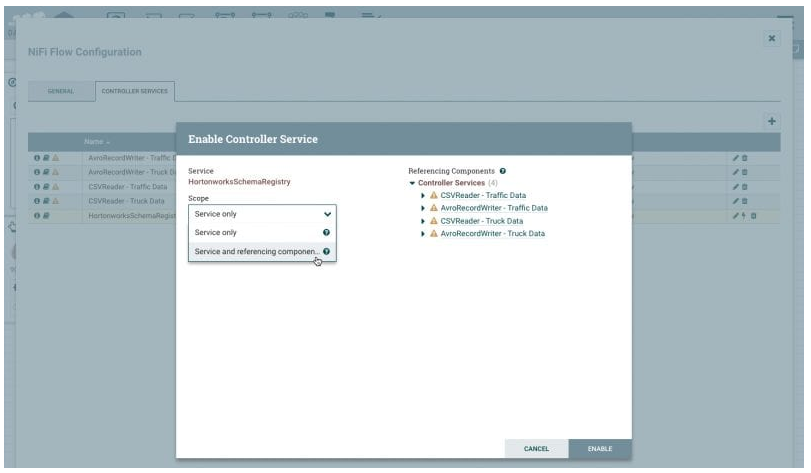


3. Sélectionnez l'icône nifi\_configuration Cliquez sur l'onglet Controller Services.

4. HortonworksSchemaRegistry doit être activé. S'il n'est pas activé, sélectionnez le symbole en forme d'éclair en regard de HortonworksSchemaRegistry.



5. Dans la fenêtre « Enable Controller Service», sous « Scope», sélectionnez « Service and referencing components». Puis Cliquez sur Activer.



Tous les services de contrôleur faisant référence à HortonworksSchemaRegistry seront également activés. Retournez au flux de données NiFi.

Vue d'ensemble des 7 processeurs du flux NiFi:

GetTruckingData - Simulator génère TruckData et TrafficData dans un fichier CSV délimité par des barres

RouteOnAttribute - filtre les données TrafficData et TruckData en flux de données distincts

|  |  |
| --- | --- |
| **DATA NAME** | **DATA FIELDS** |
| **TruckData** | eventTime, truckId, driverId, driverName, routeId, routeName, latitude, longitude, speed, eventType |
| **TrafficData** | eventTime, routeId, congestionLevel |

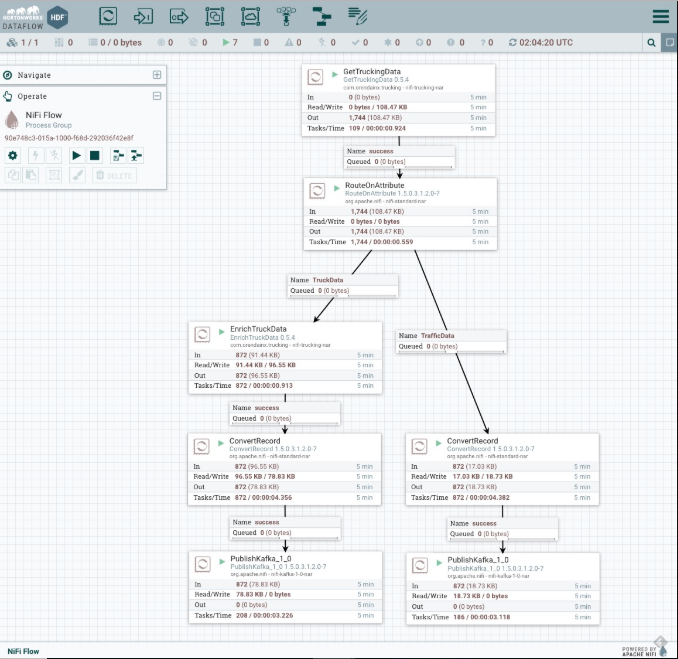
TruckData side of Flow

* **EnrichTruckData** - étiquettes sur trois champs jusqu'à la fin de TruckData: “brumeux”, “pluvieux”, “venteux”
* **ConvertRecord** - lit les données entrantes avec «CSVReader» et écrit les données Avro avec «AvroRecordSetWriter» en incorporant un schéma «trucking\_data\_truck\_enriched» dans chaque fichier de flux.
* **PublishKafka\_1\_0** - stocke les données Avro dans la rubrique Kafka «trucking\_data\_truck\_enriched», côté trafic de FlowData
* **ConvertRecord** - convertit les données CSV en données Avro intégrant un schéma «trucking\_data\_traffic» sur chaque fichier de flux
* **PublishKafka\_1\_0** - stocke les données Avro dans le sujet Kafka «trucking\_data\_traffic»

Vue d'ensemble des **5 controller services** utilisés dans le flux NiFi:

* **AvroRecordSetWriter** - Enriched Truck Data - écrit le contenu de RecordSet au format binaire Avro (schéma trucking\_data\_truck\_enriched)
* **AvroRecordSetWriter** - Données de trafic - écrit le contenu de RecordSet au format binaire Avro (schéma trucking\_data\_traffic)
* **CSVReader** - Enriched Truck Data - renvoie chaque ligne du fichier CSV sous forme d'enregistrement distinct (schéma trucking\_data\_truck\_enriched)
* **CSVReader** - Données de trafic - renvoie chaque ligne du fichier CSV sous la forme d'un enregistrement séparé (schéma trucking\_data\_traffic)
* **HortonworksSchemaRegistry** - fournit un service de registre de schéma pour une interaction avec le registre de schémas Hortonworks

6. Appuyez sur command + A ou sur control + A pour sélectionner tous les processeurs du flux de données NiFi, puis cliquez sur le bouton Démarrer.

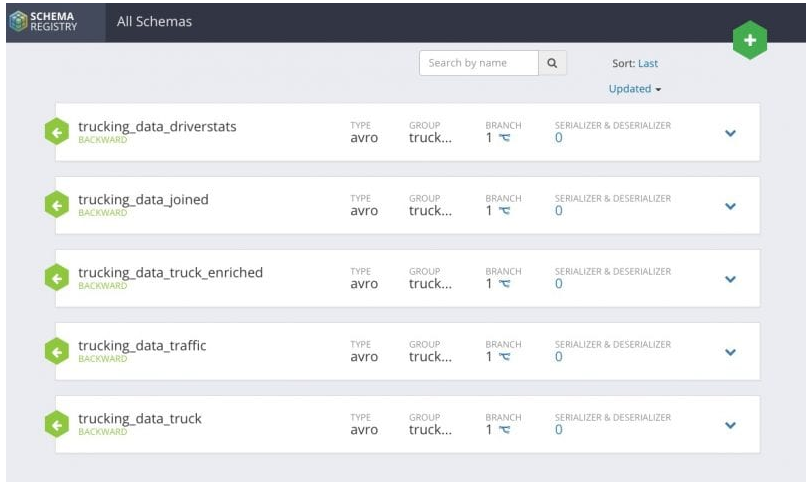


7. Pour réduire la consommation de ressources et l'encombrement, lorsque les processeurs PublishKafka\_1\_0 atteignent environ 500 enregistrements d'entrée, cliquez sur le bouton d'arrêt nifi\_stop. Cela prendra environ 1 à 2 minutes.

8. Arrêt du service NiFi**: Ambari -> NiFi -> Actions -> Arrêter**

## ÉTAPE 2: VIEW SCHEMA REGISTRY

1. Ouvrez l'interface utilisateur du SCHEMA REGISTRY à l'adresse <http://sandbox-hdf.hortonworks.com:7788/>.



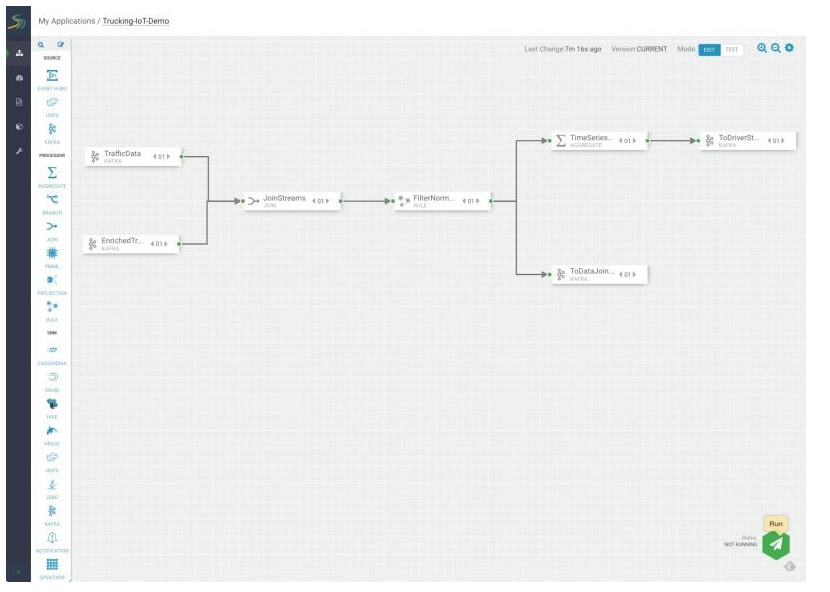
Vue d'ensemble des schémas essentiels du SCHEMA REGISTRY:

* trucking\_data\_joined - modèle pour un événement de camion provenant d’un ordinateur de bord (EnrichedTruckAndTrafficData)
* trucking\_data\_truck\_enriched - modèle d’événement de camion provenant de l’ordinateur de bord d’un camion (EnrichedTruckData)
* Modèle trucking\_data\_traffic pour eventTime, routeId, congestionLevel (TrafficData)

## ETAPE 3: ANALYSER L’ APPLICATION STREAM ANALYTICS

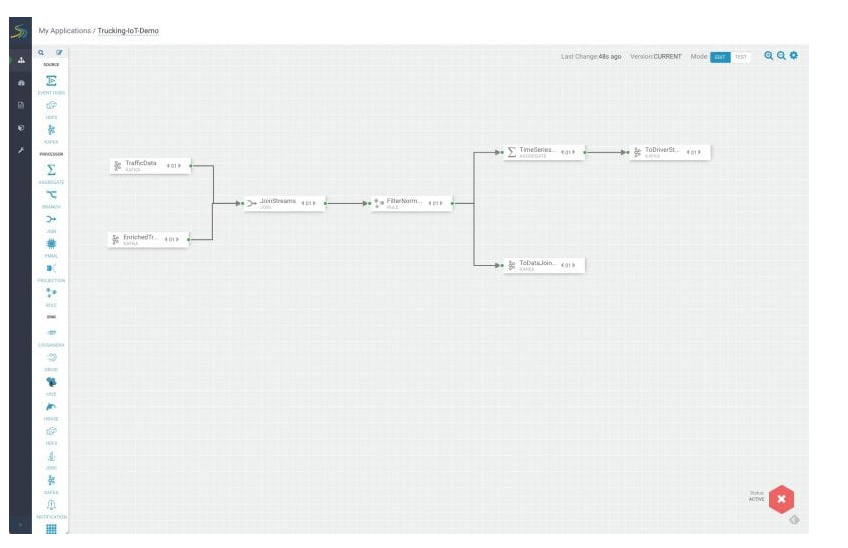
1. Ouvrez Stream Analytics Manager (SAM) à l’adresse http://sandbox-hdf.hortonworks.com:7777/

2. Cliquez sur Trucking-IoT\_Demo, puis sur le crayon vert dans le coin supérieur droit. Cela devrait afficher une image similaire à celle ci-dessous, cliquez sur le bouton Exécuter pour déployer la topologie:



Une fenêtre apparaît vous demandant si vous souhaitez continuer le déploiement, cliquez sur OK.

3. Vous recevrez une notification indiquant que l'application de topologie SAM a été déployée avec succès et votre topologie affichera Statut actif dans le coin inférieur droit.



### Vue d'ensemble de la toile SAM:

* My Applications: différents projects de topologie
* 1re barre latérale gauche: My Applications, Dashboard, Schema Registry, Model Registry, Configuration 2e barre latérale gauche: différents composants du flux (source, processor, sink)
* Icône d'engrenage: Configurer les paramètres de topologie
* Icône d'état: Démarrer ou arrêter la topologie

### Vue d'ensemble de la topologie SAM:

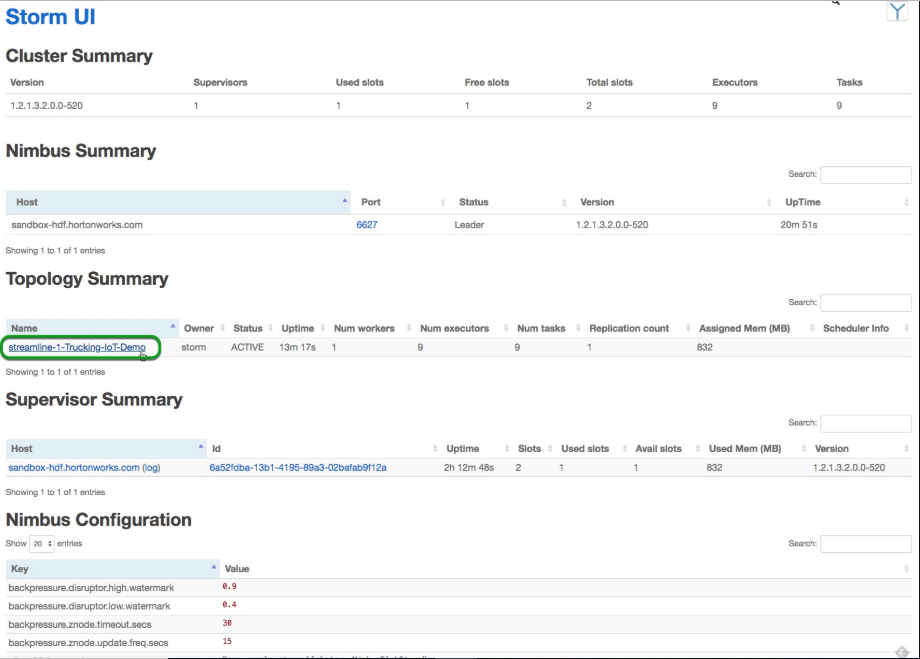
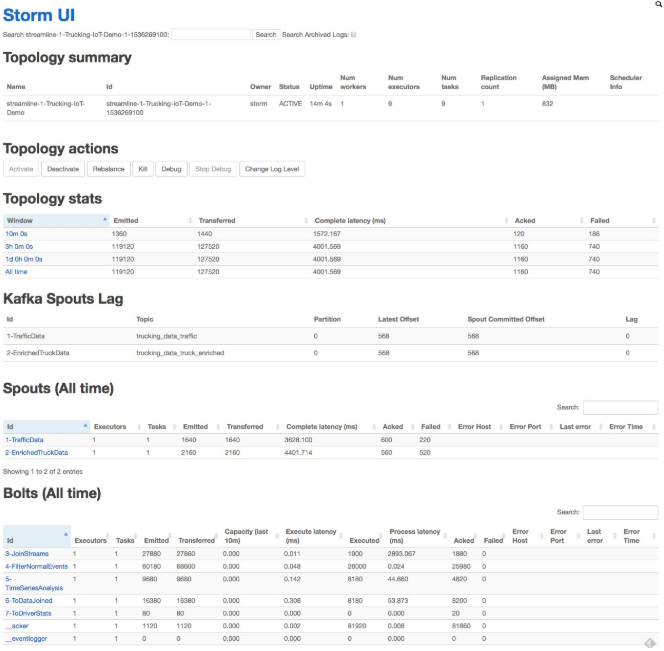
* **TrafficData** est le nom du composant source, qui extrait les données de la rubrique Kafka «trucking\_data\_traffic».
* **EnrichedTruckData** est le nom du composant source, extrait des données de la rubrique Kafka «trucking\_data\_truck\_enriched».
* **JoinStreams** rejoint les flux «TrafficData» et «EnrichedTruckData» par «routeId».
* **FilterNormalEvents** vérifie si des types d'événement non « Normal » se produisent, puis il les émettra.
* **TimeSeriesAnalysis** calcule la moyenne de 10 échantillons de vitesse sur une période de fenêtre de 10 secondes, calcule la somme sur la même période de fenêtre qu'avant pour les périodes brumeuses, pluvieuses, venteuses et eventTime individuellement.
* **ToDriverStats** stocke l'entrée de «TimeSeriesAnalysis»: driveId, routeId, averageSpeed, totalFog, totalRain, totalWind et totalViolations dans le sujet Kafka «trucking\_data\_driverstats».
* **ToDataJoined** stocke l'entrée de «FilterNormalEvents»: eventTime, congestionLevel, truckId, driverId, driverName, routeId, routeName, latitude, longitude, vitesse, eventType, brumeux, pluvieux et venteux dans le sujet de Kafka «trucking\_data\_joined».

## ÉTAPE 4: VISUALISEZ LE MOTEUR STORM DANS SAM

1. Depuis Ambari, cliquez sur **Storm> Storm UI**.

2. Cliquez sur Nom de la topologie: **streamline-1-Trucking-IoT-Demo** sous **Topology Summary**

Vue sur Storm Topology



Vous pouvez voir le nombre total de tuples émis (1360) et transférés (1440) après 10m 0 sous les statistiques de topologie pour l'ensemble de la topologie. Vous pouvez également voir des n-uplets individuels émis et transférés pour chaque bec et boulon individuels dans l’augmentation de la topologie. Si nous survolons l'un des becs ou des boulons du graphique, nous pouvons voir la quantité de données traitées et leur temps de latence.

* Topology Summary
* Topology Stats
* Topology Static Visualization
* Spout
* Bolts
* Topology Configuration

# RESUME

félicitations! Vous avez déployé la démo Trucking IoT qui traite les données d'événement de camion en utilisant l'application de flux de données NiFi pour séparer les données en deux flux: TruckData et TrafficData. Ces deux flux sont ensuite transmis dans deux files d'attente robustes Kafka balisées avec des schémas Schema Registry: trucking\_data\_traffic et \_trucking\_data\_truck\_enriched. La topologie SAM (Stream Analytics Manager) extrait ces données pour relier les deux flux (ou flux) par routId et filtrer les événements non normaux qui sont ensuite divisés en deux flux. Un flux est envoyé directement à un récepteur Kafka, l’autre flux est ensuite filtré avec un processeur global puis envoyé à un autre récepteur Kafka.