



LAB

02 – Cloudera Data Engineering

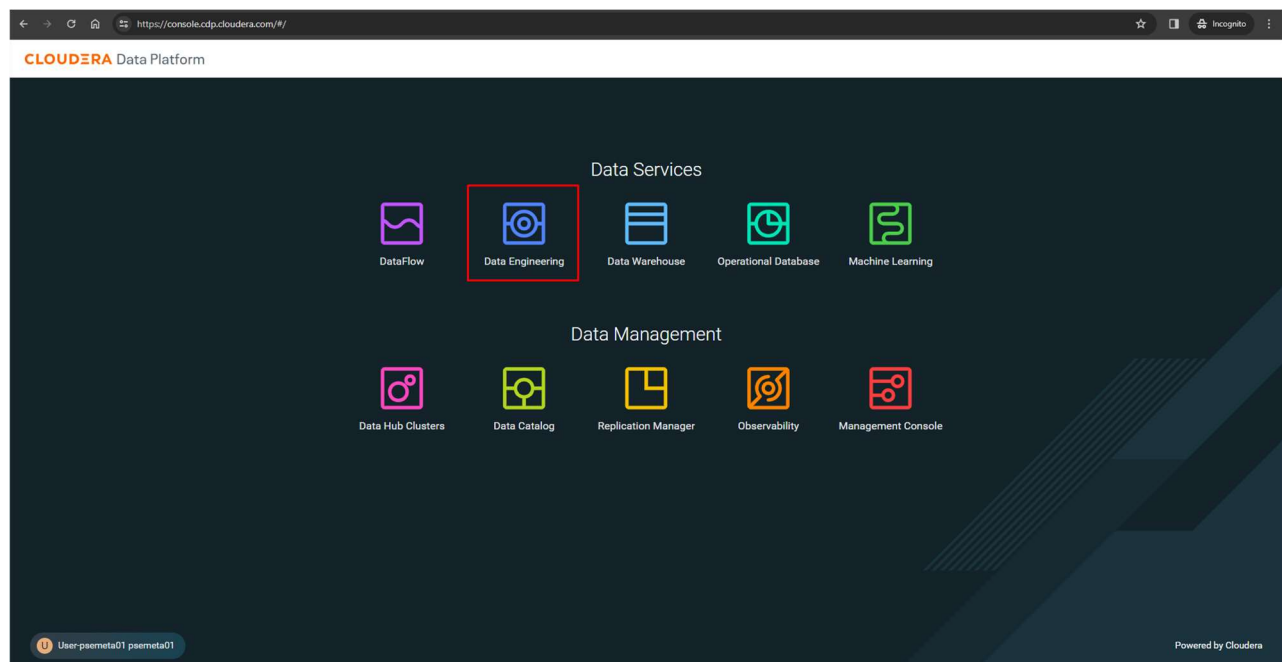
Data Lifecycle on CDP Public Cloud

Data Engineering Lab

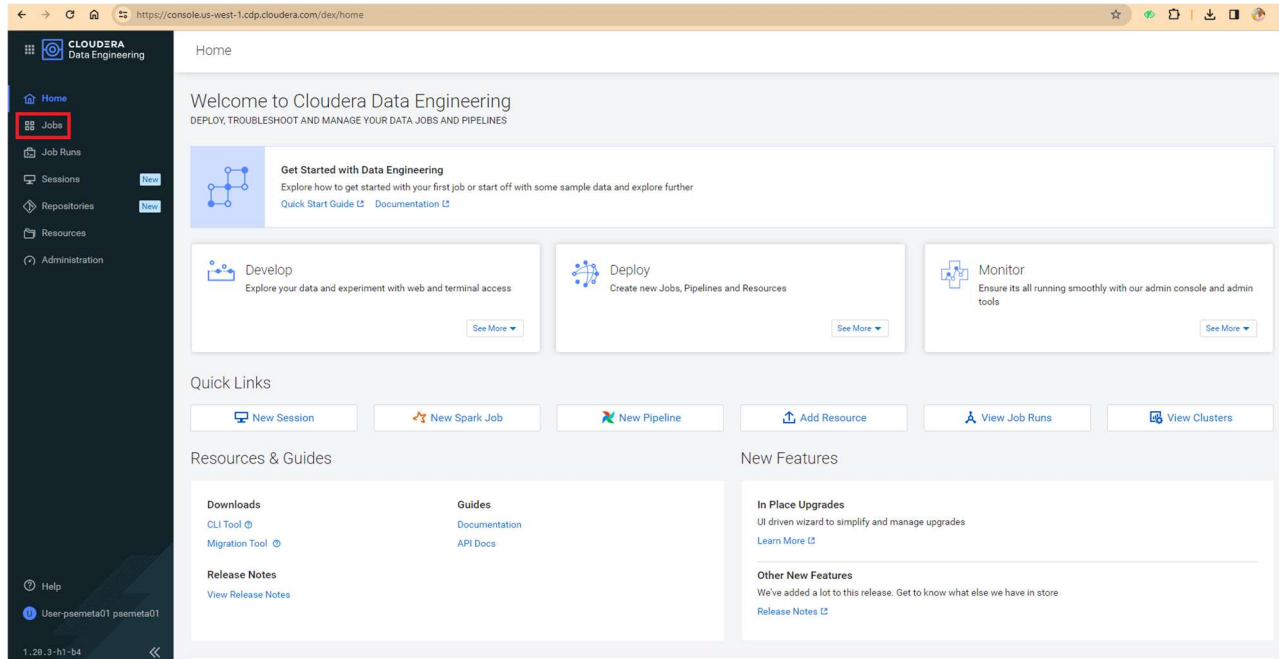
Goals:

- Run a data enrichment process
- Run a process to simulate changes to the data
- Configure the execution of a pipeline using low-code/no-code tools

1. Click on **Data Engineering** from CDP PC Home:



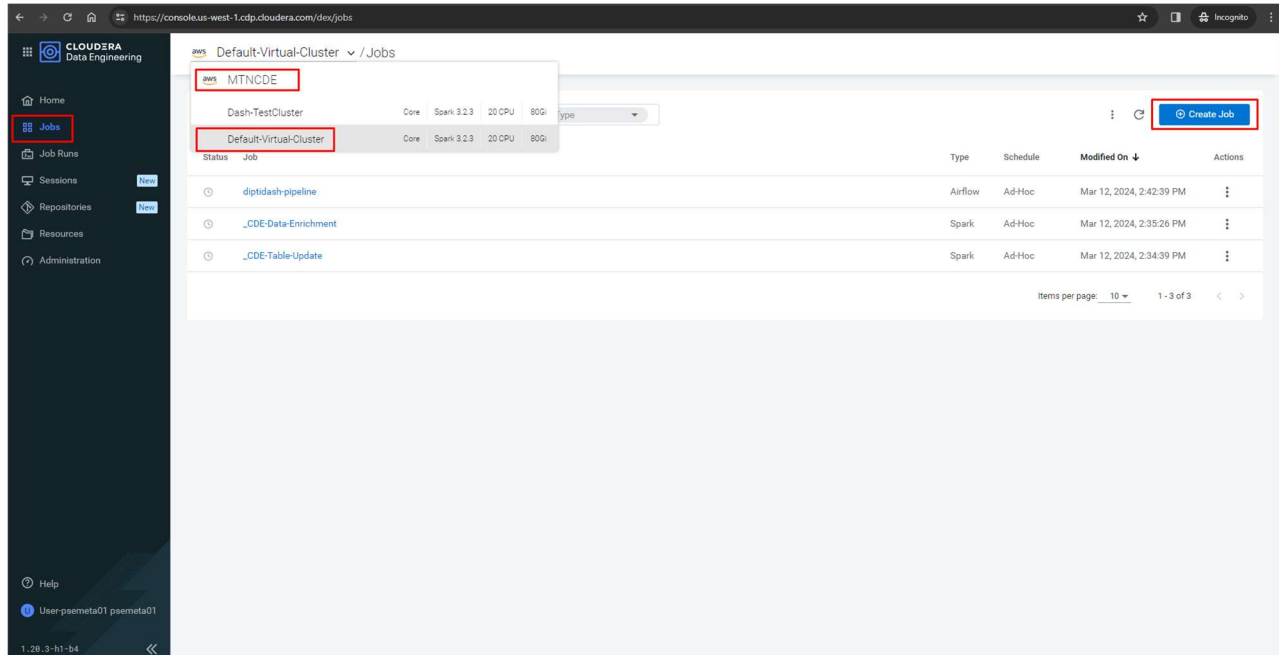
2. The Data Engineering Home shows all the actions that can be done, such as Jobs in Spark and pipelines in Airflow, Resources and useful information/documentation. Click on the option **Jobs** from the left menu to create a dataflow in Airflow.



3. Here the available tasks are listed. For the purposes of this workshop, two Jobs have been configured:

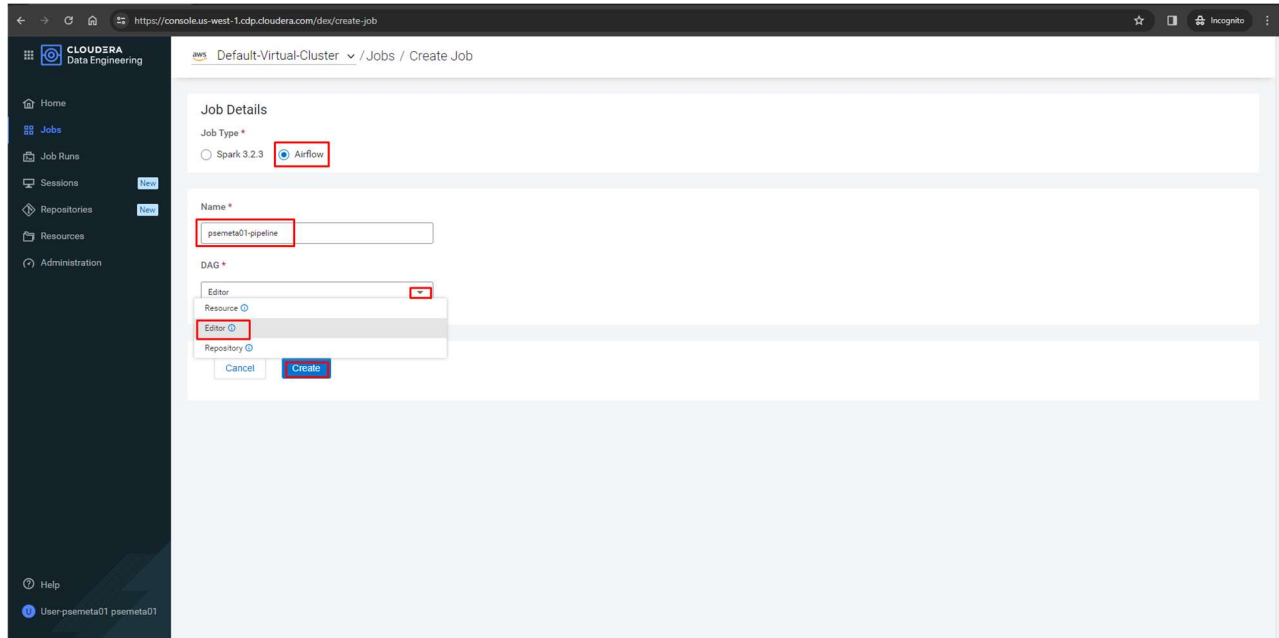
- **_CDE-Table-Update**, generate random changes and enrich table to visualize LakehouseTime Travel functionality.
- **_CDE-Data-Enrichment**, process in Spark (Python) to enrich the data ingested fromKafka and save to a new table.

It is time to create our Job in Airflow. Click on **Create Job**.

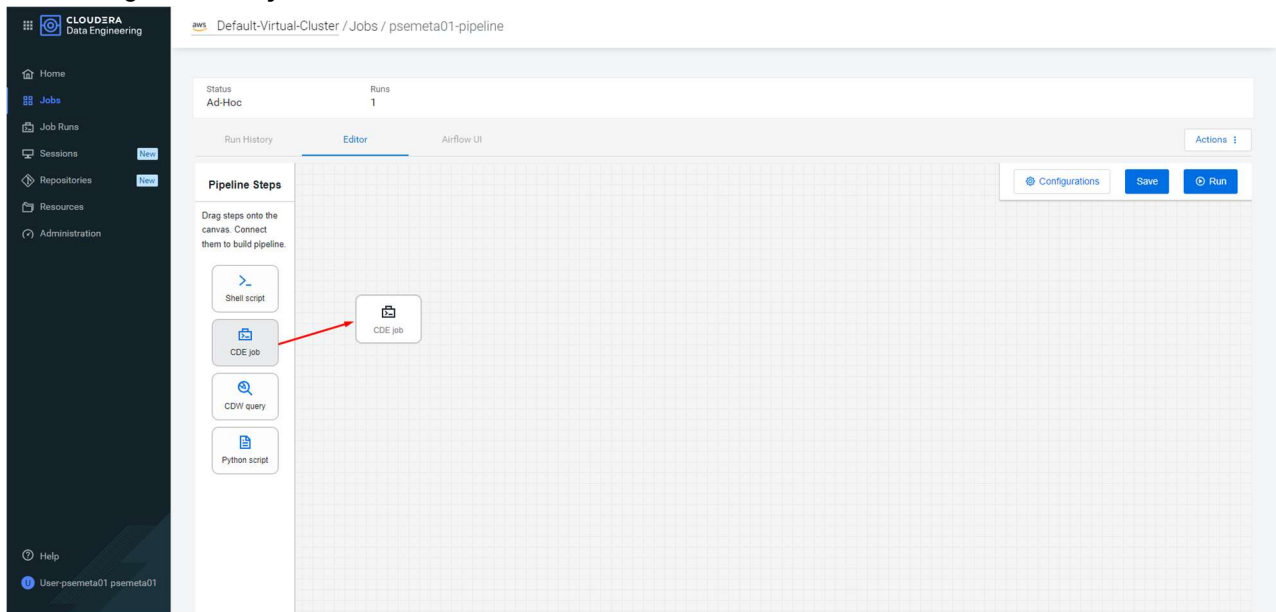


4. In the Job creation form, you must enter the following information:

- Job Type: **Airflow**
- Name: Use the naming <assigned user>-pipeline. Replace <assigned user> with the user assigned to you. For example, **psemeta01**
- DAG: **Editor**, to graphically configure the task. Then, click **Create**.

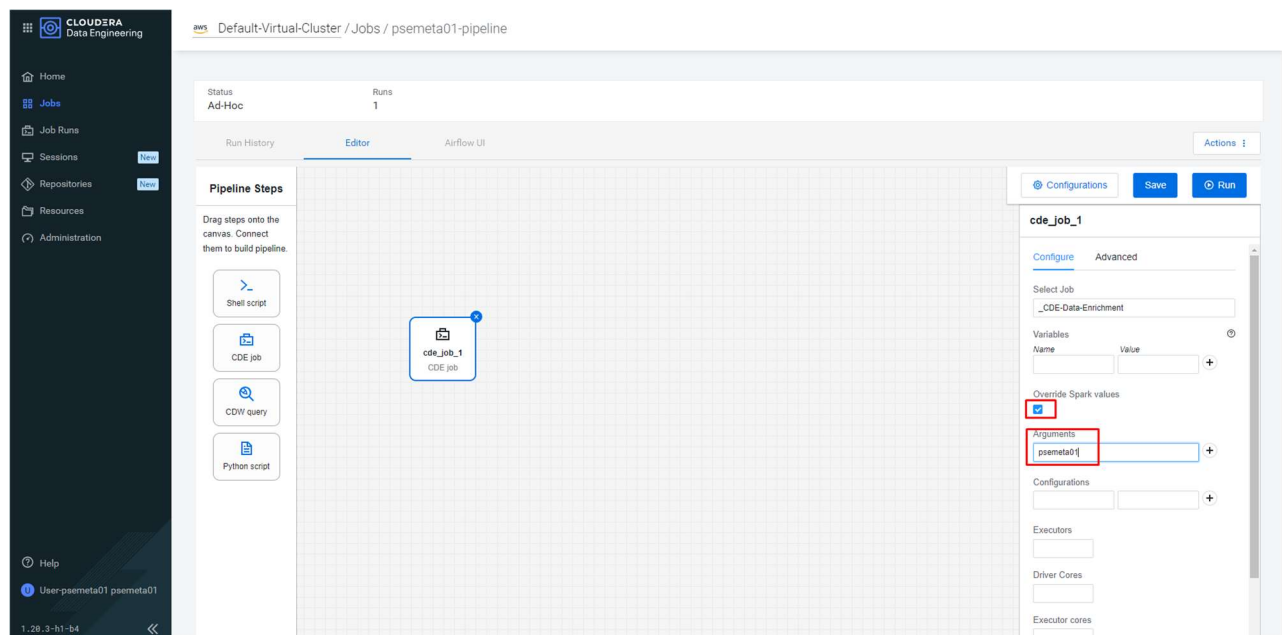
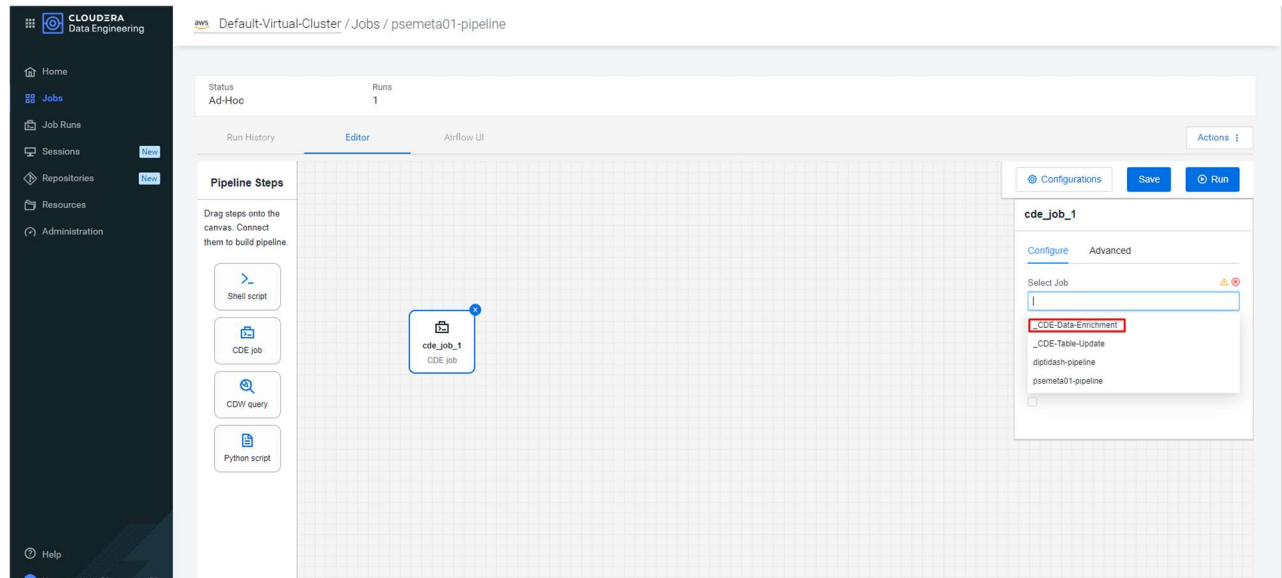


5. On the Job editing screen, select the Editor tab, and you will see the following canvas to drag the steps of the pipeline that we are going to create. In our case, we are going to create two CDE Jobs and relate them. Drag the **CDE job** into the canvas.



6. Let's start with the first Job. Click on the CDE Job button and drag onto the canvas, entering the following settings:

- **Select Job:** select the Job **_CDE-Data-Enrichment**
- Check the checkbox **Override Spark values**. Additional options will appear below.
- **Arguments:** <assigned user>. Use the username assigned to you. For example, **psemeta01**



7. Configure the second Job. Click on the **CDE Job** button and drag onto the canvas, entering the following settings:

- **Select Job:** select the Job **_CDE-Table-Update**
- Check the checkbox **Override Spark values**. Additional options will appear below.
- **Arguments:** <assigned user>. Use the username assigned to you. For example, **psemeta01**

CloudERA Data Engineering console interface showing the pipeline editor for "psemeta01-pipeline".

Header: Status: Ad-Hoc, Runs: 1

Left Sidebar: Home, Jobs, Job Runs, Sessions, Repositories, Resources, Administration, Help, User: psemeta01 psemeta01

Editor View:

- Pipeline Steps:** Drag steps onto the canvas. Connect them to build pipeline. Available steps: Shell script, CDE job, CDW query, Python script.
- Canvas:** A workflow diagram showing a sequence of three "CDE job" steps connected by red arrows. The first step is labeled "cde_job_1" and the second is labeled "CDE job".
- Right Panel:** Actions: Configurations, Save, Run.

CloudERA Data Engineering console interface showing the pipeline editor for "psemeta01-pipeline" with the configuration panel open.

Header: Status: Ad-Hoc, Runs: 1

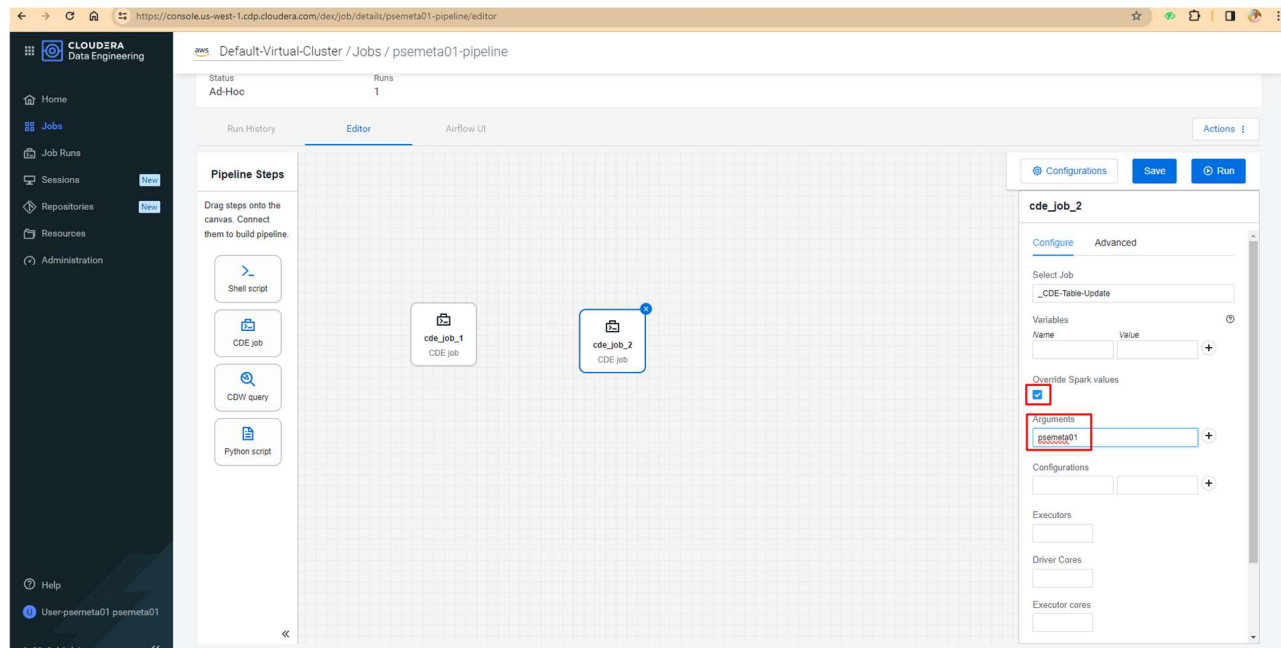
Left Sidebar: Home, Jobs, Job Runs, Sessions, Repositories, Resources, Administration, Help, User: psemeta01 psemeta01

Editor View:

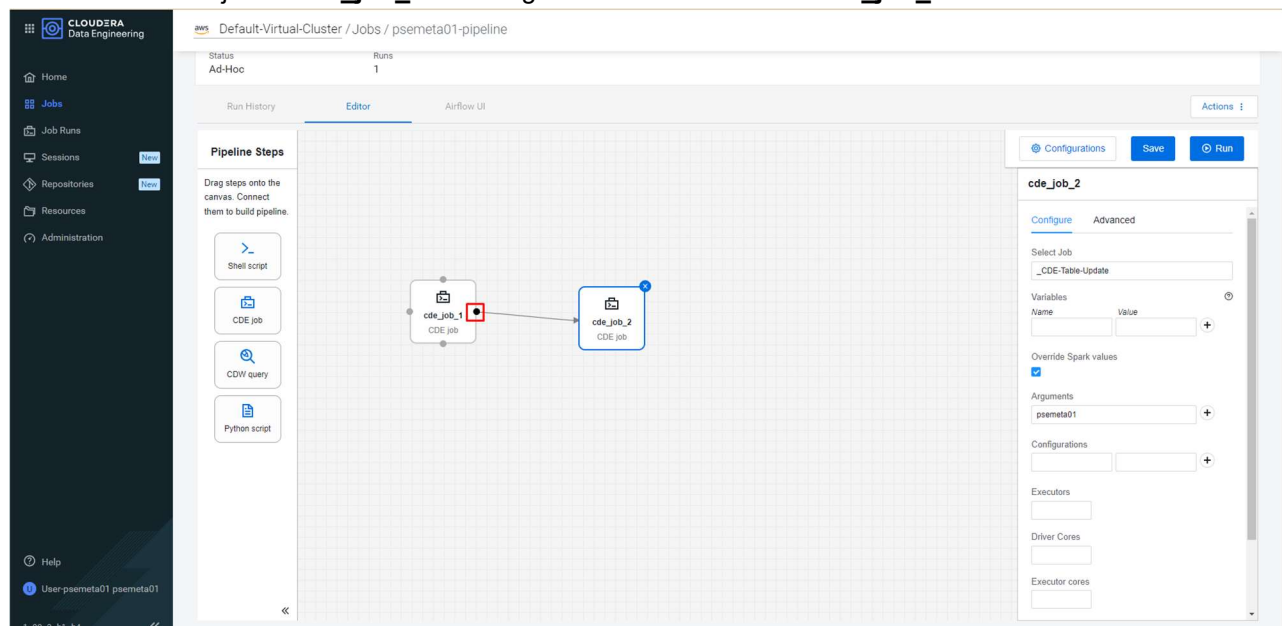
- Pipeline Steps:** Drag steps onto the canvas. Connect them to build pipeline. Available steps: Shell script, CDE job, CDW query, Python script.
- Canvas:** A workflow diagram showing a sequence of two "CDE job" steps connected by a red arrow. The first step is labeled "cde_job_1" and the second is labeled "cde_job_2".
- Right Panel:** Actions: Configurations, Save, Run. The configuration panel for "cde_job_2" is open, showing the "Configure" tab. The "Select Job" dropdown is set to "CDE-Table-Update".

Configuration Panel (cde_job_2):

- Configure** (selected) | Advanced
- Select Job: CDE-Table-Update
- Available Jobs: _CDE-Data-Enrichment, _CDE-Table-Update, dipkdash, CDE-Table-Update, psemeta01-pipeline



8. To set up the execution sequence, bind **cde_job_1** with **cde_job_2**. For that, click on the right connector of the job of **cde_job_1** and drag to the left connector of **cde_job_2**.



Once the Jobs are linked let's rename the jobs. Click on **cde_job_1** and then rename it as **Data Enrichment**.

https://console.us-west-1.cdp.cloudera.com/dev/job/details/psemeta01-pipeline/editor

aws Default-Virtual-Cluster / Jobs / psemeta01-pipeline

Status: Ad-Hoc Runs: 1

Run History Editor Airflow UI

Pipeline Steps

Drag steps onto the canvas. Connect them to build pipeline.

- Shell script
- CDE job
- CDW query
- Python script

Canvas: **cde_job_1** (CDE job) → **cde_job_2** (CDE job)

Data Enrichment

Configure Advanced

Select Job: _CDE-Data-Enrichment

Variables: Name Value

Override Spark values: ☒

Arguments: psemeta01

Configurations:

Executors:

Driver Cores:

Executor cores:

Click on **cde_job_2** and then rename it as **Table Update**.

https://console.us-west-1.cdp.cloudera.com/dev/job/details/psemeta01-pipeline/editor

aws Default-Virtual-Cluster / Jobs / psemeta01-pipeline

Status: Ad-Hoc Runs: 1

Run History Editor Airflow UI

Pipeline Steps

Drag steps onto the canvas. Connect them to build pipeline.

- Shell script
- CDE job
- CDW query
- Python script

Canvas: **Data Enrichment** (CDE job) → **cde_job_2** (CDE job)

Table Update

Configure Advanced

Select Job: _CDE-Table-Update

Variables: Name Value

Override Spark values: ☒

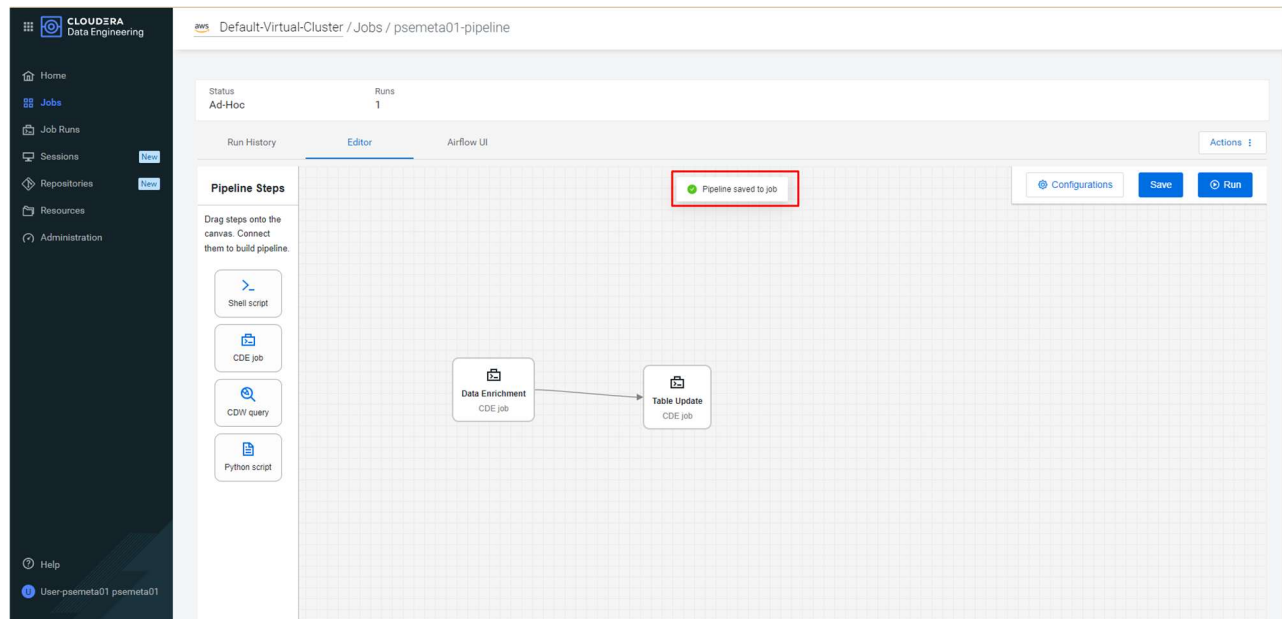
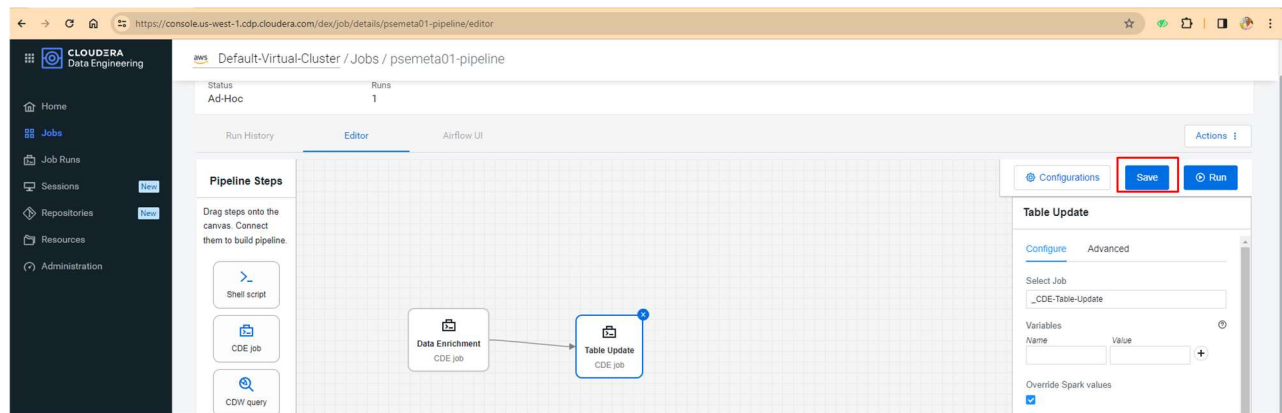
Arguments: psemeta01

Configurations:

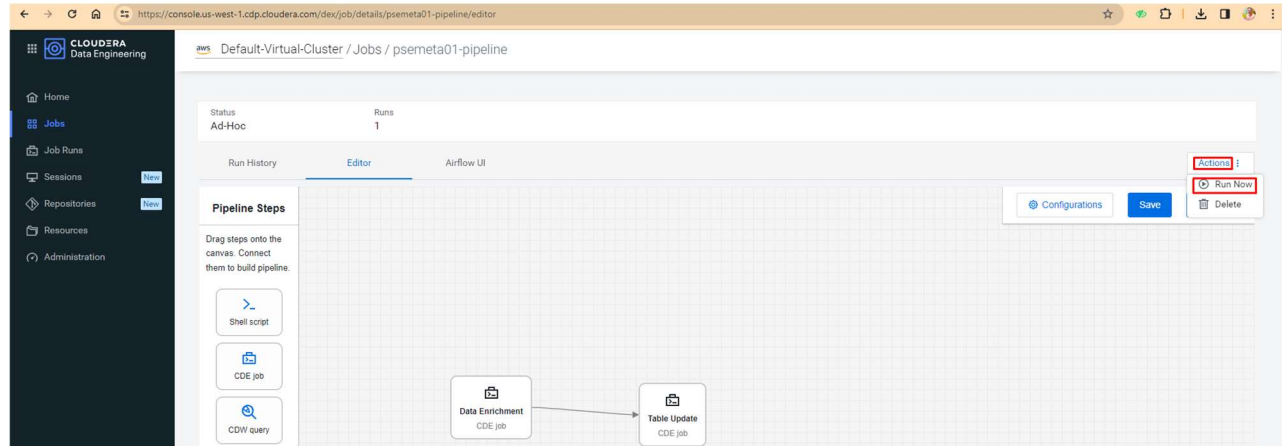
Executors:

Driver Cores:

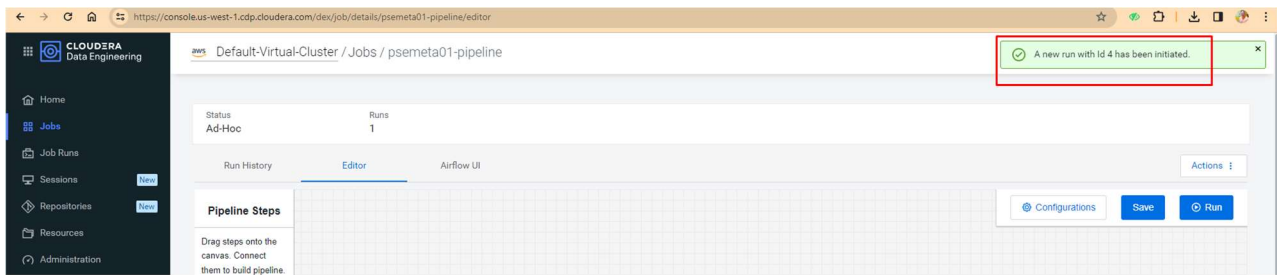
9. Once the Jobs have been joined, click on **Save** to save the settings made. You should see a message indicating **Pipeline saved to job**.



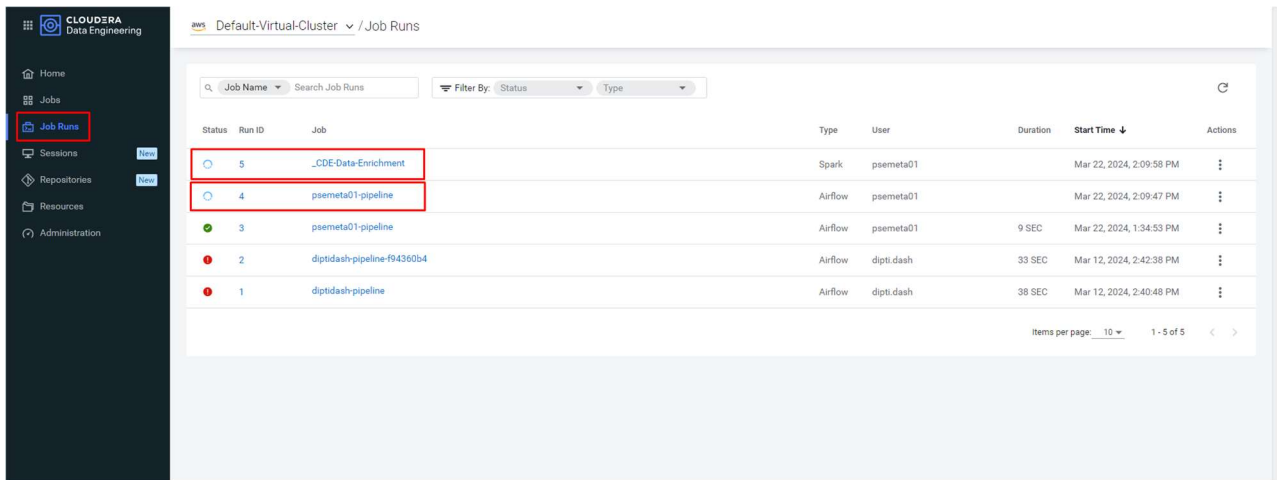
10. The time has come to run the pipeline. On the upper right side of the canvas, click **Actions**
-> **Run Now**.



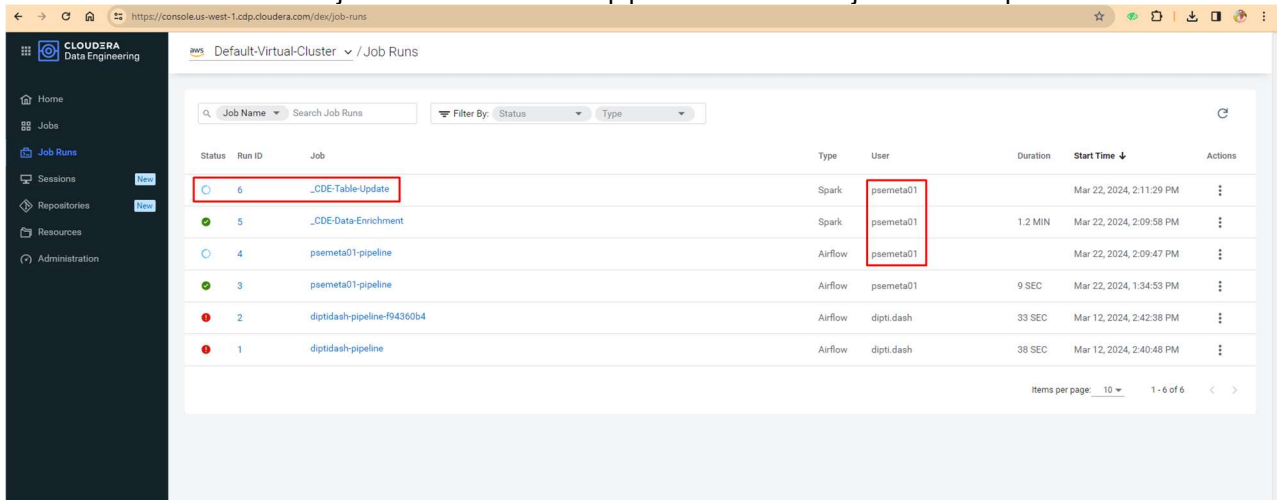
11. You should see the pipeline execution screen, indicating that the execution has been initialized.



Also, on the Job Runs tab you can see the pipeline and the very first job of the pipeline getting started.



After some time the second job starts and then the pipeline and the two jobs are completed.



Status	Run ID	Job	Type	User	Duration	Start Time	Actions
Success	6	_CDE Table-Update	Spark	psemeta01	57 SEC	Mar 22, 2024, 2:11:29 PM	
Success	5	_CDE Data-Enrichment	Spark	psemeta01	1.2 MIN	Mar 22, 2024, 2:09:58 PM	
Success	4	psemeta01-pipeline	Airflow	psemeta01	2.7 MIN	Mar 22, 2024, 2:09:47 PM	
Success	3	psemeta01-pipeline	Airflow	psemeta01	9 SEC	Mar 22, 2024, 1:34:53 PM	
Failed	2	diptidash-pipeline-f94360b4	Airflow	dipti.dash	33 SEC	Mar 12, 2024, 2:42:38 PM	
Failed	1	diptidash-pipeline	Airflow	dipti.dash	38 SEC	Mar 12, 2024, 2:40:48 PM	

12. Click on the Airflow UI tab to see the execution detail of each step in the pipeline. The configured Data Enrichment and Table Update jobs are listed at the bottom left. The colours indicate the status of each job. Make sure the radio button **Auto-refresh** is enabled to automatically display the status of jobs.

Default-Virtual-Cluster / Jobs / psemeta01-pipeline

Status: Ad-Hoc | Runs: 2

Run History | Editor | **Airflow UI** | Actions

DAG: psemeta01_pipeline

Grid | Graph | Calendar | Task Duration | Task Times | Landing Times | Gantt | Details | Code | Audit Log

22-03-2024 10:14:28 | 25 | All Run Types | All Run States | Clear Filters | Auto-refresh: ☒

Deferred | Failed | Queued | Removed | Restarting | **Running** | Scheduled | Skipped | Success | Up_for_reschedule | Up_for_retry | Upstream_failed | No status

Duration: 00:02:41

00:01:35

Data_Enrichment
Table_Update

More Details

DAG Runs Summary

Total Runs Displayed	1
Total success	1
First Run Start	2024-03-22, 10:09:48 UTC
Last Run Start	2024-03-22, 10:09:48 UTC
Max Run Duration	00:02:41
Mean Run Duration	00:02:41

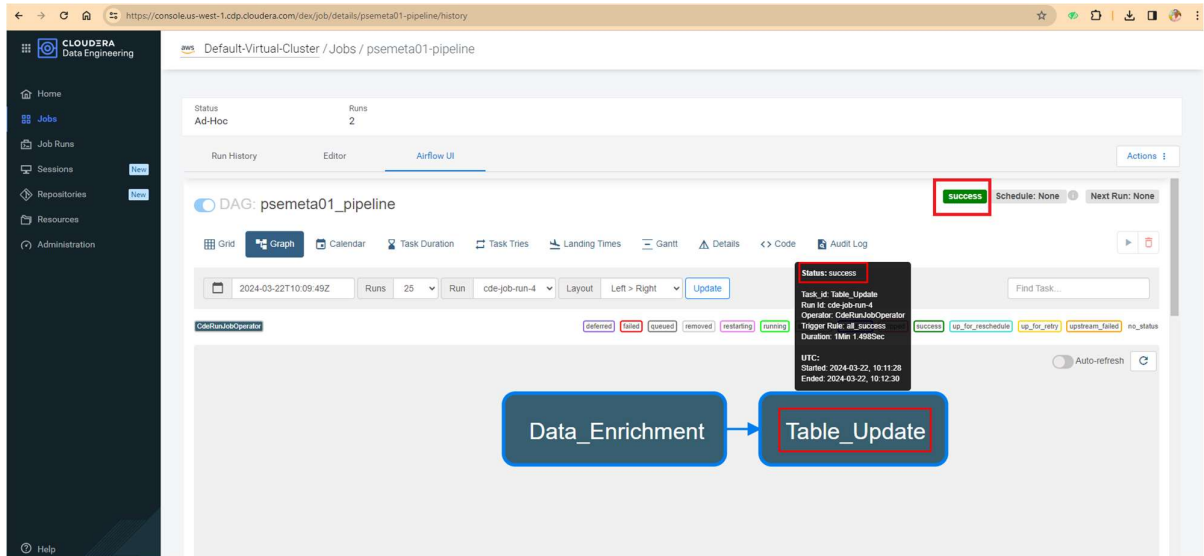
13. You can see more information about the execution by clicking on the view **Graph**. Hovering the mouse over the Job name displays specific information for each step in the pipeline. Make sure the pipeline status is Success, which indicates that the entire pipeline was able to run without issue.

The screenshot shows the Cloudera Data Engineering console interface. On the left is a sidebar with navigation links: Home, Jobs, Job Runs, Sessions, Repositories, Resources, and Administration. The main panel displays the 'psemeta01_pipeline' DAG. At the top, the status is 'Ad-Hoc' and 'Runs: 2'. Below this, there are tabs for 'Run History', 'Editor', and 'Airflow UI'. The 'Airflow UI' tab is active, showing the DAG graph. The DAG has two tasks: 'Data_Enrichment' and 'Table_Update'. A tooltip for 'Data_Enrichment' is open, showing its status as 'success' and execution details: Task Id: Data_Enrichment, Run Id: cde-job-run-4, Operator: CodeRunJobOperator, Trigger Rule: all_success, Duration: 1Min 21.965Sec, UTC: Started: 2024-03-22, 10:09:57, Ended: 2024-03-22, 10:11:19. The pipeline status is indicated by a green 'success' label at the top right.

This screenshot is similar to the one above, showing the same DAG. However, the tooltip is now for the 'Table_Update' task. The tooltip shows its status as 'success' and execution details: Task Id: Table_Update, Run Id: cde-job-run-4, Operator: CodeRunJobOperator, Trigger Rule: all_success, Duration: 1Min 1.498Sec, UTC: Started: 2024-03-22, 10:11:28, Ended: 2024-03-22, 10:12:30. The pipeline status remains 'success'.

The execution status appears next to the name of the pipeline (marked in red). If it is green and indicates **Success**, it means that the execution was successful.

CloudERA Data Engineering console showing the DAG for the `psemeta01_pipeline`. The pipeline is in a **success** state. The DAG consists of two tasks: `Data_Enrichment` and `Table_Update`, connected by a dependency arrow. The `Table_Update` task is highlighted with a red box. The console also shows the task's status as **success** and the UTC time of completion.



14. The data enrichment and table update flows essentially updates the underlying table with more descriptive columns. You may check the same by going to CDW instance.

Impala console screenshot showing the execution of a query to select data from the `master_data.misc` table. The query is highlighted with a red box. The results show 6 rows of data, including columns `id` and `description`.

id	description
1	Y
2	N
3	F
4	M
5	1
6	0

Impala console screenshot showing the execution of a query to select data from the `master_data.contract` table. The query is highlighted with a red box. The results show 3 rows of data, including columns `id` and `description`.

id	description
1	1
2	2
3	3

Search data and saved documents...

Impala

Add a name... Add a description...

0.73s psemeta99

Tables

Documents

telco_data_curated

telco_iceberg_kafka

select * from psemeta99.telco_data_curated;

Query 5b4965f86e71a621:e8053df400000000 100% Complete (3 out of 3)

Query 5b4965f86e71a621:e8053df400000000 100% Complete (3 out of 3)

Query History

Saved Queries

Results (100+)

	multiplexlines	paperlessbilling	gender	onlinesecurity	internetservice	techsupport	contract	churn	seniorcitizen	deviceprotection
1	No phone service	Yes	Female	No	DSL	No	Month-to-month	No	0	No
2	No	No	Male	Yes	DSL	No	One year	No	0	Yes
3	No phone service	No	Male	Yes	DSL	Yes	One year	No	0	Yes
4	Yes	Yes	Male	No	Fiber optic	No	Month-to-month	No	0	No
5	No phone service	No	Female	Yes	DSL	No	Month-to-month	No	0	No
6	No	No	Male	Yes	DSL	No	One year	No	0	No
7	No	Yes	Male	Yes	DSL	No	Month-to-month	No	0	No
8	No	No	Male	No internet service	No	No internet service	Two year	No	0	No internet service
9	Yes	No	Male	No	Fiber optic	No	One year	No	0	Yes
10	No	Yes	Male	Yes	Fiber optic	Yes	Month-to-month	No	0	Yes
11	Yes	No	Female	Yes	Fiber optic	Yes	Two year	No	0	Yes
12	No	No	Female	No internet service	No	No internet service	One year	No	0	No internet service
13	Yes	No	Male	Yes	Fiber optic	No	Two year	No	0	Yes