Seamless Transition:

Migrating from commercial to open source databases

SQL Server to Aurora PostgreSQL with Babelfish

Alvaro Costa-Neto
Sr. Database Specialist SA, AWS

Minesh Chande
Sr. Database Specialist SA, AWS

Agenda

- SQL Server Migration Options
- Purpose-built Databases
- Move to Opensource
- Babelfish
- How does it works?
- Adoption
- Migration using Bablefish
- Demo
- QA

SQL Server Migration Options

6 Rs of application migration

- Retain (don’t move)
- Retire (decommission)
- Repurchase (move to software as a service)

- Rehost (lift and shift)
- Replatform (move to managed)
- Refactor (convert and/or rearchitect)

SQL Server options on AWS

Rehost to Amazon EC2

"Lift and shift"

Managed physical infrastructure, OS installation, and scaling

OS-level control

Linux support

Replatform to Amazon RDS

Fully managed with single click high availability, auto-scaled storage, and automated backups

Business innovation focused

Refactor to purpose-built databases

Eliminate SQL Server licensing costs

Broadest selection of AWS purpose-built databases

Performance and availability of commercial-grade databases at 1/10th the cost

Purpose-built Databases

AWS-managed database services

Amazon RDS

Amazon Aurora

KEY-VALUE

Amazon DynamoDB

DOCUMENT

Amazon DocumentDB

CACHING

Amazon ElastiCache

GRAPH

Amazon Neptune

TIME-SERIES

Amazon Timestream

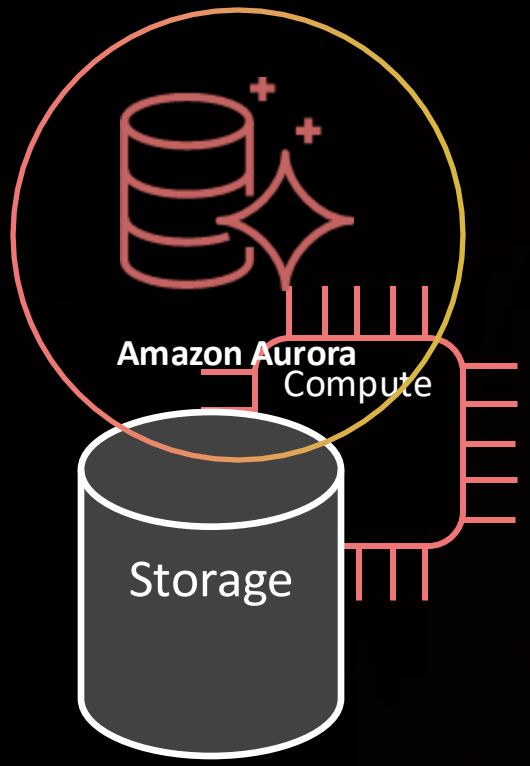
MEMORY

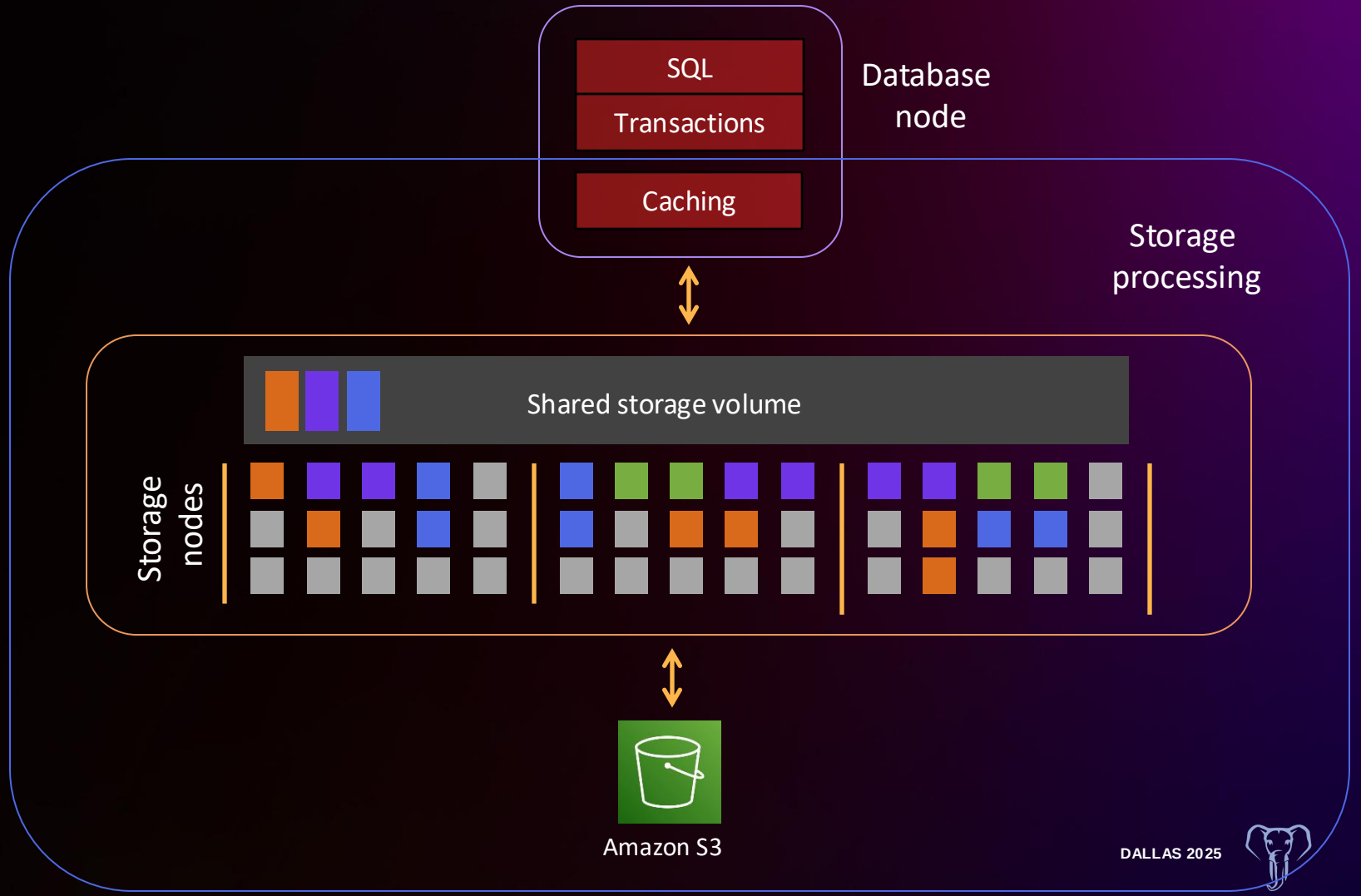
Amazon MemoryDB

WIDE COLUMN

Amazon Keyspaces

Aurora architecture





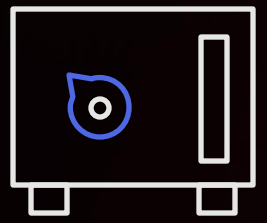
Move to Opensource

Why Customers are Migrating to Opensource

BREAK FREE FROM LEGACY DATABASES

Costly

Proprietary



Lock-in

Punitive licensing

You’ve got mail

Challenges in migrating from commercial to open source

Application

TDS Endpoint
T-SQL

Microsoft SQL Server

Previously…
Rewrite T-SQL
Rewrite app code
Switch drivers

Tools

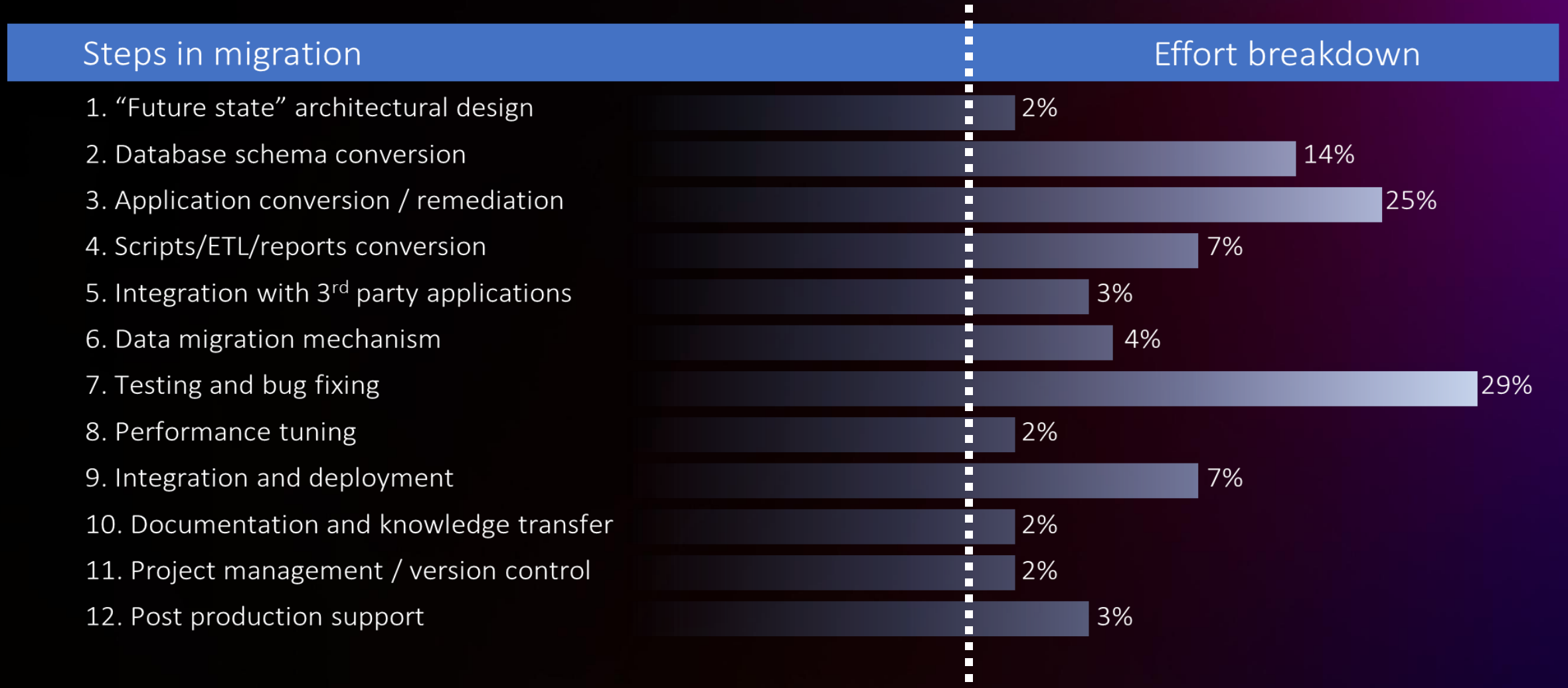
1. AWS Schema Conversion Tool
2. AWS Database Migration Service (DMS)

Application

Postgre Endpoint
PL/pgSQL

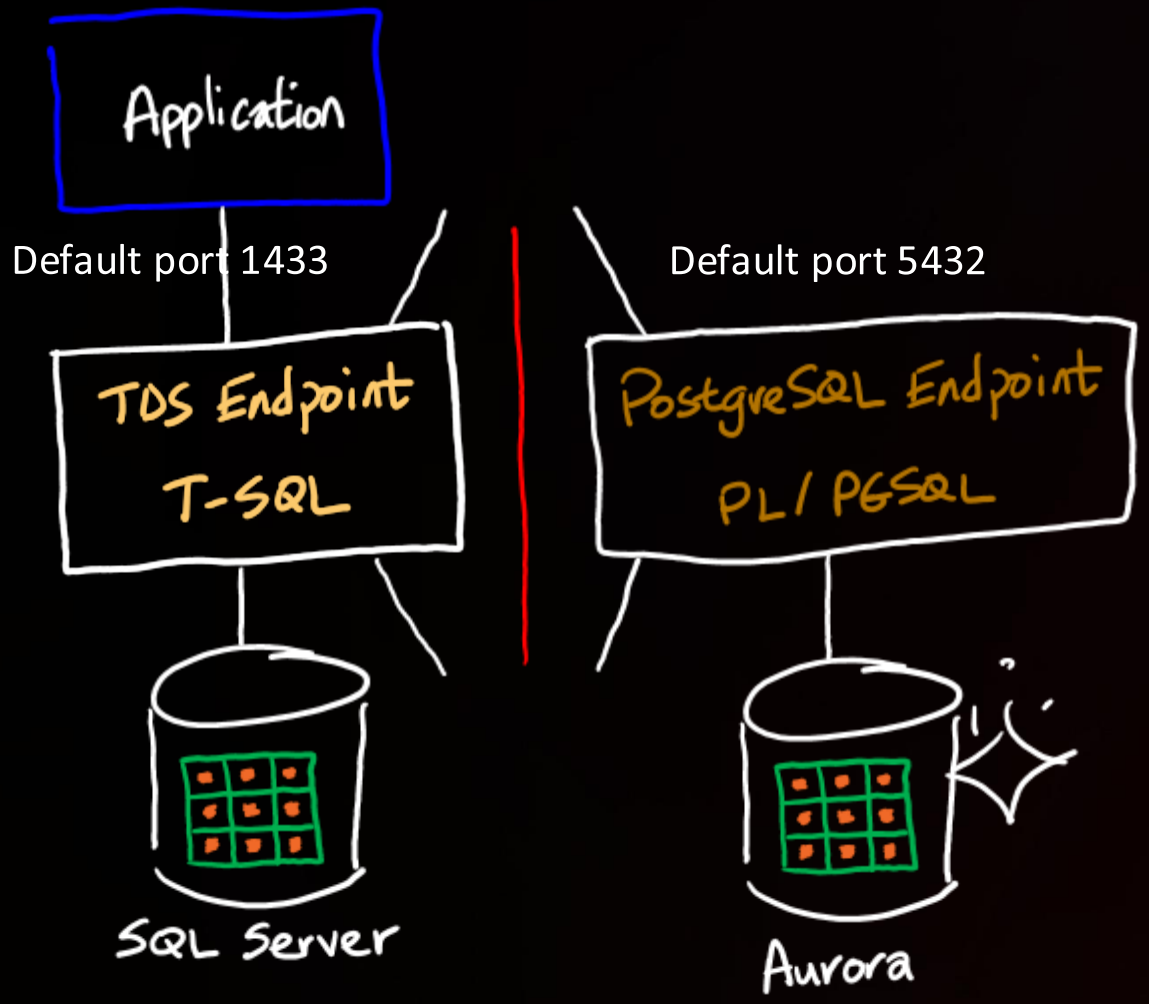
Amazon Aurora PostgreSQL

Overview of a typical migration effort



Babelfish

Imagine if you could . . .



1) A major portion of your existing application code remains in T-SQL
2) No need to change client drivers
3) Write new application code with T-SQL or PL/PGSQL

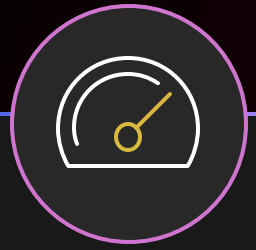
Introducing Babelfish for Aurora PostgreSQL

Migrate SQL Server applications to PostgreSQL in a fraction of the time, compared to a traditional migration

Keep existing queries

Language extension enables Aurora PostgreSQL to understand Microsoft SQL Server's proprietary T-SQL

Accelerate migrations



Lower risk and complete migrations faster, saving you months to years of work

Freedom to innovate

Run T-SQL code side-by-side with open source functionality and continue developing with familiar tools

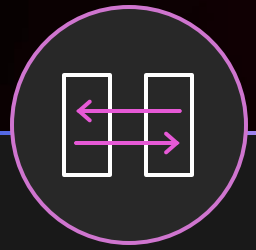
Babelfish for PostgreSQL (Open-Source Project)

An open-source project for Babelfish source code

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

What is Babelfish?

Babelfish is:

- *“Babelfish is a migration accelerator providing semantically correct execution of T-SQL over the TDS protocol, natively implemented in PostgreSQL.”*
- A native implementation of TDS and T-SQL, using PG building blocks
- A PostgreSQL extension (in fact, 3 extensions)
- A second endpoint in an Aurora cluster (TDS + PG ports)
- Open-source

Babelfish is not:

- A SQL ‘mapping’ proxy between the client app and PG
- A separate server
- A temporary solution for customers
- Replacing PG

Babelfish for PostgreSQL design tenets

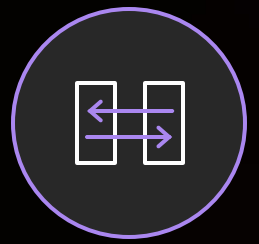
GUIDING PRINCIPLES

No compromises on correctness

Database calls either work or return an error

Wire protocol compatibility

Applications work without changing database drivers



Interoperability

Use both the PostgreSQL port and TDS port for your development

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

Transactional semantics: SQL Server vs. PostgreSQL

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
Babelfish behaves in the same way (when using the TDS port)

Transactional semantics: SQL Server vs. PostgreSQL

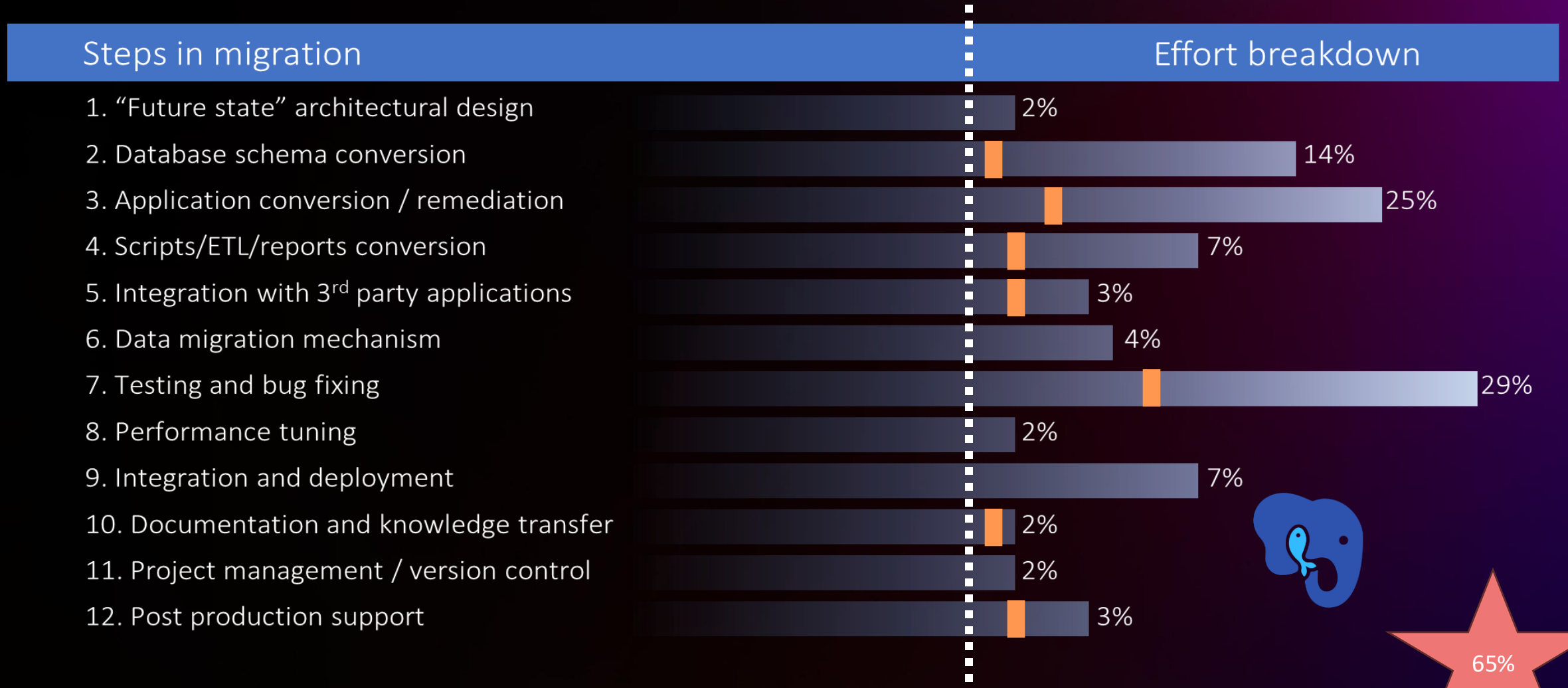
PostgreSQL:

```
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
When connecting over the PostgreSQL port, this behavior is maintained.

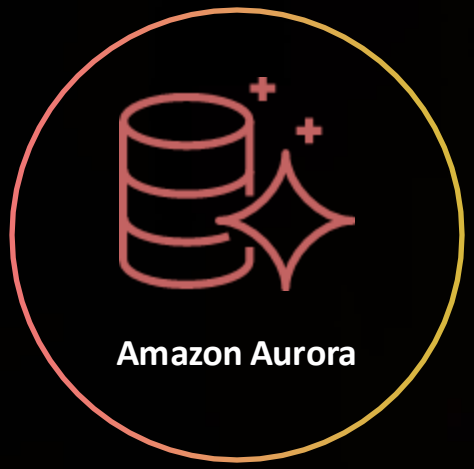
Reducing the migration costs with Babelfish



Amazon Aurora Advantages

An enterprise database at open-source prices

Babelfish includes Aurora's powerful capabilities



Serverless V2 allows automatic scaling of your workload

High availability using read replicas

Integration with AWS services such as S3, Lambda, Cloudwatch

Storage capabilities such as Dynamic Resizing, Clone & Snapshot

How does it work?

Babelfish for PostgreSQL (Open-Source Project)

Migrate SQL Server applications to PostgreSQL in a fraction of the time, compared to a traditional migration

Keep existing queries

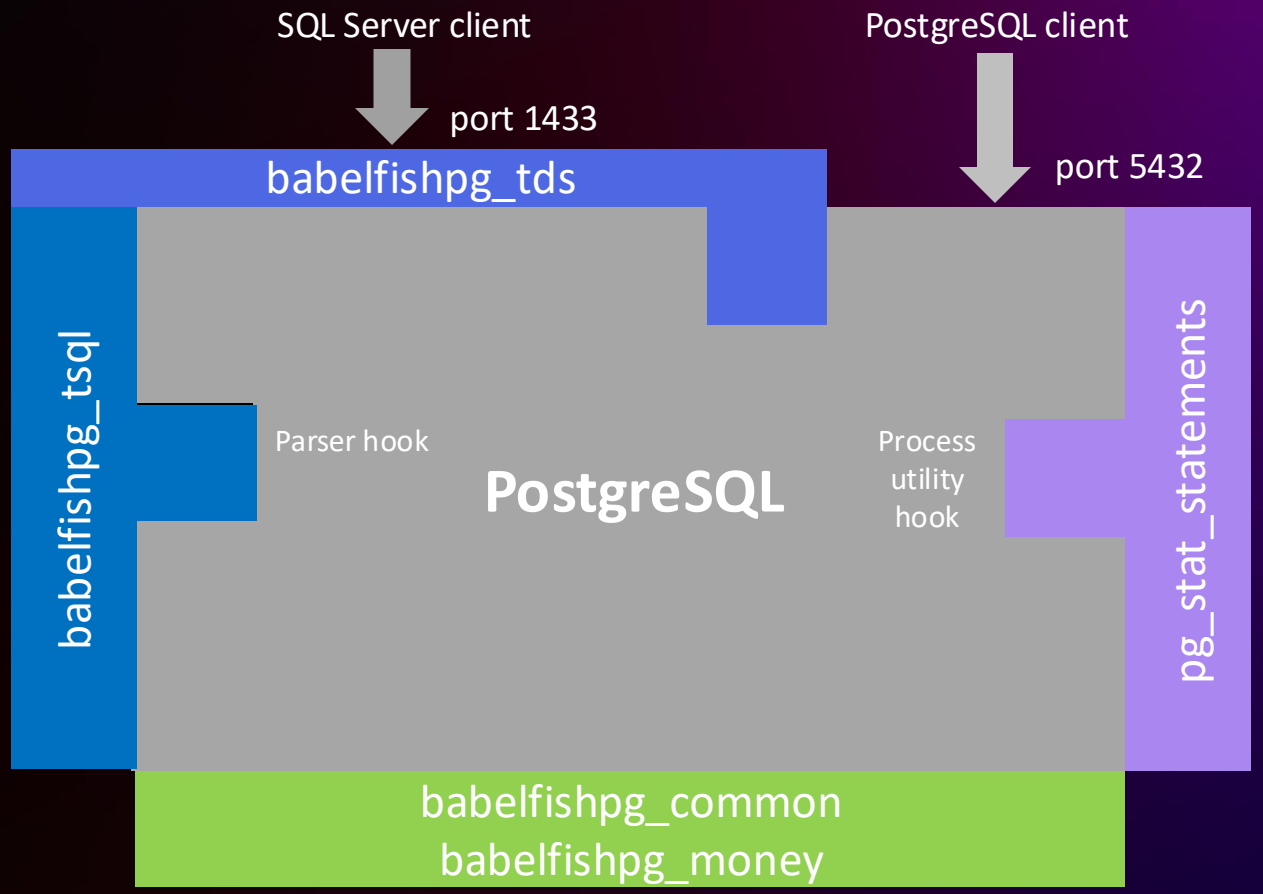
Language extension enables Aurora PostgreSQL to understand Microsoft SQL Server's proprietary T-SQL

Accelerate migrations

Lower risk and complete migrations faster, saving you months to years of work

Freedom to innovate

Run T-SQL code side-by-side with open source functionality and continue developing with familiar tools



Support for...

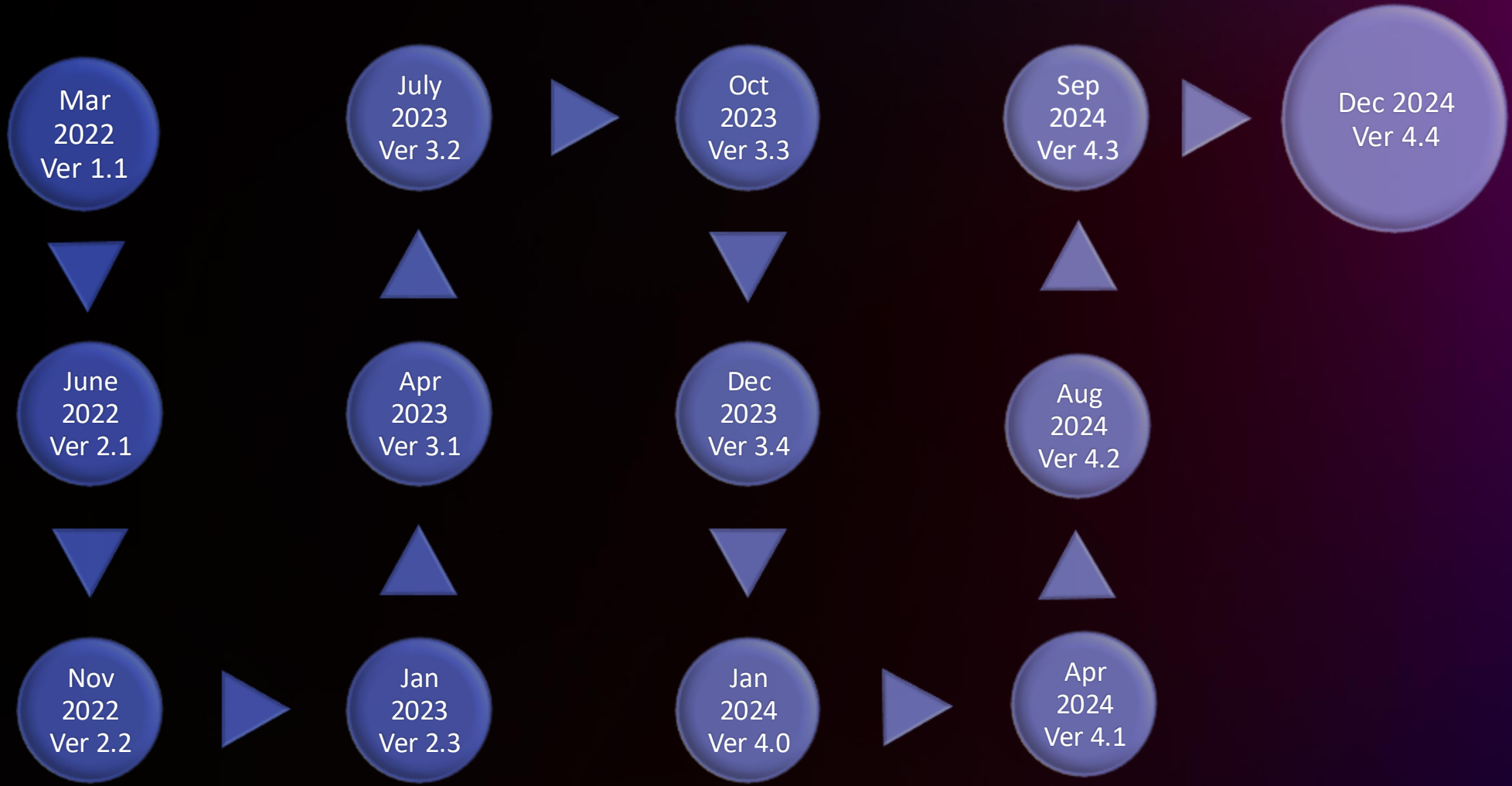
SQL SERVER-SPECIFIC FEATURES

- Triggers and Stored Procedures
- Full Text Search
- Linked Server
- Nested transactions
- Table Valued and Scalar Functions
- Data types (money, sql_variant, table, user defined)
- Savepoints
- Static cursors
- Control-of-Flow statements (e.g. GOTO, TRY/CATCH)
- Case-insensitive identifiers
- Much, much more.

Support for SQL Development Tools

- Improved support for SQL Server Management Studio with latest Babelfish version
- Command line tools: Sqlcmd, Invoke-Sqlcmd, BCP
- Azure Data Studio has limited Babelfish support and also supports connections to the PostgreSQL endpoint
- DBeaver now has a Babelfish-specific endpoint in preview

Babelfish Release Timeline



Babelfish feature support enhancements

Babelfish 1.2 (APG 13.6) March 2022

- **DMS support with Babelfish target endpoint –** Includes system catalog objects to support using the new DMS Babelfish target endpoint for full load operations only.
- **CREATE / ALTER / DROP USER –** Supports low privileged users for individual virtual databases
- **GRANT / REVOKE -** Support for granting and revoking (GRANT/REVOKE) permissions for database principals only (not database roles). Support includes GRANT OPTION and REVOKE..CASCADE options for SELECT, INSERT, UPDATE, DELETE, REFERENCES, EXECUTE, and ALL [PRIVILEGES].
- **New or improved built-in functions –** Adds or improves support for lock_timeout, servicename, columns_updated, update, fulltextserviceproperty, isjson, json_query, json_value, has_dbaccess, suser_sid, suser_sname, is_srvrolemember, checksum, schema_id, connection_property, and serverproperty
- **New information_schema views** – Adds support for tables, columns, domains, and table_constraints
- **New catalog views and stored procedures** – Adds support for sys.dm_os_host_info, sys.dm_exec_sessions, sys.dm_exec_connections, sys.endpoints, sys.table_types, sys.database_principals, sys.sysprocesses, sys.sysconfigures, sys.syscurconfigs, and sys.configurations, sp_table_privileges, sp_column_privileges, sp_special_columns, sp_fkeys, sp_pkeys, sp_stored_procedures, xp_qv, sp_describe_undeclared_parameters, and sp_helpuser

Babelfish 2.1 (APG 14.3) June 2022

- **DMS support for the PostgreSQL target endpoint –** Customers can now use the PostgreSQL target endpoint for full load and continuous loads using CDC
- **Cross-database DML** – Provides three-part name support for select, select..into, insert, update, delete to tables and views for virtual databases in the same instance
- **Database roles** – Support create, drop, and alter database roles to simplify assignment of permissions
- **Explain plan support for T-SQL –** Uses SET BABELFISH_SHOWPLAN_ALL ON (and OFF) and SET BABELFISH_STATISTICS PROFILE ON (OFF) to generate PostgreSQL explain plans for troubleshooting
- **SQL Server Management Studio** – Supports for connecting to the Object explorer to database objects
- **New built-in functions –** Adds support for is_member, is_rolemember, has_perms_by_name, and patindex
- **New catalog views and stored procedures** – Adds support for syslanguages, sys.indexes, sys.all_views, sys.database_files, sys.sql_modules, sys.system_sql_modules, sys.all_sql_modules, sys.xml_schema_collections, sys.dm_hadr_database_replica_states, sys.data_spaces, sys.database_mirroring, sys.database_role_members, sp_sproc_columns, sp_sproc_columns_100, sp_helprole, and sp_helprolemember

Babelfish 2.2 (APG 14.5) November 2022

- **DMS Babelfish target endpoint improvements** – Includes performance and removes table column limitations
- **SQL Server BCP utility** – Now supports -E flag (for identity columns) and -b flag (for batching inserts)
- **Cross-database stored procedure calls** – Provides three-part name support for calling stored procedures in other virtual databases in the same instance
- **New built-in functions** – Adds support for host_name(), objectproperty(), objectpropertyex(), system_user(), session_user(), original_login(), servername(), and dbts()
- **New information_schema views** – Adds support for check_constraints, constraint_column_usage, column_domain_usage, views, and routines
- **New query join types** – Provides support for OUTER APPLY and CROSS APPLY syntax
- **New SET statements** - Adds support for showplan_all, statistics profile, and parseonly

Babelfish 2.3 (APG 14.6) January 2023

- **Upgrade and patching –** Supports upgrades from 13.8, 13.7.1, and 13.6.4 and higher versions to Aurora PostgreSQL 14.6. Provides zero-downtime patching for minor version upgrade and engine patching.
- **New or improved built-in functions –** Adds or improves support for format, json_modify, values, datetimeoffsetfromparts, datediff, dateadd, datepart, datename, try_convert, substring, objectproperty, sum, like
- **New or improved catalog views and stored procedures** – Adds support for sp_addrole, sp_droprole, ap_addrolemember, sp_helpfixedrole, sys.all_parameters, sysobjects.crdate, sp_helpsrvrolemember, sys.all_objects, sp_tables
- **Query troubleshooting –** Adds support for T-SQL hints for INDEX, JOIN, FORCE ORDER, and MAXDOP. Adds execution plan support for THROW, PRINT, USE, and RAISEERROR along with
- **Security model improvements –** Supports guest user for databases along with GRANT/CONNECT TO/FROM user including guest
- **Performance improvements –** Improved performance for INSERT statement with IDENTITY_INSERT=ON, execution of sp_sproc_columns

Babelfish 3.1 (APG 15.2) April 2023

- **Windows Active Directory –** Provides support for individual user logins
- **Linked server support –** Provides support to create linked server to an external SQL Server or Babelfish database and selecting data using OPENQUERY()
- **New or improved built-in functions –** str, atn2, datediff_big, object_schema_name, app_name, degrees, radians, power, next value for, openjson, avg, serverproperty, object_definition, object_name, object_id, scope_identity, session_context, try_convert, try_cast
- **New or improved catalog views and stored procedures** – Adds support for sp_rename, sp_fkeys, sys.systypes, sys.all_views, sys.indexes, sys.sysindexes, sys.has_perms_by_name, sys.types, sp_babelfish_volatility, sys.check_constraints, sp_set_session_context, sys.partitions, sp_executesql
- **New information_schema views** – Adds support for schemata, routines, sequences
- **Framework improvements** – Support for ASP.net SQLServerMembershipProvider, address issues with .net Entity Framework
- **SQL Server Management Studio** – Support for scripting Babelfish objects from the Object Explorer

Continuous Babelfish Updates with every PG Release

Babelfish 3.2 (APG 15.3) July 2023

- **New features –**Supports for remote SELECT with 4-part object name. Remote references in UPDATE/DELETE only for reading, i.e. in FROM-clause. Supports Babelfish instance as a linked server from SQL server instance.
- **New aggregate functions –** Support for STDEV(), STDEVP(), VAR() , VARP()
- **New Functions -** Supports TIMEFROMPARTS(), DATETIME2FROMPARTS(), ROWCOUNT_BIG(), DATABASE_PRINCIPAL_ID() and CONTEXT_INFO() T-SQL functions
- Supports SET rowcount and SET CONTEXT_INFO T-SQL syntax
- Adds support for TOP clause with INSERT/UPDATE/DELETE commands
- Supports sp_rename for COLUMN , TRIGGER, TABLE TYPE and USER DEFINED DATATYPE objects.

Babelfish 3.3 (APG 15.4) October 2023

- Added support for pg_stat_statements extension with Babelfish.
- Added support for the T-SQL square bracket syntax with the LIKE predicate.
- Added support for CREATE or ALTER or DROP EXTENSION statements in sp_execute_postgresql procedure.
- Added support for extended properties for object types database, schema, table, view, column, sequence, function, procedure.
- New built-in functions – Adds support for HOST_ID(), EOMONTH(), PARSENAME(), SMALLDATETIMEFROMPARTS(), fn_listextendedproperty()
- New catalog views and stored procedures – Adds support for sp_enum_oledb_providers, sp_testlinkedserver, sp_who, sys.extended_properties, sp_addextendedproperty, sp_updateextendedproperty, sp_dropextendedproperty
- Added support for connect_timeout option in sp_serveroption.

Babelfish 3.4 (APG 15.5) December 2023

- Added support for TSQL Isolation Level SERIALIZABLE and REPEATABLE READ with PostgreSQL semantics.
- Added support for enable or disable triggers.
- Added support for TSQL functions DATETRUNC(), DATE_BUCKET(), SWITCHOFFSET(), TODATETIMEOFFSET(), AT TIME ZONE clause, TYPE_ID(), TYPE_NAME(), COL_LENGTH() and COL_NAME()
- Added support for PIVOT in limited scope (not supported when used in a view definition, a common table expression, or a join)
- Added support for DEFAULT keyword in calls to stored procedures and functions.
- Added support for casting DATETIME to numeric types.
- Added support for DBCC CHECKIDENT for ability to reset IDENTITY columns.
- Added support for PRIMARY KEY NOT NULL IDENTITY clause in CREATE/ALTER TABLE.
- Added support for double-quoted strings containing single-quote, embedded double quotes in a double-quoted string, and unquoted string parameters.
- Added support for ALTER AUTHORIZATION syntax to change database owner.
- Added support for TSQL KILL command.
- Added support for TSQL catalog objects Information_schema.key_column_usage, sys.server_role_members, sys.database_permissions, sp_changedbowner
- Added support of variable as input for SET ROWCOUNT and SET DATEFIRST.
- Added support for IDENTITY() function in a SELECT-INTO statement
- Added support for ALTER USER…WITH LOGIN syntax.
- Added support for change in transaction isolation from inside transaction block with well defined behavior.
- Added support for casting datetime and smalldatetime to numeric types.

Babelfish 4.0 (APG 16.1) January 2024

- Limited support for Full Text Search in Babelfish for CONTAINS function and CREATE/DROP FULLTEXT INDEX. No FULLTEXT catalog is required.
- Added support for creating INSTEAD OF Triggers on Views.
- Changed the default Babelfish migration mode from single database to multiple databases.
- Supports major version upgrade from Aurora PostgreSQL version 15 to version 16. Major version upgrades from Aurora PostgreSQL 14 need to go to version 15 first.
- Supports parsename, session_context and sp_set_session_context when using with non-default server collation.

Babelfish 4.1 (APG 16.2) April 2024

- Support the ability to perform similarity search using embeddings vector through Babelfish. The ability to use HNSW and IVFLAT indexes is also supported
- Support the ability to access Amazon ML services such as Amazon Comprehend, Amazon Sagemaker and Amazon Bedrock through AWS_ML extension
- DMS support with version 3.6.3 for Babelfish 15.5 or newer as a source using the Aurora PostgreSQL endpoint
- SELECT … FOR JSON AUTO
- INSTEAD OF trigger on view
- Comparison operators !> and !<
- GEOGRAPHY, GEOMETRY datatypes
- Geospatial functions STX, STY, LAT, LONG, STASTEXT, STASBINARY, STDISTANCE
- Full-text search: CONTAINS() (limited supported), CREATE FULLTEXT INDEX
- Catalogs sys.availability_groups, sys.availability_replicas (SSMS support)
- sp_procedure_params_100_managed
- DROP INDEX index ON schema.table; DROP INDEX [schema.]table.index
- Expressions in OFFSET…FETCH… clauses in SELECT
- Comparison operators with embedded whitespace (e.g. a < > b)
- Optional AS keyword in CREATE FUNCTION
- Support for bbf_dump and bbf_dumpall on EC2 instances using Amazon Linux 2023(AL2023). Requires installing psql and pg_restore clients on instances.

New Features

Version 4.2	Version 4.3	Version 4.4
• PIVOT • Accent sensitive collation • **Group Security based AD authentication for Babelfish** • **Support DMS migration out from Babelfish as source using PG endpoint** • Logical replication support for Babelfish • Blue Green Deployment • Alter Procedure • Grant Schema Permissions • Support for unique indexes on columns allows NULLS	• Table Partitioning • Non-deterministic CHARINDEX, PATINDEX and REPLACE • Geometry/Geography	• ALTER FUNCTION syntax • pgaudit extension support • XML method .EXIST() for XML Datatypes • Create database with specific collations • Support of sys.sp_reset_connection stored procedure to reset connection. • Enabled cross-database references of objects (tables/views/functions) in views • Support of **sys.dm_os_sys_info** view to provides information about the instance like **server_start_time** and **ms_ticks** • Support of **user connection** and **network packet size** information in sys.configurations view • Correlated subquery transformation

Adoption

Customer References

FundApps provides automated compliance monitoring services to financial institutions, including some of the world's largest asset managers, hedge funds, and investment banks. To achieve their mission of making regulatory compliance simple, FundApps builds its services on AWS to cost-effectively handle the scaling demands of a growing business and maximize their ability to invest in innovation, while meeting carbon emission reduction objectives.

> "Babelfish for Amazon Aurora PostgreSQL-Compatible Edition has enabled us to modernize our databases to utilize Amazon Aurora Serverless v2 without having to make significant changes to our application. We used the open-source Babelfish Compass tool to generate a compatibility assessment report, which identified 90% of our SQL commands as compatible with Aurora PostgreSQL. The remaining commands were addressed with simple workarounds leading to a complete migration within weeks. With Babelfis able to avoid rewriting large portions of the application, accelerating time to delivery and allowing the team to focus their ti product innovation instead of the migration. We chose Aurora Serverless v2 to manage background services, which have hug demand and long periods of relative inactivity. Aurora Serverless v2 allows us to scale to meet peak demand when required having to massively over-provision. This allows us to achieve better overall utilisation, lower costs, and a smaller carbon foot Overall, the migration from self-managed MS SQL Server to Aurora PostgreSQL has reduced our database costs by over 60% no longer need to undertake a majority of the traditional, undifferentiated maintenance work associated with relational data means our engineers have more time to invest in building valuable new products."

Toby O'Rourke, CTO - FundApps

FactSet

FactSet enables tens of thousands of investment professionals around the world with the data and analytics they need to make crucial decisions. FactSet creates flexible data and software solutions and leverages Babelfish for Amazon Aurora PostgreSQL-Compatible Edition in production as part of its technology stack.

> "FactSet is excited about Babelfish for Aurora PostgreSQL. Babelfish materially accelerated the pace of a set of migration initiatives from commercial relational databases to PostgreSQL on Amazon Aurora and modernized our data infrastructure without the burden of converting all of our application code as we normally would. As a result, our database and application teams can focus on revenue generating product features rather than rewriting queries."

Wilson Tsai, Senior Director of Engineering, FactSet

Diligent is the leading Governance, Risk, and Compliance (GRC) SaaS company, empowering more than 1 million users and members and leaders to make better decisions, faster. The Diligent One Platform helps organizations connect their entire including governance, risk, compliance, audit and ESG — to bring clarity to complex risk, stay ahead of regulatory changes and deliver impactful insights, in one consolidated view.

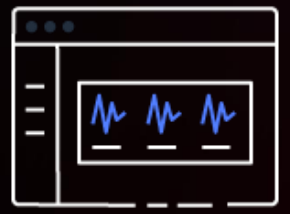
> "A large majority of Diligent products are built on SQL Server databases. With Babelfish we were able to accelerate our SQL Server application migrations to Aurora PostgreSQL-Compatible Edition, saving us thousands of developer hours which would have been required for manual migrations. Babelfish enables Aurora PostgreSQL to understand SQL Server dialect (T-SQL) and protocol, so our SQL Server applications could be migrated with little or no code change, and AWS offers the Babelfish capability free of charge. We also used Babelfish Compass, an open source standalone tool, which quickly analyzed our T-SQL SQL/DDL scripts for compatibility. Using Babelfish allowed us to drastically reduce the overall infrastructure cost associated with Diligent's migration to AWS. In addition, with our migration to Aurora PostgreSQL using Babelfish, we are now utilizing Amazon Aurora Serverless v2. By leveraging Aurora Serverless v2, we modernized our databases to scale automatically, reduced operational overhead, and lowered our backup costs by 78%."

Satheesh Ravala, CTO - Diligent

Development model for Babelfish

HOW DO I ADD NEW FUNCTIONALITY IN MY MIGRATED APPLICATIONS?

Develop new functionality in T-SQL using SQL Server database drivers



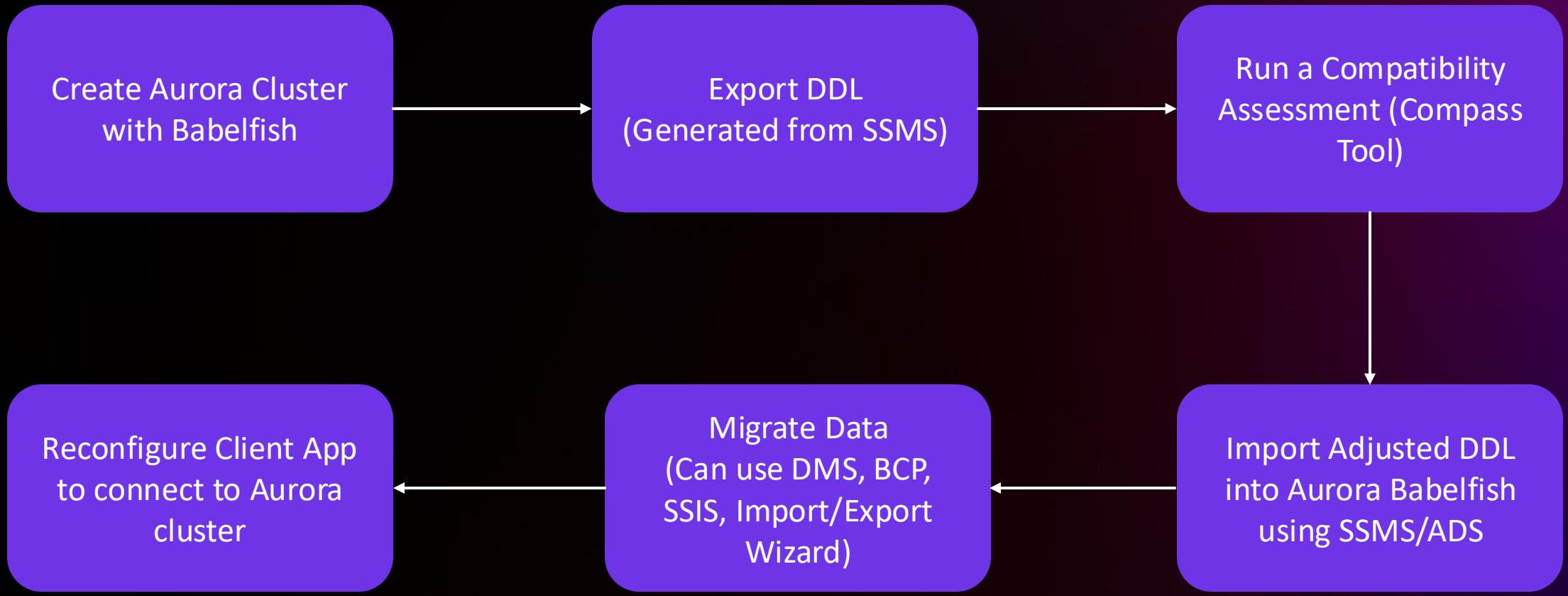
Develop new functionality in PostgreSQL using PostgreSQL database drivers

Develop new functionality in both languages, depending on best fit.**

** PostgreSQL and SQL Server have different transactional rules, such as for rollbacks. Avoid mixing languages inside a single database call to avoid unexpected results. Watch for triggers, etc.

Migration

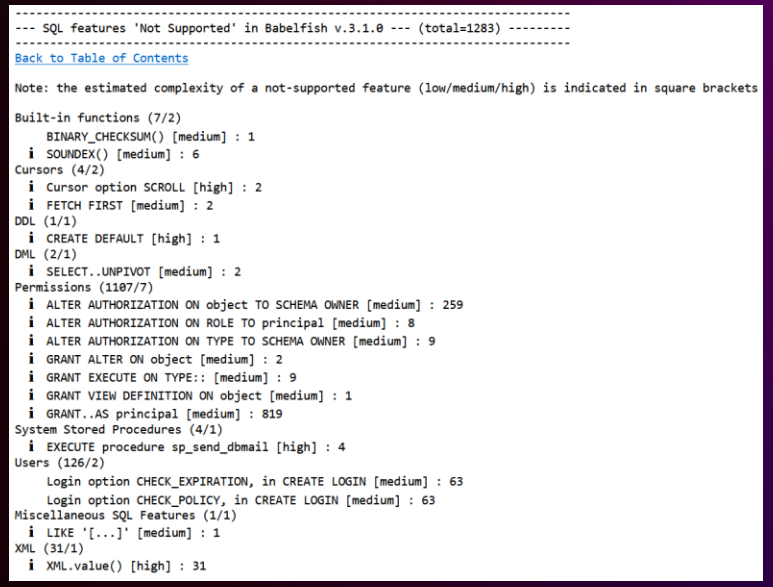
Migration steps



Migration Assessment Tool

BABELFISH COMPASS

- Determines what part of your app/T-SQL is not supported by Babelfish using open-source tool - Babelfish Compass'
  - Stand-alone, command-line tool
  - GitHub Download | User guide
  - Updated regularly. Always use the latest version
  - Consider SQL Profiler / Extended Event traces for app use cases
  - Includes rewrite option to workaround many unsupported commands
  - No Babelfish cluster is needed to run the assessment

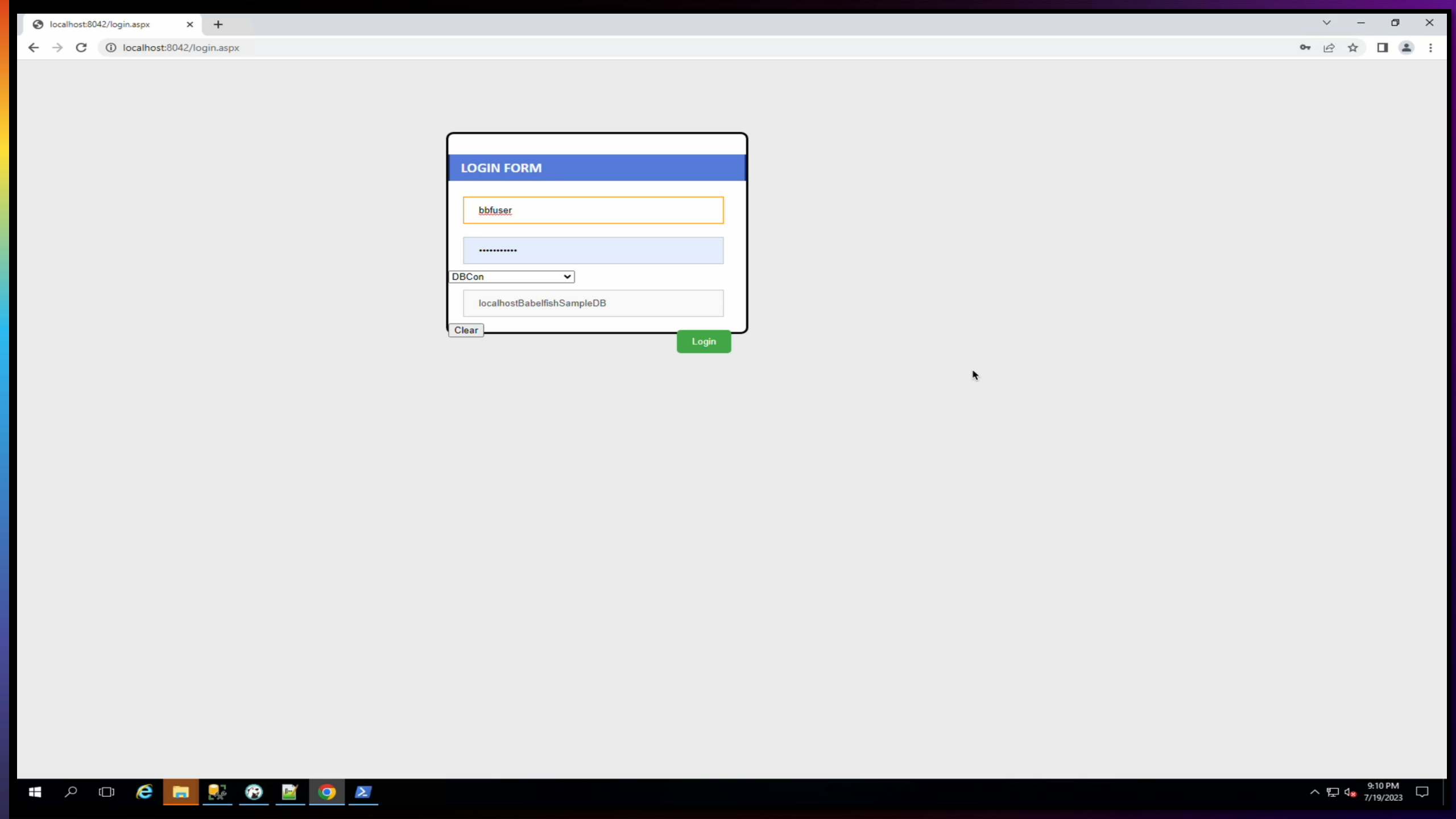


```
-----------------------------------------------------------------------------
--- SQL features 'Not Supported' in Babelfish v.3.1.0 --- (total=1283) ---------
-----------------------------------------------------------------------------
Back to Table of Contents

Note: the estimated complexity of a not-supported feature (low/medium/high) is indicated in square brackets

Built-in functions (7/2)
    BINARY_CHECKSUM() [medium] : 1
  ℹ SOUNDEX() [medium] : 6
Cursors (4/2)
  ℹ Cursor option SCROLL [high] : 2
  ℹ FETCH FIRST [medium] : 2
DDL (1/1)
  ℹ CREATE DEFAULT [high] : 1
DML (2/1)
  ℹ SELECT..UNPIVOT [medium] : 2
Permissions (1107/7)
  ℹ ALTER AUTHORIZATION ON object TO SCHEMA OWNER [medium] : 259
  ℹ ALTER AUTHORIZATION ON ROLE TO principal [medium] : 8
  ℹ ALTER AUTHORIZATION ON TYPE TO SCHEMA OWNER [medium] : 9
  ℹ GRANT ALTER ON object [medium] : 2
  ℹ GRANT EXECUTE ON TYPE:: [medium] : 9
  ℹ GRANT VIEW DEFINITION ON object [medium] : 1
  ℹ GRANT..AS principal [medium] : 819
System Stored Procedures (4/1)
  ℹ EXECUTE procedure sp_send_dbmail [high] : 4
Users (126/2)
    Login option CHECK_EXPIRATION, in CREATE LOGIN [medium] : 63
    Login option CHECK_POLICY, in CREATE LOGIN [medium] : 63
Miscellaneous SQL Features (1/1)
  ℹ LIKE '[...]' [medium] : 1
XML (31/1)
  ℹ XML.value() [high] : 31
```

Example command line:

```
./BabelfishCompass.bat MyAppName store-objects.sql –rewrite –reportoption xref -optimistic
```

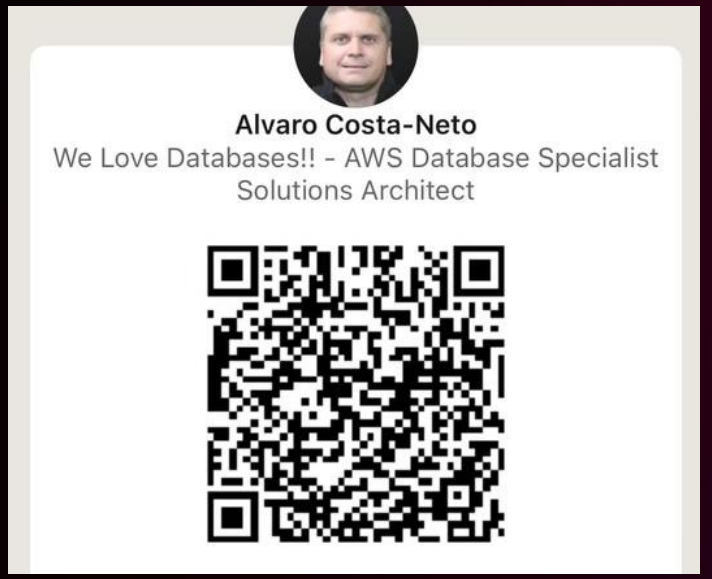
DEMO


localhost:8042/login.aspx
localhost:8042/login.aspx
LOGIN FORM
bbfuser
DBCon
localhostBabelfishSampleDB
Clear
Login
9:10 PM
7/19/2023

Q&A

We value your feedback!

Thank You!!



agcosta@amazon.com