
AWS
re:Invent
NOV. 28 – DEC. 2, 2022 | LAS VEGAS, NV

PRT099

SPONSORED BY COCKROACH LABS

How Starburst uses CockroachDB to power Starburst Galaxy

Matt Fuller

Co-Founder
Starburst

aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

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

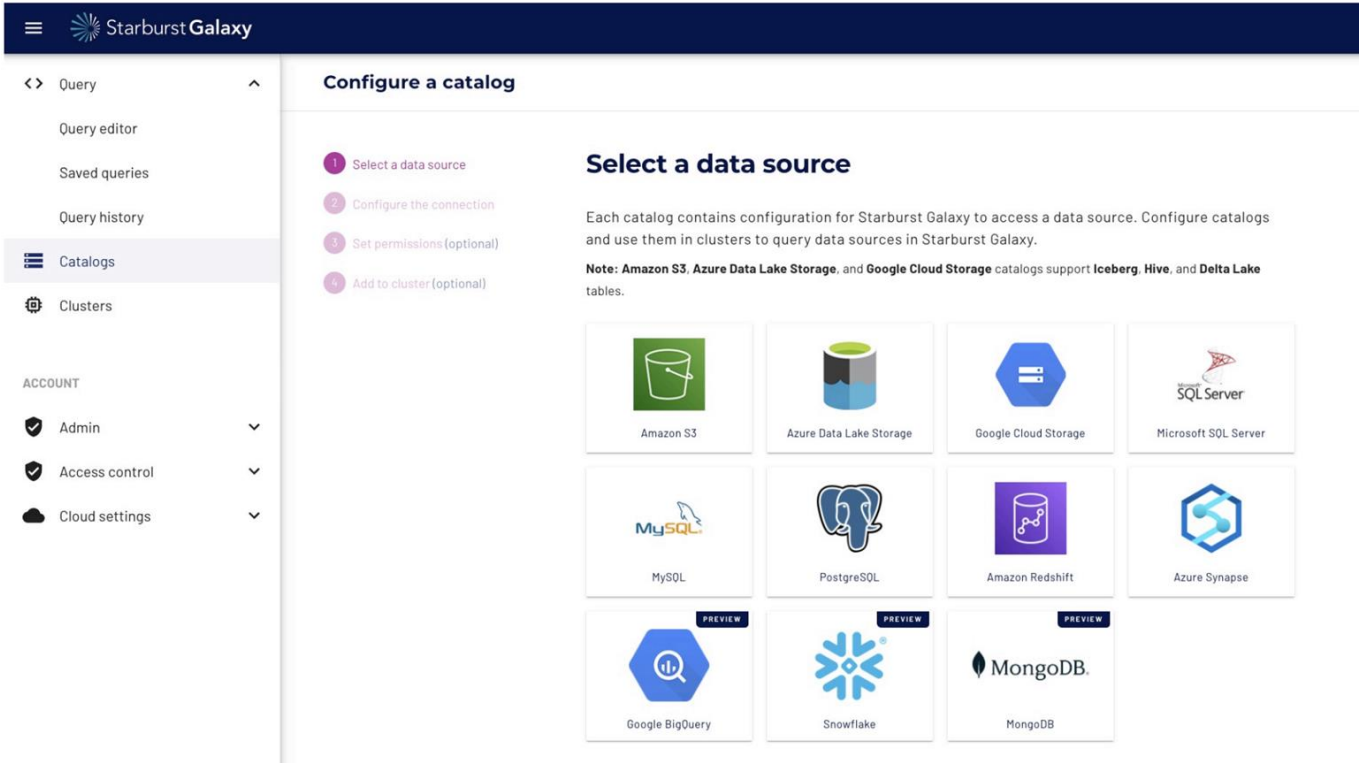
Starburst

Starburst Galaxy

Trino software as a service

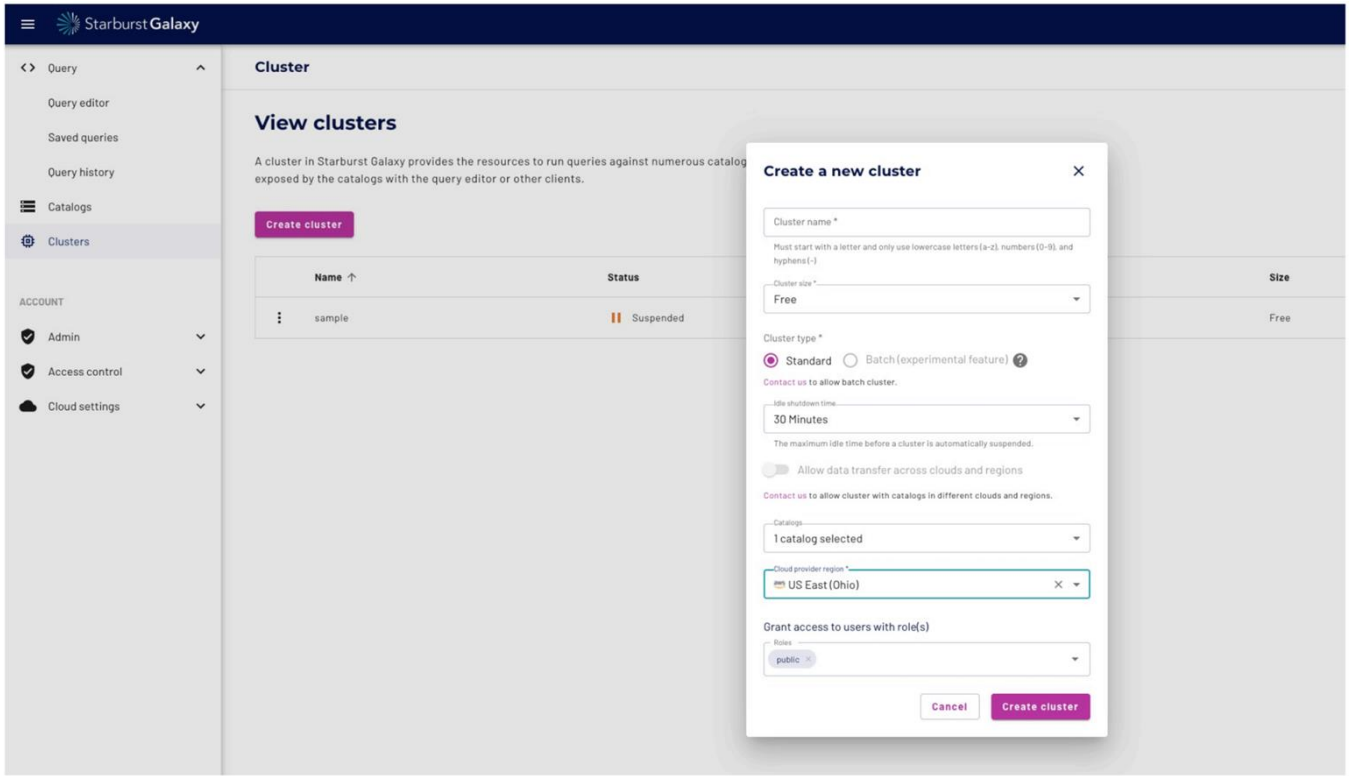
Starburst

Starburst Galaxy



Starburst

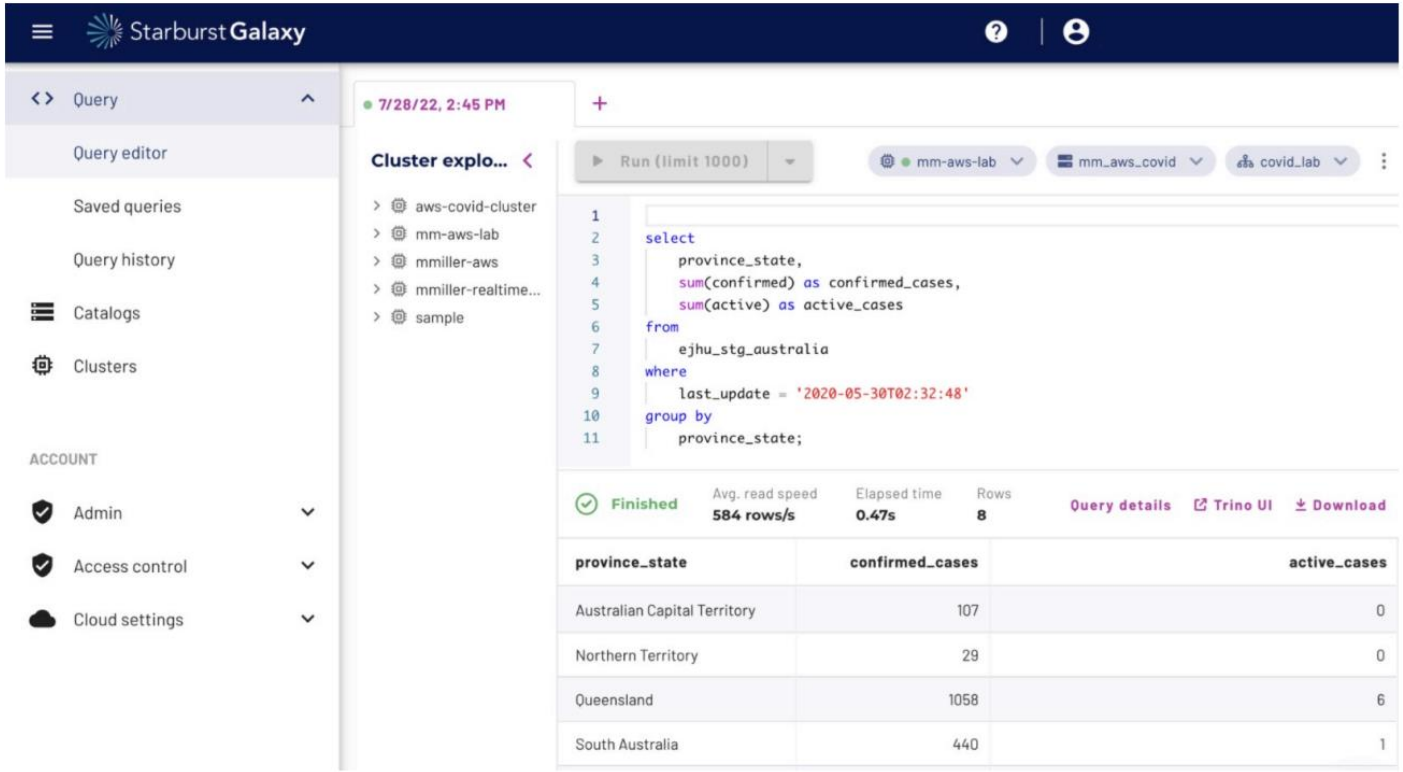
Starburst Galaxy



Starburst

Starburst

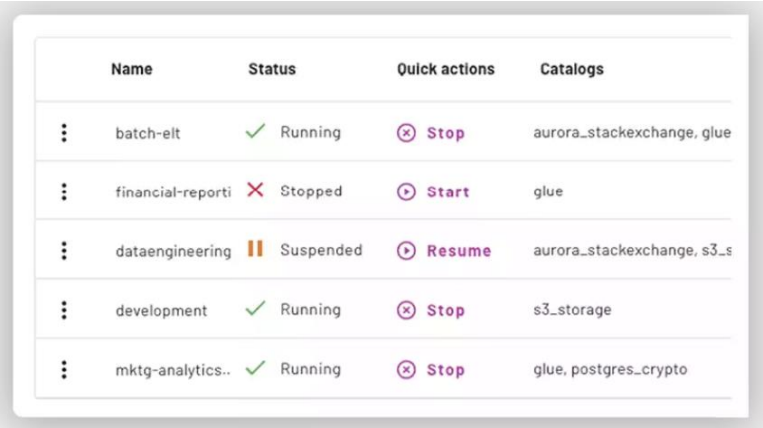
Starburst Galaxy

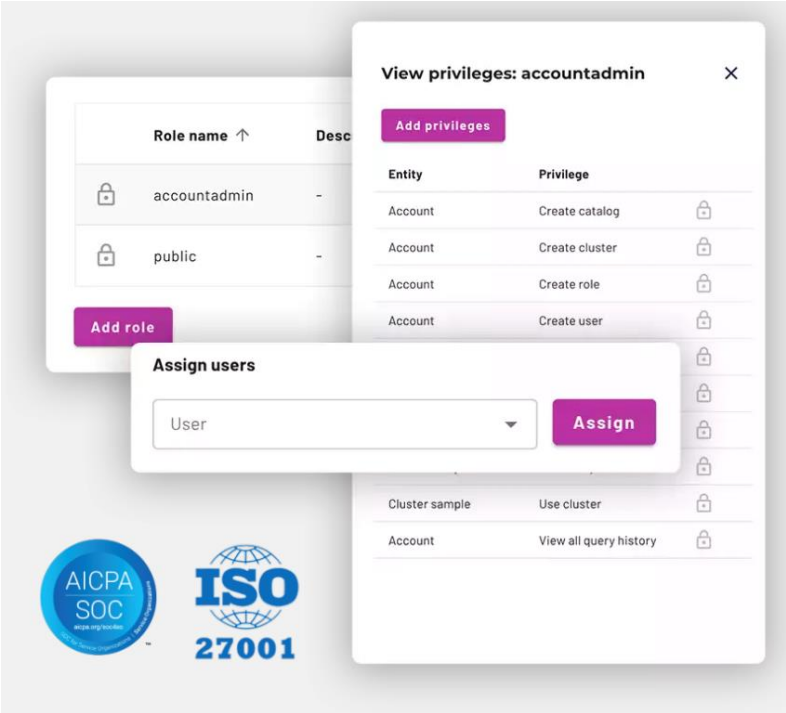


Starburst

Starburst

Starburst Galaxy

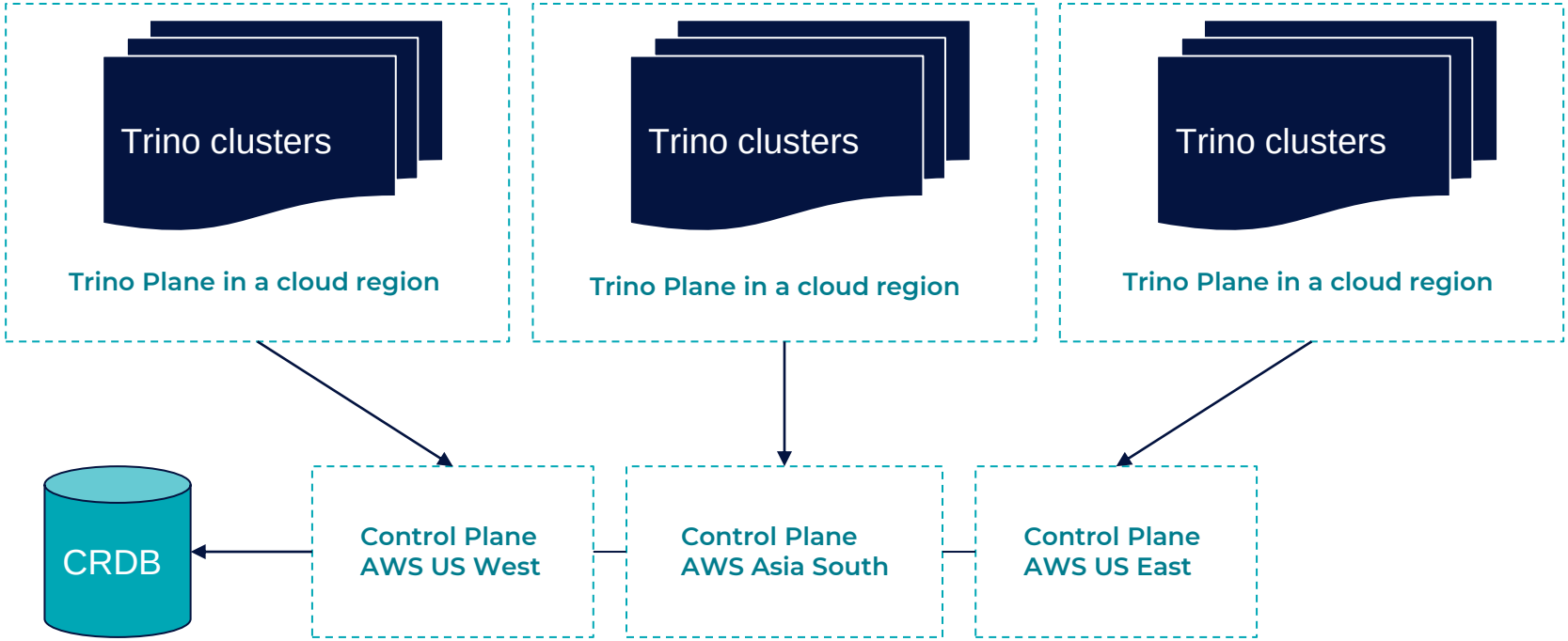




Starburst

Starburst

Galaxy architecture



Starburst

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

Starburst

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

Starburst

Operations

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

Starburst

How we use CockroachDB

Usage: What is the data?

- Trino cluster management
- Users and permissions
- Metastore for user data
- Trino query history (~1,000x larger than other data)
- Transient state for stateless services

Starburst

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

Starburst

Schema migrations

Starburst

Using Flyway with CockroachDB

- Run each migration in separate transaction
- Update `flyway_schema_history` table within transaction
- New version: *`https://github.com/electrum/flyway-cockroachdb`*

Starburst

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=...
```


Starburst

Beware of mixing DML with DDL

```
> SELECT * FROM abc;
   x
--------
      6
      8
   NULL
(3 rows)
```


Starburst

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
```


Starburst

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

Starburst

Thank you!



Please complete the session survey in the **mobile app**

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.