

PostgreSQL でペタバイト・リアルタイム? UK の COVID-19 ダッシュボードに見る Cosmos DB for PostgreSQL の活用

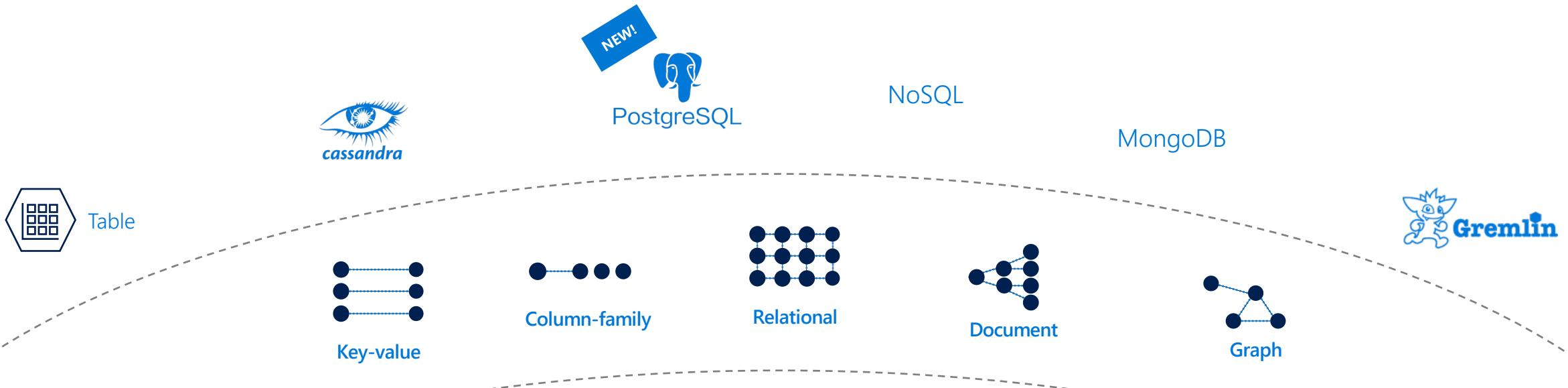
Rio Fujita

GBB OSS Data Senior Specialist



Azure Cosmos DB

Azure Cosmos DB



完全マネージド、
サーバレス

瞬時の柔軟な
スケーラビリティ

10 ms未満のレイテンシー
と99.999%の可用性を保証
する SLA (NoSQL データ)

全ての Azure リー
ジョンへのデータ
レプリケーション

ターンキー
マルチマスター
書き込み

Microsoft Windows relies on Cosmos DB for PostgreSQL for mission-critical decisions

"Ship/no-ship decisions for Microsoft Windows are made using Cosmos DB for PostgreSQL where our team runs on-the-fly analytics on billions of JSON events with sub-second responses.

Distributed SQL with Cosmos DB for PostgreSQL is a game changer."

2.0 PB+ data (8TB / day)

6M queries / day

Real-time analytics: 95% queries execute < 4s

75% queries execute < 200ms

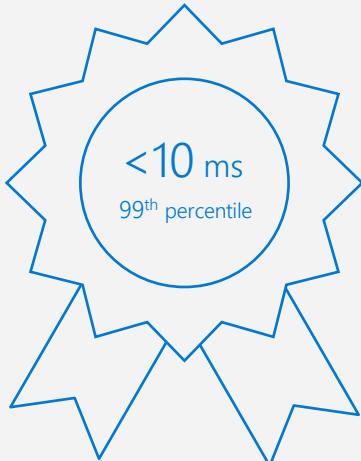


包括的SLA（NOSQLデータ）

全世界のインフラでアプリを実行

Azure Cosmos DBは、99パーセンタイルで1桁msの読み書きと、99.999%の高可用性、確約されたスループットと整合性を、課金で保証されたSLAを持つ唯一のサービス

Latency



High Availability



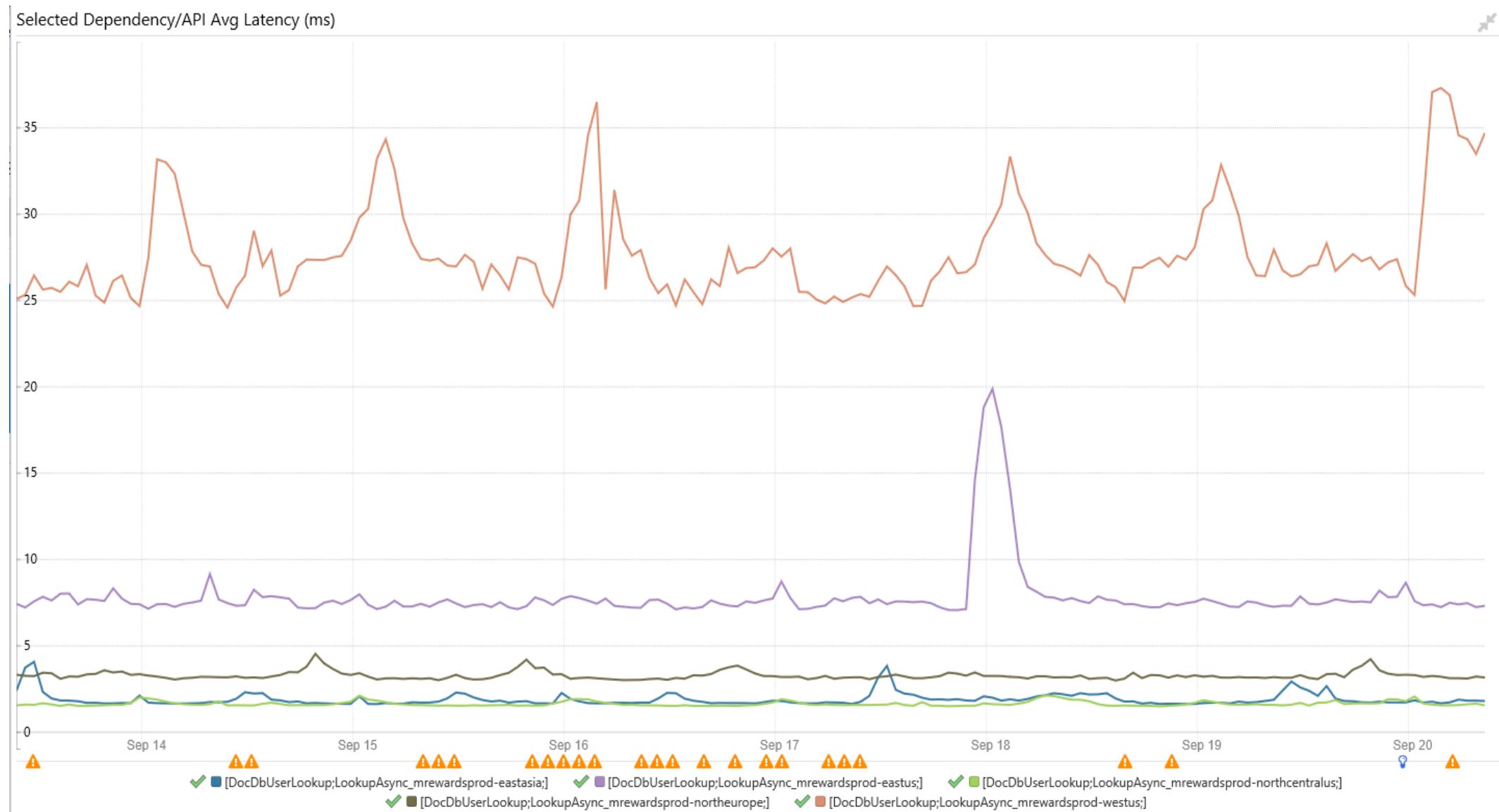
Throughput



Consistency



レイテンシーの実例

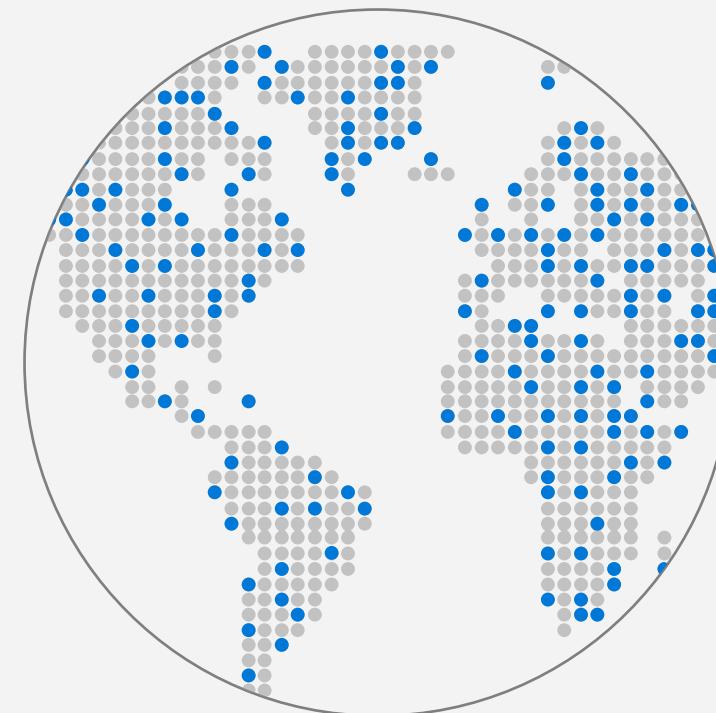


ターンキー グローバル分散

ユーザのいる場所に迅速にデータを置く

AWSやGoogleを合わせたよりも多くのリージョン、
世界中に、自動的にデータをレプリケーション

- [全ての Azure regions](#) で利用可能
- 手動・自動のフェイルオーバー
- 自動化かつ同期した複数リージョンの
レプリケーション



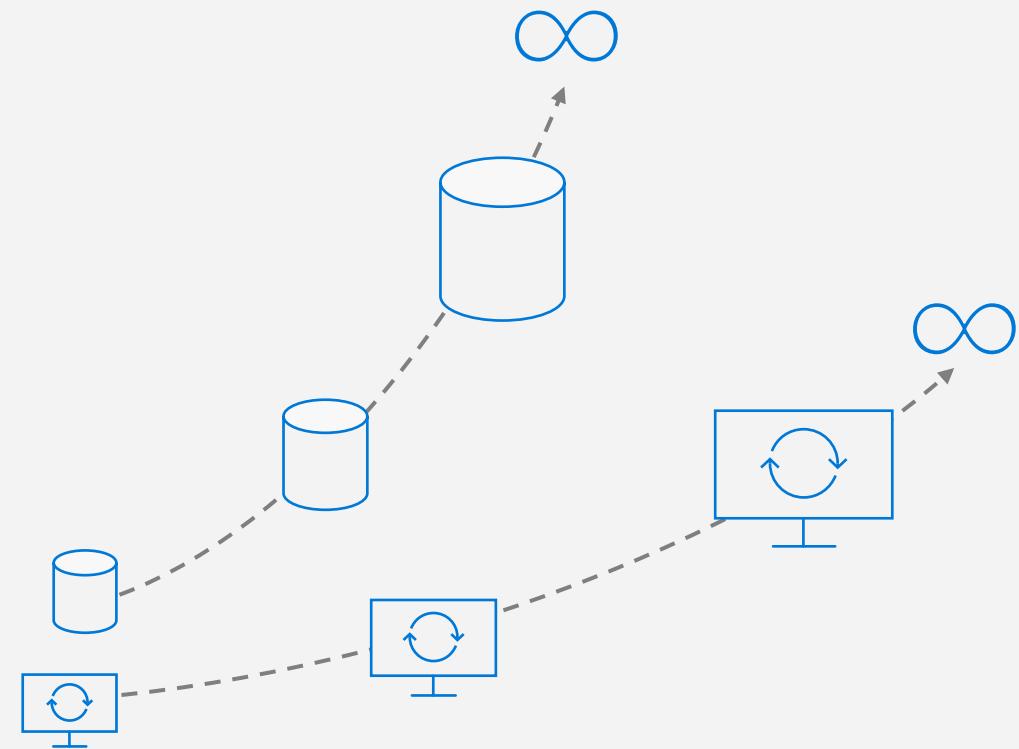
<https://infrastructuremap.microsoft.com>

ストレージとスループットの柔軟なスケールアウト

アプリの要求の変化に合わせてスケール

独立に柔軟にリージョンにまたがってストレージと
スループットをアプリの要求に合わせてスケール -
予測不可能なトラフィックのバースト

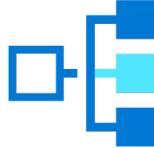
- 複数のリージョンで秒間1,000万から數十億リクエストに対して、柔軟にスループットをスケール
- 異なるワークロードの秒間リクエストをサポート
- 必要とするスループットとストレージに対してのみの支払い



Azure Cosmos DB for PostgreSQL

Azure Cosmos DB for PostgreSQL

完全なマネージドの分散リレーショナルデータベースで、エンタープライズで実証済みの SQL 領域の ACID 特性、グローバル分散、Cosmos DB の柔軟性を併せ持つ



分散 Postgres

単一ノードの PostgreSQL の制約から解放され、[100ノード](#)にスケールアウト



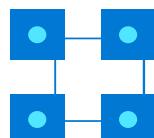
SaaS に最適

スケールする構成、データ分離の双方を、個別のシャードでの詳細な制御により、簡単に実装



グローバル
スケーラビリティー

リージョン間レプリカによるデータのグローバル利用



OLTP と HTAP の汎用的なソースに

時間とコストの削減。[1つの DB](#) でトランザクションと分析を実行。手動でのシャーディングの面倒を回避。



並列化
パフォーマンス

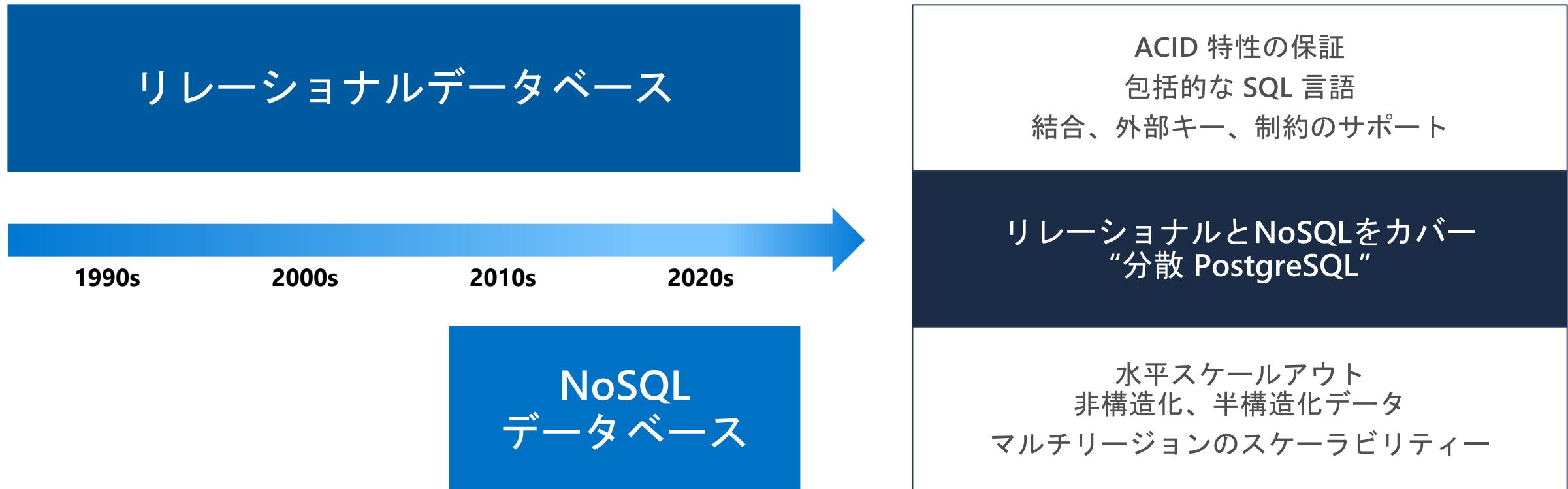
数十億行のデータに対する1秒未満のレスポンスでリアルタイムに DB に投入、クエリー



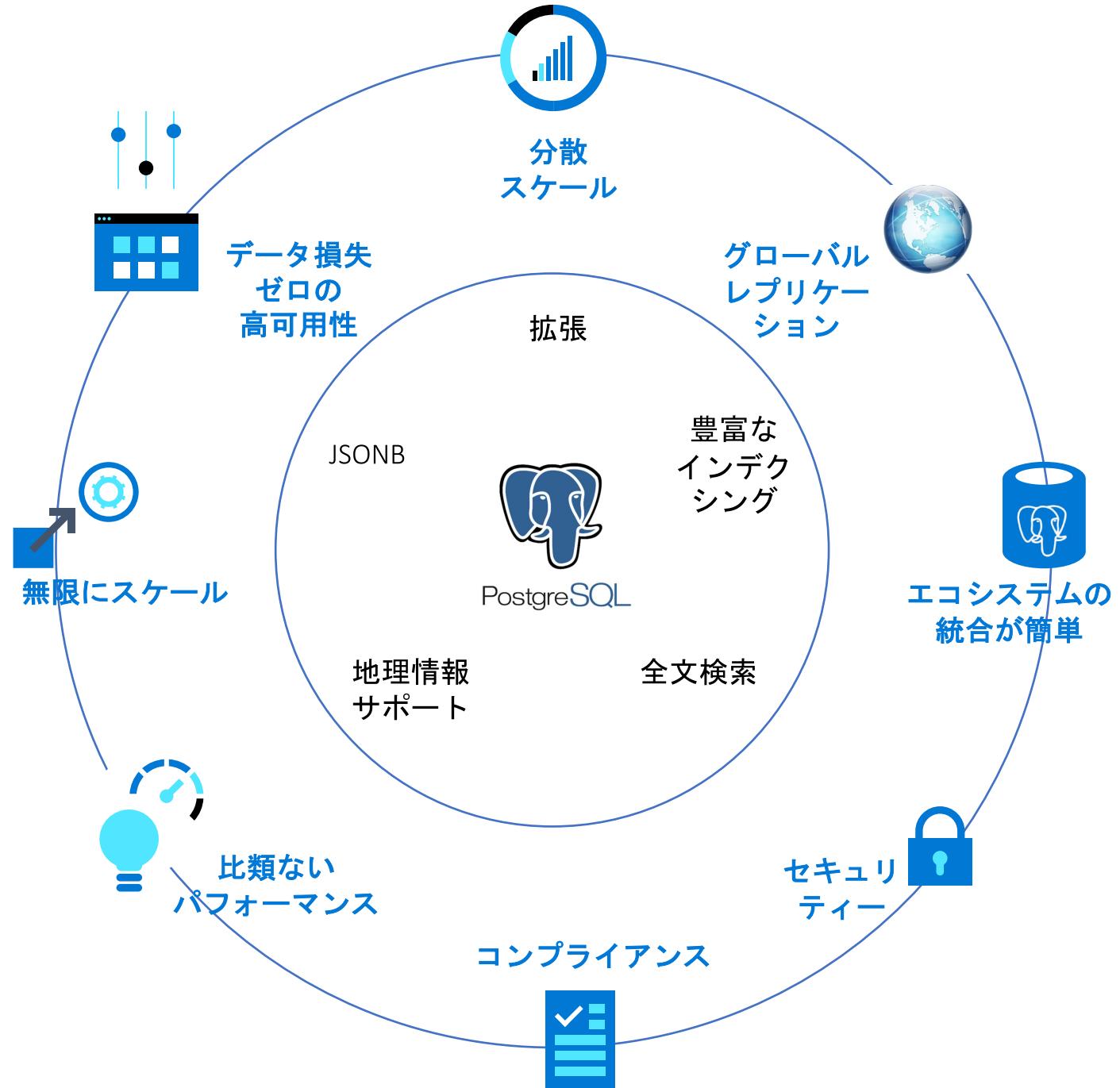
PostgreSQL の革新を追い続ける

オープンソースの拡張（フォークではない）として開発され、PostgreSQL のスキルと最新の機能を活用。

分散 PostgreSQL: リレーショナルと NoSQL の進化



オープンソースの PostgreSQL で構築された 完全マネージドの 分散データベース



運用アプリケーションへの最適化

分析への最適化ではない

データウェアハウスに特化したものではない！



マルチテナント SaaS

- ・ テナントのワークロードを分離
- ・ 詳細な行セキュリティ



IoT

- ・ 時系列のパーティショニング
- ・ 高速なデータ投入
- ・ ストリーミング分析の統合



リアルタイム分析

- ・ 莫大な IOPS
- ・ 列指向ストレージ
- ・ クエリーのスケールアウト



アプリケーション

- ・ 参照整合性
- ・ 制約
- ・ 強力なデータ型

柔軟なクラウドの
コンピュート + ストレージ

Provisioning

Create a Cosmos DB account

Microsoft



Which API best suits your workload?

Azure Cosmos DB is a fully managed NoSQL and relational database service for building scalable, high performance applications. [Learn more](#)

To start, select the API to create a new account. The API selection cannot be changed after account creation.

Azure Cosmos DB for NoSQL

Azure Cosmos DB's core, or native API for working with documents. Supports fast, flexible development with familiar SQL query language and client libraries for .NET, JavaScript, Python, and Java.

[Create](#)[Learn more](#)

Azure Cosmos DB for MongoDB

Fully managed database service for apps written for MongoDB. Recommended if you have existing MongoDB workloads that you plan to migrate to Azure Cosmos DB.

[Create](#)[Learn more](#)

Azure Cosmos DB for PostgreSQL

Fully-managed relational database service for PostgreSQL with distributed query execution, powered by the Citus open source extension. Build new apps on single or multi-node clusters—with support for JSONB, geospatial, rich indexing, and high-performance scale-out.

[Create](#)[Learn more](#)

Azure Cosmos DB for Apache Cassandra

Fully managed Cassandra database service for apps written for Apache Cassandra. Recommended if you have existing Cassandra workloads that you plan to migrate to Azure Cosmos DB.

[Create](#)[Learn more](#)

Azure Cosmos DB for Apache Gremlin

Fully managed graph database service using the Gremlin query language, based on Apache TinkerPop project. Recommended for new workloads that need to store relationships between data.

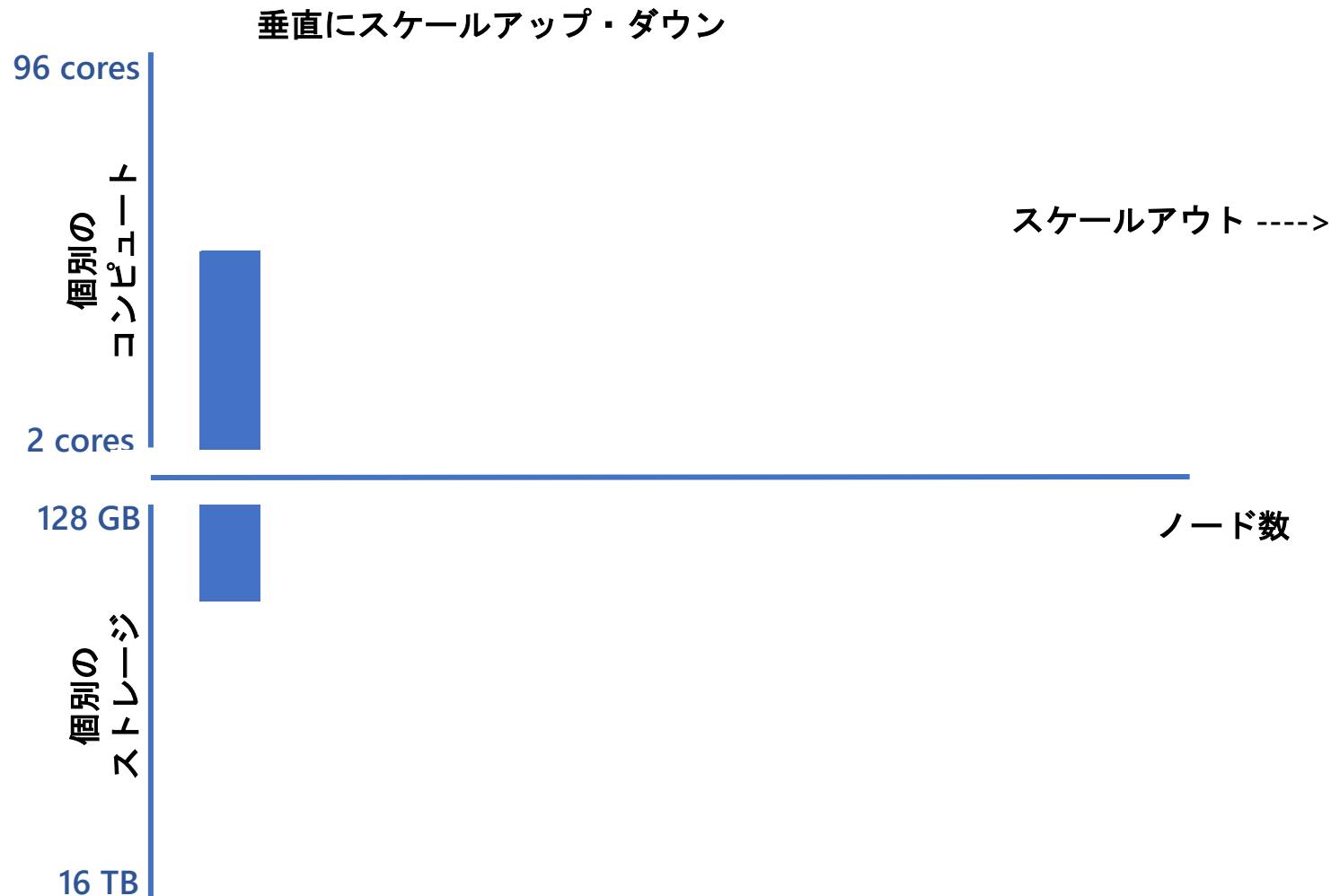
[Create](#)[Learn more](#)

Azure Cosmos DB for Table

Fully managed database service for apps written for Azure Table storage. Recommended if you have existing Azure Table storage workloads that you plan to migrate to Azure Cosmos DB.

[Create](#)[Learn more](#)

柔軟なコンピュートとストレージのオプション

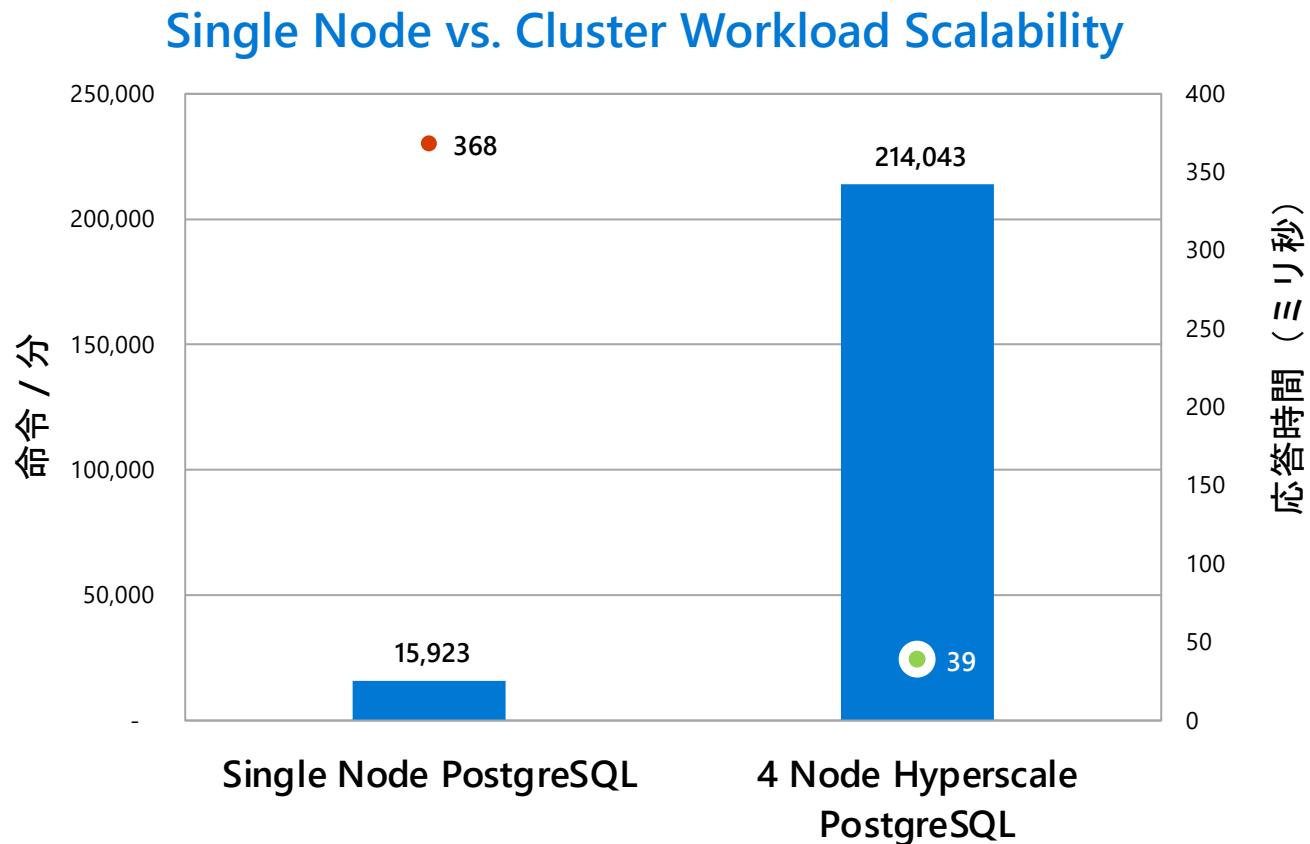


スケールアウトする パフォーマンスの メリット

リニア「以上」に
スループットが改善

- 13倍のスループット
- コストは5倍で収まる

レイテンシーは劇的に減少



Developing



✓ Servers (3)

- cfstdsg
- Databases (2)
 - citus
 - Casts
 - Catalogs
 - Event Triggers
 - Extensions
 - Foreign Data Wrappers
 - Languages
 - Publications
 - Schemas (8)
 - azure_storage
 - citus
 - citus_internal
 - columnar
 - columnar_internal
 - cron
 - partman
 - public
 - Aggregates
 - Collations
 - Domains
 - FTS Configurations
 - FTS Dictionaries
 - FTS Parsers
 - FTS Templates
 - Foreign Tables
 - Functions
 - Materialized Views
 - Operators
 - Procedures

citus/citus@cfstdsg

No limit

Query History

```
5
6 CREATE TABLE EcomSalesTransactions (
7     TenantId int,
8     CustomerId int,
9     TransactionId int,
10    ProductId int,
11    TransactionTime timestamp,
12    UnitPrice decimal(10,2),
13    UnitQuantity int,
14    PromotionCode varchar(20),
15    SalesAmount decimal(10,2),
16    TaxAmount decimal(10,2),
17    PRIMARY KEY (TenantId,CustomerId,TransactionId));
18
19 CREATE INDEX ix_sales
20 ON EcomSalesTransactions (tenantid, transactionid)
21 INCLUDE (customerid, unitprice,unitquantity)
22
```

ADD STORAGE ACCOUNT

Data output Messages Notifications

DROP TABLE

Query returned successfully in 111 msec.

PostgreSQL そのもの

全ての PostgreSQL アプリケーションをサポート

あなたの PostgreSQL の
スキルを活用

ストアドプロシージャー、ト
リガー、拡張、分離レベルの
設定などが「そのまま動く」

The screenshot shows the pgAdmin 4 interface. On the left is the 'Browser' pane, which displays a tree view of database objects. Under 'Extensions (30)', several extensions are listed: btree_gin, btree_gist, citext, citus, cube, dblink, earthdistance, fuzzystrmatch, hll, hstore, intarray, ltree, pg_buffercache, pg_cron, pg_freespacemap, pg_partman, pg_prewarm, pg_stat_statements, pg_trgm, pgcrypto, pgrowlocks, pgstattuple, plpgsql, sslinfo, tablefunc, tdigest, topn, unaccent, uuid-ossp, and xml2. At the bottom of the browser pane, there are sections for 'Foreign Data Wrappers' and 'Languages'. The main area contains a query editor with the following SQL code:

```
1 CREATE TABLE EcomSalesTransactions (
2     TenantId int,
3     CustomerId int,
4     TransactionId int,
5     ProductId int,
6     TransactionTime timestamp,
7     UnitPrice decimal(10,2),
8     UnitQuantity int,
9     PromotionCode varchar(20),
10    SalesAmount decimal(10,2),
11    TaxAmount decimal(10,2),
12    PRIMARY KEY (TenantId,CustomerId,TransactionId));
13
14 SELECT * FROM EcomSalesTransactions LIMIT 10;
15
```

Below the query editor is a 'Data output' tab, which displays a table with 10 rows of transaction data. The columns are: tenantid, customerid, transactionid, productid, transactiontime, unitprice, unitquantity, promotioncode, salesamount, and taxamount.

tenantid	customerid	transactionid	productid	transactiontime	unitprice	unitquantity	promotioncode	salesamount	taxamount	
1	8	53507	791	480	2021-07-14 15:47:58.063	2.29	2	Disc_2021	4.58	0.41
2	8	101007	792	231	2021-07-18 04:15:39.88	49.99	4	Disc_2021	199.96	18.00
3	8	156507	793	231	2021-07-18 04:15:39.88	49.99	4	Disc_2021	199.96	18.00
4	8	119507	794	231	2021-07-18 04:15:39.88	49.99	4	Disc_2021	199.96	18.00
5	8	82507	795	231	2021-07-18 04:15:39.88	49.99	4	Disc_2021	199.96	18.00

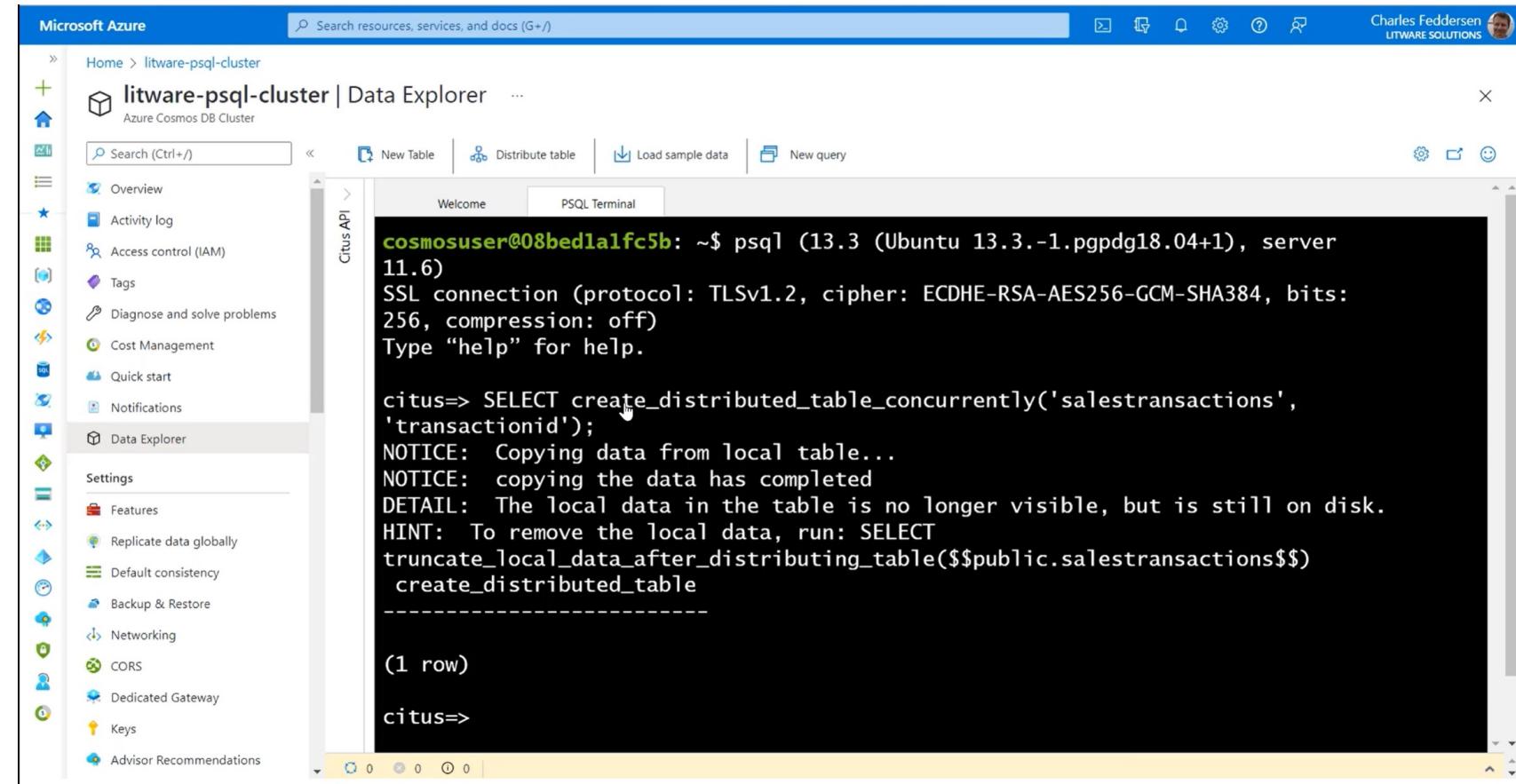
Total rows: 10 of 10 Query complete 00:00:00.878 Ln 8, Col 17

Cosmos DB は 50超の オープンソースの PostgreSQL 拡張を サポート

広く利用される
PostgreSQL 拡張の
業界をリードする
サポート

Category	Extension
Data Types	citext cube hll hstore isn lo ltree seg tdigest topn
Full-text Search	dict_int dict_xsyn unaccent
Index Types	Bloom Btree_gin Btree_gist
Language	plpgsql
PostGIS	postgis_topology Postgis_tiger_geocode Postgis_sfsgal Address_standardizer Address_standardizer_us
Functions	Autoinc Earthdistance Fuzzystrmatch Insert_username Intagg Intarray Moddatetime Pg_partman Pg_trgm Pgcrypto Refint Tablefunc Tcn Timetravel Uuid-ossp
Miscellaneous	Adminpack Amcheck Dblink File_fdw Pageinspect Pg_buffercache Pg_cron Pg_stat_statement

Azure ポータル上の PSQL ターミナル



The screenshot shows the Microsoft Azure portal interface for managing an Azure Cosmos DB cluster named "litware-psql-cluster". The left sidebar provides navigation options like Overview, Activity log, and Data Explorer. The main area displays the "Data Explorer" interface for the Citus API. The "PSQL Terminal" tab is selected, showing a terminal session with the following output:

```
cosmosuser@08bed1a1fc5b: ~$ psql (13.3 (Ubuntu 13.3.-1.pgpdg18.04+1), server 11.6)
SSL connection (protocol: TLSv1.2, cipher: ECDHE-RSA-AES256-GCM-SHA384, bits: 256, compression: off)
Type "help" for help.

citus=> SELECT create_distributed_table_concurrently('salestransactions',
'transactionid');
NOTICE: Copying data from local table...
NOTICE: copying the data has completed
DETAIL: The local data in the table is no longer visible, but is still on disk.
HINT: To remove the local data, run: SELECT
truncate_local_data_after_distributing_table($$public.salestransactions$$)
create_distributed_table
-----
(1 row)

citus=>
```

Scaling

pgAdmin 4

File Object Tools Help

Browser Properties SQL Dependencies Dependents CreateLoadTable... CreateDistributedTable.sql IsolateTenant.sql

Servers (3) cfstdsg Databases (2) citus Casts Catalogs Event Triggers Extensions Foreign Data Wrap Languages Publications Schemas (8) azure_storage citus citus_internal columnar columnar_internal cron partman public Aggregates Collations Domains FTS Configuration FTS Dictionary FTS Parsers FTS Template Foreign Table Functions Materialized View Operators Procedures

citus/citus@cfstdsg

No limit

Query History

1
2 **SELECT * FROM citus_get_active_worker_nodes();**
3
4 **SELECT * FROM citus_shards;** →
5
6 **SELECT COUNT(*) FROM EComSalesTransactions**
7
8 **EXPLAIN SELECT COUNT(*) FROM EComSalesTransactions**
9
10 -- START APP
11
12 **SELECT COUNT(*) FROM EComSalesTransactions**

Data output Messages Notifications

QUERY PLAN text

Aggregate (cost=250.00..250.02 rows=1 width=0)
→ Custom Scan (Citus Adaptive) (cost=0.0...
Task Count: 32
Tasks Shown: One of 32
→ Task
Node: host=private-c.cfstdsg.postgres.dat...
→ Aggregate (cost=314.89..314.90 rows=1 width=0)
→ Seq Scan on ecomsalestransactions_10...

Total rows: 8 of 8 Query complete 00:00:00.191 Ln 12, Col 41

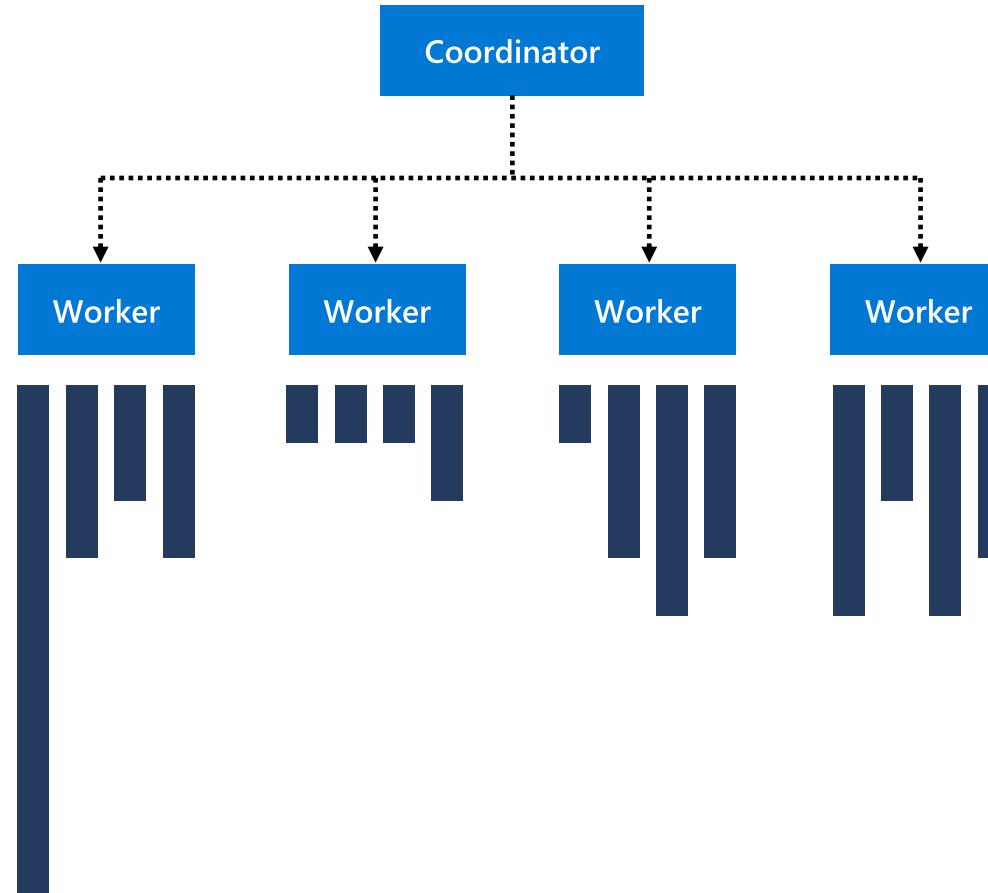
オンラインの テーブルのスケールアウト

ローカルの PostgreSQL から
シャードしたクラスターへの
スケールアウト

実行中のアプリへの影響は無
し、アプリの修正も必要無し

```
CREATE TABLE Orders (
    order_id int,
    customer_id int,
    region_name varchar(20),
    amount decimal(10,2))
```

```
SELECT
create_distributed_table('Orders', 'c
ustomer_id')
```



データの分散

```
/* Create Postgres table */
```

```
CREATE TABLE SalesTxn
(
    SalesTxnID      int
    CustomerID      bigint
    Date            datetime
    Amount          decimal(19,4)
)
```

```
/* Inbuilt Automatic Online Sharding */
```

```
SELECT create_distributed_table_concurrently
(
    'SalesTxn' , -- Table_name
    'CustomerID' -- Shard Key
)
```

SalesTxn	
SalesTxnID	int
CustomerID	bigint
Date	datetime
Amount	decimal(19,4)

コーディネーターに作成された
ローカルテーブル

SalesTxn_111	
SalesTxnID	int
CustomerID	bigint
Date	datetime
Amount	decimal(19,4)

SalesTxn_112	
SalesTxnID	int
CustomerID	bigint
Date	datetime
Amount	decimal(19,4)

SalesTxn_113	
SalesTxnID	int
CustomerID	bigint
Date	datetime
Amount	decimal(19,4)

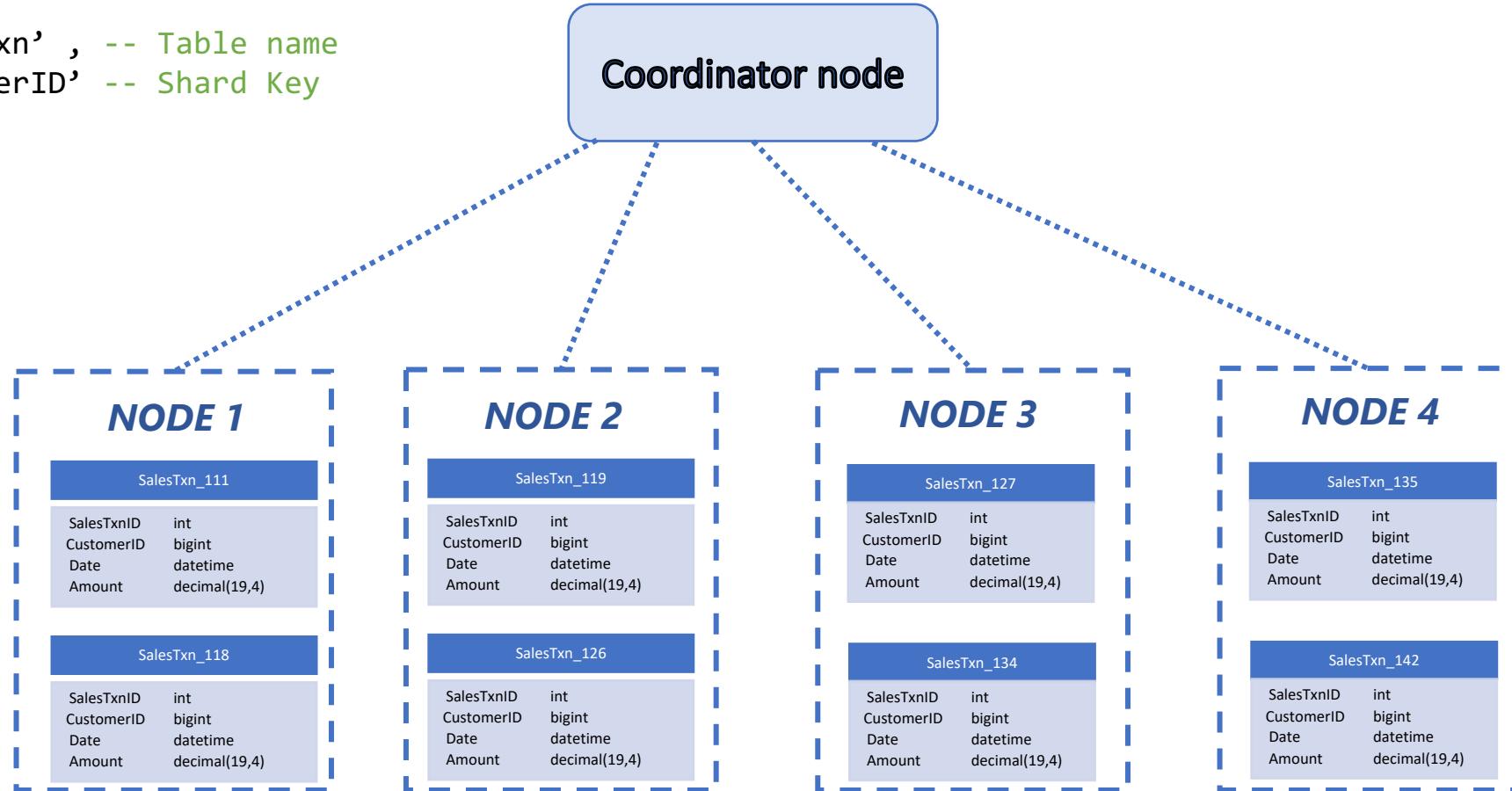
SalesTxn_114	
SalesTxnID	int
CustomerID	bigint
Date	datetime
Amount	decimal(19,4)

.....

SalesTxn_142	
SalesTxnID	int
CustomerID	bigint
Date	datetime
Amount	decimal(19,4)

データ分散(ノードレベル)

```
SELECT create_distributed_table_concurrently
(
    'SalesTxn' , -- Table name
    'CustomerID' -- Shard Key
)
```

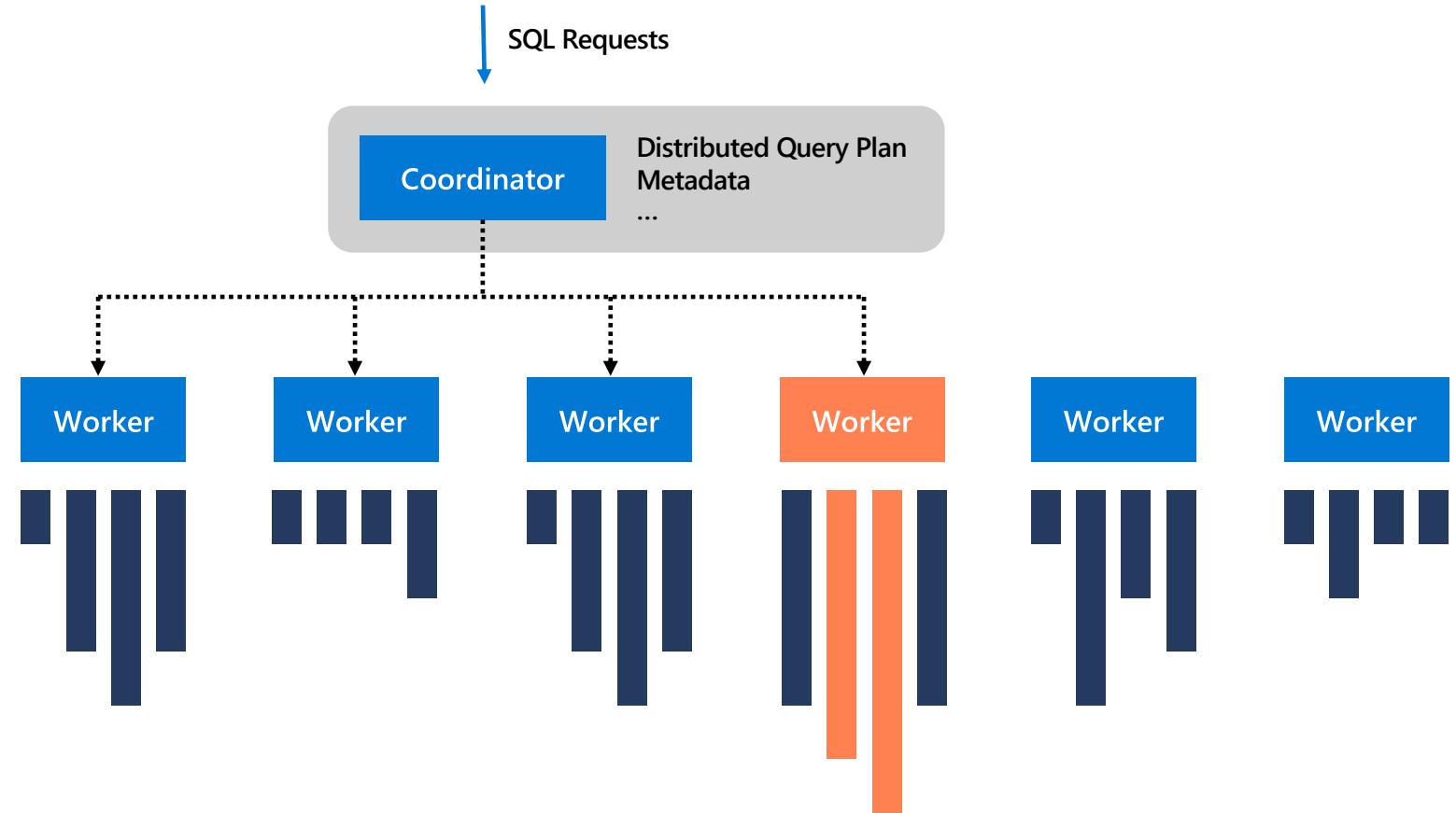


Distribution¥Shard key – カラム (上図では **CustomerID**) がノード間のデータ分散に使われる

シャードリバランサー

クラスター内でシャードを再分配する単一のコマンド

クラスターをより効率的に利用することで、価格を変えずにパフォーマンスを改善



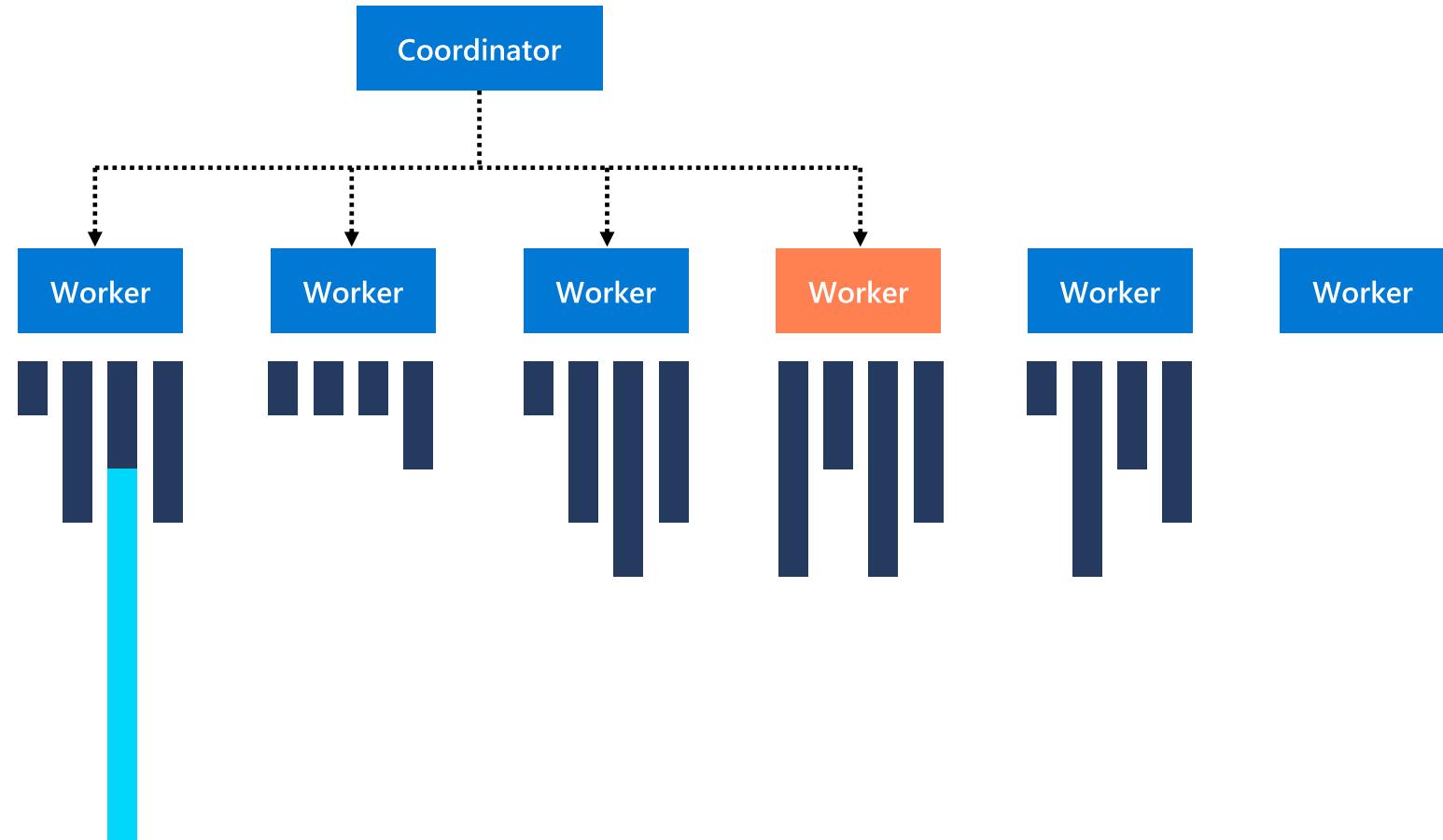
オンライン テナント分離

SaaS のワークフロー向け設計

新しいノードにオンラインで
テナントを移動することを可
能に

大きなあるいは高負荷のテナ
ントを専用のノードに分離し
パフォーマンスを最大化

アプリのコード修正やダウン
タイムは不要



UK COVID-19 Dashboard

COVID-19ダッシュボード - UK

<https://coronavirus.data.gov.uk>

「大臣や科学者は一般人より先に個々のデータセットを見ることができますが、ダッシュボード自体は真に民主化されたオープンアクセステータの例です。

ニューカッスルの自宅に座っている人は、ダウニングストリートのオフィスにいるボリス・ジョンソン(当時、首相)と同じ瞬間、つまりデータが更新される午後4時に初めて最新のトレンドとグラフを見ることが可能です。」

- 75億レコード
- 150万ユーザー/日
- ピーク時に毎分8.5~10万ユーザーが利用
- 16vCPU/2TB SSD x 12ワークロード
- 64vCPUコーディネーターノード

<https://techcommunity.microsoft.com/t5/azure-database-for-postgresql/uk-covid-19-dashboard-built-using-postgres-and-citus-for/ba-p/3036276>

Last updated on Thursday, 13 October 2022 at 4:00pm

Daily update

England Summary

The official UK government website for data and insights on coronavirus (COVID-19).

See the [simple summary](#) for England.

[Testing](#)[Cases](#)[Healthcare](#)[Vaccinations](#)[Deaths](#)[Interactive maps](#)[Metrics documentation](#)[Download data](#)[What's new](#)[Developer's guide](#)[About](#)

Vaccinations

People vaccinated in England

NATION

Up to and including 9 October 2022

Last 7 days – first dose

12,418

Last 7 days – second dose

18,538

Last 7 days – booster or third dose

28,444

#1

Percentage of population aged 12+

93.6%

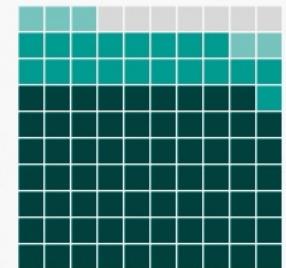
88.3%

69.5%

First dose

Second dose

Booster or third dose



Total – first dose

45,288,388

Total – second dose

42,737,455

Total – booster or third dose

33,641,741[All vaccinations data in England](#)

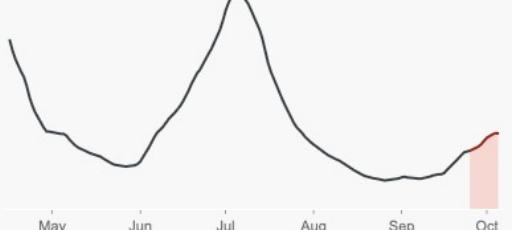
Cases

People tested positive in England

NATION

Up to and including 8 October 2022

Last 7 days

61,809 ↑ 10,763 (21.1%)[All cases data in England](#)

#2

Deaths

Deaths within 28 days of positive test in England

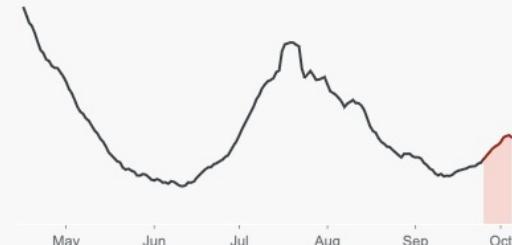
NATION

Up to and including 8 October 2022

Last 7 days

631 ↑ 82 (14.9%)

► Rate per 100,000 people: 1.1

[All deaths data in England](#)

#3

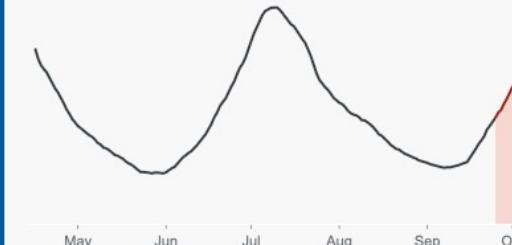
Healthcare

Patients admitted in England

NATION

Up to and including 10 October 2022

Last 7 days

8,198 ↑ 294 (3.7%)[All healthcare data in England](#)

#4

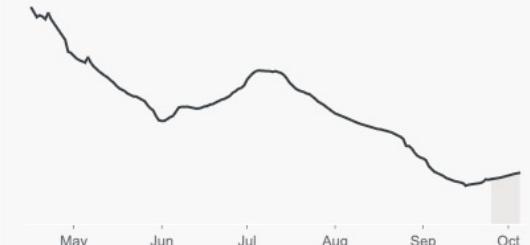
Testing

Virus tests conducted in England

NATION

Up to and including 12 October 2022

Last 7 days

499,251 ↑ 10,079 (2.1%)[All testing data in England](#)

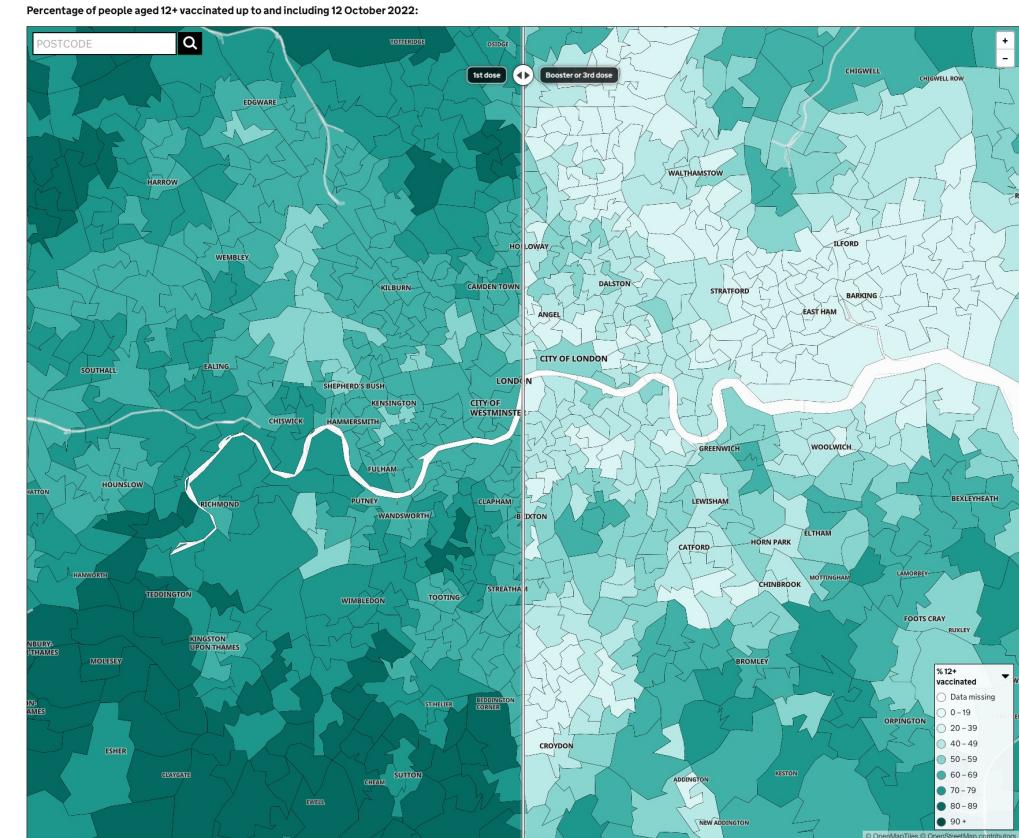
#5

1,500人から8万人に

ダッシュボードがリリースされた直後：1,500ユーザー
首相のツイート後：8万ユーザー

ワクチン接種者数の対話型地図

- 1, 2, 3回目の接種者比率を地図上で比較できる
- マップとのやり取りは1秒未満で2～3回のクエリーが実行される

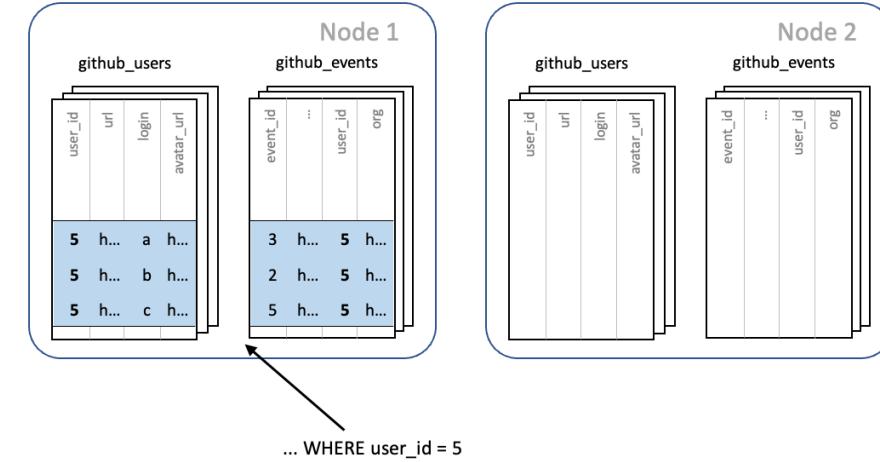


英国がCosmos DB for PostgreSQLを選択した理由

- RDBMSでありながらJSON/JSONBも格納できる
- ORマッパーのサポートがある
- 高可用性（HA）構成
- Azureの各種サービスとの連携
 - Azure Functions
 - Azure Cache for Redis
 - Azure Front Door
- スケーラビリティー

Cosmos DB for PostgreSQLの「コツ」

- 適切なクラスターサイズ
 - 例：4vCPU x 8ノード vs 8vCPU x 4ノード
 - コーディネーターのvCPU数
- 適切なシャードキーの選択
 - ノード跨ぎのJOINが発生しないこと
 - ワーカーノードに平均的に分散すること
- 時系列データの場合はパーティションを活用
 - PostgreSQLネイティブではなく、Citusが提供するパーティション機能を利用
- 列形式ストレージによるデータ容量の削減
 - USING COLUMNARキーワード
 - INSERT INTO table_name (col1, col2...) VALUES ('a', 1...), ('b', 2...), ('c', 3...)



Call To Action!



Azure Cosmos DB for PostgreSQL

General Availability

無料で試せます！ aka.ms/trycosmosdb