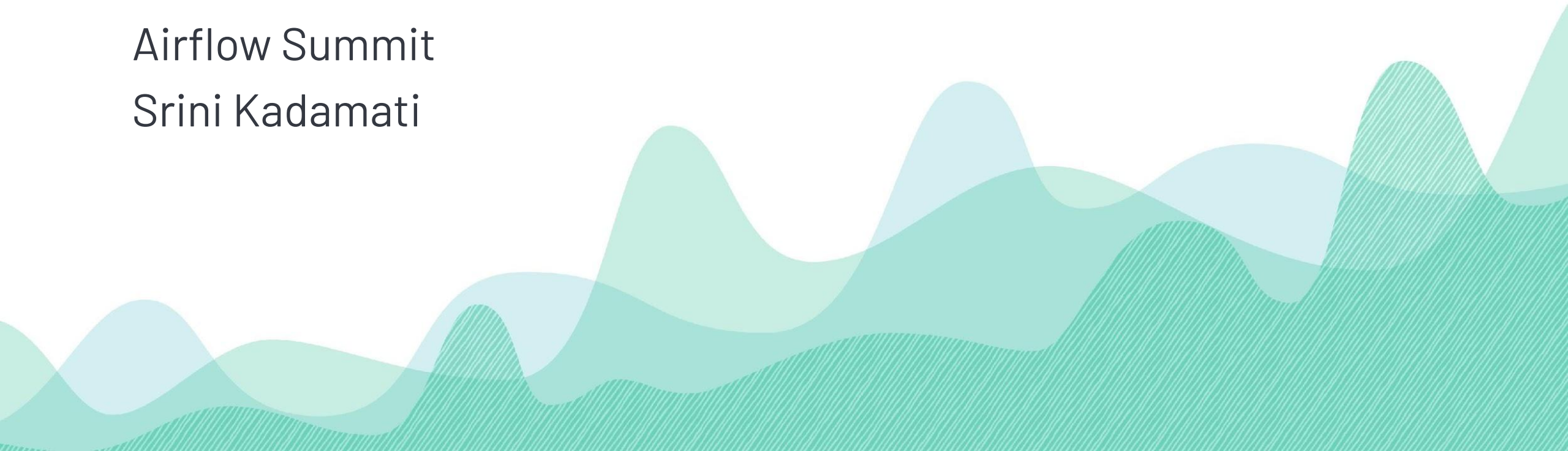




Advanced Superset for Engineers

Airflow Summit

Srini Kadamati



Overview

Sections

- Tour of Superset
- Superset API
- Version Controlling Charts & Dashboards
- The Future
- How to Contribute!

About Me

Previously

Data Scientist for 2 years

Data Science Educator / Product Manager at Dataquest for 5 years

RADIUS



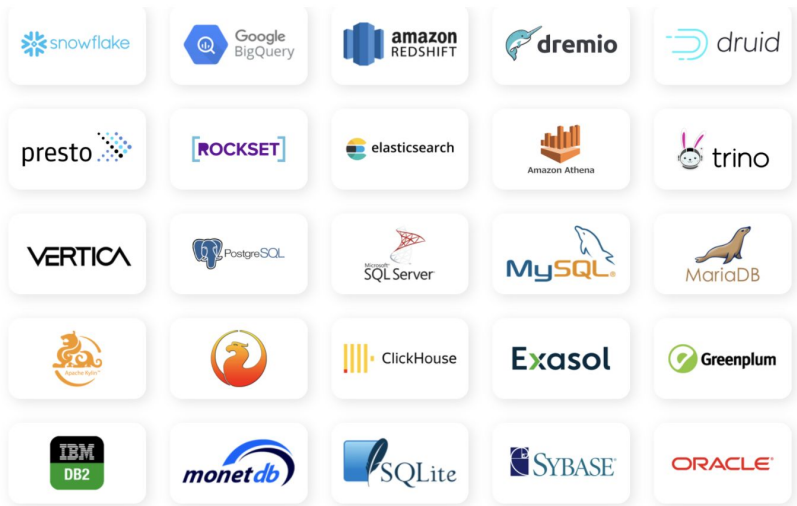
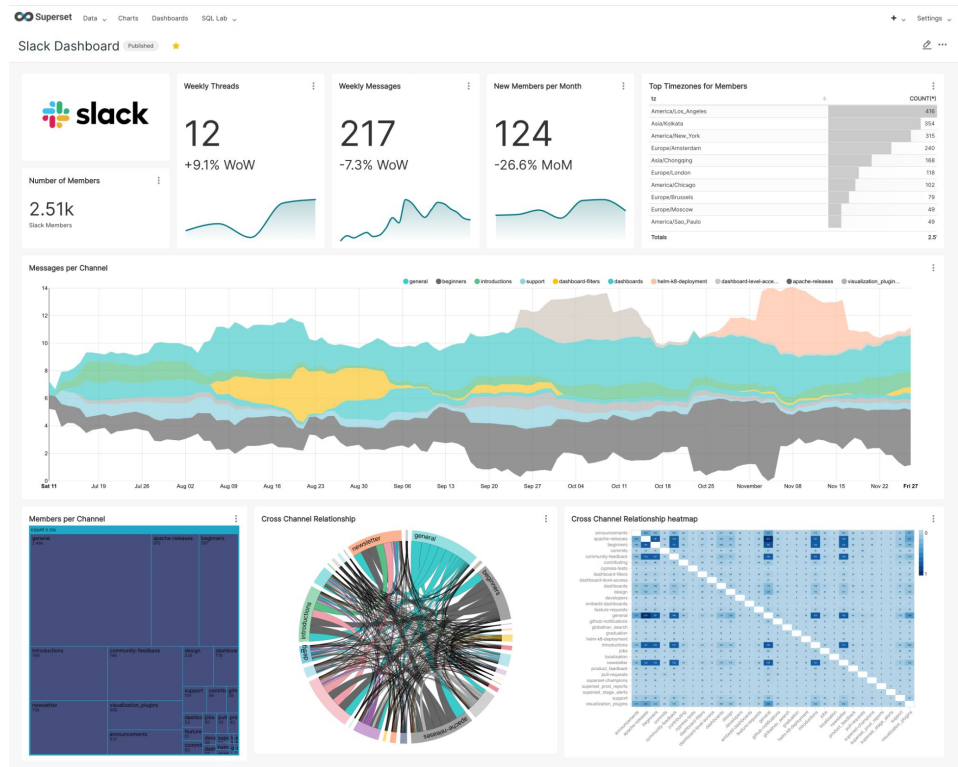
DATAQUEST

Currently

Senior Developer Advocate at Preset
+Committer to Superset

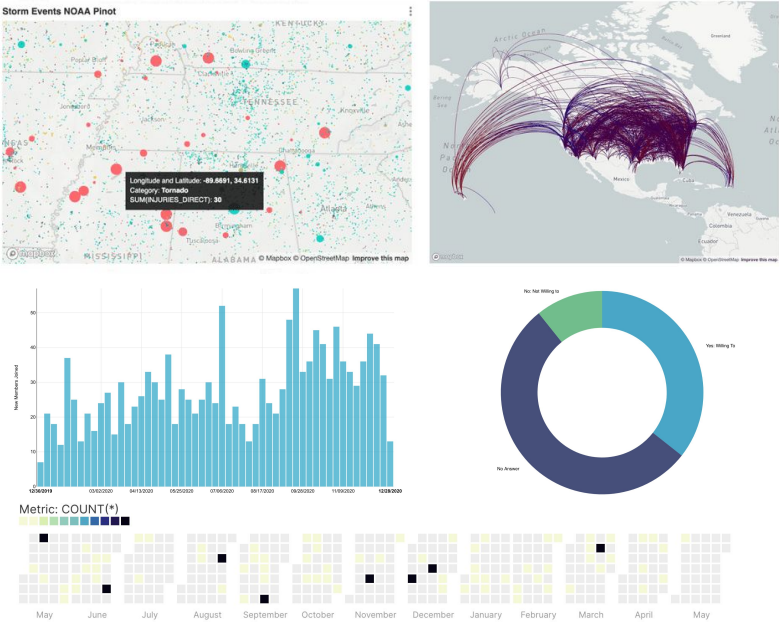


What is Apache Superset?



Works with
nearly any
SQL speaking
data engine

Modern open source BI
platform



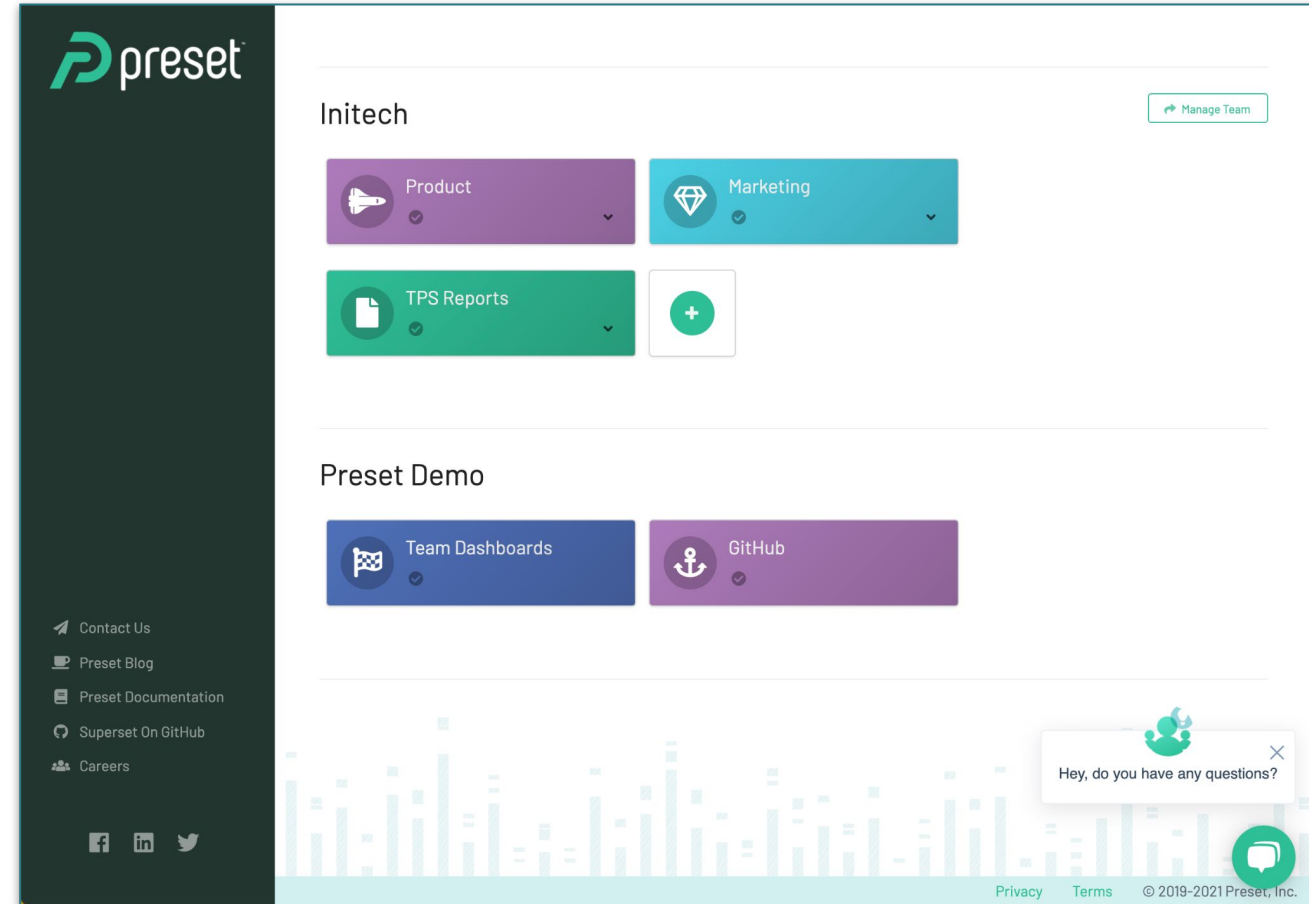
Large
diversity of
charts

What is Preset?

Preset Cloud is a multi-workspace, cloud-hosted Apache Superset (with all the batteries, bells, & whistles included)

We also layer on:

- SOC2 Compliance
- End-user documentation at docs.preset.io
- RBAC and SSO



Tour of Superset

Homepage

Databases

Datasets

Saved Queries

SQL Lab

Charts

Dashboards

Semantic Layer

- Metrics
- Calculated Columns
- Virtual datasets

Jinja Filters

Alerts & Reports

Row Level Security

Superset API

API Demo!

Jupyter

Superset API Examples

Last Checkpoint: 29 minutes ago (autosaved)

Logout

FileEditViewInsertCellKernelWidgetsHelp

Not TrustedPython 3

+

Run

Code

```
In [47]: import pandas as pd
import json
import requests

Authenticate with API

In [48]: payload = {
    'username': 'admin',
    'password': 'admin',
    'provider': 'db'
}

r = requests.post('http://localhost:9000/api/v1/security/login', json=payload)

In [49]: access_token = r.json()
access_token

Out[49]: {'access_token': 'eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpYXQiOiJlY2MjY0NDY0MTESIm5iziI6MTYyNjQ0NjQxMSwianRpIjoiotE1NTdmZjgtMjc4ZS00NTdmLTljNTgtMDJkNWFKODYxZjQwIiwiaXNjaXoxNjI2NDQ3MzExLCJpZGVudGl0eSI6MSwiZnJlc2giOnRydWUsInR5cGU6Ij0iJhY2Nlc3MifQ.qc9MQmaAPcEHd5FddeVyR-PmMjkqF7rtmm4maLvWVvK'}

In [50]: headersAuth = {
    'Authorization': 'Bearer ' + str(access_token['access_token'])
}
```


Tech Specs

Follows OpenAPI specification and is RESTful

Built mostly using [Flask App Builder](#)

Interactive Swagger documentation lives at **/swagger/v1** endpoint

Blog:

preset.io/blog/2020-10-01-superset-api/



API Use Cases

Invalidate cache upon data refresh

- **GET** /chart/data{cache_key}
- **POST** /cachekey/invalidate

Import, export dashboards

- **GET** /dashboard/export
- **POST** /dashboard/import

Create new database

- **POST** /database/

Return SELECT * from a db table

- **GET**
/database/{pk}/select_star/{table_name}/

Schedule a new report

- **POST** /report

Version Control

Superset CLI

Version control workflow is hacky, and is more of a PUSH-into-Superset workflow than a PULL one

1. Export Dashboard (from UI, *shell coming soon*)
2. Import Dashboard (from **superset shell**)

Beto Dealmeida opened a [PR \(#15686\)](#) to enable a CLI command for importing a dashboard:

```
superset import-directory -o  
~/Downloads/dashboard_export_20210715T205336/
```

Version Control Demo!

```
(venv) → superset git:(import_dir_command) superset import-directory -o ~/Downloads/dashboard_export_20210715T205336/
Loaded your LOCAL configuration at [/Users/srinify/Documents/superset/superset_config.py]
logging was configured successfully
2021-07-15 21:16:55,107:INFO:superset.utils.logging_configurator:logging was configured successfully
2021-07-15 21:16:55,113:INFO:root:Configured event logger of type <class 'superset.utils.log.DBEventLogger'>
/Users/srinify/Documents/superset/venv/lib/python3.8/site-packages/flask_caching/__init__.py:201: UserWarning: Flask-Caching: CACHE_TYPE is set to null, caching is effectively disabled.
  warnings.warn(
/Users/srinify/Documents/superset/superset/examples/utils.py:85: YAMLLoadWarning: calling yaml.load() without Loader=... is deprecated, as the default Loader is unsafe. Please read https://msg.pyyaml.org/load for full details.
  metadata = yaml.load(contents.get(METADATA_FILE_NAME, "{}"))
Updating dbs examples
2021-07-15 21:16:56,268:INFO:superset.models.helpers:Updating dbs examples
Updating tables birth_names
2021-07-15 21:16:56,279:INFO:superset.models.helpers:Updating tables birth_names
Updating sql_metrics <superset.connectors.sqla.models.SqlMetric object at 0x11c0b93a0>
2021-07-15 21:16:56,284:INFO:superset.models.helpers:Updating sql_metrics <superset.connectors.sqla.models.SqlMetric object at 0x11c0b93a0>
Updating sql_metrics <superset.connectors.sqla.models.SqlMetric object at 0x11c0b9820>
2021-07-15 21:16:56,286:INFO:superset.models.helpers:Updating sql_metrics <superset.connectors.sqla.models.SqlMetric object at 0x11c0b9820>
Updating table_columns num_california
2021-07-15 21:16:56,291:INFO:superset.models.helpers:Updating table_columns num_california
Updating table_columns state
2021-07-15 21:16:56,293:INFO:superset.models.helpers:Updating table_columns state
Updating table_columns gender
2021-07-15 21:16:56,295:INFO:superset.models.helpers:Updating table_columns gender
Updating table_columns name
2021-07-15 21:16:56,297:INFO:superset.models.helpers:Updating table_columns name
Updating table_columns ds
2021-07-15 21:16:56,298:INFO:superset.models.helpers:Updating table_columns ds
Updating table_columns sum_boys
2021-07-15 21:16:56,300:INFO:superset.models.helpers:Updating table_columns sum_boys
Updating table_columns sum_girls
2021-07-15 21:16:56,301:INFO:superset.models.helpers:Updating table_columns sum_girls
Updating table_columns num
2021-07-15 21:16:56,303:INFO:superset.models.helpers:Updating table_columns num
Updating slices Pivot Table
2021-07-15 21:16:56,391:INFO:superset.models.helpers:Updating slices Pivot Table
Importing new dashboards Dashboard<None>
2021-07-15 21:16:56,399:INFO:superset.models.helpers:Importing new dashboards Dashboard<None>
```

Version Control Workflow

1. Use Superset to iterate on dashboard
2. Export dashboard, as static files
3. Check in static files to Github
4. Setup CI workflow to

Next Steps

Export dashboards (as ZIP) via API

Better error messages, documentation, & developer experience

Eventually:

- Implement design & UI for version control within Superset itself

Embedding Superset

Expose individual standalone charts

Expose individual standalone dashboards

...anddd IFrame!

How?

- Use public role as gamma [\(docs\)](#)

Help us improve this by joining **#embedding-superset** in our Slack!

Other Advanced Features

Lots of nifty, experimental [feature flags](#) to play with:

- new native filters
- omnibar
- cross filtering

Adding custom viz plugins:

- Creating a Custom Viz Plugin ([tutorial](#))
- How Nielsen Manages Custom Viz Plugins in Production ([blog](#))

Superset as a Platform

Expose all the legos!

- API for everything
- Highly custom embedded dashboards
- Easily build applications & data products
- Viz plugin packs (bring your own viz library + charts)

How to Contribute



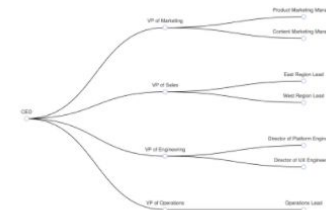
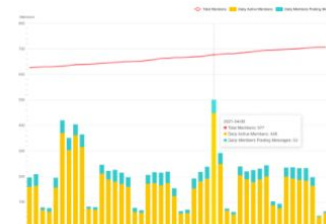
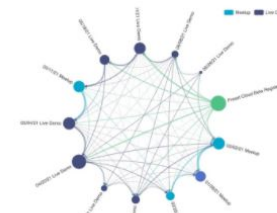
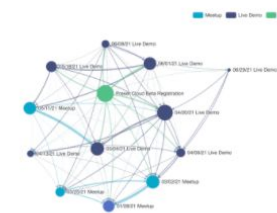
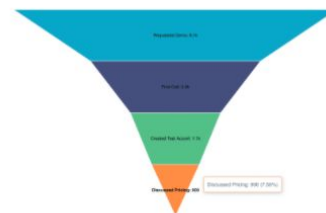
More databases!

- *Recently:* CrateDB, Netezza, & Databricks Cloud

More charts!

- *Recently:* Force directed graphs, funnel, radar charts, tree chart, etc

[Learn more about how to contribute](#)



Resources

- Preset Docs: <https://docs.preset.io/docs>
- Preset User Corner (YT):
<https://www.youtube.com/channel/UCzg8opP7sG8n0Mi0e8yeqAg>
- Apache Superset Community Docs:
<https://superset.apache.org/docs/intro>
- CONTRIBUTING.MD:
<https://github.com/apache/superset/blob/master/CONTRIBUTING.md>

Twtr: **@apachesuperset**