

Składowanie danych w systemach Big Data

Dokumentacja projektu zespołu KOALE

Adam Majczyk, Sabina Sidarovich, Damian Skowroński

Spis treści

Spis treści	2
Cel projektu i potencjalne korzyści z wdrożenia	3
Wykorzystywany stos architektoniczny	3
Planowany podział pracy w zespole	3
Opis zbiorów danych planowanych do wykorzystania w projekcie	4
Pobieranie danych	4
Dane lotnicze	4
Dane pogodowe	4
Przetwarzanie danych	4
Poziom brązowy	5
Poziom srebrny	5
Poziom złoty	6
Wgrywanie danych	6
Przykłady danych dostępnych dla warstwy prezentacyjnej	6
Testy	8
Podsumowanie finalnej wersji rozwiązania	8
Wymagania	9

Cel projektu i potencjalne korzyści z wdrożenia

Celem projektu jest stworzenie systemu pozyskiwania, przetwarzania i składowania danych związanych z lotami samolotów oraz danymi pogodowymi na terenie Polski. Projekt ma na celu dostarczanie aktualnych informacji na temat stanu pogody i lokalizacji samolotów, które mogą być wykorzystane do stworzenia wizualizacji sytuacji na mapie Polski w czasie rzeczywistym.

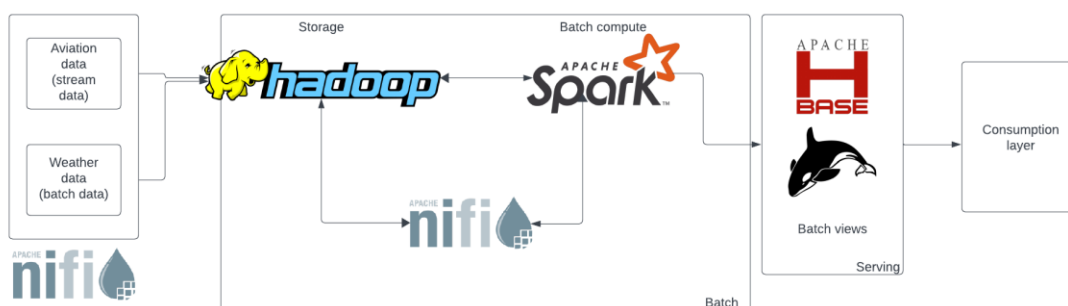
Przy wykorzystaniu odpowiedniej ilości danych i częstotliwości aktualizacji, system mógłby zostać wykorzystany do zarządzania ruchem lotniczym.

Dostęp do danych dotyczących pogody pozwala na lepsze planowanie tras lotów, co może przyczynić się do zwiększenia bezpieczeństwa lotów.

Zgromadzone dane mogą zostać wykorzystane do prowadzenia analiz, na przykład analizy trendów pogodowych.

Wykorzystywany stos architektoniczny

W projekcie wykorzystana została modyfikacja architektury lambda przedstawiona na poniższym diagramie (patrz Rys. 1).



Rysunek 1: Planowany schemat architektury

Dane są pozyskiwane przy użyciu skryptów napisanych w języku Python. Z uwagi na pewne ograniczenia techniczne, aktualnie nie jest możliwe zautomatyzowanie tego etapu. W kolejnych fazach systemu wykorzystane zostało narzędzie Apache NiFi. W celu przechowywania danych użyto Apache Hadoop (batch layer) oraz Apache HBase (serving layer). Apache Spark został zastosowany do przetwarzania danych. Ostatecznie, dane pobrane z Apache HBase zostały zwizualizowane za pomocą bibliotek języka Python.

Planowany podział pracy w zespole

Adam Majczyk:

- Stworzenie przepływów danych w Apache NiFi.
- Wizualizacje.
- Dokumentacja.

Sabina Sidarovich:

- Przygotowanie HDFS.
- Integracja Apache Hive.
- Dokumentacja.

Damian Skowroński:

- Wczytywanie danych.
- Kod w Apache Spark.

- Dokumentacja.

Opis zbiorów danych planowanych do wykorzystania w projekcie

Użyte zostały dane lotnicze ze źródła 1 oraz dane pogodowe ze źródła 2:

1. Dane o lotach:

[The OpenSky Network API documentation](#)

Informacje takie jak:

- Timestamp.
- Koordynaty.
- Nachylenie.
- Prędkość.
- Lotnisko startu i końca podróży.
- Inne.

Dostępne poprzez REST API. Pobierane dane są w formacie JSON. Każdy plik zawiera 100 rekordów. Pliki zawierające informacje dot. lotów są pobierane co 30 minut, natomiast pliki zawierające informacje o lotniskach, samolotach i ich rodzajach są pobierane co miesiąc. Pobierano jedynie loty z oraz do polskich lotnisk.

2. Dane pogodowe w Polsce

[Dobowe dane meteorologiczne](#) dostarczane przez Instytut Meteorologii i Gospodarki Wodnej - Państwowy Instytut Badawczy

Informacje takie jak:

- Miejsce pomiaru (nazwa stacji)
- Temperatury: rosy, minimalna, maksymalna, średnia (w ciągu dnia)
- Opady: rodzaj opadów, ilość opadów (mm)

Dostępne w postaci plików CSV oraz są pobierane co miesiąc. Jeden plik CSV zawiera od tysiąca do dziesięciu tysięcy wierszy. Dla każdego miesiąca pobierane są dwa pliki z różnymi zestawami informacji pogodowych, które później tworzą jedno źródło danych.

Dane są udostępniane z opóźnieniem dwóch miesięcy. Wobec tego wykorzystane zostały dane z poprzedniego roku aby zasymulować dane aktualne. Docelowo należałoby wykorzystać płatne rozwiązanie z danymi pogodowymi wyższej jakości i aktualizowanymi w czasie rzeczywistym.

Pobieranie danych

Dane lotnicze

Stworzony został skrypt w języku Python do pobierania danych lotniczych, który korzysta z API AviationStack przy użyciu spersonalizowanego klucza API. Dane są aktualizowane co miesiąc dla informacji o samolotach, ich typach, lotniskach oraz liniach lotniczych, natomiast dane dotyczące lotów są pobierane co 30 minut. Taki harmonogram aktualizacji wynika z ograniczeń związanych z korzystaniem z bezpłatnej wersji API. API w wersji darmowej ograniczone było do 1000 zapytań. Aby poradzić sobie z tym ograniczeniem (co pół godziny dokonywano ~500 zapytań) stworzono 33 konta (a co za tym idzie 33 klucze API, 33000 dostępne zapytania).

Finalnie dane okazały się niskiej jakości – koordynaty nie zmieniały się w trakcie trwania lotu. Wpłynęło to na finalną postać wizualizacji w sposób negatywny.

Dane pogodowe

Dane pogodowe są pobierane ręcznie co miesiąc.

Przetwarzanie danych

Podczas przetwarzania danych użyta została architektura medalionu (medallion architecture) - wzorec projektowania danych służący do logicznej organizacji danych o różnym stopniu jakości oraz agregacji. Każdy poziom systemu jest obsługiwany przez odrębną grupę wątków w narzędziu NiFi. Aby zapewnić bezkolizyjne uruchamianie procesorów, każda z warstw jest inicjowana o ustalonej godzinie. Proces rozpoczyna się od warstwy brązowej, a pomiędzy kolejnymi poziomami istnieje 30-minutowe okno czasowe.

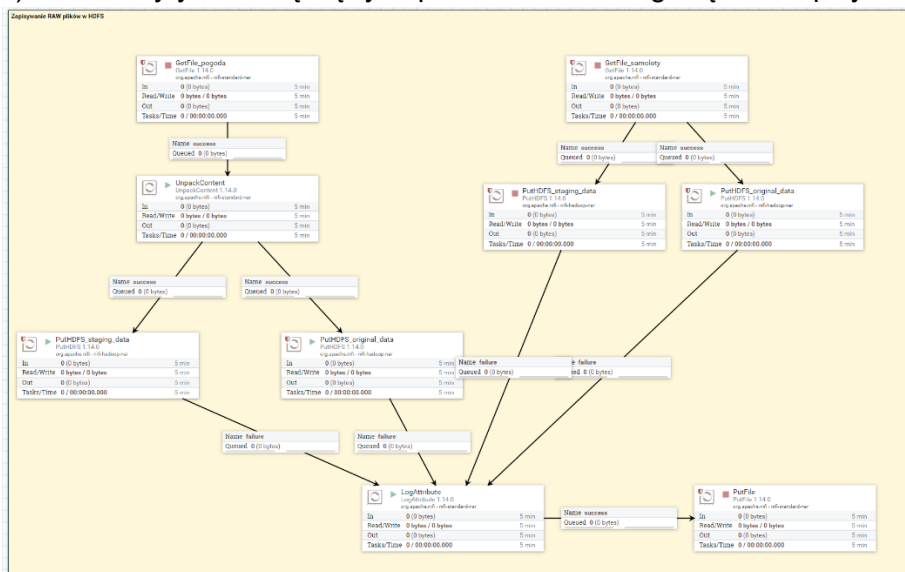
Błędy i wyjątki są zarządzane z poziomu NiFi poprzez agregację logów, umożliwiając deweloperom bieżący wgląd i skuteczną reakcję na potencjalne problemy. W skryptach pysparkowych dodawane były wiadomości do loggera, jednak odpalając skrypty z poziomu NiFi przestawały działać z ich powodu. Wobec tego, te wiadomości zostały zakomentowane.

Poziom brązowy

Na tym etapie opracowany został przepływ NiFi, który składa się z dwóch odrębnych strumieni. Pierwszy z nich zajmuje się pobieraniem danych pogodowych z lokalnego systemu, rozpakowywaniem archiwów zawierających dane (oryginalne dane są przechowywane w formacie .zip). Następnie przetworzone dane są zapisywane w HDFS w dwóch kopiach. Jedna kopia służy do archiwizacji, natomiast druga - staging - jest wykorzystywana do dalszej obróbki danych.

W katalogu "stage" znajdują się podkatalogi odpowiadające plikom/tabelom w "bronze". Tam tymczasowo przechowywane są nowe pliki w celu ich przetworzenia do poziomu "silver". Po pomyślnym przetworzeniu plików one są usuwane z katalogu "stage".

Strumień przetwarzający dane lotnicze wygląda identycznie, z wyjątkiem etapu rozpakowywania (dane nie są spakowane). Przechwytywane są błędy zapisu na HDFS. Logi błędów zapisywane do pliku txt.



Poziom srebrny

Na niniejszym etapie, dane staging są ekstrahowane z HDFS w celu utworzenia tabel danych.

W procesie tworzenia tych tabel, spłaszczana jest także struktura danych dla plików JSON, czyli plików zawierających informacje o lotach.

Ustalane są odpowiednie typy danych dla wszystkich kolumn. Nadawane są również odpowiednie nazwy kolumn, aby ułatwić zrozumienie struktury danych. Ponadto, niektóre wartości są modyfikowane z myślą o polepszeniu czytelności danych.

Folder stage jest podzielony na dwa główne podfoldery: "samoloty" oraz "pogoda". Każdy z tych podfolderów zawiera specyficzne dla siebie dane, które zostaną poddane dalszej obróbce w procesie ELT.

W podfolderze "stage/samoloty" gromadzone są dane dotyczące lotów (aktualizowane codziennie) oraz informacje związane z lotniskami, liniami lotniczymi, samolotami, ich modelami. Tutaj pojawia się potencjalny problem z parsowaniem różnych plików, odnoszących się do różnych rodzajów danych o zróżnicowanych formatach. Aby skutecznie radzić sobie z tym wyzwaniem, w skrypcie Sparka stosowana jest filtracja ścieżek na podstawie nazw plików. W ten sposób odpowiednie pliki trafiają do odpowiadających im tabel.

Dane o lotach są przechowywane w formacie zagnieżdżonym JSON - Spark nie radził sobie z jego schematem. Rozwiązaniem tego problemu jest iterowanie przez każdy plik w procesie przetwarzania, co pozwala na utworzenie obiektów `pyspark.sql.DataFrame`, a następnie łączenie ich w jedną dużą ramkę danych za pomocą metody `union`. W tym miejscu nie występuje problem z łączeniem danych, ponieważ definiowane są schematy z typami kolumn, co eliminuje ewentualne niejednorodności. Dodatkowo, dodawana jest kolumna z nazwą pliku, w której przechowywany jest znacznik czasu (timestamp), co umożliwia uzyskanie dokładnego czasu dla tych danych. Jest to istotne, ponieważ pozycja samolotu zmienia się w czasie.

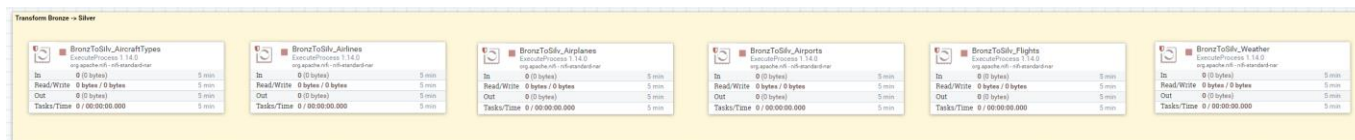
Dla danych niezwiązanych z lotami (np. o lotniskach) nie ma potrzeby spłaszczania, ponieważ ich struktura JSON jest łatwa do wczytania. Jednak nadal każda z tabeli powstaje z wielu plików, więc pliki te przetwarzane są w pętli. Dla tych danych nie definiuje się schematu, ponieważ format JSON jest wystarczająco klarowny, aby uniknąć konieczności precyzyjnego definiowania struktury danych.

W przypadku danych pogodowych, występują dwa rodzaje plików, które zawierają różne informacje. Pierwszy rodzaj plików posiada przedrostek w nazwie "k_d", natomiast drugi rodzaj zawiera przedrostek "k_d_t". Oba rodzaje plików przechowują dane pogodowe dla każdej stacji, z podziałem na rok, miesiąc i dzień. Ze względu na zróżnicowane informacje dostępne w tych plikach, definiowane są dla nich różne schematy, a następnie pliki łączone są na podstawie kluczy stacji, roku, miesiąca i dnia.

Oba rodzaje plików są w formacie CSV, który jest obsługiwany przez Sparka natywnie, umożliwiając łatwe przetwarzanie całych folderów.

Po złączeniu tabel, przeprowadzane jest kilka transformacji na kolumnach. Na przykład, zamieniane są binarne wartości tekstowe na format True/False oraz inne wartości dostosowywane są do bardziej zrozumiałych dla nas formatów, szczególnie statusy pomiarów pogodowych.

Ostatecznie, przekształcone tabele są konwertowane do formatu Parquet i zapisywane z powrotem do HDFS.



Poziom złoty

Z danych lotniczych w warstwie złotej przetwarzane są wyłącznie dane dotyczące lotów, które były wówczas aktywne, czyli znajdowały się w powietrzu, pochodzące z warstwy srebrnej.

Początkowe założenie dotyczące danych lotniczych z API polegały na otrzymywaniu aktualnych informacji na temat pozycji i prędkości samolotu w czasie rzeczywistym z przerwami 30 minut w trakcie jego lotu. Niestety okazało się, że API w darmowej wersji wcale nie aktualizuje tych informacji, co wymusiło znaczną zmianę pierwotnej koncepcji projektu.

W praktyce, dla danych lotów wybierane są jedynie te, które są aktualnie w powietrzu.

Zachowuje się unikalne loty, uwzględniając kolumny z timestampami odlotu, przylotu i kodem ICAO dla danej trasy.

W praktyce, podczas transformacji lotów z warstwy srebrnej, wiersze są przefiltrowane tylko do tych, które opisują loty będące w powietrzu w trakcie wykonywania skryptów pobierających. W celu pozbycia się duplikatów wierszy z tego samego lotu, które pojawiły się ze względu na wcześniej opisane nieoczekiwane działanie API, pozostawiony został jeden wiersz na lot na podstawie kolumn dotyczących unikalnego kodu ICAO dla trasy, a także czasu odlotu i przylotu.

Ponadto, ekstrahuje się dzień, miesiąc i rok z czasów odlotu i przylotu. Tworzona jest binarna kolumna informująca o rozpoczęciu lub zakończeniu lotu w Polsce. Następnie tabela zostaje podzielona na dwie partycje: odloty i przyloty z Polski, opierając się na nowej kolumnie.

W kontekście danych pogodowych, przyporządkowuje się najbliższe stacje pogodowe do lotnisk, dla których obecne są dane. Ramka danych jest zduplikowana na dwie: *arrival_weather* i *departure_weather*, z dodanymi przedrostkami odpowiednio *arrival* i *departure*. Następnie łączy się te ramki danych według kodu lotniska z danymi dotyczącymi lotów.

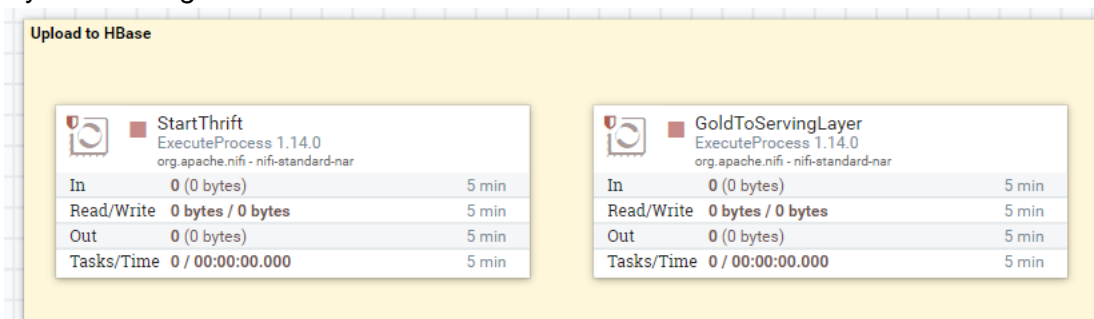
W rezultacie uzyskiwane są dwie tabele zawierające informacje o odlotach i przylotach do Polski oraz związane z nimi dane pogodowe. Wiersze tych tabel są następnie łączone w jedną, uzupełniając ewentualne brakujące kolumny wartościami null.



Transform Silver -> Gold		
SilvToGold_Flights_Weather ExecuteProcess 1.14.0 org.apache.nifi - nifi-standard-nar		
In	0 (0 bytes)	5 min
Read/Write	0 bytes / 0 bytes	5 min
Out	0 (0 bytes)	5 min
Tasks/Time	0 / 00:00:00.000	5 min

Wgrywanie danych do Serving Layer

Przetworzone dane finalnie wgrywane są do HBase. W tym celu używany jest NiFi flow zawierający 2 procesory, gdzie jeden odpowiada za uruchomienie potrzebnych narzędzi, a drugi uruchamia kod napisany w Pythonie z użyciem biblioteki HappyBase. Dane są wgrywane do jednej tabeli, zawierającej dwie rodziny kolumn: *flight* oraz *weather*.

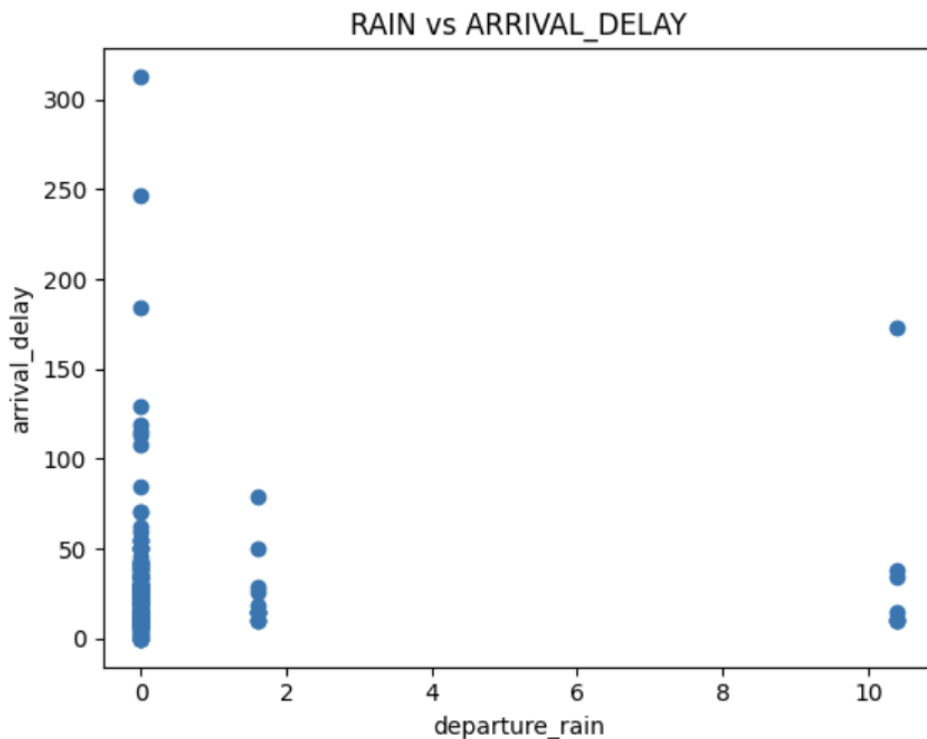


Upload to HBase		
StartThrift ExecuteProcess 1.14.0 org.apache.nifi - nifi-standard-nar		
In	0 (0 bytes)	5 min
Read/Write	0 bytes / 0 bytes	5 min
Out	0 (0 bytes)	5 min
Tasks/Time	0 / 00:00:00.000	5 min
GoldToServingLayer ExecuteProcess 1.14.0 org.apache.nifi - nifi-standard-nar		
In	0 (0 bytes)	5 min
Read/Write	0 bytes / 0 bytes	5 min
Out	0 (0 bytes)	5 min
Tasks/Time	0 / 00:00:00.000	5 min

Przykłady danych dostępnych dla warstwy prezentacyjnej

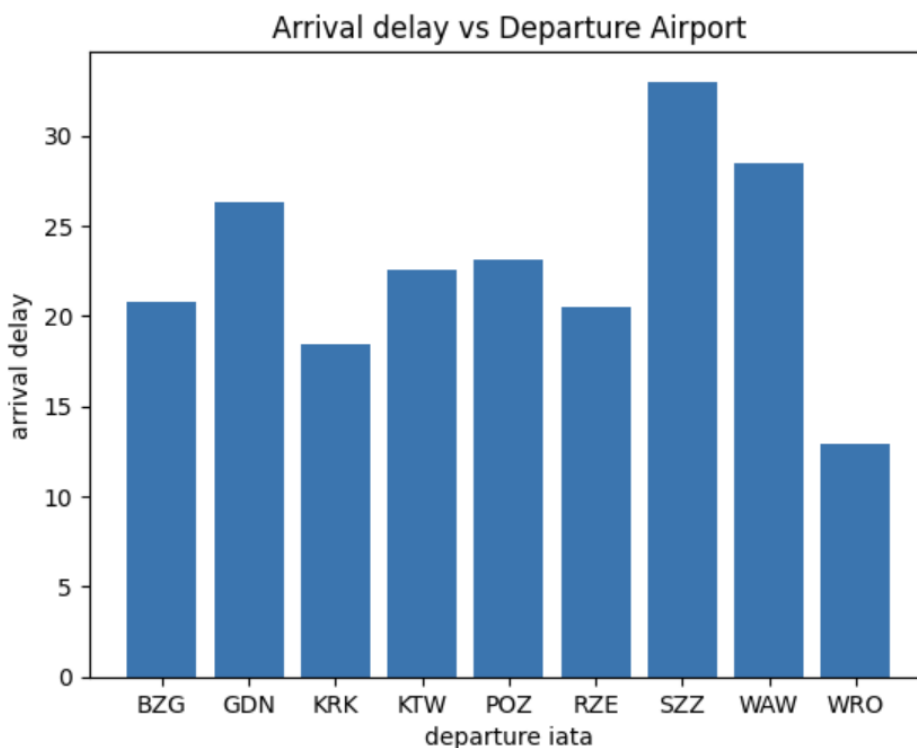
W celu przedstawienia potencjału pozyskanych i przetworzonych danych zostały wygenerowane 2 przykładowe wizualizacje.

Pierwszy wykres (Rys. 2) przedstawia zależność opóźnienia lotu od natężenia opadów na wybranym przez użytkownika lotnisku.



Rysunek 2: Zależność opóźnienia lotu od natężenia opadów (WAW)

Drugi wykres (Rys. 3) natomiast przedstawia średnie opóźnienie przylotu w zależności od lotniska startowego.



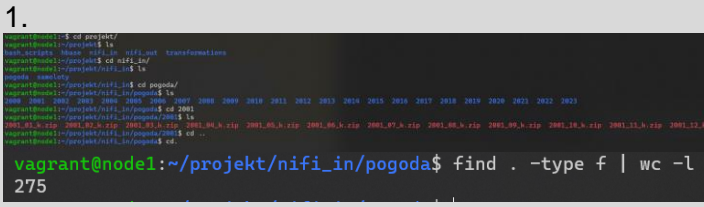

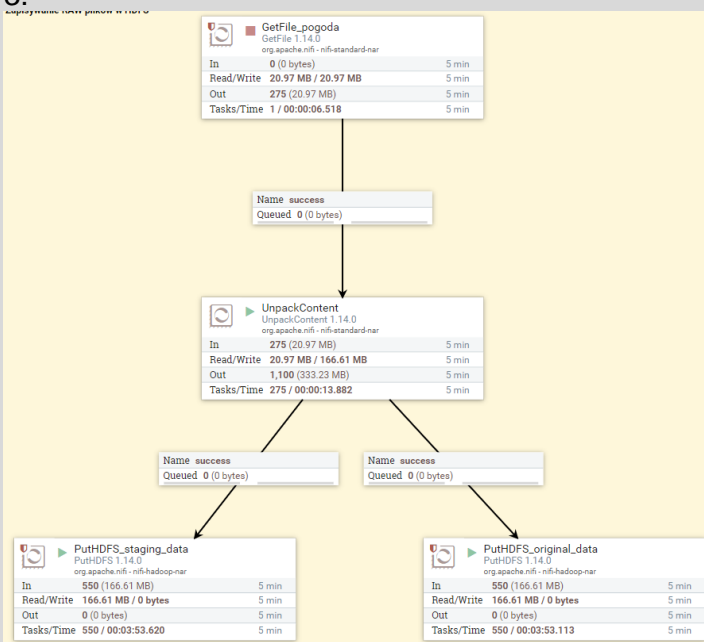
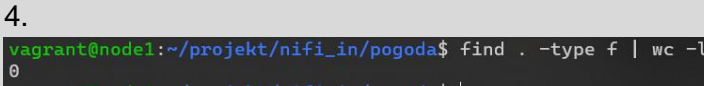
Rysunek 3: Średnie opóźnienie przylotu w zależności od lotniska startowego

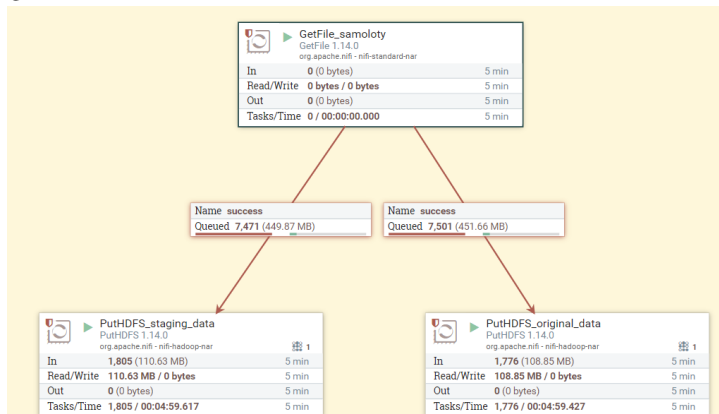
Podsumowanie finalnej wersji rozwiązania

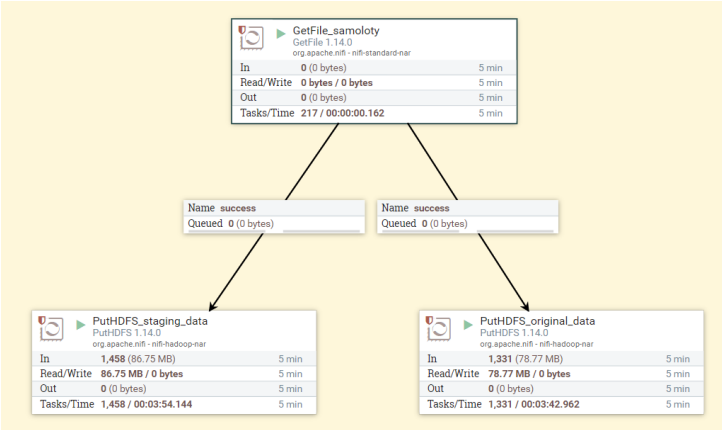
Projekt obejmuje kompleksowy proces przetwarzania danych, rozpoczynając od pobierania informacji za pomocą skryptów napisanych w języku Python. Dane te po odpowiednim ładowaniu są wczytywane przez narzędzie NiFi, oraz przepuszczane przez przepływ uruchamiany zgodnie z ustalonym

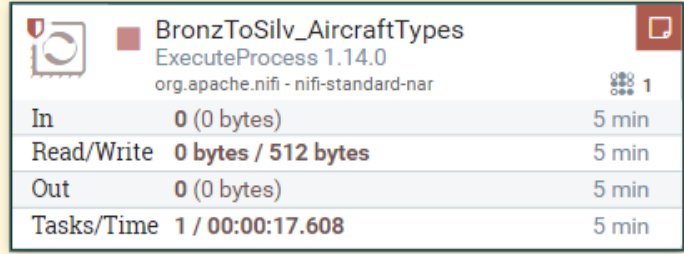
harmonogramem. Dane przechodzą przez etapy od ładowania do tak zwanej warstwy brązowej, gdzie są poddawane procesom takim jak staging i archiwizacja, a następnie przechodzą do warstwy srebrnej, gdzie są modyfikowane poprzez spłaszczenie i dopasowanie do zaproponowanego schematu. Z warstwy srebrnej dane przechodzą do warstwy złotej, gdzie są dalej modyfikowane poprzez grupowanie, filtry. Ostatecznie dane w postaci gotowej do wykorzystania przez użytkowników biznesowych trafiają do HBase. W warstwie obsługi (serving layer) skrypt NiFi automatycznie uruchamia proces aktualizacji, dodając nowe wiersze do HBase. Kluczowym elementem jest logiczny klucz, który ułatwia obsługę i zapewnia unikalność (`row{flight_iata}_{timestamp_odlotu_planowego}`). Ostatecznie, w warstwie obsługi, użytkownik może korzystać z notebooka do uruchamiania wizualizacji lub samodzielnie tworzyć nowe analizy. Projektem zarządza kompleksowy system, który umożliwia efektywne zarządzanie i analizę danych z wykorzystaniem różnych narzędzi Big Data

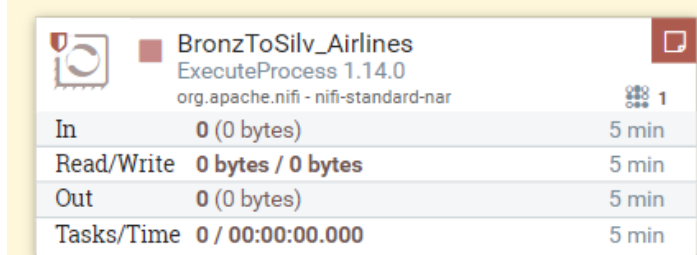
Testy

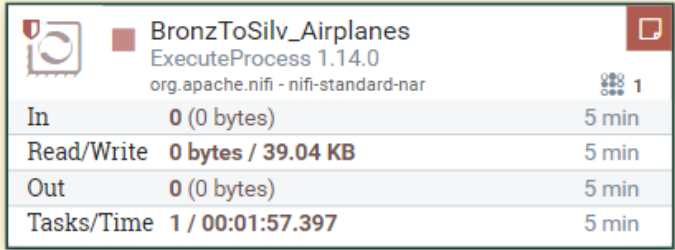
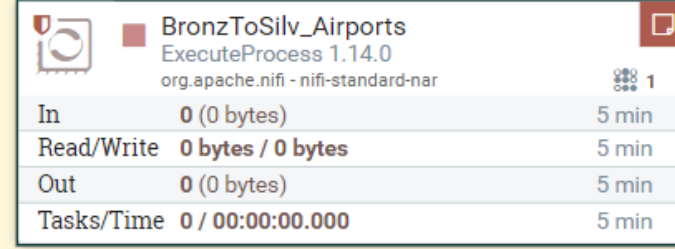
Cel testu	Kroki	Oczekiwany wynik	Potwierdzenie
Zweryfikowanie działania importowania za pomocą NiFi danych pogodowych z lokalnego systemu plików do HDFS	<ol style="list-style-type: none"> 1. Sprawdzenie zawartości lokalnego systemu plików 2. Sprawdzenie zawartości odpowiednich folderów na HDFS 3. Włączenie flow w NiFi 4. Ponowne sprawdzenie zawartości folderu na lokalnym systemie plików 5. Ponowne sprawdzenie zawartości folderów /user/projekt/bronze/pogoda oraz /user/projekt/bronze/stage/pogoda 6. Pokazanie przykładowych plików w HFDS 	<ol style="list-style-type: none"> 1. 275 plików 2. 0 plików 3. Flow działa 4. 0 plików 5. Po 550 plików w /user/projekt/bronze/pogoda a oraz /user/projekt/bronze/stage/pogoda 6. Pliki rozpakowane 	<ol style="list-style-type: none"> 1.  <pre>vagrant@node1:~/projekt/nifi_in/pogoda\$ find . -type f wc -l 275</pre> 2.  <pre>vagrant@node1:~/projekt/nifi_in/pogoda\$ hadoop fs -ls /user/projekt/bronze/pogoda vagrant@node1:~/projekt/nifi_in/pogoda\$ hadoop fs -ls /user/projekt/bronze/stage/pogoda vagrant@node1:~/projekt/nifi_in/pogoda\$</pre> 3.  <pre>graph TD GetFile[GetFile 1.14.0] --> UnpackContent[UnpackContent 1.14.0] UnpackContent --> PutHDFS_staging[PutHDFS_staging_data] UnpackContent --> PutHDFS_original[PutHDFS_original_data]</pre> 4.  <pre>vagrant@node1:~/projekt/nifi_in/pogoda\$ find . -type f wc -l 550</pre> 5.

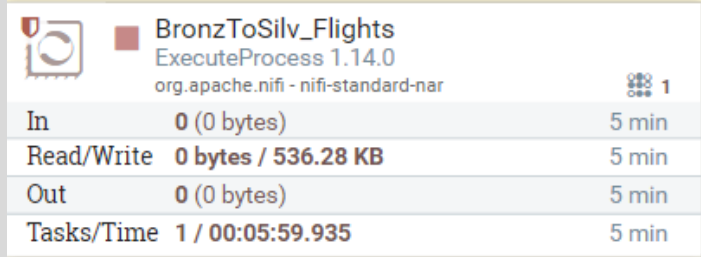
Cel testu	Kroki	Oczekiwany wynik	Potwierdzenie
			<pre>vagrant@node1:~\$ hadoop fs -count /user/projekt/bronze/stage/pogoda 1 550 174706403 /user/projekt/bronze/stage/pogoda vagrant@node1:~\$ hadoop fs -count /user/projekt/bronze/pogoda 1 550 174706403 /user/projekt/bronze/pogoda</pre> <p>6.</p> <pre>-rw-r--r-- 1 root supergroup 159333 2024-01-06 20:20 /user/projekt/bronze/pogoda/k_d_t_12_2016.c sv -rw-r--r-- 1 root supergroup 161185 2024-01-06 20:20 /user/projekt/bronze/pogoda/k_d_t_12_2017.c sv -rw-r--r-- 1 root supergroup 154719 2024-01-06 20:17 /user/projekt/bronze/pogoda/k_d_t_12_2018.c sv -rw-r--r-- 1 root supergroup 143610 2024-01-06 20:18 /user/projekt/bronze/pogoda/k_d_t_12_2019.c sv -rw-r--r-- 1 root supergroup 135738 2024-01-06 20:19 /user/projekt/bronze/pogoda/k_d_t_12_2020.c sv -rw-r--r-- 1 root supergroup 131513 2024-01-06 20:20 /user/projekt/bronze/pogoda/k_d_t_12_2021.c sv -rw-r--r-- 1 root supergroup 118289 2024-01-06 20:19 /user/projekt/bronze/pogoda/k_d_t_12_2022.c sv -rw-r--r-- 1 root supergroup 4229472 2024-01-06 20:18 /user/projekt/bronze/pogoda/k_d_t_2000.csv vagrant@node1:~\$</pre>
Zweryfikowanie działania importowania za pomocą NiFi danych lotniczych z lokalnego systemu plików do HDFS	<ol style="list-style-type: none"> 1. Sprawdzenie zawartości lokalnego systemu plików 2. Sprawdzenie zawartości odpowiednich folderów na HDFS 3. Włączenie flow w NiFi 4. Ponowne sprawdzenie zawartości folderu na lokalnym systemie plików 5. Ponowne sprawdzenie zawartości folderów [/bronze/samoloty oraz [/bronze/stage/samoloty 6. Pokazanie przykładowych plików w HFDS 	<ol style="list-style-type: none"> 1. 9403 plików 2. 0 plików 3. Flow działa 4. 0 plików 5. Po 9403 plików w /user/projekt/bronze/samoloty oraz /user/projekt/bronze/stage/samoloty 6. Pliki rozpakowane 	<ol style="list-style-type: none"> 1. <pre>vagrant@node1:~/projekt/nifi_in/samoloty\$ ls aircraft_types airlines airplanes airports flights taxes vagrant@node1:~/projekt/nifi_in/samoloty\$ find . -type f wc -l 9403</pre> 2. (zamiast tego kroku wykonano komendę hadoop fs -rm /user/projekt/bronze/samoloty/* oraz hadoop fs -rm /user/projekt/bronze/stage/samoloty/*) 3.  <pre>graph TD GetFile[GetFile_samoloty] --> Name1[Name success] GetFile --> Name2[Name success] Name1 --> PutStaging[PutHDFS_staging_data] Name2 --> PutOriginal[PutHDFS_original_data]</pre> <p>The diagram shows a NiFi flow. At the top is a 'GetFile_samoloty' processor (org.apache.nifi-nifi-standard-nar) with 0 bytes in/out and 0 tasks. It connects to two 'Name success' processors (org.apache.nifi-nifi-standard-nar), each with 7,471 MB queued. These connect to two 'PutHDFS' processors (org.apache.nifi-nifi-hadoop-nar). The left 'PutHDFS_staging_data' processor has 1,805 MB in, 110.63 MB written, and 1 task. The right 'PutHDFS_original_data' processor has 1,776 MB in, 108.85 MB written, and 1 task.</p>

Cel testu	Kroki	Oczekiwany wynik	Potwierdzenie
			<p>Zajmuje około 30 minut przy 9,5k plików, ale dziennie jest ich ~600 normalnie, więc jest dużo szybciej normalnie. Tu po załadowaniu około ¼ screen.</p>  <p>Tu po całości, ale usunęło się z historii procesorów</p> <p>4.</p> <pre>vagrant@node1:~/projekt/nifi_in/samoloty\$ find . -type f wc -l 0</pre> <p>5.</p> <pre>vagrant@node1:~/projekt/nifi_in/samoloty\$ hadoop fs -count /user/projekt/bronze/samoloty/ 1 9403 595274884 /user/projekt/bronze/samoloty vagrant@node1:~/projekt/nifi_in/samoloty\$ hadoop fs -count /user/projekt/bronze/stage/samoloty/ 1 9403 595274884 /user/projekt/bronze/stage/samoloty</pre> <p>6.</p>

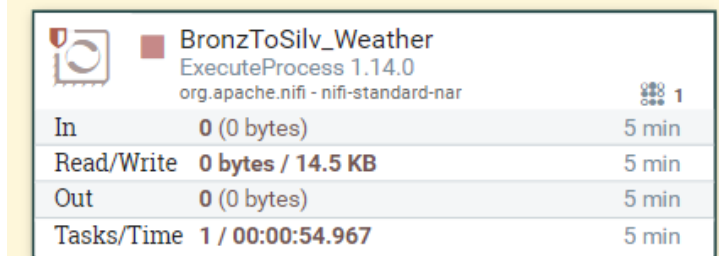
Cel testu	Kroki	Oczekiwany wynik	Potwierdzenie
			<pre> -rw-r--r-- 1 root supergroup 16.8 K 2024-01-06 20:40 /user/projekt/bronze/samoloty/flights_WRO_D EP_2024-01-04-09-23-56_offset100_.json -rw-r--r-- 1 root supergroup 91.3 K 2024-01-06 20:47 /user/projekt/bronze/samoloty/flights_WRO_D EP_2024-01-04-09-58-18_offset0_.json -rw-r--r-- 1 root supergroup 18.7 K 2024-01-06 20:51 /user/projekt/bronze/samoloty/flights_WRO_D EP_2024-01-04-09-58-21_offset100_.json -rw-r--r-- 1 root supergroup 91.4 K 2024-01-06 20:39 /user/projekt/bronze/samoloty/flights_WRO_D EP_2024-01-04-10-34-27_offset0_.json -rw-r--r-- 1 root supergroup 18.7 K 2024-01-06 21:02 /user/projekt/bronze/samoloty/flights_WRO_D EP_2024-01-04-10-34-29_offset100_.json -rw-r--r-- 1 root supergroup 91.5 K 2024-01-06 20:42 /user/projekt/bronze/samoloty/flights_WRO_D EP_2024-01-04-11-08-41_offset0_.json -rw-r--r-- 1 root supergroup 18.7 K 2024-01-06 20:52 /user/projekt/bronze/samoloty/flights_WRO_D EP_2024-01-04-11-08-44_offset100_.json -rw-r--r-- 1 root supergroup 9.1 K 2024-01-06 20:41 /user/projekt/bronze/samoloty/taxes_offset0 _.json -rw-r--r-- 1 root supergroup 9.2 K 2024-01-06 20:51 /user/projekt/bronze/samoloty/taxes_offset1 00_.json -rw-r--r-- 1 root supergroup 9.2 K 2024-01-06 20:58 /user/projekt/bronze/samoloty/taxes_offset2 00_.json -rw-r--r-- 1 root supergroup 9.2 K 2024-01-06 20:44 /user/projekt/bronze/samoloty/taxes_offset3 00_.json -rw-r--r-- 1 root supergroup 9.1 K 2024-01-06 20:55 /user/projekt/bronze/samoloty/taxes_offset4 00_.json -rw-r--r-- 1 root supergroup 2.0 K 2024-01-06 20:53 /user/projekt/bronze/samoloty/taxes_offset5 00_.json </pre>
Zweryfikowanie działania przetworzenia danych AircraftTypes do warstwy Srebrnej	<ol style="list-style-type: none"> 1. Sprawdzenie zawartości folderu [/silver na HDFS 2. Sprawdzenie zawartości folderu [/bronze/stage/samoloty na HDFS 3. Włączenie flow w nifi (włączenie skryptu sparkowego) 4. Sprawdzenie katalogu [/silver dla aircraftTypes na HDFS 5. Sprawdzenie [/stage/bronze/samoloty na HDFS 	<ol style="list-style-type: none"> 1. 0 plików 2. 9403 plików 3. Flow (skrypt) działa 4. Pojawiły się pliki Parquet 5. Zniknęło kilka plików 	<ol style="list-style-type: none"> 1. <pre> vagrant@node1:~\$ hadoop fs -count /user/projekt/silver/ 11 0 0 /user/projekt/silver vagrant@node1:~\$ </pre> 2. <pre> vagrant@node1:~/projekt/nifi_in/samoloty\$ hadoop fs -count /user/projekt/bronze/stage/samoloty/ 1 9403 595274884 /user/projekt/bronze/stage/samoloty </pre> 3.  4.

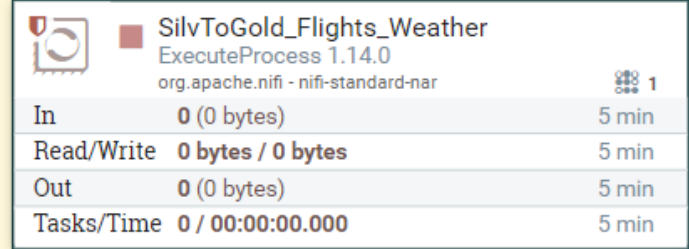
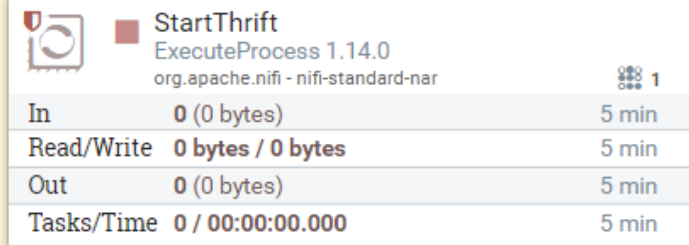
Cel testu	Kroki	Oczekiwany wynik	Potwierdzenie
			<pre>vagrant@node1:~\$ hadoop fs -count /user/projekt/silver/aircraft_types 1 9 16586 /user/projekt/silver/aircraft_types vagrant@node1:~\$ hadoop fs -ls /user/projekt/silver/aircraft_types Found 9 items -rw-r--r-- 1 root supergroup 0 2024-01-06 21:32 /user/projekt/silver/aircraft_types/_SUCCESS\$ -rw-r--r-- 1 root supergroup 2342 2024-01-06 21:32 /user/projekt/silver/aircraft_types/part-0000-2ab3c3a7-70fb-4c8f-8f1d-e6e11449d3c3-c000.snappy.parquet -rw-r--r-- 1 root supergroup 2514 2024-01-06 21:32 /user/projekt/silver/aircraft_types/part-0001-2ab3c3a7-70fb-4c8f-8f1d-e6e11449d3c3-c000.snappy.parquet -rw-r--r-- 1 root supergroup 2103 2024-01-06 21:32 /user/projekt/silver/aircraft_types/part-0002-2ab3c3a7-70fb-4c8f-8f1d-e6e11449d3c3-c000.snappy.parquet -rw-r--r-- 1 root supergroup 2113 2024-01-06 21:32 /user/projekt/silver/aircraft_types/part-0003-2ab3c3a7-70fb-4c8f-8f1d-e6e11449d3c3-c000.snappy.parquet -rw-r--r-- 1 root supergroup 2419 2024-01-06 21:32 /user/projekt/silver/aircraft_types/part-0004-2ab3c3a7-70fb-4c8f-8f1d-e6e11449d3c3-c000.snappy.parquet -rw-r--r-- 1 root supergroup 2490 2024-01-06 21:32 /user/projekt/silver/aircraft_types/part-0005-2ab3c3a7-70fb-4c8f-8f1d-e6e11449d3c3-c000.snappy.parquet -rw-r--r-- 1 root supergroup 1316 2024-01-06 21:32 /user/projekt/silver/aircraft_types/part-0006-2ab3c3a7-70fb-4c8f-8f1d-e6e11449d3c3-c000.snappy.parquet -rw-r--r-- 1 root supergroup 1289 2024-01-06 21:32 /user/projekt/silver/aircraft_types/part-0007-2ab3c3a7-70fb-4c8f-8f1d-e6e11449d3c3-c000.snappy.parquet</pre> <p>5.</p> <pre>vagrant@node1:~/projekt/transformations\$ hadoop fs -count /user/projekt/bronze/stage/samoloty/ 1 9399 595243849 /user/projekt/bronze/stage/samoloty</pre>
<p>Zweryfikowanie działania przetworzenia danych Airlines do warstwy Srebrnej</p>	<ol style="list-style-type: none"> 1. Sprawdzenie zawartości folderu [/bronze/stage/samoloty na HDFS 2. Włączenie flow w nifi (włączenie skryptu sparkowego) 3. Sprawdzenie katalogu [/silver dla airlines na HDFS 4. Sprawdzenie [/stage/bronze/samoloty na HDFS 	<ol style="list-style-type: none"> 1. 9399 plików 2. Flow (skrypt) działa 3. Pojawiły się pliki Parquet 4. Zniknęło trochę plików 	<p>1.</p> <pre>vagrant@node1:~/projekt/transformations\$ hadoop fs -count /user/projekt/bronze/stage/samoloty/ 1 9399 595243849 /user/projekt/bronze/stage/samoloty</pre> <p>2.</p>  <p>3.</p> <pre>vagrant@node1:~\$ hadoop fs -count /user/projekt/silver/airlines 1 265 1578766 /user/projekt/silver/airlines</pre> <p>4.</p> <pre>vagrant@node1:~/projekt/transformations\$ hadoop fs -count /user/projekt/bronze/stage/samoloty/ 1 9267 591085138 /user/projekt/bronze/stage/samoloty</pre>

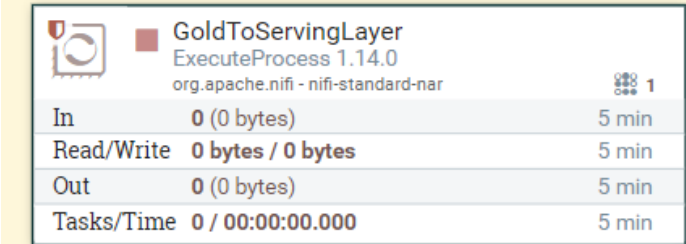
Cel testu	Kroki	Oczekiwany wynik	Potwierdzenie
Zweryfikowanie działania przetworzenia danych Airplanes do warstwy Srebrnej	<ol style="list-style-type: none"> 1. Sprawdzenie zawartości folderu [/bronze/stage/samoloty na HDFS 2. Włączenie flow w nifi (włączenie skryptu sparkowego) 3. Sprawdzenie katalogu [/silver dla airplanes na HDFS 4. Sprawdzenie [/stage/bronze/samoloty na HDFS 	<ol style="list-style-type: none"> 1. 9267 plików 2. Flow (skrypt) działa 3. Pojawiły się pliki Parquet 4. Zniknęło trochę plików 	<ol style="list-style-type: none"> 1. <pre>vagrant@node1:~/projekt/transformations\$ hadoop fs -count /user/projekt/bronze/stage/samoloty/ 1 9267 591085138 /user/projekt/bronze/stage/samoloty</pre> 2.  3. <pre>@acaecb-371f-4723-8983-1d70765e3001-c000.snappy.parquet -rw-r--r-- 1 root supergroup 18071 2024-01-06 21:42 /user/projekt/silver/airplanes/part-00378-5 @acaecb-371f-4723-8983-1d70765e3001-c000.snappy.parquet -rw-r--r-- 1 root supergroup 9898 2024-01-06 21:42 /user/projekt/silver/airplanes/part-00379-5 @acaecb-371f-4723-8983-1d70765e3001-c000.snappy.parquet -rw-r--r-- 1 root supergroup 9484 2024-01-06 21:42 /user/projekt/silver/airplanes/part-00380-5 @acaecb-371f-4723-8983-1d70765e3001-c000.snappy.parquet -rw-r--r-- 1 root supergroup 9652 2024-01-06 21:42 /user/projekt/silver/airplanes/part-00381-5 @acaecb-371f-4723-8983-1d70765e3001-c000.snappy.parquet vagrant@node1:~\$ hadoop fs -count /user/projekt/silver/airplanes 1 383 3915103 /user/projekt/silver/airplanes</pre> 4. <pre>vagrant@node1:~/projekt/transformations\$ hadoop fs -count /user/projekt/bronze/stage/samoloty/ 1 9076 577769424 /user/projekt/bronze/stage/samoloty</pre>
Zweryfikowanie działania przetworzenia danych Airports do warstwy Srebrnej	<ol style="list-style-type: none"> 1. Sprawdzenie zawartości folderu [/bronze/stage/samoloty na HDFS 2. Włączenie flow w nifi (włączenie skryptu sparkowego) 3. Sprawdzenie katalogu [/silver dla airports na HDFS 4. Sprawdzenie [/stage/bronze/samoloty na HDFS 	<ol style="list-style-type: none"> 1. 9076 plików 2. Flow (skrypt) działa 3. Pojawiły się pliki Parquet 4. Zniknęło trochę plików 	<ol style="list-style-type: none"> 1. <pre>vagrant@node1:~/projekt/transformations\$ hadoop fs -count /user/projekt/bronze/stage/samoloty/ 1 9076 577769424 /user/projekt/bronze/stage/samoloty</pre> 2.  3.

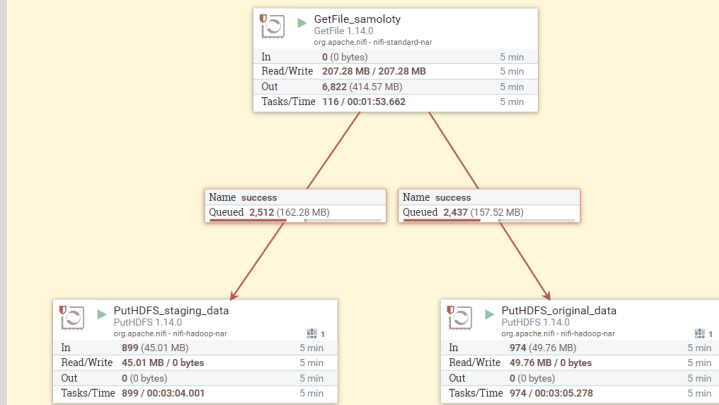
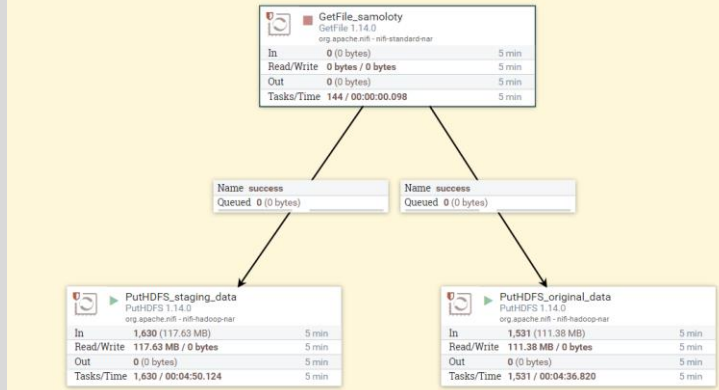
Cel testu	Kroki	Oczekiwany wynik	Potwierdzenie
			<pre> -rw-r--r-- 1 root supergroup 7.5 K 2024-01-06 21:46 /user/projekt/silver/airports/part-00128-16 64463c-abe3-4b3d-be59-8ca71aca202c-c000.snappy.parquet -rw-r--r-- 1 root supergroup 7.2 K 2024-01-06 21:46 /user/projekt/silver/airports/part-00129-16 64463c-abe3-4b3d-be59-8ca71aca202c-c000.snappy.parquet -rw-r--r-- 1 root supergroup 7.3 K 2024-01-06 21:46 /user/projekt/silver/airports/part-00130-16 64463c-abe3-4b3d-be59-8ca71aca202c-c000.snappy.parquet -rw-r--r-- 1 root supergroup 7.4 K 2024-01-06 21:46 /user/projekt/silver/airports/part-00131-16 64463c-abe3-4b3d-be59-8ca71aca202c-c000.snappy.parquet -rw-r--r-- 1 root supergroup 7.3 K 2024-01-06 21:46 /user/projekt/silver/airports/part-00132-16 64463c-abe3-4b3d-be59-8ca71aca202c-c000.snappy.parquet -rw-r--r-- 1 root supergroup 7.3 K 2024-01-06 21:46 /user/projekt/silver/airports/part-00133-16 64463c-abe3-4b3d-be59-8ca71aca202c-c000.snappy.parquet -rw-r--r-- 1 root supergroup 7.5 K 2024-01-06 21:46 /user/projekt/silver/airports/part-00134-16 64463c-abe3-4b3d-be59-8ca71aca202c-c000.snappy.parquet -rw-r--r-- 1 root supergroup 7.4 K 2024-01-06 21:46 /user/projekt/silver/airports/part-00135-16 64463c-abe3-4b3d-be59-8ca71aca202c-c000.snappy.parquet vagrant@node1:~\$ hadoop fs -count /user/projekt/silver/airports 1 137 1009768 /user/projekt/silver/airports vagrant@node1:~\$ </pre> <p>4.</p> <pre> vagrant@node1:~/projekt/transformations\$ hadoop fs -count /user/projekt/bronze/stage/samoloty/ 1 9008 575754684 /user/projekt/bronze/stage/samoloty </pre>
Zweryfikowanie działania przetworzenia danych Flights do warstwy Srebrnej	<ol style="list-style-type: none"> 1. Sprawdzenie zawartości folderu [/bronze/stage/samoloty na HDFS 2. Włączenie flow w nifi (włączenie skryptu sparkowego) 3. Sprawdzenie katalogu [/silver dla flights na HDFS 4. Sprawdzenie [/stage/bronze/samolot na HDFS 	<ol style="list-style-type: none"> 1. 9008 plików 2. Flow (skrypt) działa 3. Pojawiły się pliki Parquet 4. Zniknęło trochę plików 	<ol style="list-style-type: none"> 1. <pre> vagrant@node1:~/projekt/transformations\$ hadoop fs -count /user/projekt/bronze/stage/samoloty/ 1 9008 575754684 /user/projekt/bronze/stage/samoloty </pre> 2.  3.

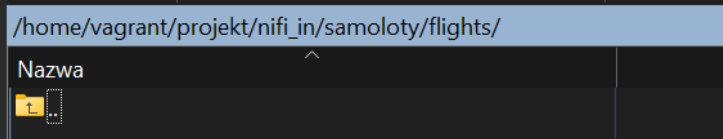
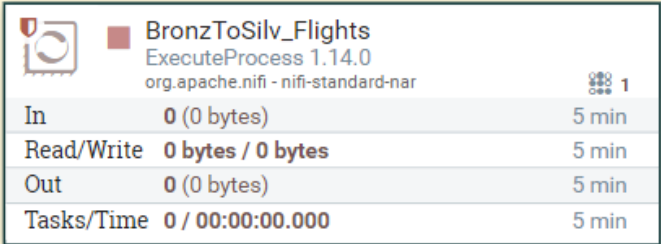
Cel testu	Kroki	Oczekiwany wynik	Potwierdzenie
			<pre>vagrant@node1:~\$ hadoop fs -ls -h /user/projekt/silver/flights Found 3 items -rw-r--r-- 1 root supergroup 0 2024-01-06 21:57 /user/projekt/silver/flights/_SUCCESS drwxr-xr-x - root supergroup 0 2024-01-06 21:53 /user/projekt/silver/flights/year=2023 drwxr-xr-x - root supergroup 0 2024-01-06 21:54 /user/projekt/silver/flights/year=2024 vagrant@node1:~\$ hadoop fs -ls -h /user/projekt/silver/flights/year=2024 Found 1 items drwxr-xr-x - root supergroup 0 2024-01-06 21:54 /user/projekt/silver/flights/year=2024/month=1 vagrant@node1:~\$ hadoop fs -ls -h /user/projekt/silver/flights/year=2024/month=1 Found 4 items drwxr-xr-x - root supergroup 0 2024-01-06 21:57 /user/projekt/silver/flights/year=2024/month=1/day=1 drwxr-xr-x - root supergroup 0 2024-01-06 21:57 /user/projekt/silver/flights/year=2024/month=1/day=2 drwxr-xr-x - root supergroup 0 2024-01-06 21:57 /user/projekt/silver/flights/year=2024/month=1/day=3 drwxr-xr-x - root supergroup 0 2024-01-06 21:57 /user/projekt/silver/flights/year=2024/month=1/day=4 vagrant@node1:~\$ hadoop fs -ls -h /user/projekt/silver/flights/year=2024/month=1/day=4 Found 18 items -rw-r--r-- 1 root supergroup 131.6 K 2024-01-06 21:56 /user/projekt/silver/flights/year=2024/month=1/day=4/part-00000-336e593e-991c-4c1b-84e3-e66a327b2948.c000.snappy.parquet -rw-r--r-- 1 root supergroup 112.5 K 2024-01-06 21:55 /user/projekt/silver/flights/year=2024/month=1/day=4/part-00000-3bd08de8-cf2f-48e9-a591-6dc8298b4254.c000.snappy.parquet -rw-r--r-- 1 root supergroup 119.2 K 2024-01-06 21:54 /user/projekt/silver/flights/year=2024/month=1/day=4/part-00000-495d0268-dce3-4e68-931b-afb6c0145835.c000.snappy.parquet -rw-r--r-- 1 root supergroup 139.0 K 2024-01-06 21:56 /user/projekt/silver/flights/year=2024/month=1/day=4/part-00000-58c8a9a4-009c-4237-9cbc-8eefc29ca27d.c000.snappy.parquet -rw-r--r-- 1 root supergroup 120.6 K 2024-01-06 21:57 /user/projekt/silver/flights/year=2024/month=1/day=4/part-00000-601d19c7-b200-42c6-add4-f01132a454d4.c000.snappy.parquet -rw-r--r-- 1 root supergroup 136.1 K 2024-01-06 21:55 /user/projekt/silver/flights/year=2024/month=1/day=4/part-00000-9c8a37ee-2a55-4eff-bc77-116fee4f4bfff.c000.snappy.parquet -rw-r--r-- 1 root supergroup 130.3 K 2024-01-06 21:55 /user/projekt/silver/flights/year=2024/month=1/day=4/part-00000-a222dfc8-cb14-4639-83f6-caa305830d66.c000.snappy.parquet -rw-r--r-- 1 root supergroup 121.6 K 2024-01-06 21:56 /user/projekt/silver/flights/year=2024/month=1/day=4/part-00000-acl57e4d-a93d-4745-9efc-83c3385ec4b5.c000.snappy.parquet -rw-r--r-- 1 root supergroup 118.7 K 2024-01-06 21:54 /user/projekt/silver/flights/year=2024/month=1/day=4/part-00000-ee5965ca-e730-4c37-bf69-66c9a9d3dd07.c000.snappy.parquet -rw-r--r-- 1 root supergroup 156.5 K 2024-01-06 21:56 /user/projekt/silver/flights/year=2024/month=1/day=4/part-00001-336e593e-991c-4c1b-84e3-e66a327b2948.c000.snappy.parquet -rw-r--r-- 1 root supergroup 117.9 K 2024-01-06 21:55 /user/projekt/silver/flights/year=2024/month=1/day=4/part-00001-3bd08de8-cf2f-48e9-a591-6dc8298b4254.c000.snappy.parquet -rw-r--r-- 1 root supergroup 136.1 K 2024-01-06 21:54 /user/projekt/silver/flights/year=2024/month=1/day=4/part-00001-495d0268-dce3-4e68-931b-afb6c0145835.c000.snappy.parquet -rw-r--r-- 1 root supergroup 165.6 K 2024-01-06 21:56 /user/projekt/silver/flights/year=2024/month=1/day=4/part-00001-58c8a9a4-009c-4237-9cbc-8eefc29ca27d.c000.snappy.parquet -rw-r--r-- 1 root supergroup 146.6 K 2024-01-06 21:57 /user/projekt/silver/flights/year=2024/month=1/day=4/part-00001-601d19c7-b200-42c6-add4-f01132a454d4.c000.snappy.parquet -rw-r--r-- 1 root supergroup 129.5 K 2024-01-06 21:55 /user/projekt/silver/flights/year=2024/month=1/day=4/part-00001-9c8a37ee-2a55-4eff-bc77-116fee4f4bfff.c000.snappy.parquet -rw-r--r-- 1 root supergroup 146.4 K 2024-01-06 21:55 /user/projekt/silver/flights/year=2024/month=1/day=4/part-00001-a222dfc8-cb14-4639-83f6-caa305830d66.c000.snappy.parquet -rw-r--r-- 1 root supergroup 141.0 K 2024-01-06 21:56 /user/projekt/silver/flights/year=2024/month=1/day=4/part-00001-acl57e4d-a93d-4745-9efc-83c3385ec4b5.c000.snappy.parquet -rw-r--r-- 1 root supergroup 143.0 K 2024-01-06 21:54 /user/projekt/silver/flights/year=2024/month=1/day=4/part-00001-ee5965ca-e730-4c37-bf69-66c9a9d3dd07.c000.snappy.parquet vagrant@node1:~\$ hadoop fs -count -h /user/projekt/silver/flights 12 129 22.2 M /user/projekt/silver/flights vagrant@node1:~\$</pre>
			<p>Pliki są, popartycjonowane</p> <p>4.</p> <pre>vagrant@node1:~/projekt/transformations\$ hadoop fs -count /user/projekt/bronze/stage/samoloty/ 1 6 49021 /user/projekt/bronze/stage/samoloty/ vagrant@node1:~/projekt/transformations\$ hadoop fs -ls /user/projekt/bronze/stage/samoloty/ Found 6 items -rw-r--r-- 1 root supergroup 9365 2024-01-06 20:40 /user/projekt/bronze/stage/samoloty/taxes_o ffset0_.json -rw-r--r-- 1 root supergroup 9459 2024-01-06 20:50 /user/projekt/bronze/stage/samoloty/taxes_o ffset100_.json -rw-r--r-- 1 root supergroup 9411 2024-01-06 20:59 /user/projekt/bronze/stage/samoloty/taxes_o ffset200_.json -rw-r--r-- 1 root supergroup 9397 2024-01-06 20:44 /user/projekt/bronze/stage/samoloty/taxes_o ffset300_.json -rw-r--r-- 1 root supergroup 9303 2024-01-06 20:55 /user/projekt/bronze/stage/samoloty/taxes_o ffset400_.json -rw-r--r-- 1 root supergroup 2086 2024-01-06 20:53 /user/projekt/bronze/stage/samoloty/taxes_o ffset500_.json</pre> <p>Zostały tylko taxes, ale ich dalej nie wykorzystujemy, stąd nie będą usunięte, gdyż nie ma dla nich</p>

Cel testu	Kroki	Oczekiwany wynik	Potwierdzenie
			dedykowanych skryptów/flow. Tak czy inaczej pliki o lotach się usunęły :)
Zweryfikowanie działania przetworzenia danych Weather do warstwy Srebrnej	<ol style="list-style-type: none"> 1. Sprawdzenie zawartości folderu [/bronze/stage/pogoda na HDFS 2. Włączenie flow w nifi (włączenie skryptu sparkowego) 3. Sprawdzenie katalogu [/silver dla weather na HDFS 4. Sprawdzenie [/stage/bronze/pogoda na HDFS 	<ol style="list-style-type: none"> 1. 550plików 2. Flow (skrypt) działa 3. Pojawiły się pliki Parquet 4. Zniknęły wszystkie pliki 	<ol style="list-style-type: none"> 1. <pre>vagrant@node1:~/projekt/transformations\$ hadoop fs -count /user/projekt/bronze/stage/pogoda/ 1 550 174706403 /user/projekt/bronze/stage/pogoda</pre> 2.  3. <pre>-rw-r--r-- 1 root supergroup 69.2 K 2024-01-06 21:49 /user/projekt/silver/weather/part-00190-570 3080b-62a7-446e-aaf5-6651d69d6533-c000.snappy.parquet -rw-r--r-- 1 root supergroup 69.8 K 2024-01-06 21:49 /user/projekt/silver/weather/part-00191-570 3080b-62a7-446e-aaf5-6651d69d6533-c000.snappy.parquet -rw-r--r-- 1 root supergroup 71.8 K 2024-01-06 21:49 /user/projekt/silver/weather/part-00192-570 3080b-62a7-446e-aaf5-6651d69d6533-c000.snappy.parquet -rw-r--r-- 1 root supergroup 70.3 K 2024-01-06 21:49 /user/projekt/silver/weather/part-00193-570 3080b-62a7-446e-aaf5-6651d69d6533-c000.snappy.parquet -rw-r--r-- 1 root supergroup 69.9 K 2024-01-06 21:49 /user/projekt/silver/weather/part-00194-570 3080b-62a7-446e-aaf5-6651d69d6533-c000.snappy.parquet -rw-r--r-- 1 root supergroup 69.9 K 2024-01-06 21:49 /user/projekt/silver/weather/part-00195-570 3080b-62a7-446e-aaf5-6651d69d6533-c000.snappy.parquet -rw-r--r-- 1 root supergroup 70.2 K 2024-01-06 21:49 /user/projekt/silver/weather/part-00196-570 3080b-62a7-446e-aaf5-6651d69d6533-c000.snappy.parquet -rw-r--r-- 1 root supergroup 69.4 K 2024-01-06 21:49 /user/projekt/silver/weather/part-00197-570 3080b-62a7-446e-aaf5-6651d69d6533-c000.snappy.parquet -rw-r--r-- 1 root supergroup 70.4 K 2024-01-06 21:49 /user/projekt/silver/weather/part-00198-570 3080b-62a7-446e-aaf5-6651d69d6533-c000.snappy.parquet -rw-r--r-- 1 root supergroup 69.0 K 2024-01-06 21:49 /user/projekt/silver/weather/part-00199-570 3080b-62a7-446e-aaf5-6651d69d6533-c000.snappy.parquet vagrant@node1:~\$ hadoop fs -count /user/projekt/silver/weather 1 201 10330921 /user/projekt/silver/weather</pre> 4. <pre>vagrant@node1:~/projekt/transformations\$ hadoop fs -count /user/projekt/bronze/stage/pogoda/ 1 0 0 /user/projekt/bronze/stage/pogoda</pre>
Zweryfikowanie działania przetworzenia danych Weather oraz Flights do warstwy Złotej	<ol style="list-style-type: none"> 1. Sprawdzenie, że folder [/gold na HDFS jest pusty 2. Włączenie Flow w nifi (skryptu sparkowego) 3. Sprawdzenie, że dodały się pliki do [/gold na HDFS 	<ol style="list-style-type: none"> 1. Folder jest pusty 2. Flow (skrypt) działa 3. Pliki dodały się 	<ol style="list-style-type: none"> 1. <pre>vagrant@node1:~\$ hadoop fs -ls -h /user/projekt/gold/flights_weather Found 1 items drwxr-xr-x - root supergroup 0 2024-01-06 22:08 /user/projekt/gold/flights_weather/_tempora ry</pre> <p>To w trakcie działania skryptu, zapomniałem złapać screena przed, ale jest tylko temporary (czyli wcześniej nic nie było)</p> 2.

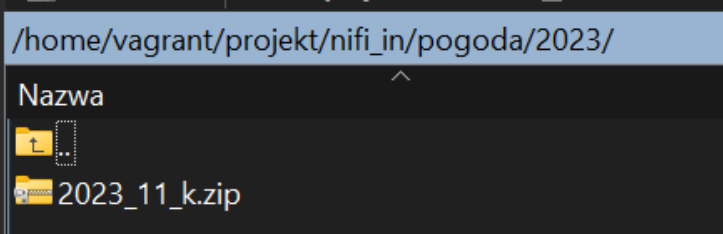
Cel testu	Kroki	Oczekiwany wynik	Potwierdzenie
			 <p>3.</p> <pre> 49-75665f78-5f9e-4f38-98d8-9f1afd69ea06-c000.snappy.parquet -rw-r--r-- 1 root supergroup 32.0 K 2024-01-06 22:09 /user/projekt/gold/flights_weather/part-003 51-75665f78-5f9e-4f38-98d8-9f1afd69ea06-c000.snappy.parquet -rw-r--r-- 1 root supergroup 27.0 K 2024-01-06 22:09 /user/projekt/gold/flights_weather/part-003 52-75665f78-5f9e-4f38-98d8-9f1afd69ea06-c000.snappy.parquet -rw-r--r-- 1 root supergroup 25.5 K 2024-01-06 22:09 /user/projekt/gold/flights_weather/part-003 53-75665f78-5f9e-4f38-98d8-9f1afd69ea06-c000.snappy.parquet -rw-r--r-- 1 root supergroup 38.3 K 2024-01-06 22:09 /user/projekt/gold/flights_weather/part-003 56-75665f78-5f9e-4f38-98d8-9f1afd69ea06-c000.snappy.parquet -rw-r--r-- 1 root supergroup 26.6 K 2024-01-06 22:09 /user/projekt/gold/flights_weather/part-003 61-75665f78-5f9e-4f38-98d8-9f1afd69ea06-c000.snappy.parquet -rw-r--r-- 1 root supergroup 31.9 K 2024-01-06 22:09 /user/projekt/gold/flights_weather/part-003 69-75665f78-5f9e-4f38-98d8-9f1afd69ea06-c000.snappy.parquet -rw-r--r-- 1 root supergroup 26.3 K 2024-01-06 22:09 /user/projekt/gold/flights_weather/part-003 74-75665f78-5f9e-4f38-98d8-9f1afd69ea06-c000.snappy.parquet -rw-r--r-- 1 root supergroup 26.0 K 2024-01-06 22:09 /user/projekt/gold/flights_weather/part-003 76-75665f78-5f9e-4f38-98d8-9f1afd69ea06-c000.snappy.parquet -rw-r--r-- 1 root supergroup 29.0 K 2024-01-06 22:09 /user/projekt/gold/flights_weather/part-003 92-75665f78-5f9e-4f38-98d8-9f1afd69ea06-c000.snappy.parquet -rw-r--r-- 1 root supergroup 25.5 K 2024-01-06 22:09 /user/projekt/gold/flights_weather/part-003 96-75665f78-5f9e-4f38-98d8-9f1afd69ea06-c000.snappy.parquet -rw-r--r-- 1 root supergroup 26.4 K 2024-01-06 22:09 /user/projekt/gold/flights_weather/part-003 97-75665f78-5f9e-4f38-98d8-9f1afd69ea06-c000.snappy.parquet -rw-r--r-- 1 root supergroup 25.7 K 2024-01-06 22:09 /user/projekt/gold/flights_weather/part-003 99-75665f78-5f9e-4f38-98d8-9f1afd69ea06-c000.snappy.parquet vagrant@node1:~\$ hadoop fs -count -h /user/projekt/gold/flights_weather 1 210 5.1 M /user/projekt/gold/flights_weather vagrant@node1:~\$ </pre>
Sprawdzenie dodania danych z Gold do Serving Layer na HBase	<ol style="list-style-type: none"> 1. Włączenie servera thrift za pomocą skryptu python w NiFi 2. Sprawdzenie, że tabela jest pusta 3. Włączenie (z poziomu NiFi) skryptu python dodającego wiersze do HBase 4. Sprawdzenie, że dodały się jakieś wiersze 	<ol style="list-style-type: none"> 1. Server włącza się (procesor w NiFi jest aktywny) 2. Tabela nie ma wierszy 3. Skrypt włącza się 4. Dodały się wiersze 	<ol style="list-style-type: none"> 1.  2.

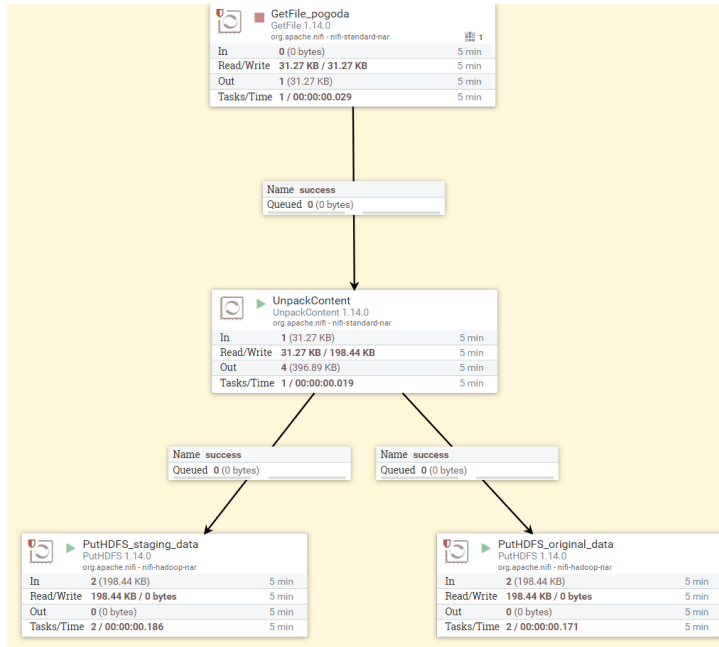
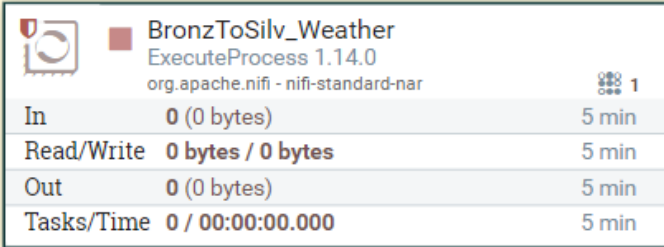
Cel testu	Kroki	Oczekiwany wynik	Potwierdzenie
			<pre>hbase(main):008:0> scan 'projekt' ROW 0 row(s) Took 0.0759 seconds hbase(main):009:0> </pre>
			<p>3.</p>  <p>4.</p> <pre>hbase(main):012:0> scan 'projekt', {'LIMIT' => 1} row COLUMNCELL rowF81147.2023-12-11 11: column=flight:aircraft_1ata, timestamp=2024-01-06T22:17:53.316, value=BC53 05:00 rowF81147.2023-12-11 11: column=flight:aircraft_1icao, timestamp=2024-01-06T22:17:53.316, value=BC53 05:00 rowF81147.2023-12-11 11: column=flight:aircraft_1icao20, timestamp=2024-01-06T22:17:53.316, value=39E 05:00 rowF81147.2023-12-11 11: column=flight:aircraft_registration, timestamp=2024-01-06T22:17:53.316, val 05:00 rowF81147.2023-12-11 11: column=flight:airline_1ata, timestamp=2024-01-06T22:17:53.316, value=FR 05:00 rowF81147.2023-12-11 11: column=flight:airline_1icao, timestamp=2024-01-06T22:17:53.316, value=AFR 05:00 rowF81147.2023-12-11 11: column=flight:airline_name, timestamp=2024-01-06T22:17:53.316, value=air fr 05:00 rowF81147.2023-12-11 11: column=flight:arrival_actual_runway, timestamp=2024-01-06T22:17:53.316, val 05:00 rowF81147.2023-12-11 11: column=flight:arrival_1airport, timestamp=2024-01-06T22:17:53.316, value=FR 05:00 rowF81147.2023-12-11 11: column=flight:arrival_1day, timestamp=2024-01-06T22:17:53.316, value=31 05:00 rowF81147.2023-12-11 11: column=flight:arrival_1day, timestamp=2024-01-06T22:17:53.316, value=31 05:00 rowF81147.2023-12-11 11: column=flight:arrival_delay, timestamp=2024-01-06T22:17:53.316, value=18 05:00 rowF81147.2023-12-11 11: column=flight:arrival_estimated, timestamp=2024-01-06T22:17:53.316, value=2 05:00 rowF81147.2023-12-11 11: column=flight:arrival_1ata, timestamp=2024-01-06T22:17:53.316, value=CG 05:00 rowF81147.2023-12-11 11: column=flight:arrival_1icao, timestamp=2024-01-06T22:17:53.316, value=DFG 05:00 rowF81147.2023-12-11 11: column=flight:arrival_month, timestamp=2024-01-06T22:17:53.316, value=12 05:00 rowF81147.2023-12-11 11: column=flight:arrival_scheduled, timestamp=2024-01-06T22:17:53.316, value=2 05:00 rowF81147.2023-12-11 11: column=flight:arrival_terminal, timestamp=2024-01-06T22:17:53.316, value=2F 05:00 rowF81147.2023-12-11 11: column=flight:arrival_timezone, timestamp=2024-01-06T22:17:53.316, value=fr 05:00 rowF81147.2023-12-11 11: column=flight:arrival_year, timestamp=2024-01-06T22:17:53.316, value=2023 05:00 rowF81147.2023-12-11 11: column=flight:departure_actual_runway, timestamp=2024-01-06T22:17:53.316, v 05:00 rowF81147.2023-12-11 11: column=flight:departure_1airport, timestamp=2024-01-06T22:17:53.316, value= 05:00 rowF81147.2023-12-11 11: column=flight:departure_estimated, timestamp=2024-01-06T22:17:53.316, value 05:00 rowF81147.2023-12-11 11: column=flight:departure_gate, timestamp=2024-01-06T22:17:53.316, value=2 05:00 rowF81147.2023-12-11 11: column=flight:departure_1ata, timestamp=2024-01-06T22:17:53.316, value=MM 05:00 rowF81147.2023-12-11 11: column=flight:departure_1icao, timestamp=2024-01-06T22:17:53.316, value=EMA 05:00 rowF81147.2023-12-11 11: column=flight:departure_scheduled, timestamp=2024-01-06T22:17:53.316, value 05:00 rowF81147.2023-12-11 11: column=flight:departure_terminal, timestamp=2024-01-06T22:17:53.316, value 05:00 rowF81147.2023-12-11 11: column=flight:departure_timezone, timestamp=2024-01-06T22:17:53.316, value 05:00 rowF81147.2023-12-11 11: column=flight:flight_delay, timestamp=2024-01-06T22:17:53.316, value=2023-12 05:00 rowF81147.2023-12-11 11: column=flight:flight_1ata, timestamp=2024-01-06T22:17:53.316, value=AF1107 05:00 rowF81147.2023-12-11 11: column=flight:flight_1icao, timestamp=2024-01-06T22:17:53.316, value=AF1107 05:00 rowF81147.2023-12-11 11: column=flight:flight_number, timestamp=2024-01-06T22:17:53.316, value=1147 05:00 rowF81147.2023-12-11 11: column=flight:flight_status, timestamp=2024-01-06T22:17:53.316, value=activ 05:00 rowF81147.2023-12-11 11: column=weather:arrival_act_piled, timestamp=2024-01-06T22:17:53.316, value 05:00 rowF81147.2023-12-11 11: column=weather:departure_cloud_coverage_avg, timestamp=2024-01-06T22:17:41.979, value=0 05:00 rowF81147.2023-12-11 11: column=weather:departure_cloud_coverage_avg_status, timestamp=2024-01-06T22 05:00 rowF81147.2023-12-11 11: column=weather:departure_humidity_avg, timestamp=2024-01-06T22:17:41.979, v 05:00 rowF81147.2023-12-11 11: column=weather:departure_humidity_avg_status, timestamp=2024-01-06T22:17:41 05:00 rowF81147.2023-12-11 11: column=weather:departure_rain, timestamp=2024-01-06T22:17:41.979, value=0 05:00 rowF81147.2023-12-11 11: column=weather:departure_rain_status, timestamp=2024-01-06T22:17:41.979, va 05:00 rowF81147.2023-12-11 11: column=weather:departure_snow_height, timestamp=2024-01-06T22:17:41.979, v 05:00 rowF81147.2023-12-11 11: column=weather:departure_snow_height_status, timestamp=2024-01-06T22:17:41 05:00 rowF81147.2023-12-11 11: column=weather:departure_station_name, timestamp=2024-01-06T22:17:41.979, v 05:00 rowF81147.2023-12-11 11: column=weather:departure_station_name, timestamp=2024-01-06T22:17:41.979, v 05:00 rowF81147.2023-12-11 11: column=weather:departure_temperature_avg, timestamp=2024-01-06T22:17:41.979 05:00 rowF81147.2023-12-11 11: column=weather:departure_temperature_avg_status, timestamp=2024-01-06T22:17 05:00 rowF81147.2023-12-11 11: column=weather:departure_temperature_ground_min, timestamp=2024-01-06T22:17 05:00 rowF81147.2023-12-11 11: column=weather:departure_temperature_ground_min_status, timestamp=2024-01-06T22:17:41.979, value=11.4 05:00 rowF81147.2023-12-11 11: column=weather:departure_temperature_max, timestamp=2024-01-06T22:17:41.979 05:00 rowF81147.2023-12-11 11: column=weather:departure_temperature_max_status, timestamp=2024-01-06T22:17 05:00 rowF81147.2023-12-11 11: column=weather:departure_temperature_min, timestamp=2024-01-06T22:17:41.979 05:00 rowF81147.2023-12-11 11: column=weather:departure_temperature_min_status, timestamp=2024-01-06T22:17 05:00 rowF81147.2023-12-11 11: column=weather:departure_wind_speed_avg, timestamp=2024-01-06T22:17:41.979, 05:00 rowF81147.2023-12-11 11: column=weather:departure_wind_speed_avg_status, timestamp=2024-01-06T22:17 05:00 1 row(s) Took 0.0601 seconds hbase(main):013:0> count 'projekt' 01 row(s)</pre>

Cel testu	Kroki	Oczekiwany wynik	Potwierdzenie
Sprawdzenie, czy przy już dodanych danych, dobrze dodadzą się nowe dane flights	<ol style="list-style-type: none"> 1. Dodanie danych do lokalnego systemu plików 2. Włączenie flow pobierającego dane do poziomu brązowego 3. Sprawdzenie, że pliki zostały pobrane i umieszczone w odpowiednich folderach 4. Włączenie flow poziomu srebrnego 5. Sprawdzenie, że pliki usunęły się z [/bronze/stage/samoloty oraz dodały do poziomu srebrnego na HDFS 6. Włączenie skryptu ładującego dane do poziomu złotego 7. Sprawdzenie zawartości folderu [/gold na HDFS 8. Włączenie skryptu aktualizującego tabelę w HBase 9. Sprawdzenie, zawartości tabeli w HBase 	<ol style="list-style-type: none"> 1. Pliki są w lokalnym FS 2. Flow działa 3. Pliki zniknęły z lokalnego FS, pojawiły się w bronze oraz stage/bronze 4. Flow działa 5. Pliki zniknęły z stage/bronze i pojawiły się w nowe w silver/flights 6. Flow działa 7. Dodały się nowe pliki w gold 8. Flow działa 9. Dowwały się nowe wiersze 	<ol style="list-style-type: none"> 1. Wklejono 4520 plików z lotami do odpowiedniego folderu 2.  <p>W trakcie</p>  <p>Po</p> <ol style="list-style-type: none"> 3. <pre> vagrant@node1:~/projekt/nifi_in/samoloty/flights\$ hadoop fs -count /user/projekt/bronze/samoloty/ 1 13923 901161918 /user/projekt/bronze/samoloty vagrant@node1:~/projekt/nifi_in/samoloty/flights\$ hadoop fs -count /user/projekt/bronze/stage/samoloty/ 1 4526 305936855 /user/projekt/bronze/stage/samoloty </pre>

Cel testu	Kroki	Oczekiwany wynik	Potwierdzenie
			<p>Po (dodały się; jest 4526, nie 4520, przez te 6 plików taxes)</p>  <p>4.</p> <pre>vagrant@node1:~\$ hadoop fs -count -h /user/projekt/silver/flights 12 129 22.2 M /user/projekt/silver/flights vagrant@node1:~\$ </pre> <p>Przed</p>  <p>W trakcie</p> <pre>vagrant@node1:~/projekt/nifi_in/samoloty/flights\$ hadoop fs -count /user/projekt/bronze/stage/samoloty 1 6 49021 /user/projekt/bronze/stage/samoloty</pre> <p>Po w lokalnym (te 6 to taxes)</p> <p>5.</p>

Cel testu	Kroki	Oczekiwany wynik	Potwierdzenie
			<pre> vagrant@node1:~\$ hadoop fs -ls -h /user/projekt/silver/flights Found 3 items -rw-r--r-- 1 root supergroup 0 2024-01-06 22:51 /user/projekt/silver/flights/_SUCCESS drwxr-xr-x - root supergroup 0 2024-01-06 21:53 /user/projekt/silver/flights/year=2023 drwxr-xr-x - root supergroup 0 2024-01-06 21:54 /user/projekt/silver/flights/year=2024 vagrant@node1:~\$ hadoop fs -ls -h /user/projekt/silver/flights/year=2024/month=1 Found 5 items drwxr-xr-x - root supergroup 0 2024-01-06 21:57 /user/projekt/silver/flights/year=2024/month=1/day=1 drwxr-xr-x - root supergroup 0 2024-01-06 21:57 /user/projekt/silver/flights/year=2024/month=1/day=2 drwxr-xr-x - root supergroup 0 2024-01-06 21:57 /user/projekt/silver/flights/year=2024/month=1/day=3 drwxr-xr-x - root supergroup 0 2024-01-06 22:51 /user/projekt/silver/flights/year=2024/month=1/day=4 drwxr-xr-x - root supergroup 0 2024-01-06 22:51 /user/projekt/silver/flights/year=2024/month=1/day=5 drwxr-xr-x - root supergroup 0 2024-01-06 22:51 /user/projekt/silver/flights/year=2024/month=1/day=6 vagrant@node1:~\$ hadoop fs -count -h /user/projekt/silver/flights 14 159 31.5 M /user/projekt/silver/flights vagrant@node1:~\$ hadoop fs -ls -h /user/projekt/silver/flights/year=2024/month=1/day=6 Found 10 items -rw-r--r-- 1 root supergroup 284.8 K 2024-01-06 22:51 /user/projekt/silver/flights/year=2024/month=1/day=6/part-00000-1610ba06-e378 -ae4c-bc3d-73b00c7bcafl.c000.snappy.parquet -rw-r--r-- 1 root supergroup 388.2 M 2024-01-06 22:50 /user/projekt/silver/flights/year=2024/month=1/day=6/part-00000-30e2e0bd-1e3e -4f87-8761-06668f1fbd3.c000.snappy.parquet -rw-r--r-- 1 root supergroup 374.9 K 2024-01-06 22:49 /user/projekt/silver/flights/year=2024/month=1/day=6/part-00000-73fb52bd-3736 -06fd-b89b-0515d859dc97.c000.snappy.parquet -rw-r--r-- 1 root supergroup 335.6 K 2024-01-06 22:50 /user/projekt/silver/flights/year=2024/month=1/day=6/part-00000-cc98a280-a7f1 -1173-06fb-fed14f3097fc.c000.snappy.parquet -rw-r--r-- 1 root supergroup 352.2 K 2024-01-06 22:51 /user/projekt/silver/flights/year=2024/month=1/day=6/part-00000-d1ffdbd1-abf2 -4e9a-ab2e-e75764f13bf.c000.snappy.parquet -rw-r--r-- 1 root supergroup 246.9 K 2024-01-06 22:51 /user/projekt/silver/flights/year=2024/month=1/day=6/part-00001-1610ba06-e378 -ae4c-bc3d-73b00c7bcafl.c000.snappy.parquet -rw-r--r-- 1 root supergroup 409.2 K 2024-01-06 22:50 /user/projekt/silver/flights/year=2024/month=1/day=6/part-00001-30e2e0bd-1e3e -4f87-8761-06668f1fbd3.c000.snappy.parquet -rw-r--r-- 1 root supergroup 287.4 K 2024-01-06 22:49 /user/projekt/silver/flights/year=2024/month=1/day=6/part-00001-73fb52bd-3736 -06fd-b89b-0515d859dc97.c000.snappy.parquet -rw-r--r-- 1 root supergroup 410.2 K 2024-01-06 22:50 /user/projekt/silver/flights/year=2024/month=1/day=6/part-00001-cc98a280-a7f1 -1173-06fb-fed14f3097fc.c000.snappy.parquet -rw-r--r-- 1 root supergroup 380.8 K 2024-01-06 22:51 /user/projekt/silver/flights/year=2024/month=1/day=6/part-00001-d1ffdbd1-abf2 -4e9a-ab2e-e75764f13bf.c000.snappy.parquet vagrant@node1:~\$ vagrant@node1:~\$ hadoop fs -count -h /user/projekt/silver/flights 14 159 31.5 M /user/projekt/silver/flights vagrant@node1:~\$ </pre> <p>Wcześniej były pliki do 04.01, teraz z części 04.01 do 06.01 około 23:20 - są jak widać</p> <p>6. (nie zrobiono rzutu)</p> <p>7.</p> <pre> vagrant@node1:~\$ hadoop fs -count -h /user/projekt/gold/flights_weather 1 210 5.1 M /user/projekt/gold/flights_weather vagrant@node1:~\$ </pre> <p>Przed</p> <pre> vagrant@node1:~/projekt/nifi_in/samoloty/flights\$ hadoop fs -count -h /user/projekt/gold/flights_weather 3 457 11.2 M /user/projekt/gold/flights_weather </pre> <p>Po</p> <p>8.(nie zrobiono rzutu)</p> <p>9.</p>

Cel testu	Kroki	Oczekiwany wynik	Potwierdzenie
			<pre> Took 0.2709 seconds > ? hbase(main):023:0> scan 'projekt', {'LIMIT' => 1} ROW COLUMN=CELL rowACE871-2024-01-05 11 column:flight:aircraft_data, timestamp=2024-01-06T23:00:03.369, value=130:00 rowACE871-2024-01-05 11 column:flight:aircraft_icar, timestamp=2024-01-06T23:00:03.369, value=130:00 rowACE871-2024-01-05 11 column:flight:aircraft_icar2, timestamp=2024-01-06T23:00:03.369, val=130:00 rowACE871-2024-01-05 11 column:flight:aircraft_registration, timestamp=2024-01-06T23:00:03.369, value=130:00 rowACE871-2024-01-05 11 column:flight:airline_data, timestamp=2024-01-06T23:00:03.369, value=130:00 rowACE871-2024-01-05 11 column:flight:airline_icar, timestamp=2024-01-06T23:00:03.369, value=130:00 rowACE871-2024-01-05 11 column:flight:airline_name, timestamp=2024-01-06T23:00:03.369, value=130:00 rowACE871-2024-01-05 11 column:flight:arrival_actual_runway, timestamp=2024-01-06T23:00:03.369, value=130:00 0:00 rowACE871-2024-01-05 11 column:flight:arrival_airport, timestamp=2024-01-06T23:00:03.369, val=130:00 rowACE871-2024-01-05 11 column:flight:arrival_day, timestamp=2024-01-06T23:00:03.369, value=5:130:00 rowACE871-2024-01-05 11 column:flight:arrival_delay, timestamp=2024-01-06T23:00:03.369, value=130:00 rowACE871-2024-01-05 11 column:flight:arrival_estimated, timestamp=2024-01-06T23:00:03.369, v=130:00 rowACE871-2024-01-05 11 column:flight:arrival_data, timestamp=2024-01-06T23:00:03.369, value=130:00 rowACE871-2024-01-05 11 column:flight:arrival_icar, timestamp=2024-01-06T23:00:03.369, value=130:00 rowACE871-2024-01-05 11 column:flight:arrival_month, timestamp=2024-01-06T23:00:03.369, value=130:00 rowACE871-2024-01-05 11 column:flight:arrival_scheduled, timestamp=2024-01-06T23:00:03.369, v=130:00 rowACE871-2024-01-05 11 column:flight:arrival_tutorial, timestamp=2024-01-06T23:00:03.369, va=130:00 rowACE871-2024-01-05 11 column:flight:arrival_timezone, timestamp=2024-01-06T23:00:03.369, va=130:00 rowACE871-2024-01-05 11 column:flight:arrival_year, timestamp=2024-01-06T23:00:03.369, value=130:00 rowACE871-2024-01-05 11 column:flight:departure_actual_runway, timestamp=2024-01-06T23:00:03.369, value=130:00 +00:00 rowACE871-2024-01-05 11 column:flight:departure_airport, timestamp=2024-01-06T23:00:03.369, v=130:00 rowACE871-2024-01-05 11 column:flight:departure_estimated, timestamp=2024-01-06T23:00:03.369, value=130:00 rowACE871-2024-01-05 11 column:flight:departure_gate, timestamp=2024-01-06T23:00:03.369, valu=130:00 rowACE871-2024-01-05 11 column:flight:departure_data, timestamp=2024-01-06T23:00:03.369, valu=130:00 rowACE871-2024-01-05 11 column:flight:departure_icar, timestamp=2024-01-06T23:00:03.369, valu=130:00 rowACE871-2024-01-05 11 column:flight:departure_scheduled, timestamp=2024-01-06T23:00:03.369, value=130:00 rowACE871-2024-01-05 11 column:flight:departure_timezone, timestamp=2024-01-06T23:00:03.369, value=Europe/Warsaw 130:00 rowACE871-2024-01-05 11 column:flight:flight_data, timestamp=2024-01-06T23:00:03.369, value=2:130:00 rowACE871-2024-01-05 11 column:flight:flight_icar, timestamp=2024-01-06T23:00:03.369, value=A:130:00 rowACE871-2024-01-05 11 column:flight:flight_number, timestamp=2024-01-06T23:00:03.369, value=130:00 rowACE871-2024-01-05 11 column:flight:flight_status, timestamp=2024-01-06T23:00:03.369, value=130:00 rowACE871-2024-01-05 11 column:weather:arrival_to_poland, timestamp=2024-01-06T23:00:03.369, value=230:00 rowACE871-2024-01-05 11 column:weather:departure_cloud_coverage_avg, timestamp=2024-01-06T23:130:00 rowACE871-2024-01-05 11 column:weather:departure_cloud_coverage_avg_status, timestamp=2024-01-06T23:59:51.979, value=true rowACE871-2024-01-05 11 column:weather:departure_humidity_avg, timestamp=2024-01-06T23:59:51.979, value=0.0 rowACE871-2024-01-05 11 column:weather:departure_humidity_avg_status, timestamp=2024-01-06T23:59:51.979, value=true rowACE871-2024-01-05 11 column:weather:departure_rain_status, timestamp=2024-01-06T23:59:51.979, val=130:00 rowACE871-2024-01-05 11 column:weather:departure_rain_status, timestamp=2024-01-06T23:59:51.979, value=0 rowACE871-2024-01-05 11 column:weather:departure_rain_status, timestamp=2024-01-06T23:59:51.979, value=0 rowACE871-2024-01-05 11 column:weather:departure_rain_type, timestamp=2024-01-06T23:59:51.979, value=79 rowACE871-2024-01-05 11 column:weather:departure_snow_height_status, timestamp=2024-01-06T23:59:51.979, value=8 rowACE871-2024-01-05 11 column:weather:departure_snow_height_status, timestamp=2024-01-06T23:59:51.979, value=8 rowACE871-2024-01-05 11 column:weather:departure_station_code, timestamp=2024-01-06T23:59:51.979, value=252200150 rowACE871-2024-01-05 11 column:weather:departure_station_name, timestamp=2024-01-06T23:59:51.979, value=0002200-BIELAWY rowACE871-2024-01-05 11 column:weather:departure_temperature_avg, timestamp=2024-01-06T23:59:51.979, value=5.7 rowACE871-2024-01-05 11 column:weather:departure_temperature_avg_status, timestamp=2024-01-06T23:59:51.979, value=true rowACE871-2024-01-05 11 column:weather:departure_temperature_ground_avg, timestamp=2024-01-06T23:59:51.979, value=4 rowACE871-2024-01-05 11 column:weather:departure_temperature_ground_avg_status, timestamp=2024-01-06T23:59:51.979, value=true rowACE871-2024-01-05 11 column:weather:departure_temperature_max, timestamp=2024-01-06T23:59:51.979, value=9 rowACE871-2024-01-05 11 column:weather:departure_temperature_max_status, timestamp=2024-01-06T23:59:51.979, value=true rowACE871-2024-01-05 11 column:weather:departure_temperature_min, timestamp=2024-01-06T23:59:51.979, value=9 rowACE871-2024-01-05 11 column:weather:departure_temperature_min_status, timestamp=2024-01-06T23:59:51.979, value=true rowACE871-2024-01-05 11 column:weather:departure_wind_speed_avg, timestamp=2024-01-06T23:59:51.979, value=5.3 rowACE871-2024-01-05 11 column:weather:departure_wind_speed_avg_status, timestamp=2024-01-06T23:59:51.979, value=true 1 row(s) Took 0.0213 seconds hbase(main):024:0> </pre>
			<p>Jak widać dodały się nowe wiersze (jest ich ponad 700, było ~500, są wiersze z 06.01)</p>
<p>Sprawdzenie, czy przy już dodanych danych, dobrze dodadzą się nowe dane weather (do poziomu silver, gdyż są one z 11.2023 -> u nas poprzez dodanie roku będzie to</p>	<ol style="list-style-type: none"> Wygenerowanie sztuczne danych pogodowych poprzez zamianę 10 na 11 w danych z 10.2023 Dodanie danych do folderu na lokalnym FS Uruchomienie Flow w nifi Sprawdzenie zawartości folderów [/bronze/pogoda oraz bronze/stage/pogoda Uruchomienie kolejnego (srebrnego) Flow w NiFi Sprawdzenie zawartości 	<ol style="list-style-type: none"> Dane mamy Dane są na lokalnym FS Flow działa Pojawiły się po 2 nowe pliki (2 w stage, sumarycznie 552 w archiwum) Flow działa Dodały się nowe pliki (było 201) Usun 	<ol style="list-style-type: none">  <p>To są po prostu dane z 2023_10 z zamienionym na 10 na 11 wszędzie (czyli 2024, bo jest +rok)</p> <p>Nic nie zmieni w goldzie (do jest left do flights, a flights jest do 2024.01.06 włącznie w danych)</p>

Cel testu	Kroki	Oczekiwany wynik	Potwierdzenie
11.2024, a nie mamy takich danych z flights, więc nie wpłyną one na golda)	folderu [/silver/weather na HDFS 6. Sprawdzenie zawartości katalogu [/bronze/stage/pogoda na HDFS		<p>Więc analizujemy do silvera</p> <p>2.</p>  <p>3.</p> <pre>vagrant@node1:~/projekt/nifi_in/samoloty/flights\$ hadoop fs -count -h /user/projekt/bronze/pogoda 1 552 166.8 M /user/projekt/bronze/pogoda vagrant@node1:~/projekt/nifi_in/samoloty/flights\$ hadoop fs -count -h /user/projekt/bronze/stage/pogoda 1 2 198.4 K /user/projekt/bronze/stage/pogoda</pre> <p>4.</p> <pre>vagrant@node1:~\$ hadoop fs -count /user/projekt/silver/weather 1 281 14338921 /user/projekt/silver/weather</pre> <p>Przed</p> 

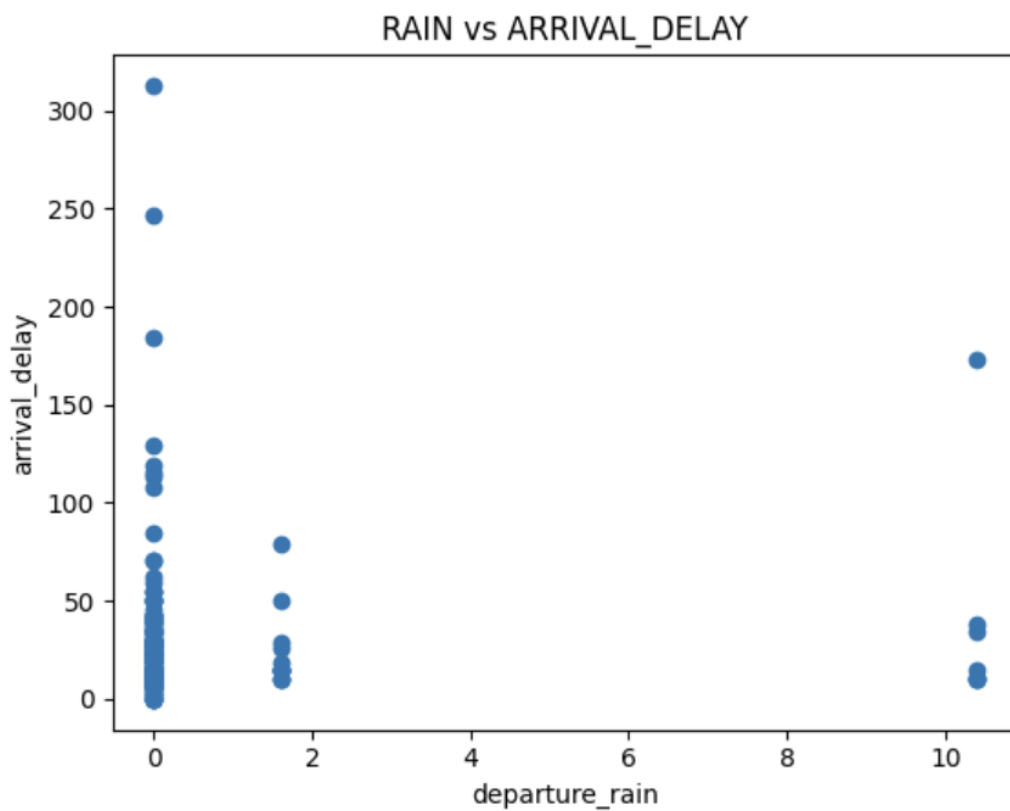
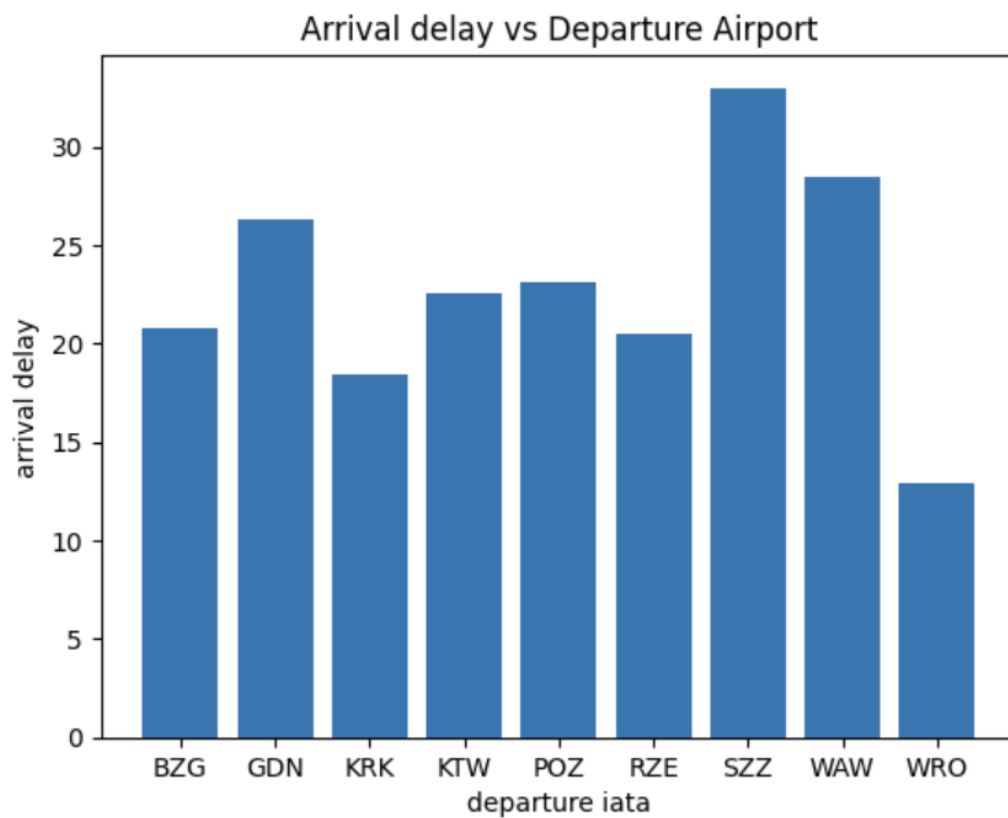
Cel testu	Kroki	Oczekiwany wynik	Potwierdzenie
			<p>5.</p> <pre>vagrant@node1:~/projekt/nifi_in/samoloty/flights\$ hadoop fs -count -h /user/projekt/silver/weather 1 202 13.7 M /user/projekt/silver/weather vagrant@node1:~/projekt/nifi_in/samoloty/flights\$ hadoop fs -count /user/projekt/silver/weather 1 202 14353893 /user/projekt/silver/weather</pre> <p>6.</p> 

UWAGA

[] - /user/projekt

Wizualizacje

Wizualizacje z serving layera na danych do 04.01



Wykresy dla nowych danych (do 06.01 włącznie)

