

What is Rill?

<https://github.com/rilldata/rill>

Rill = an open-source platform that simplifies real-time data processing and analytics. It's key capabilities are:

- Real-time Data Processing: Rill allows you to process data as it's generated, making it ideal for applications requiring up-to-the-second insights
- Data Transformation: You can transform and enrich data streams using various built-in functions, ensuring your data is in the desired format for analysis
- Integration with Popular Data Stores: Rill seamlessly integrates with data stores like Elasticsearch, Kafka, and more
- Scalability: It's designed to handle high volumes of data, making it suitable for both small-scale and large-scale projects

Why is it so cool?

- Dashboards-as-code: everything can be version controlled & easily shared using Git
- Blazing fast & a pleasure to interact with: powered by DuckDB & Sveltekit
- Automatic data profiling
- A plethora of data sources supported
- No need to compile/wait for computations to finish: everything re-calculates on-the-fly

311 US Call Center Operational Analytics with Rill

- What is "311 service"? <https://en.wikipedia.org/wiki/3-1-1>
- Data used (Boston example): <https://data.boston.gov/dataset/8048697b-ad64-4bfc-b090-ee00169f2323/resource/e6013a93-1321-4f2a-bf91-8d8a02f1e62f/download/tmphniigcid.csv>

Let's spin up & publish a dashboard about 311 Service!

0. In case you're running Windows, install WSL2 (Windows Subsystem for Linux) first:
<https://learn.microsoft.com/en-us/windows/wsl/>

1. Install Rill

```
curl -s https://cdn.rilldata.com/install.sh | bash
```

2. Explore **sources/** that we'll be working with in **sources/** directory. There're 4 cities => 4 .yaml files

More on processing different sources here: <https://docs.rilldata.com/develop/import-data>

3. Explore **models/** We're using DuckDB's SQL to address the sources configured in **sources/** directly: have a look at how `...from...` clauses look like.

Using `with` pattern we form 1 big table that will power our dashboard

4. Explore **dashboards/**

Inside a .yaml file we have multiple blocks:

- Model — data model that creates One Big Table that will power our dashboard
- Timeseries — a column from our model that will serve as x-axis in line charts. Time will get truncated into different time periods
- Measures — numerical aggregates of columns from our data model shown on the y-axis of the line charts and numerical summaries
- Dimensions — categorical columns from our data model whose values are shown in leaderboards and allow us to apply categorical filtering to data

5. Spin up the Dashboard:

```
rill start
```

6. In case the web UI didn't open automatically, navigate to <http://localhost:9009>

7. Explore the dashboard

8. Stop the dashboard by hitting Ctrl+C

9. Create a GitHub repository "rill_311_service" & push code stored locally there:

```
git init
git add .
git commit -m"added rill project files"
git branch -M main
git remote add origin git@github.com:migumax/rill_311_service.git
```

10. Deploy you project to Rill Cloud:

```
rill deploy
```

11. Wait for all the steps to finish

12. Explore your Dashboard in the Cloud & Share it!

To share access with 1 new user:

```
rill user add --project rill_311_service --role viewer
```

To make your project publicly accessible:

```
rill project edit --public=true
```

Bonus: setting up CICD

13. Update your project's code base. Let's say we'd like to set up a project-wide schedule for once-every-30-mins Data Sources updates: add this snippet to rill.yaml:

```
sources:
  refresh:
    cron: '*/*30 * * * *
```

```
git add => git commit -m"updated sources refresh" => git push
```

CICD in action with no hassle!

More on CICD settings here: <https://docs.rilldata.com/deploy/existing-project>