



``git push`` your data stack with  
Airbyte, Airflow, and dbt

Airflow Summit 2022



# Hello!



## Evan Tahler

Staff Engineer

 @evantahler

 evantahler

 /in/evantahler



## Marcos Marx

User Success Engineer

 marcosmarxm

 /in/marcosmarxmllnitz



## Airbyte

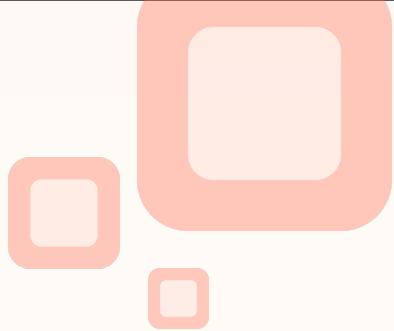
Open-Source Data Integration

 20,000 users

 7,500 Slack members

 6,500 GitHub stars

# Sample Code!



[www.github.com/airbytehq/airflow-summit-airbyte-2022](https://www.github.com/airbytehq/airflow-summit-airbyte-2022)

*The goal of this talk is to share an example of how we can use today's tools to automate your data stack and gain the benefits of Continuous Integration and Continuous Deployment (CI/CD)*

# What is Airbyte?



# E\_xtract

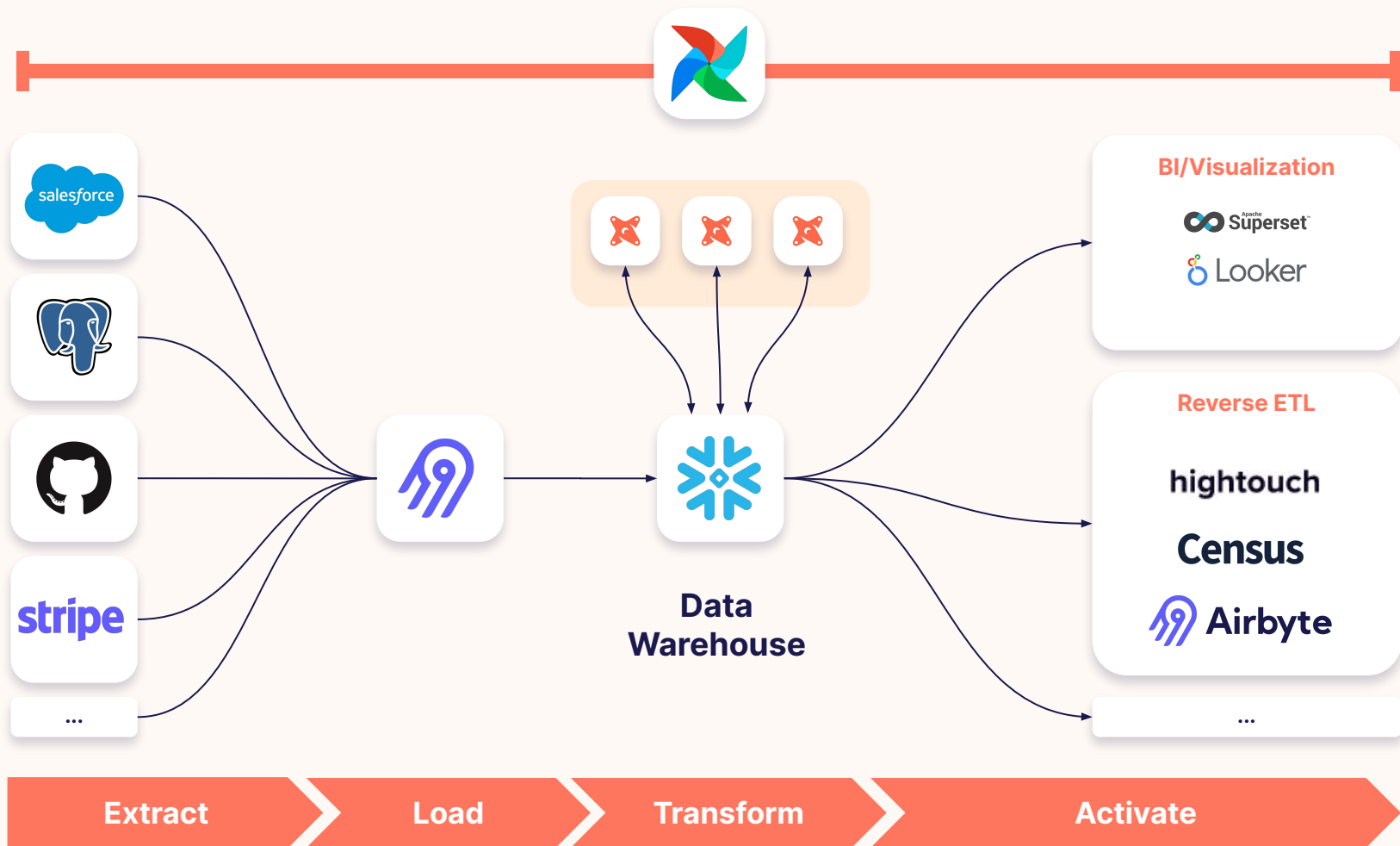
**General-purpose** routines to pull selected data from a source.

# L\_oad

**General-purpose** routines to push raw data where it is going to be consumed.

# T\_ransform

Business logic specific to your organization to serve an analytics or operational use case with SQL / dbt / ...



# The Modern Data Stack



# ~~The Modern Data Stack~~

## The Modern Data Engineering Workflow





# Data Teams == Engineering Teams



Modern Data Teams are Engineering Teams. We make data products consumed by others. To be successful, reliable, and fast, modern engineering workflows are needed.

- *Version Control*
- *Tests*
- *Continuous Deployment*

# Data Teams == Engineering Teams



Modern Data Teams are Engineering Teams. We make data products consumed by others. To be successful, reliable, and fast, modern engineering workflows are needed.

- **Version Control**

- *Track and test changes before they are deployed*
- *Rollback when problems occur*
- *Manage approvals & reviews*

- **Tests**

- *... with the confidence that nothing broke*


- **Continuous Deployment**

- *... deployed quickly and automatically, perhaps to multiple environments*

# Version Controlled Airbyte: The Octavia CLI



This is Octavia ->



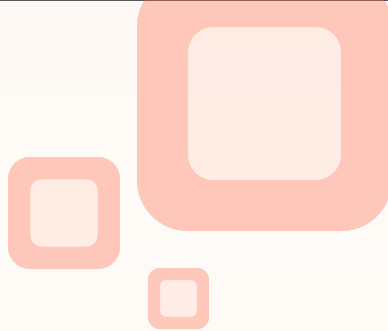
```
$ octavia --help
Usage: octavia [OPTIONS] COMMAND [ARGS]...

Options:
  --airbyte-url TEXT  The URL of your Airbyte instance.
  --workspace-id TEXT The id of the workspace on which you want octavia-cli
                      to work. Defaults to the first one found on your
                      Airbyte instance.
  --help              Show this message and exit.

Commands:
  apply  Create or update Airbyte remote resources according local...
  delete [NOT IMPLEMENTED] Delete resources
  generate Generate a YAML template for a source, destination or a...
  import  [NOT IMPLEMENTED] Import an existing resources from the...
  init    Initialize required directories for the project.
  list    List existing Airbyte resources.
```

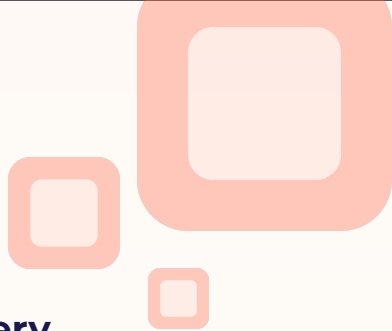
# `octavia`

*Released in April, 2022!*



- **generate YAML files that can be version controlled in Git**
- **leverage any scripting or templating tool to generate configurations dynamically for creation of data sources, destinations, and connections**
- **deploy configurations on multiple Airbyte instances, in order to manage several Airbyte environments**
- **integrate into a CI workflow to automate the deployment of data integration configurations**
  - This allows for the same configuration to be run against multiple environments - running the same ELT process against staging or test data by changing environment variables
  - Secrets can be in the system/shell Environment, or a local dotfile

# Coming Soon for `octavia`



- **Deterministic Source and Destination IDs for idempotent recovery**
- **Deterministic Connection IDs to orchestrate with other applications**

For now, we have some custom code to make this work - `change_resource_id.py`

# Demo Time!



[www.github.com/airbytehq/airflow-summit-airbyte-2022](https://www.github.com/airbytehq/airflow-summit-airbyte-2022)

# Our Demo Stack 100% Open Source, of course



**Airflow**

Orchestrator



**Airbyte**

Extract & Load

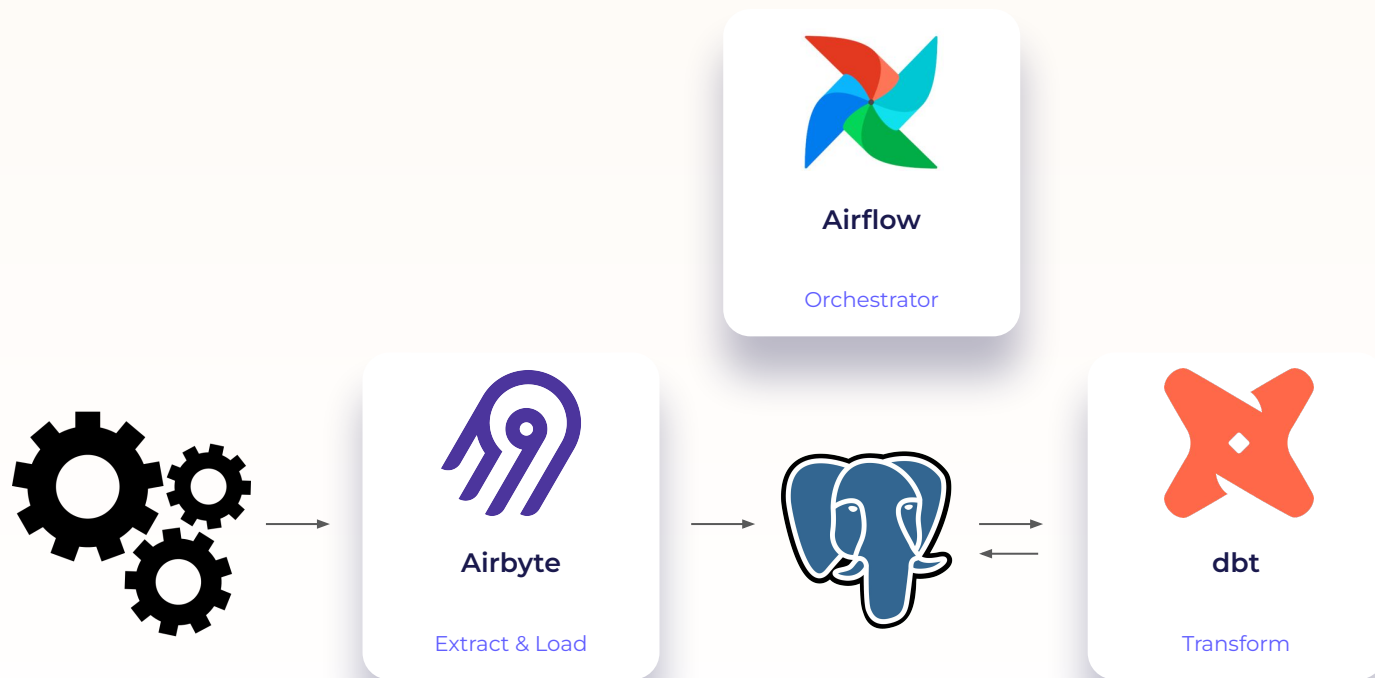


**dbt**

Transform



# Our Demo Stack 100% Open Source, of course



# DEMO PART 1

*In which everything works great*

# Demo: Docker Compose

- For easy developer setup
- For easy CI setup

**Just run `./tools/start.sh` and everything will be configured and the Airflow DAG will run**

 .github/workflows

 airbyte

 dags

 dbt

 docker

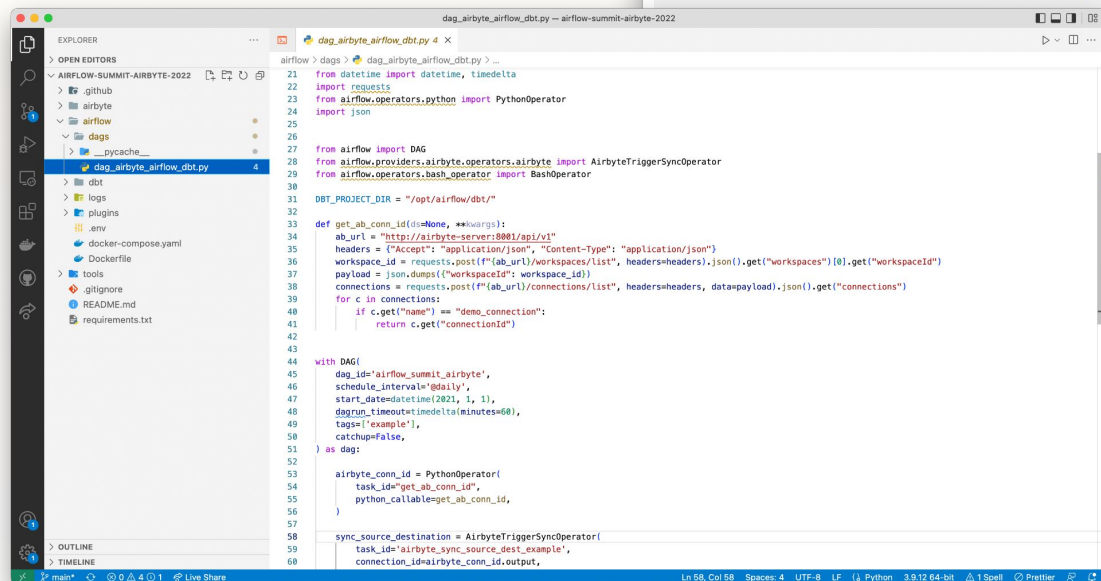
 .gitignore

 Makefile

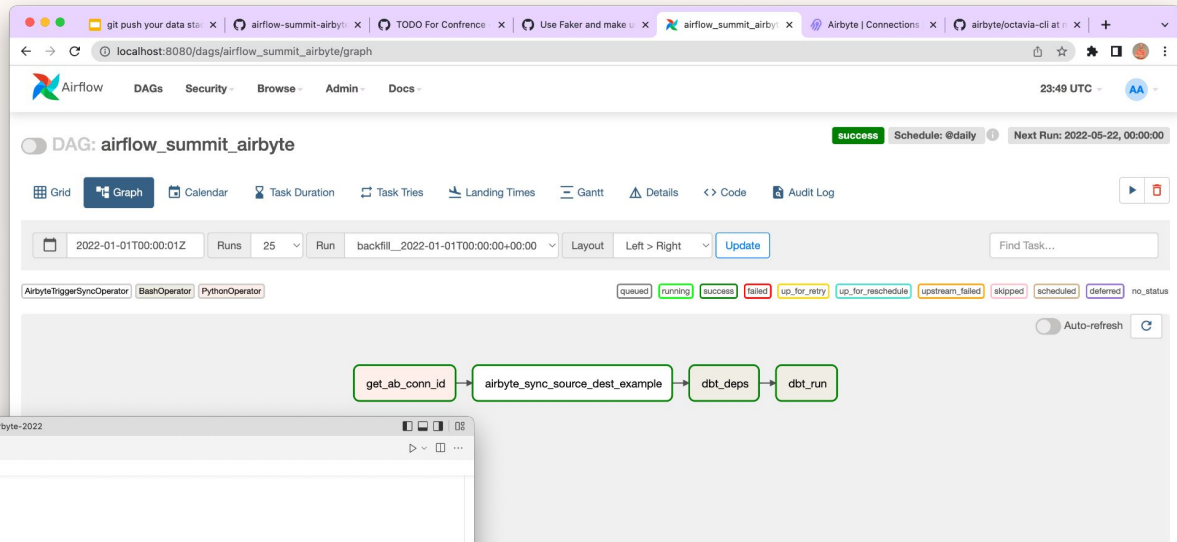
 README.md

# Demo:

## Orchestrate w/ Airflow



```
21 from datetime import datetime, timedelta
22 import requests
23 from airflow.operators.python import PythonOperator
24 import json
25
26
27
28 from airflow import DAG
29 from airflow.providers.airbyte.operators.airbyte import AirbyteTriggerSyncOperator
30 from airflow.providers.bash.operators.bash import BashOperator
31
32 DBT_PROJECT_DIR = "/opt/airflow/dbt/"
33
34 def get_ab_conn_id(ds=None, **kwargs):
35     ab_url = "http://airbyte-server:8080/api/v1/"
36     headers = {"Accept": "application/json", "Content-Type": "application/json"}
37     workspace_id = requests.post(f"{ab_url}/workspaces/list", headers=headers).json()[0].get("workspaceId")
38     payload = json.dumps({"workspaceId": workspace_id})
39     connections = requests.post(f"{ab_url}/connections/list", headers=headers, data=payload).json().get("connections")
40     for c in connections:
41         if c.get("name") == "demo_connection":
42             return c.get("connectionId")
43
44 with DAG(
45     dag_id='airflow_summit_airbyte',
46     schedule_interval='@daily',
47     start_date=datetime(2021, 1, 1),
48     dagrun_timeout=timedelta(minutes=60),
49     tags=['example'],
50     catchup=False,
51 ) as dag:
52
53     airbyte_conn_id = PythonOperator(
54         task_id='get_ab_conn_id',
55         python_callable=get_ab_conn_id,
56     )
57
58     sync_source_destination = AirbyteTriggerSyncOperator(
59         task_id='airbyte_sync_source_dest_example',
60         connection_id=airbyte_conn_id.output,
```



# Demo:

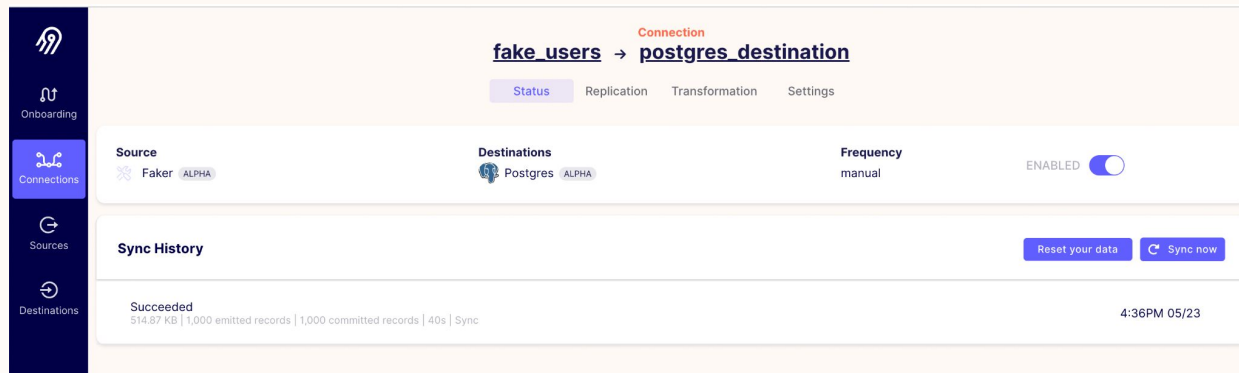
## Extract & Load w/ Airbyte

```
configuration.yaml .../demo_connection X configuration.yaml
1 configuration:
2   namespace_definition: destination
3   namespace_format: ${SOURCE_NAMESPACE}
4   prefix: ''
5   resource_requirements:
6     cpu_limit: ''
7     cpu_request: ''
8     memory_limit: ''
9     memory_request: ''
10  status: active
11  sync_catalog:
12  streams:
13    - config:
14        alias_name: Users
15        cursor_field: []
16        destination_sync_mode: append
17        primary_key: []
18        selected: true
19        sync_mode: full_refresh
20      stream:
21        default_cursor_field: []
22      json_schema:
23        $schema: http://json-schema.org/draft-07/schema#
24      properties:
25        address:
26          type: string
27        birthdate:
28          format: date
29          type: string
30        blood_group:
31          type: string
32        company:
33          type: string
34        created_at:
35          airbyte_type: timestamp_without_timezone
36          format: date-time
37          type: string
38        current_location:
39          type: array
40        id:
41          type: number
42        job:
43          type: string
44        mail:
45          type: string
46        name:
47          type: string
48        residence:
49          type: string
50        sex:
51          type: string
52        ssn:
53          type: string
54        updated_at:
55          airbyte_type: timestamp_without_timezone
56          format: date-time
57          type: string
58        username:
59          type: string
60        website:
61          type: array
62        type: object
63      name: Users
64      namespace: null

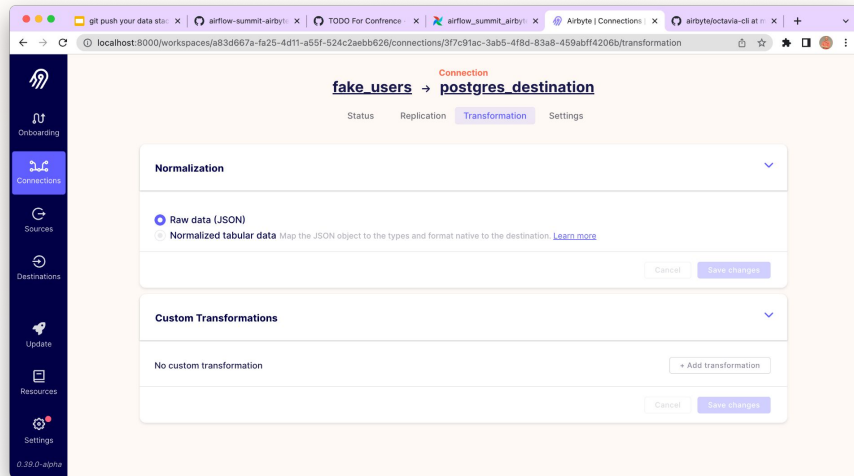
configuration.yaml X configuration.yaml
1 # Configuration for airbyte/destination-postgres
2 # Documentation about this connector can be found at https://docs.airbyte.io/integrations/destinations/postgres
3 resource_name: postgres_destination
4 definition_type: destination
5 definition_id: 25c5221d-dce2-4163-ade9-739ef79ef503
6 definition_image: airbyte/destination-postgres
7 definition_version: 0.3.19
8
9 # EDIT THE CONFIGURATION BELOW!
10 configuration:
11   ssl: false # OPTIONAL | boolean | Encrypt data using SSL.
12   host: ${POSTGRES_HOST} # REQUIRED | string | Hostname of the database.
13   port: 5432 # REQUIRED | integer | Port of the database. | Example: 5432
14   schema: "public" # REQUIRED | string | The default schema tables are written to.
15   database: postgres # REQUIRED | string | Name of the database.
16   password: ${POSTGRES_PASSWORD} # SECRET (please store in environment variables) |
17   username: ${POSTGRES_USERNAME} # REQUIRED | string | Username to use to access the
18   tunnel_method:
19     # ----- Pick one valid structure among the examples below: -----
20     tunnel_method: "NO_TUNNEL" # REQUIRED | string | No ssh tunnel needed to connect
21     # ----- Another valid structure for tunnel_method: -----
22     # ssh_key: ${SSH_KEY} # SECRET (please store in environment variables) | REQUIRED
23     # tunnel_host: # REQUIRED | string | Hostname of the jump server host that allow
24     # tunnel_port: 22 # REQUIRED | integer | Port on the proxy/jump server that allow
25     # tunnel_user: # REQUIRED | string | OS-level username for logging into the jump
26     # tunnel_method: "SSH_KEY_AUTH" # REQUIRED | string | Connect through a jump server
27     # ----- Another valid structure for tunnel_method: -----
28     # tunnel_host: # REQUIRED | string | Hostname of the jump server host that allow
29     # tunnel_port: 22 # REQUIRED | integer | Port on the proxy/jump server that allow
30     # tunnel_user: # REQUIRED | string | OS-level username for logging into the jump
31     # tunnel_method: "SSH_PASSWORD_AUTH" # REQUIRED | string | Connect through a jump
32     # tunnel_user_password: ${TUNNEL_USER_PASSWORD} # SECRET (please store in environment variables)
```

# Demo:

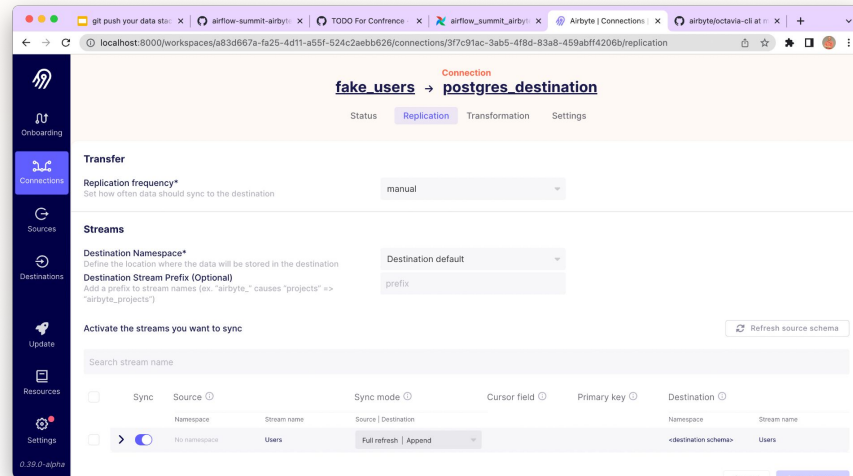
## Extract & Load w/ Airbyte



The screenshot shows the Airbyte Connections overview page. The left sidebar contains navigation links: Onboarding, Connections (active), Sources, and Destinations. The main content area is titled "fake\_users → postgres\_destination" with a "Connection" label. Below the title are tabs for Status, Replication, Transformation, and Settings. The "Status" tab is selected, showing a "Source" of "Faker" (ALPHA) and "Destinations" of "Postgres" (ALPHA). The "Frequency" is set to "manual" and is "ENABLED". A "Sync History" section shows a successful sync: "Succeeded 514.97 kB | 1,000 emitted records | 1,000 committed records | 40s | Sync" at "4:36PM 05/23". Buttons for "Reset your data" and "Sync now" are present.



The screenshot shows the "Transformation" settings page for the "fake\_users → postgres\_destination" connection. The left sidebar is the same as the overview page. The main content area has tabs for Status, Replication, Transformation (active), and Settings. The "Transformation" tab is selected, showing "Normalization" and "Custom Transformations" sections. Under "Normalization", "Raw data (JSON)" is selected, and "Normalized tabular data" is an option with a link to "Learn more". Under "Custom Transformations", there is a button to "Add transformation". Buttons for "Cancel" and "Save changes" are at the bottom of each section.



The screenshot shows the "Replication" settings page for the "fake\_users → postgres\_destination" connection. The left sidebar is the same as the overview page. The main content area has tabs for Status, Replication (active), Transformation, and Settings. The "Replication" tab is selected, showing "Transfer" and "Streams" sections. Under "Transfer", "Replication frequency\*" is set to "manual". Under "Streams", "Destination Namespace\*" is set to "Destination default" and "Destination Stream Prefix (Optional)" is set to "prefix". A section "Activate the streams you want to sync" has a "Refresh source schema" button and a search bar. Below is a table with columns: Sync, Source, Stream name, Sync mode, Cursor field, Primary key, Destination, and Stream name. The table shows one stream: "Users" with "Full refresh | Append" sync mode.

# Demo: Extract & Load w/ Airbyte

Don't worry, it's fake data from the `faker` package

The screenshot shows a database management tool interface. At the top, it indicates 'Connected to database "postgres"' and 'PostgreSQL 10.21 (Debian 10.21-1.pgdg90+1)'. The main area displays a table with columns: `_airbyte_ab_id`, `_airbyte_data`, and `_airbyte_emitted_at`. The table contains 20 rows of data, each representing a fake user profile. The `_airbyte_data` column contains JSON objects with fields like `id`, `job`, `sex`, `ssn`, `mail`, `name`, `address`, `company`, `website`, `username`, `birthdate`, `residence`, `created_at`, `updated_at`, `blood_group`, and `current_location`. A right-hand pane shows the selected JSON object for the first row, with a blue box highlighting the `current_location` field. The bottom of the interface shows a search bar, a '1 of 1000 selected' status, and a 'Page 1 of 1' indicator.

_airbyte_ab_id	_airbyte_data	_airbyte_emitted_at
0009f7df-c57d-4bb9-9540-52e73d7ee879	{ "id": 109, "job": "Fisheries officer", "sex": "F", "ssn": "714-13-0935", "mail": ... }	2022-05-23 23:36:34+00
0048d4c7-c918-46ce-8bc7-142b1aa84a3b	{ "id": 37, "job": "Therapist, nutritional", "sex": "F", "ssn": "432-97-0391", "mail": ... }	2022-05-23 23:36:33+00
005bb4a4-24b7-4acc-a598-c7b8d81b7248	{ "id": 901, "job": "Scientific laboratory technician", "sex": "M", "ssn": "710-50-... }	2022-05-23 23:36:46+00
0105b822-0cc7-4ae4-83f7-d708562ae9e9	{ "id": 356, "job": "Tourism officer", "sex": "F", "ssn": "707-51-1788", "mail": "iyou... }	2022-05-23 23:36:38+00
015aefbe-ea31-43a7-a9b7-d9b33d7cd03f	{ "id": 254, "job": "Programmer, systems", "sex": "M", "ssn": "246-06-5239", "mail": ... }	2022-05-23 23:36:37+00
018f7c36-13e1-4bec-91bf-e7bed777579d	{ "id": 292, "job": "Ranger/warden", "sex": "F", "ssn": "337-75-7152", "mail": "tho... }	2022-05-23 23:36:37+00
0234cda4-09ee-4508-a1cb-74b4e12ddd81	{ "id": 554, "job": "Forensic scientist", "sex": "M", "ssn": "877-89-1261", "mail": ... }	2022-05-23 23:36:41+00
030147d4-27ba-4b4b-8f54-da423cf11b70	{ "id": 303, "job": "Investment banker, operational", "sex": "M", "ssn": "751-31-... }	2022-05-23 23:36:37+00
03225369-c04b-4446-8c1b-23a4e2950572	{ "id": 60, "job": "Neurosurgeon", "sex": "F", "ssn": "584-03-6287", "mail": "ofos... }	2022-05-23 23:36:33+00
03f4d948-6997-47c1-9658-c428b8f5f687	{ "id": 108, "job": "Civil engineer, contracting", "sex": "F", "ssn": "827-64-... }	2022-05-23 23:36:34+00
03fd641f-da09-49aa-aa85-b4df3f3c8740	{ "id": 593, "job": "International aid/development worker", "sex": "M", "ssn": "... }	2022-05-23 23:36:41+00
043c03d7-7d8f-4ff3-bf67-a365ffc826f3	{ "id": 183, "job": "Leisure centre manager", "sex": "F", "ssn": "092-91-57... }	2022-05-23 23:36:35+00
043f28d3-0656-4288-b370-cd5096950358	{ "id": 10, "job": "Economist", "sex": "F", "ssn": "654-67-0997", "mail": "jill46@g... }	2022-05-23 23:36:32+00
044bb6d5-f5d6-446f-9e44-b1d8a9464ef3	{ "id": 726, "job": "English as a second language teacher", "sex": "M", "ssn": "6... }	2022-05-23 23:36:43+00
048eec9b-4919-4d6b-94e2-5bf1e9d8aa27	{ "id": 15, "job": "Telecommunications researcher", "sex": "M", "ssn": "542-89-... }	2022-05-23 23:36:33+00
04a60152-abfe-4ca6-a854-e6483e3cfe53	{ "id": 742, "job": "Air traffic controller", "sex": "M", "ssn": "475-23-5667", "mail": ... }	2022-05-23 23:36:44+00
04b48b64-ca22-42c5-a340-8f6f36cb92a9	{ "id": 799, "job": "Nurse, mental health", "sex": "M", "ssn": "118-07-0877", "mail": ... }	2022-05-23 23:36:44+00
04b5ab9b-286f-44ff-800a-161bb2efaf15	{ "id": 517, "job": "Surveyor, mining", "sex": "F", "ssn": "709-44-4367", "mail": ... }	2022-05-23 23:36:40+00



# Demo: Transform w/ dbt

The screenshot displays a code editor with two open SQL files, `data_stream_ab1.sql` and `data_stream_ab2.sql`, within a project named `AIRFLOW-SUMMIT-AIRBYTE-2022`.

**Left Pane (Explorer):** Shows the project structure. Key folders include `connections`, `destinations`, `sources`, `temporal`, `airflow`, `macros`, and `models`. The `models` folder is expanded, showing `airbyte_ctes/public` with `data_stream_ab1.sql` and `data_stream_ab2.sql`.

**Right Pane (SQL Code):** Shows the content of the two files.

**data\_stream\_ab1.sql:**

```
1 {{ config(
2     indexes = [{ 'columns': ['_airbyte_emitted_at'], 'type': 'btree' }],
3     unique_key = '_airbyte_ab_id',
4     schema = "_airbyte_public",
5     tags = [ "top-level-intermediate" ]
6 ) }}
7 -- SQL model to parse JSON blob stored in a single column and extract into separate
8 -- depends_on: {{ source('public', '_airbyte_raw_users') }}
9 select
10     {{ json_extract_scalar('_airbyte_data', ['id'], ['id']) }} as user_id,
11     {{ json_extract_scalar('_airbyte_data', ['name'], ['name']) }} as name,
12     {{ json_extract_scalar('_airbyte_data', ['mail'], ['mail']) }} as email,
13     {{ json_extract_scalar('_airbyte_data', ['username'], ['username']) }} as username,
14     {{ json_extract_scalar('_airbyte_data', ['created_at'], ['created_at']) }} as created_at,
15     {{ json_extract_scalar('_airbyte_data', ['updated_at'], ['updated_at']) }} as updated_at,
16     _airbyte_ab_id,
17     _airbyte_emitted_at,
18     {{ current_timestamp() }} as _airbyte_normalized_at
19 from {{ source('public', '_airbyte_raw_users') }} as table_alias
20 -- data_stream
21 where 1 = 1
22 {{ incremental_clause('_airbyte_emitted_at') }}
23
24
```

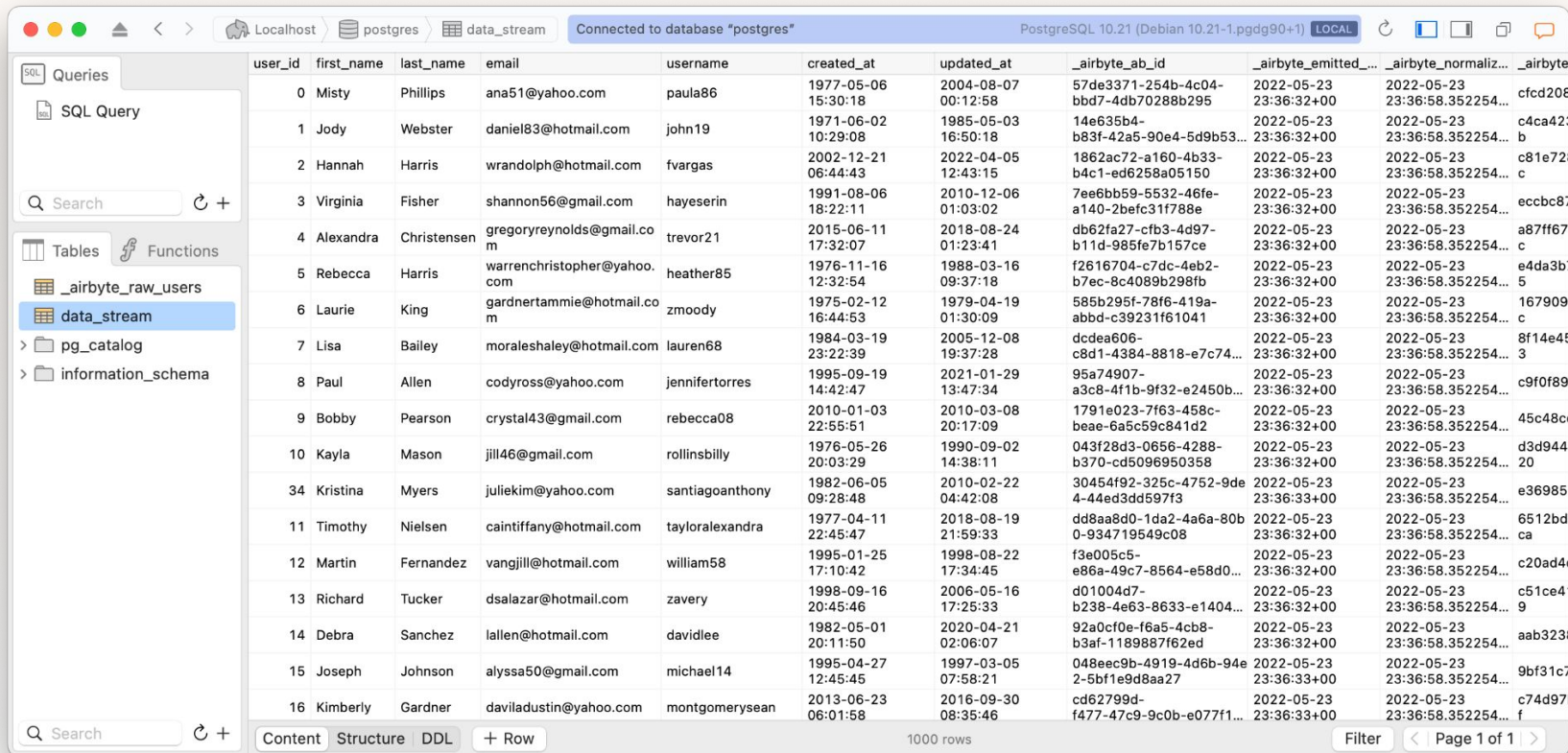
**data\_stream\_ab2.sql:**

```
1 {{ config(
2     indexes = [{ 'columns': ['_airbyte_emitted_at'], 'type': 'btree' }],
3     unique_key = '_airbyte_ab_id',
4     schema = "_airbyte_public",
5     tags = [ "top-level-intermediate" ]
6 ) }}
7 -- SQL model to cast each column to its adequate SQL type converted from the JSON sc
8 -- depends_on: {{ ref('data_stream_ab1') }}
9 select
10     cast(user_id as {{ dbt_utils.type_bigint() }}) as user_id,
11     split_part(cast(name as {{ dbt_utils.type_string() }}), ' ', 1) as first_name,
12     split_part(cast(name as {{ dbt_utils.type_string() }}), ' ', 2) as last_name,
13     cast(email as {{ dbt_utils.type_string() }}) as email,
14     cast(username as {{ dbt_utils.type_string() }}) as username,
15     cast(created_at as {{ dbt_utils.type_timestamp() }}) as created_at,
16     cast(updated_at as {{ dbt_utils.type_timestamp() }}) as updated_at,
17     _airbyte_ab_id,
18     _airbyte_emitted_at,
19     {{ current_timestamp() }} as _airbyte_normalized_at
20 from {{ ref('data_stream_ab1') }}
21 -- data_stream
22 where 1 = 1
23 {{ incremental_clause('_airbyte_emitted_at') }}
24
25
```



Don't worry, it's fake data  
from the `faker` package

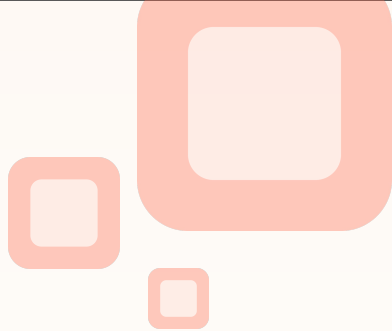
# Demo: Transform w/ dbt



The screenshot shows a database interface with a table named `data_stream` in the `pg_catalog` schema. The table contains 16 rows of user data. The interface includes a sidebar with navigation options like 'Queries', 'Tables', and 'Functions'. The main area displays the table data with columns for user information and various timestamps.

user_id	first_name	last_name	email	username	created_at	updated_at	_airbyte_ab_id	_airbyte_emitted_at	_airbyte_normaliz...	_airbyte
0	Misty	Phillips	ana51@yahoo.com	paula86	1977-05-06 15:30:18	2004-08-07 00:12:58	57de3371-254b-4c04-bbd7-4db70288b295	2022-05-23 23:36:32+00	2022-05-23 23:36:58.352254...	cfd208
1	Jody	Webster	daniel83@hotmail.com	john19	1971-06-02 10:29:08	1985-05-03 16:50:18	14e635b4-b83f-42a5-90e4-5d9b53...	2022-05-23 23:36:32+00	2022-05-23 23:36:58.352254...	b
2	Hannah	Harris	wrandolph@hotmail.com	fvargas	2002-12-21 06:44:43	2022-04-05 12:43:15	1862ac72-a160-4b33-b4c1-ed6258a05150	2022-05-23 23:36:32+00	2022-05-23 23:36:58.352254...	c
3	Virginia	Fisher	shannon56@gmail.com	hayeserin	1991-08-06 18:22:11	2010-12-06 01:03:02	7ee6bb59-5532-46fe-a140-2befc31f788e	2022-05-23 23:36:32+00	2022-05-23 23:36:58.352254...	eccbc87
4	Alexandra	Christensen	gregoryreynolds@gmail.co	trevor21	2015-06-11 17:32:07	2018-08-24 01:23:41	db62fa27-cfb3-4d97-b11d-985fe7b157ce	2022-05-23 23:36:32+00	2022-05-23 23:36:58.352254...	a87ff67
5	Rebecca	Harris	warrenchristopher@yahoo.com	heather85	1976-11-16 12:32:54	1988-03-16 09:37:18	f2616704-c7dc-4eb2-b7ec-8c4089b298fb	2022-05-23 23:36:32+00	2022-05-23 23:36:58.352254...	5
6	Laurie	King	gardnertammie@hotmail.co	zmooddy	1975-02-12 16:44:53	1979-04-19 01:30:09	585b295f-78f6-419a-abbd-c39231f61041	2022-05-23 23:36:32+00	2022-05-23 23:36:58.352254...	167909
7	Lisa	Bailey	moraleshaley@hotmail.com	lauren68	1984-03-19 23:22:39	2005-12-08 19:37:28	dcdea606-c8d1-4384-8818-e7c74...	2022-05-23 23:36:32+00	2022-05-23 23:36:58.352254...	3
8	Paul	Allen	codyross@yahoo.com	jennifertorres	1995-09-19 14:42:47	2021-01-29 13:47:34	95a74907-a3c8-4f1b-9f32-e2450b...	2022-05-23 23:36:32+00	2022-05-23 23:36:58.352254...	c9f0f89
9	Bobby	Pearson	crystal43@gmail.com	rebecca08	2010-01-03 22:55:51	2010-03-08 20:17:09	1791e023-7f63-458c-beae-6a5c59c841d2	2022-05-23 23:36:32+00	2022-05-23 23:36:58.352254...	45c48cc
10	Kayla	Mason	jill46@gmail.com	rollinsbilly	1976-05-26 20:03:29	1990-09-02 14:38:11	043f28d3-0656-4288-b370-cd5096950358	2022-05-23 23:36:32+00	2022-05-23 23:36:58.352254...	20
34	Kristina	Myers	juliekim@yahoo.com	santiagoanthony	1982-06-05 09:28:48	2010-02-22 04:42:08	30454f92-325c-4752-9de4-44ed3dd597f3	2022-05-23 23:36:32+00	2022-05-23 23:36:58.352254...	e36985
11	Timothy	Nielsen	caintiffany@hotmail.com	tayloralexandra	1977-04-11 22:45:47	2018-08-19 21:59:33	dd8aa8d0-1da2-4a6a-80b0-934719549c08	2022-05-23 23:36:32+00	2022-05-23 23:36:58.352254...	ca
12	Martin	Fernandez	vangjill@hotmail.com	william58	1995-01-25 17:10:42	1998-08-22 17:34:45	f3e005c5-e86a-49c7-8564-e58d0...	2022-05-23 23:36:32+00	2022-05-23 23:36:58.352254...	c20ad4c
13	Richard	Tucker	dsalazar@hotmail.com	zavery	1998-09-16 20:45:46	2006-05-16 17:25:33	d01004d7-b238-4e63-8633-e1404...	2022-05-23 23:36:32+00	2022-05-23 23:36:58.352254...	9
14	Debra	Sanchez	lallen@hotmail.com	davidlee	1982-05-01 20:11:50	2020-04-21 02:06:07	92a0cf0e-f6a5-4cb8-b3af-1189887f62ed	2022-05-23 23:36:32+00	2022-05-23 23:36:58.352254...	aab323c
15	Joseph	Johnson	alyssa50@gmail.com	michael14	1995-04-27 12:45:45	1997-03-05 07:58:21	048eec9b-4919-4d6b-94e2-5bf1e9d8aa27	2022-05-23 23:36:33+00	2022-05-23 23:36:58.352254...	9bf31c7
16	Kimberly	Gardner	davidalustin@yahoo.com	montgomerysean	2013-06-23 06:01:58	2016-09-30 08:35:46	cd62799d-f477-47c9-9c0b-e077f1...	2022-05-23 23:36:33+00	2022-05-23 23:36:58.352254...	f

# Demo Notes:



## Future Work:

1. Custom code to change Airbyte Connection ID: `change_resource_id.py`
2. Custom docker image for Airflow w/ dbt included
3. Using a hosted Postgres database in CI so we can inspect the data

**Pull Requests Welcome!**

# DEMO PART 2

*In which Evan breaks the build*



# CI/CD Ethos (since 2012)

- *All Code in git*
- *Git is used to deploy*
- *All configuration is done via environment variables*
- *... that's it!*

*A historical aside while we wait for tests to run...*

# Demo: A small change...

airbytehq / airflow-summit-airbyte-2022 Public

Edit Pins Unwatch 14 Fork 0 Star 1

<> Code Issues Pull requests 1 ZenHub Actions Projects Wiki Security Insights Settings

## Update last\_name to be family\_name #1

Open evantahler wants to merge 1 commit into main from an-awesome-change

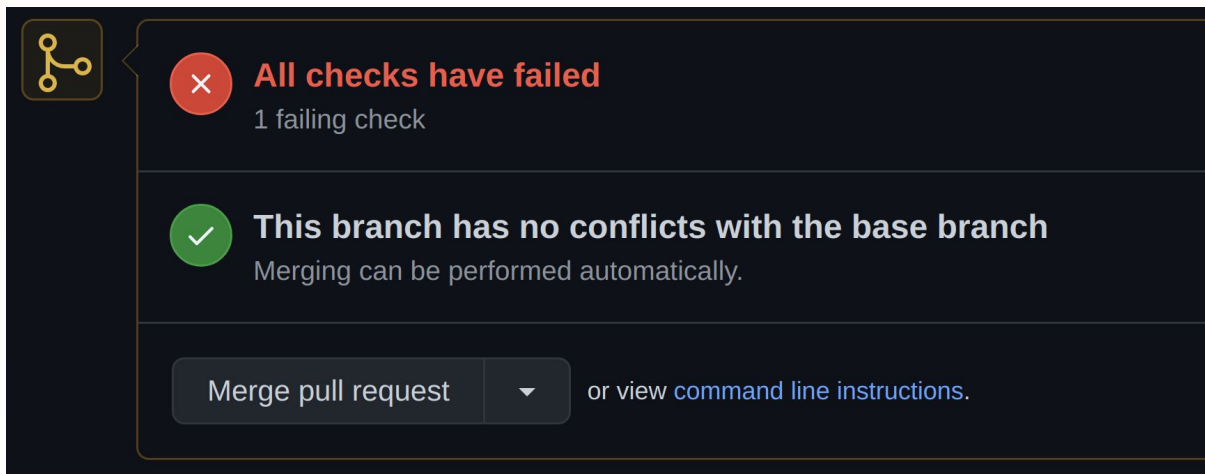
Conversation 0 Commits 1 Checks 0 Files changed 1 +1 -1

Changes from all commits File filter Conversations Jump to 0 / 1 files viewed Review changes

```
airflow/dbt/models/generated/airbyte_ctes/public/data_stream_ab2.sql
@@ -9,7 +9,7 @@
9 9 select
10 10     cast(user_id as {{ dbt_utils.type_bigint() }}) as user_id,
11 11     split_part(cast(name as {{ dbt_utils.type_string() }}), ' ', 1) as first_name,
12 -    split_part(cast(name as {{ dbt_utils.type_string() }}), ' ', 2) as last_name,
12 +    split_part(cast(name as {{ dbt_utils.type_string() }}), ' ', 2) as family_name,
13 13     cast(email as {{ dbt_utils.type_string() }}) as email,
14 14     cast(username as {{ dbt_utils.type_string() }}) as username,
15 15     cast(created_at as {{ dbt_utils.type_timestamp() }}) as created_at,
```

# Failing PRs are great.

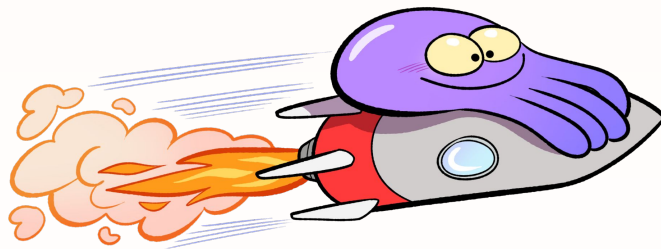
- The data product and our users are protected
- Clear and immediate feedback
- A place for the team to communicate about what to do next
- Less bugs and a better night's sleep



# What's next?

*Fix the PR 🤔... But then what?*

- *Automatic Deployment from the `main` branch*
- *Health-checks on the production system (both infrastructure and data)*
- *Better Tests:*
  - *We have an end-to-end Integration Test*
  - *Unit Tests - Checking specific known rows*
  - *Acceptance Tests - Great Expectations*
  - *Linting - SQLFluff*

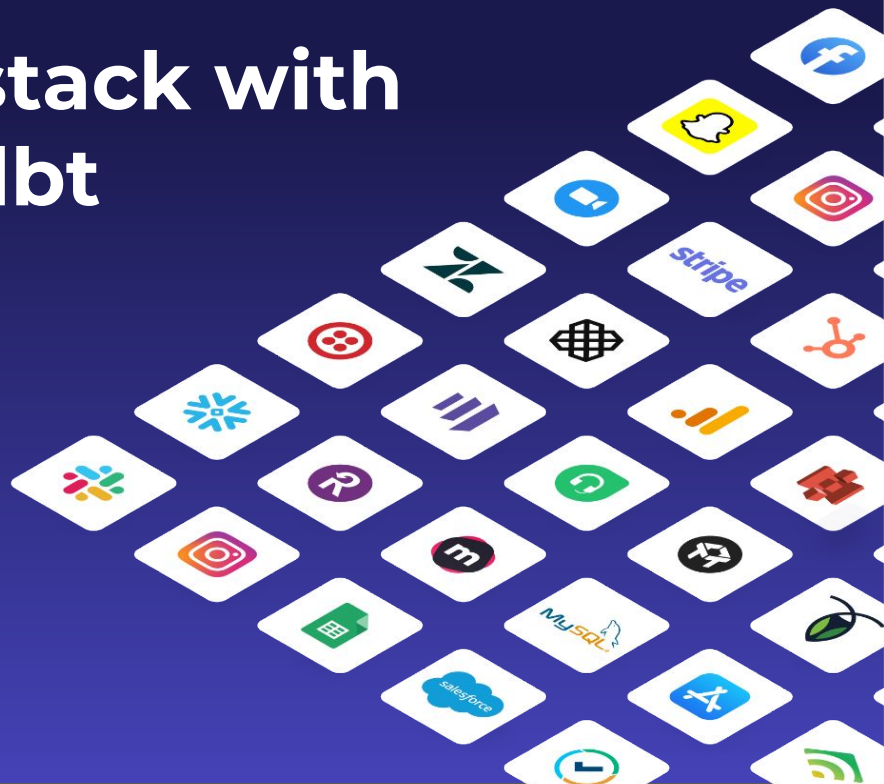


*The entry point for all of the above is another Github Action that checks PRs and each deployment*



``git push`` your data stack with  
Airbyte, Airflow, and dbt

Airflow Summit 2022





**THE END**

***Template Slides  
Below***

# 1. Section 1

2. Section 2

3. Section 3

4. Section 4

5. Section 5



# Section title



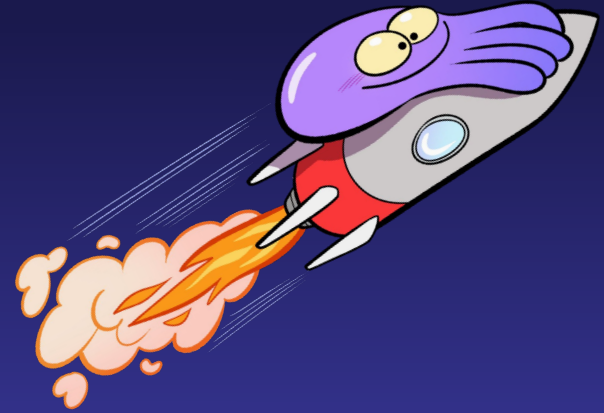
# Section title



# Section title

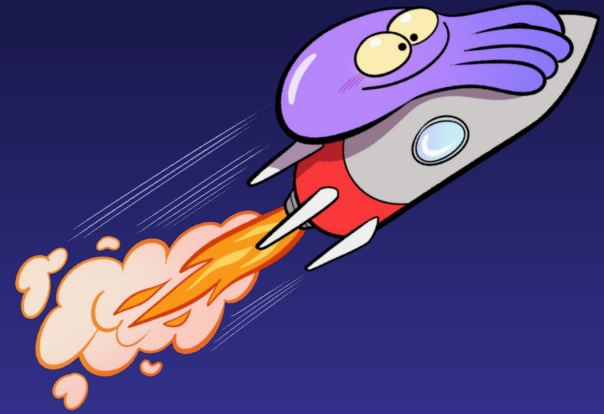


# Section title



# Summary

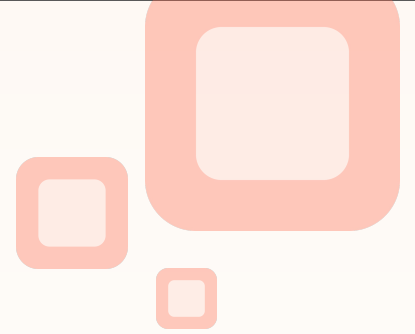
- **#troubleshooting** to Troubleshooting
- **#troubleshooting** to Troubleshooting
- **#troubleshooting** to Troubleshooting
- **#troubleshooting** to Troubleshooting
- **#troubleshooting** to Troubleshooting
- **#troubleshooting** to Troubleshooting





# Slide Main Title

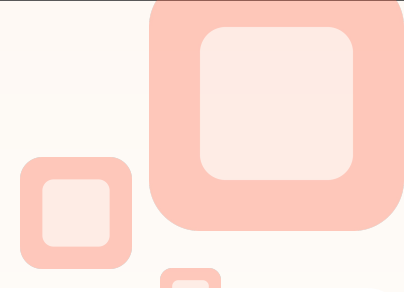
## Sub section title



- *#troubleshooting* to Troubleshooting
- *#troubleshooting* to Troubleshooting
- *#troubleshooting* to Troubleshooting
- *#troubleshooting* to Troubleshooting
- *#troubleshooting* to Troubleshooting
- *#troubleshooting* to Troubleshooting

# Slide Main Title

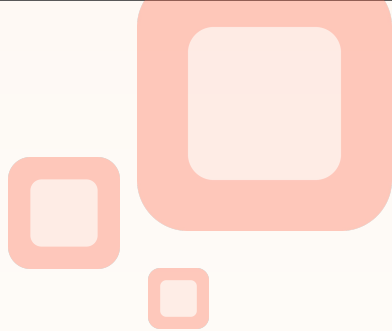
## Sub section title



<b>Mission</b>	Power organizations' data movement.
<b>Primary persona</b>	Data engineers.
<b>Tag line</b>	The data integration platform for any connection
<b>Narrative</b>	
<b>What is Airbyte?</b>	Airbyte is an open-source and hosted ELT solution that replicates data from a long-tail of APIs, databases, and files to data
<b>Mission</b>	Power organizations' data movement.
<b>Mission</b>	Power organizations' data movement.
<b>Mission</b>	Power organizations' data movement.

# Slide Main Title

## Sub section title



### **#advice channels**

*#advice-data-orchestration*

*#advice-data-ingestion*

*#advice-data-architecture*

*#advice-data-transformation*

*#advice-data-quality*

*#advice-data-warehousing*

*#advice-reverse-etl*

*#advice-data-privacy*

*#advice-data-visualization*

### **#advice channels**

*#advice-data-orchestration*

*#advice-data-ingestion*

*#advice-data-architecture*

*#advice-data-transformation*

*#advice-data-quality*

*#advice-data-warehousing*

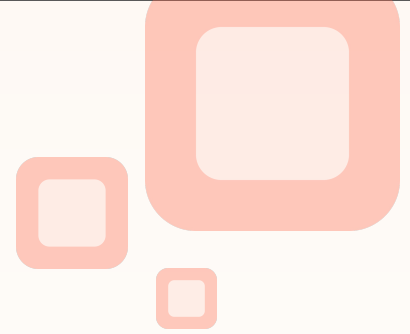
*#advice-reverse-etl*

*#advice-data-privacy*

*#advice-data-visualization*

# Slide Main Title

## Sub section title



Lorem ipsum dolor  
sit amet lorem ipsum  
dolor sit amet



Lorem ipsum dolor  
sit amet lorem ipsum  
dolor sit amet



Lorem ipsum dolor  
sit amet lorem ipsum  
dolor sit amet



Lorem ipsum dolor  
sit amet lorem ipsum  
dolor sit amet



Lorem ipsum dolor  
sit amet lorem ipsum  
dolor sit amet

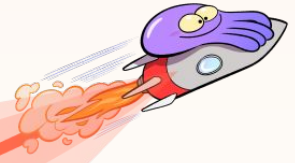


Lorem ipsum dolor  
sit amet lorem ipsum  
dolor sit amet



# Slide Main Title

## Sub section title



### Bloc title

- Conferences
- Sponsored events
- Meetup program with ambassadors
- Community calls

### Bloc title

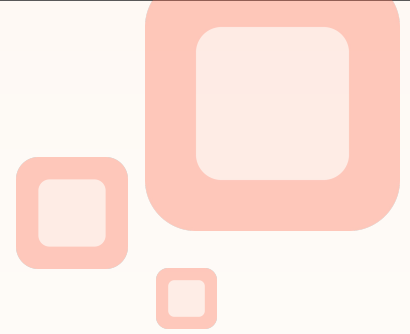
- Conferences
- Sponsored events
- Meetup program with ambassadors
- Community calls

### Bloc title

- Conferences
- Sponsored events
- Meetup program with ambassadors
- Community calls

# Slide Main Title

## Sub section title



**Abhi  
Vaidyanatha**

Senior dev advocate



**Abhi  
Vaidyanatha**

Senior dev advocate



**Abhi  
Vaidyanatha**

Senior dev advocate

# Slide Main Title

## Sub section title



**Abhi  
Vaidyanatha**

Senior dev advocate



**Abhi  
Vaidyanatha**

Senior dev advocate



**Abhi  
Vaidyanatha**

Senior dev advocate



**Abhi  
Vaidyanatha**

Senior dev advocate



**Abhi  
Vaidyanatha**

Senior dev advocate



**Abhi  
Vaidyanatha**

Senior dev advocate



**Abhi  
Vaidyanatha**

Senior dev advocate



**Abhi  
Vaidyanatha**

Senior dev advocate



**Abhi  
Vaidyanatha**

Senior dev advocate

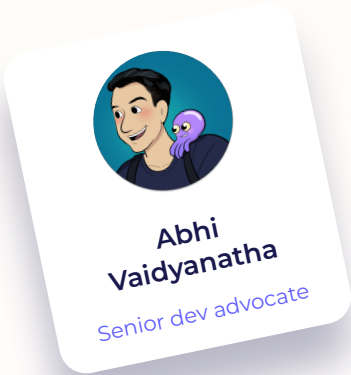


**Abhi  
Vaidyanatha**

Senior dev advocate

# Slide Main Title

## Sub section title



**“Quote lorem ipsum dolor  
sit lorem ipsum dolor sit  
amet lorem ipsum”**