

RoachFest

**Build better.
Dream bigger.**

#RoachFest22





How we use CockroachDB to power Starburst Galaxy

David Phillips, CTO



Trino (formerly known as Presto)

- Fast distributed SQL query engine for big data analytics
- Open sourced at Facebook in 2013 as Presto
- Renamed to Trino in late 2020
- Used by hundreds of organizations all over the world
- Runs on a cluster of machines



Starburst Galaxy

Trino software-as-a-service

Starburst Galaxy

☰ Starburst Galaxy

<> Query ^

- Query editor
- Saved queries
- Query history
- Catalogs**
- Clusters

ACCOUNT

- Admin
- Access control
- Cloud settings












Configure a catalog

- Select a data source
- Configure the connection
- Set permissions (optional)
- Add to cluster (optional)

Select a data source

Each catalog contains configuration for Starburst Galaxy to access a data source. Configure catalogs and use them in clusters to query data sources in Starburst Galaxy.

Note: Amazon S3, Azure Data Lake Storage, and Google Cloud Storage catalogs support Iceberg, Hive, and Delta Lake tables.

 Amazon S3	 Azure Data Lake Storage	 Google Cloud Storage	 Microsoft SQL Server
 MySQL	 PostgreSQL	 Amazon Redshift	 Azure Synapse
 Google BigQuery	 Snowflake	 MongoDB	

Starburst Galaxy

The screenshot displays the Starburst Galaxy web interface. On the left is a navigation sidebar with options like 'Query', 'Query editor', 'Saved queries', 'Query history', 'Catalogs', and 'Clusters'. The main area is titled 'Cluster' and 'View clusters', containing a table with one cluster named 'sample' in a 'Suspended' state. A modal window titled 'Create a new cluster' is open, showing configuration options for a new cluster.

Cluster

View clusters

A cluster in Starburst Galaxy provides the resources to run queries against numerous catalogs exposed by the catalogs with the query editor or other clients.

Create cluster

Name ↑	Status	Size
⋮ sample	⏸ Suspended	Free

Create a new cluster

Cluster name *

Must start with a letter and only use lowercase letters (a-z), numbers (0-9), and hyphens (-)

Cluster size *

Free

Cluster type *

Standard Batch (experimental feature) ?

Contact us to allow batch cluster.

Idle shutdown time

30 Minutes

The maximum idle time before a cluster is automatically suspended.

Allow data transfer across clouds and regions

Contact us to allow cluster with catalogs in different clouds and regions.

Catalogs

1 catalog selected

Cloud provider region *

US East (Ohio)

Grant access to users with role(s)

Roles

public

Cancel **Create cluster**

Starburst Galaxy

The screenshot displays the Starburst Galaxy web interface. On the left is a navigation sidebar with sections for 'Query' (containing Query editor, Saved queries, Query history), 'Catalogs', 'Clusters', and 'ACCOUNT' (containing Admin, Access control, Cloud settings). The main area shows a query editor with a SQL query and a results table. The query is: `select province_state, sum(confirmed) as confirmed_cases, sum(active) as active_cases from ejhu_stg_australia where last_update = '2020-05-30T02:32:48' group by province_state;`. The results table shows data for Australian states and territories.

Starburst Galaxy

7/28/22, 2:45 PM

Cluster explo...

Run (limit 1000)

mm-aws-lab mm_aws_covid covid_lab

```
1
2 select
3   province_state,
4   sum(confirmed) as confirmed_cases,
5   sum(active) as active_cases
6 from
7   ejhu_stg_australia
8 where
9   last_update = '2020-05-30T02:32:48'
10 group by
11  province_state;
```

Finished Avg. read speed 584 rows/s Elapsed time 0.47s Rows 8 Query details Trino UI Download

province_state	confirmed_cases	active_cases
Australian Capital Territory	107	0
Northern Territory	29	0
Queensland	1058	6
South Australia	440	1

Starburst Galaxy

Name	Status	Quick actions	Catalogs
⋮ batch-elt	✓ Running	⊗ Stop	aurora_stackexchange, glue
⋮ financial-reporti..	✗ Stopped	▶ Start	glue
⋮ dataengineering	⏸ Suspended	▶ Resume	aurora_stackexchange, s3_s
⋮ development	✓ Running	⊗ Stop	s3_storage
⋮ mktg-analytics..	✓ Running	⊗ Stop	glue, postgres_crypto

The screenshot displays the Starburst Galaxy user management interface. It features several overlapping panels:

- Role Management Panel:** A table listing roles with columns for 'Role name' and 'Desc'. Roles shown include 'accountadmin' and 'public'. An 'Add role' button is visible at the bottom.
- View privileges: accountadmin Panel:** A table showing the privileges assigned to the 'accountadmin' role. The table has columns for 'Entity' and 'Privilege'. Privileges listed include 'Create catalog', 'Create cluster', 'Create role', 'Create user', 'Use cluster', and 'View all query history'. Each row includes a lock icon.
- Assign users Panel:** A panel with a 'User' dropdown menu and an 'Assign' button.

At the bottom of the interface, there are two certification logos: the AICPA SOC logo and the ISO 27001 logo.

Galaxy architecture



Why CockroachDB?

Reliability

- No downtime or data loss
- Always available, even if we lose a cloud zone or region
- Support customers in regions all around the world

Development

- Standard SQL semantics
- Unique indexes, constraints, foreign keys, etc.
- Easy for engineers to understand
- Robust against bugs in application code
- Serializable transactions prevents errors

Operations

- Managed service — let someone else run it
- Allow evolution as we add new features
- Online schema changes
- Transparently scales to larger clusters

How we use CockroachDB

Usage: What is the data?

- Trino cluster management
- Users and permissions
- Metastore for user data
- Trino query history (~1000x larger than other data)
- Transient state for stateless services

Global tables

- Instant reads from anywhere
- High latency writes
- Per-statement latency for writes from non-primary region
- Forward writes to service in primary region
- Perform forwarding at service API layer

Schema migrations

Using Flyway with CockroachDB

- Existing CockroachDB support is outdated and broken
- Run each migration in separate transaction
- Update **flyway_schema_history** table within transaction
- New version: <https://github.com/electrum/flyway-cockroachdb>

Beware of mixing DML with DDL

```
> CREATE TABLE abc (x int);  
> INSERT INTO abc VALUES (3), (4), (null);  
> BEGIN;  
OPEN> UPDATE abc SET x = x * 2 WHERE x > 0;  
OPEN> ALTER TABLE abc ALTER COLUMN x SET NOT NULL;  
OPEN> COMMIT;  
ERROR: transaction committed but schema change aborted with  
error: (23514): validation of NOT NULL constraint failed:  
validation of CHECK "x IS NOT NULL" failed on row: x=NULL,  
rowid=...
```

Beware of mixing DML with DDL

```
> SELECT * FROM abc;
```

```
  X
```

```
-----
```

```
  6
```

```
  8
```

```
 NULL
```

```
(3 rows)
```

Don't be clever with transactions

```
> CREATE TABLE abc (x int NOT NULL);  
> INSERT INTO abc VALUES (5);  
> BEGIN;  
OPEN> ALTER TABLE abc ADD COLUMN y int NOT NULL DEFAULT 0;  
OPEN> ALTER TABLE abc ALTER COLUMN y DROP DEFAULT;  
OPEN> COMMIT;  
ERROR: transaction committed but schema change aborted with  
error: (23502): null value in column "y" violates not-null  
constraint
```

Rules for migrations

- DDL that interacts with existing data can fail
- Avoid mixing DML with DDL
- Mixing is only safe if the DDL cannot fail
- Only one ALTER TABLE for same table per transaction
- Use separate transactions unless atomicity is required

Thank you!