

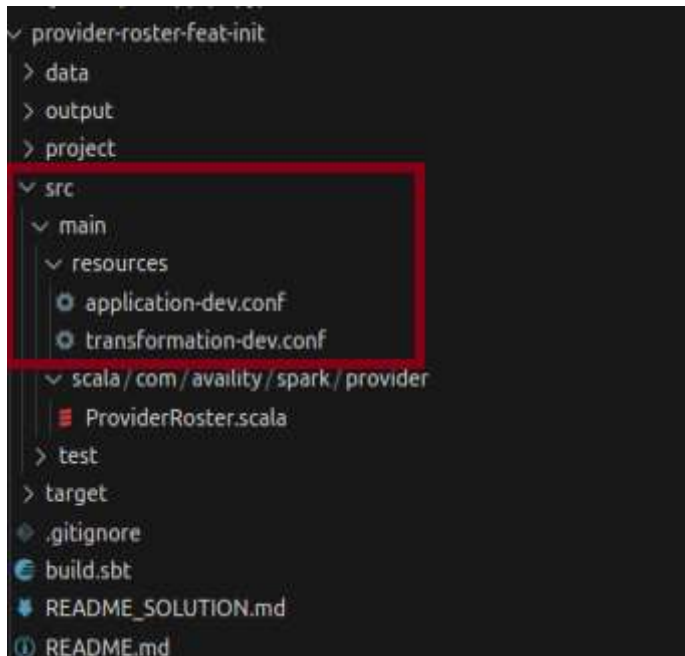
# I. DEPLOYMENT

## 1. Environment Preparation

- Created an Ubuntu 22.04 VM using VirtualBox
- Installed java, scala, spark, code
- Downloaded the zip file from [https://gitlab.com/dane.allen.3/provider-roster/-/blob/feat/init/README.md?ref\\_type=heads](https://gitlab.com/dane.allen.3/provider-roster/-/blob/feat/init/README.md?ref_type=heads)
- extracted the zip file to local folder in ubuntu as "provider-roster-feat-init"

## 2. Application Preparation

- Updated provider-roster-feat-init/build.sbt
- Updated provider-roster-feat-init/src/main/scala/com/availability/spark/provider/ProviderRoster.scala
  - conf file location



- provider-roster-feat-init/src/main/resources/application-dev.conf

```
provider-roster-feat-init > src > main > resources > application-dev.conf
1  app {
2      spark_session {
3          master = "local[*]"
4          appName = "ProviderAnalytics"
5          timeout = "300"
6          # you can add your additional config here
7      }
8
9      provider_csv {
10         csv_path = "data/providers.csv"
11         csv_header = "true"
12         csv_delimiter = "|"
13         csv_schema = [
14             { name = "provider_id", type = "LongType", nullable = false },
15             { name = "provider_specialty", type = "StringType", nullable = false },
16             { name = "first_name", type = "StringType", nullable = false },
17             { name = "middle_name", type = "StringType", nullable = false },
18             { name = "last_name", type = "StringType", nullable = false }
19         ]
20     }
21
22     visit_csv {
23         csv_path = "data/visits.csv"
24         csv_header = "false"
25         csv_delimiter = ","
26         csv_schema = [
27             { name = "visit_id", type = "LongType", nullable = false },
28             { name = "provider_id", type = "LongType", nullable = false },
29             { name = "date_of_visit", type = "DateType", nullable = false }
30         ]
31     }
32
33     output_paths {
34         answer_to_question_1_path = "output/answer_to_question_1"
35         answer_to_question_2_path = "output/answer_to_question_2"
36     }
37 }
38
```

- provider-roster-feat-init/src/main/resources/transformation-dev.conf

```
provider-roster-feat-init > src > main > resources > ⚙ transformation-dev.conf
1  transformations {
2      question_1_cte = """
3          -- CTE 1: Create `visited_provider_table` by joining provider_table and visit_table
4          WITH visited_provider_table AS (
5              SELECT
6                  p.provider_id,
7                  p.provider_specialty,
8                  CONCAT(p.first_name, ' ', p.middle_name, ' ', p.last_name) AS full_name,
9                  DATE_FORMAT(v.date_of_visit, 'yyyy-MM') AS year_month,
10                 v.visit_id
11             FROM provider_table p
12             LEFT JOIN visit_table v
13             ON p.provider_id = v.provider_id
14         ),
15
16         -- CTE 2: Aggregate visit counts for each provider and specialty
17         visited_provider_aggregated AS (
18             SELECT
19                 provider_specialty,
20                 provider_id,
21                 full_name,
22                 COUNT(provider_id) AS count_of_visit
23             FROM visited_provider_table
24             GROUP BY provider_specialty, provider_id, full_name
25             ORDER BY provider_specialty, provider_id, full_name
26         ),
27
```

### 3. Application Logics

- The application is consisted of 7 steps
- Step 1: Read application and transformation configs
- Step 2: Initialize SparkSession
- Step 3: Read input files (providers.csv and visits.csv) to DataFrames
- Step 4: Register the DataFrames as tables
- Step 5: Extract the sql transformation scripts from transformation.conf file
- Step 6: create dataframes from the query strings representing the result (solution)
- Step 7: write the dataframes as json files

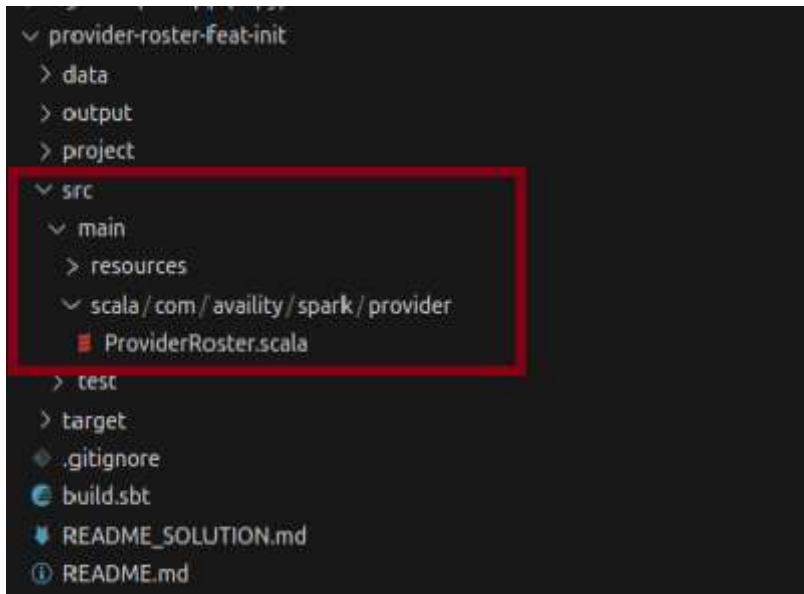
### 4. Compile + Create JAR

- Open vs code in ubuntu
- Open the folder provider-roster-feat-init
- Open the vs code integrated terminal
- Run the following from the command line "sbt clean update compile assembly"

```
• vboxuser@erick:~/udk_project/provider-roster-feat-init$ sbt clean update compile assembly
[info] welcome to sbt 1.6.2 (Private Build Java 1.8.0_432)
[info] loading settings for project provider-roster-feat-init-build from plugins.sbt ...
[info] loading project definition from /home/vboxuser/udk_project/provider-roster-feat-init/project
[info] loading settings for project provider-roster-feat-init from build.sbt ...
[info] set current project to provider (in build file:/home/vboxuser/udk_project/provider-roster-feat-init/)
[info] Executing in batch mode. For better performance use sbt's shell
[success] Total time: 0 s, completed Jan 10, 2025 2:15:47 AM
[success] Total time: 3 s, completed Jan 10, 2025 2:15:50 AM
[info] compiling 1 Scala source to /home/vboxuser/udk_project/provider-roster-feat-init/target/scala-2.12/classes ...
[success] Total time: 4 s, completed Jan 10, 2025 2:15:54 AM
[info] Strategy 'discard' was applied to a file (Run the task at debug level to see details)
[warn] Ignored unknown package option FixedTimestamp(Some(1262304000000))
[success] Total time: 1 s, completed Jan 10, 2025 2:15:55 AM
```

## 5. JSON Output

- Once the uber jar is created in the target/scala-2.12/provider.jar, then run the following command line
- "spark-submit --class com.availity.spark.provider.ProviderRoster target/scala-2.12/provider.jar"



- The above step will produce 2 output json folders representing the output files for the 2 questions
  - Question 1:  
Given the two data datasets, calculate the total number of visits per provider. The resulting set should contain the provider's ID, name, specialty, along with the number of visits. Output the report in json, partitioned by the provider's specialty.

```

1- {
2-   "provider_specialty": {
3-     "Sports Medicine": [{}],
270-     "Nephrology": [{}],
637-     "Rheumatology": [{}],
1049-     "Optometry": [{}],
1396-     "Geriatric Medicine": [{}],
1673-     "Urology": [{}],
2265-     "Chiropractic": [{}],
2497-     "Psychiatry": [{}],
2834-     "Dermatology": [{}],
3151-     "Cardiology": [{}],
3523-     "Hematology": [{}],
3855-     "Internal Medicine": [{}],
4122-     "Unknown Provider": [{}],
4449-     "Gastroenterology": [{}],
4746-     "General Practice": [
4747-       {
4748-         "provider_id": 410,
4749-         "full_name": "Raquel B Klein",
4750-         "count_of_visit": 19
4751-       },
4752-       {
4753-         "provider_id": 3292,
4754-         "full_name": "Edward B Mueller",
4755-         "count_of_visit": 25
4756-       },
4757-       {
4758-         "provider_id": 3421,
4759-         "full_name": "Ron B Keeling",
4760-         "count_of_visit": 19
4761-       },
4762-     ]

```

- Question 2:

Given the two datasets, calculate the total number of visits per provider per month. The resulting set should contain the provider's ID, the month, and total number of visits. Output the result set in json.

```
1 ▾
2 ▾ "provider_id": {
3 ▾   "57": [ ],
49 ▾   "145": [ ],
91 ▾   "204": [ ],
133 ▾   "402": [ ],
183 ▾   "410": [ ],
221 ▾   "467": [ ],
259 ▾   "499": [ ],
297 ▾   "505": [ ],
339 ▾   "509": [ ],
385 ▾   "548": [ ],
427 ▾   "574": [ ],
473 ▾   "642": [ ],
519 ▾   "667": [ ],
565 ▾   "669": [
566 ▾     {
567 ▾       "year_month": "2021_09",
568 ▾       "count_of_visit": 3
569 ▾     },
570 ▾     {
571 ▾       "year_month": "2021_10",
572 ▾       "count_of_visit": 2
573 ▾     },
574 ▾     {
575 ▾       "year_month": "2021_12",
576 ▾       "count_of_visit": 1
577 ▾     },
578 ▾     {
579 ▾       "year_month": "2022_01",
580 ▾       "count_of_visit": 2
```

- Here are the location of the files



▼ provider-roster-feat-init

> data

▼ output

▼ answer\_to\_question\_1

≡ \_SUCCESS

≡ \_SUCCESS.crc

≡ .part-00000-186f4fa3-007b-4a50-a929-4b6f0831400f-c000.json.crc

{ } part-00000-186f4fa3-007b-4a50-a929-4b6f0831400f-c000.json

▼ answer\_to\_question\_2

≡ \_SUCCESS

≡ \_SUCCESS.crc

≡ .part-00000-42881dcf-16dc-4c3f-99c6-d416b4a32412-c000.json.crc

{ } part-00000-42881dcf-16dc-4c3f-99c6-d416b4a32412-c000.json



## II. ASSUMPTIONS

1. Implemented 2 conf files, i.e.,
  - src/main/resources/application-dev.conf
    - defines the application details pertaining to
    - a. spark\_session (contains the info about the spark session)
    - b. provider.csv (source location, schema, delimiter, etc)
    - c. visit.csv (source location, schema, delimiter, etc)
    - d. output\_paths (contains the location of the output json files)
  - src/main/resources/transformation-dev.conf
    - defines the transformation logic to join the provider with the visit data
    - uses CTEs to handle the joining, aggregation and transformation logics to match the required JSON structure
2. Added "com.typesafe" % "config" % "1.4.2" on build.sbt to simplify the parsing of conf files
3. Implemented logging, i.e. import org.slf4j.LoggerFactory
4. Implemented try, catch, finally
5. Method testing not implemented

# III. DATA OBSERVATION

## 1. In the provider.csv files,

- there are a total of 1000 rows, with 0 null values
- unique provider\_id is 992
- 8 provider\_id have 2 provider\_specialty. See below:
  - 86440
  - 86202
  - 82338
  - 99487
  - 55504
  - 56273
  - 63596
  - 25817

## 2. In the visit.csv,

- there are a total of 22,348 rows, with 0 null values
- no duplicate on visit\_id
- there is provider\_id but provider\_specialty is not available,

## 3. Important Notice

- NO ADJUSTMENT WAS MADE TO CORRECT "THE IMPACT OF PROVIDER\_ID WITH 2 PROVIDER\_SPECIALTY" SINCE THERE IS NO SPECIFIC INSTRUCTION OR DESCRIPTION ABOUT IT.
- The impact will be double counting the visits for those 8 provider\_id, which can be translated as a VISIT on a provider\_id will mean visiting both PROVIDER\_SPECIALTY which can skew the total number of visits when re-aggregated by provider\_specialty by provider\_id.

## IV. DOCUMENTATION

- See the `provider-roster-feat-init/README_SOLUTION.md`.
- It describes the entire solution
- See the 7 images on `provider-roster-feat-init/images`

