AWS re:Invent

NOV. 29 – DEC. 3, 2021 | LAS VEGAS, NV

DAT312

Deep dive on Babelfish for Amazon Aurora PostgreSQL

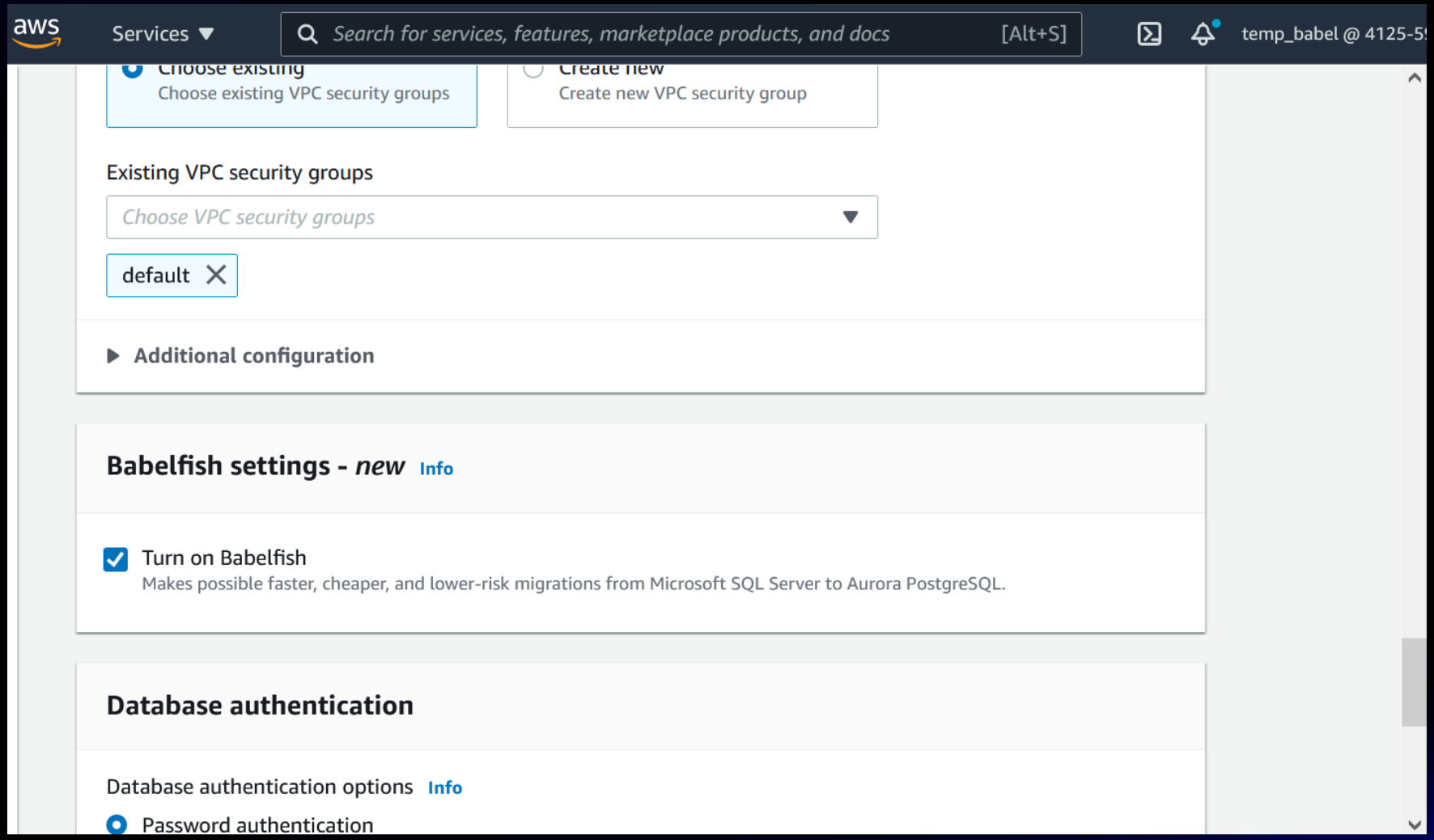
Simon Lightstone
Product Manager
AWS

Suprio Pal
Software Development Manager
AWS

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Introducing Babelfish

THE EASIEST AND FASTEST WAY TO MIGRATE FROM MICROSOFT SQL SERVER TO AURORA POSTGRESQL



© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Goal: Broad overview

BABELFISH SESSIONS

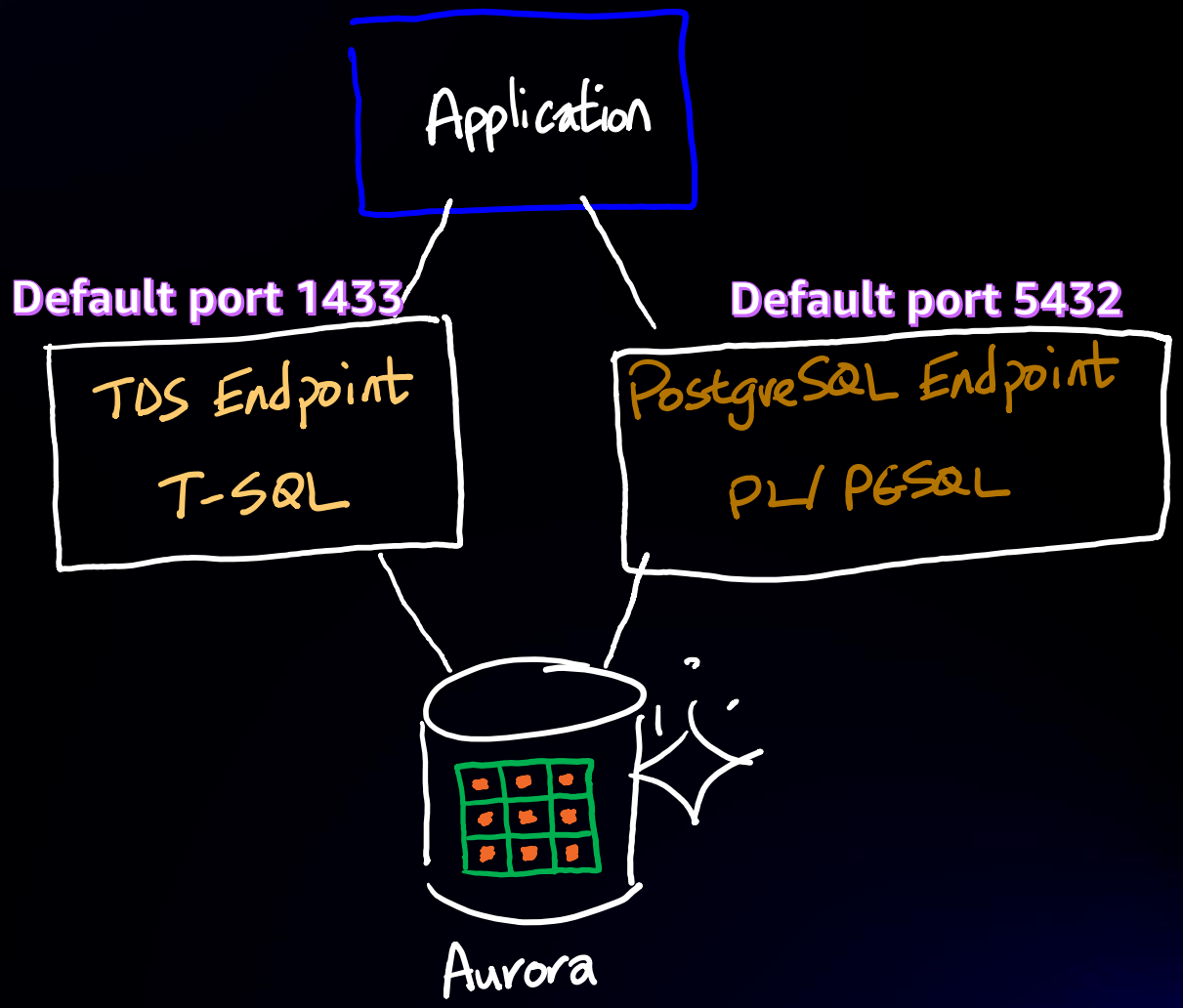
DAT312	This session – deep dive on Babelfish
DAT325	Workshop – SQL server migration
OPN202-R1	Babel architecture and design
OPN315	Build an Aurora PostgreSQL DB with Babelfish via open source



© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Connect to both ports at once

YOUR APP CAN CONNECT VIA POSTGRESQL AND TDS AT ONCE, ALLOWING INCREMENTAL MIGRATIONS



© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Support for . . .

SQL SERVER-SPECIFIC FEATURES

- Triggers, stored procedures, scalar SQL functions, and views
- T-SQL transactional semantics, including nested transactions and savepoints
- Data types (**money**, **sql_variant**, 3-millisecond **datetime**)
- Control-of-flow statements (e.g., GOTO, TRY/CATCH)
- Table data types, table parameters, and table-valued functions
- Static cursors
- Computed columns
- Dynamic SQL (**EXEC with string parameter**) and **sp_executesql**
- Application locks
- SELECT . . . FOR XML {RAW | PATH}
- **And much more . . .**



aws

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Migrate at your own pace

AMSTERDAM



© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

“Refactor” into a simpler language



© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Move to the new language over time



© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Not a translator – it naturally understands



© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Leave T-SQL quickly or stay on it forever



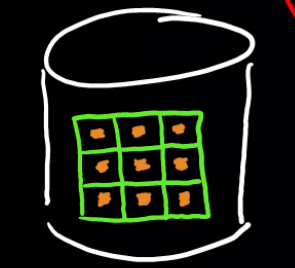
© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Application correctness

```
--example T-SQL application syntax

SELECT ProductID, ProductName, Price
   FROM dbo.Products
   WHERE Price < 30
```

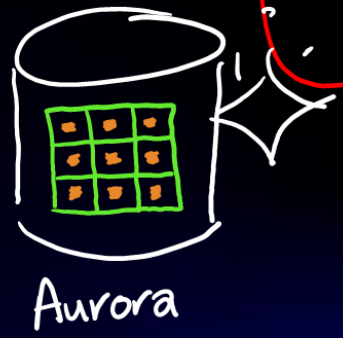
ProductID	ProductName	Price
1	Clamp	$12.8182




```
--example PGSQL application syntax

SELECT ProductID, ProductName, Price
   FROM Products
   WHERE Price < 30
```

ProductID	ProductName	Price
1	Clamp	$12.82



© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Transactional semantics: SQL Server

```
1> create table t1 (a int not null unique)
2> go
1> begin tran
2> insert into t1 values(1)
3> insert into t1 values(2)
4> update t1 set a = 3  -- sets all values to 3. Causes duplicate key error
5> commit
6> go

Msg 2627, Level 14, State 1, Server EC2AMAZ-5Q6FMIK, Line 4Violation of UNIQUE KEY constraint
'UQ__t1__3BD0198F21AFC10E'. Cannot insert duplicate key in object 'dbo.t1'. The duplicate key
value is (3).The statement has been terminated.

1> select * from t1
2> go
a
-----------
          1
          2
(2 rows affected)
```

By default, SQL Server keeps the transaction open and rolls back only the statement causing the error

Result in SQL Server: 2 rows

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Transactional semantics: Aurora PostgreSQL

```
postgres=> create table t1 (a int not null unique);
CREATE TABLE
postgres=> DO $$
postgres$> begin
postgres$> insert into t1 values (1);
postgres$> insert into t1 values (2);
postgres$> update t1 set a = 3;   -- Sets all values to 3. Causes duplicate key error
postgres$> commit;
postgres$> end$$;

ERROR:  duplicate key value violates unique constraint "t1_a_key"
DETAIL:  Key (a)=(3) already exists.
CONTEXT:  SQL statement "update t1 set a = 3"
PL/pgSQL function inline_code_block line 5 at SQL statement


postgres=> select * from t1;
a
---
(0 rows)
```

PG rolls back the entire transaction

Result in PG: 0 rows

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Error/exception handling: SQL Server

```
1> create table t2 (a int not null unique)
2> go
1> begin tran
2> insert into t2 values (123)
3> insert into t2 values (123) -- will cause duplicate key error
4> insert into t2 values (456)
5> commit
6> print 'more processing here!'
7> go

Msg 2627, Level 14, State 1, Server EC2AMAZ-EULFEOJ, Line 4
Violation of UNIQUE KEY constraint 'UQ__t2__3BD0198F04E3D22F'. Cannot insert duplicate key in object
'dbo.t2'. The duplicate key value is (123).
The statement has been terminated.

more processing here!

1> select * from t2
2> go
a
-----------
        123
        456
(2 rows affected)
```

By default, SQL Server continues with the next statement after an error; to take action, user must check @@ERROR explicitly

Result in SQL Server: after error, subsequent statements are still executed (by default)

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Error/exception handling: Aurora PostgreSQL

```
postgres=> create table t2 (a int not null unique);
CREATE TABLE
postgres=> DO $$
postgres$> begin
postgres$> insert into t2 values (123);
postgres$> insert into t2 values (123); -- will cause duplicate key error
postgres$> insert into t2 values (456);
postgres$> commit;
postgres$> raise notice 'more processing here!';
postgres$> end;
postgres$> $$;
```

When an error occurs, PostgreSQL jumps to the exception handler and leaves the block; subsequent statements are not executed



Not executed

```
ERROR: duplicate key value violates unique constraint "t2_a_key"
DETAIL: Key (a)=(123) already exists.
CONTEXT: SQL statement "insert into t2 values (123)"
PL/pgSQL function inline_code_block line 4 at SQL statement
```

```
postgres=> select * from t2;
a
---
(0 rows)
```

aws

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Easy to migrate

EVEN IF YOUR CAPS LOCK KEY IS STUCK



© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Babelfish for Aurora PostgreSQL

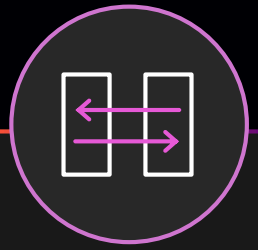
OPEN-SOURCE PROJECT FOR BABELFISH SOURCE CODE – HTTPS://BABELFISHPG.ORG/

Customize and add new features



Contribute to help steer the direction of Babelfish

Apache 2.0 license



Use it for any purpose – innovate and distribute your modifications with confidence

Available on GitHub



Is community-driven and provides transparency into the feature roadmap

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Babelfish: Aurora PostgreSQL extensions

BABELFISH IS AURORA POSTGRESQL EXTENSIONS

- Babelfishpg_money
- Babelfishpg_common
- Babelfishpg_tds
- Babelfishpg_tsql

+ Minor changes to the Aurora PostgreSQL engine, which we hope to merge

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Migration assessment tool

Determines what part of your app/DDL is not supported by Babelfish

Option 1: new open-source tool – Babelfish Compass

- Stand-alone, command-line tool
- GitHub download | User guide

Option 2: Use AWS SCT assessment feature for Babelfish assessment

- NB: Not for SQL code conversion, only for assessment

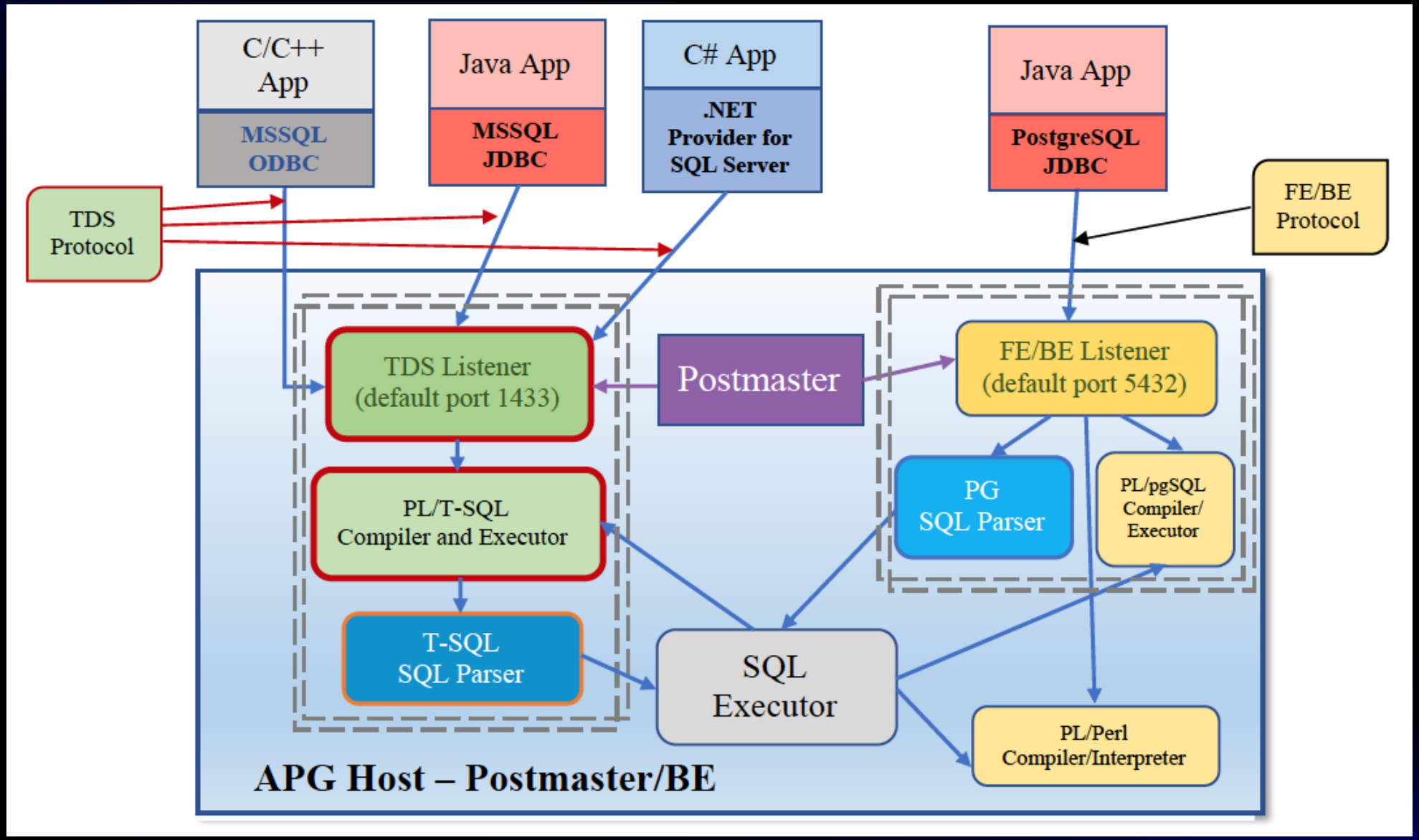
© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Dimensions of Databases (Basics)

1. Data Organization
2. Data Types
3. Query Languages
4. Procedural Languages
5. High-level Languages
6. Data Access APIs and Frameworks
7. Data Consistency
8. Database Metadata
9. Database Administration

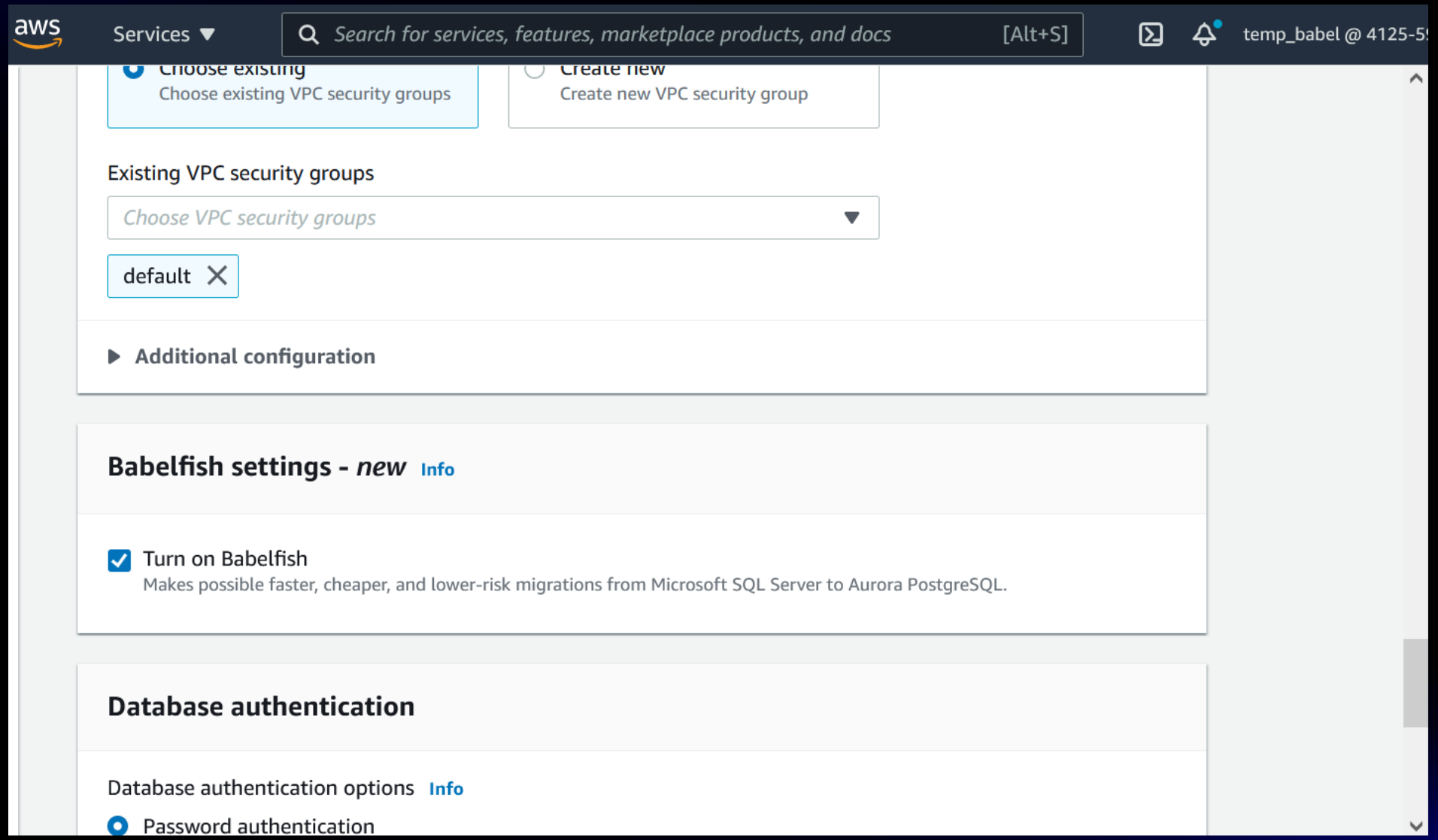
© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

The Runtime



© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

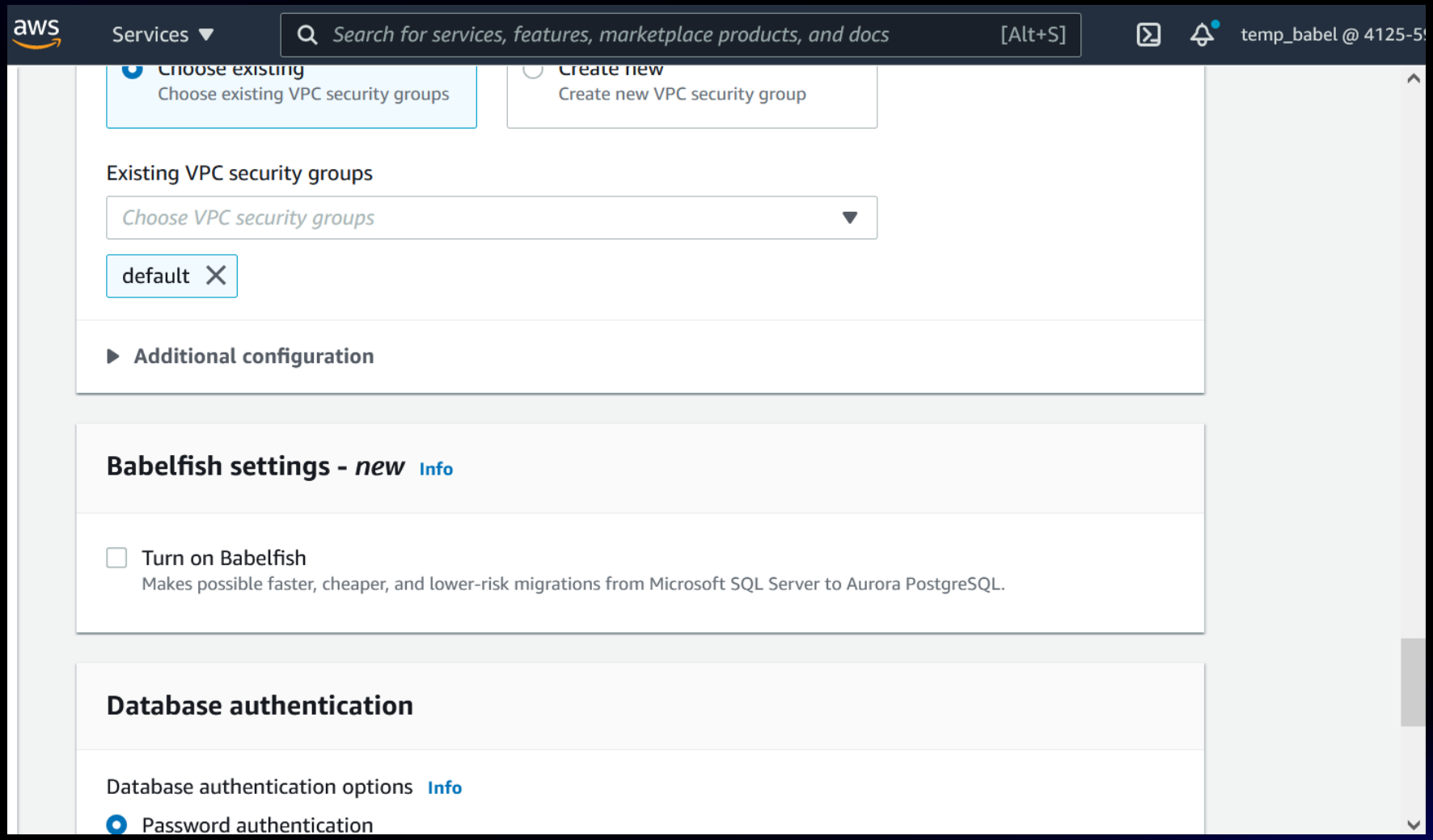
Deployment demo



© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

It’s easy

CREATE A DATABASE, TURN ON BABELFISH, AND YOU’RE READY



© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Thank you!

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.