**Assessment**

A. Migration of sample database

1. Configure the Northwind sample database from this link (https://github.com/Microsoft/sql-

server-samples/tree/master/samples/databases/northwind-pubs) and configure that locally.

2. Use any method or toolkit to migrate this to PostGRE. Create a runbook to explain the steps

involved in configuration, mentioning the special aspects to be considered based on your own

experience.

3. Any incompatibility needs to be noted and an approach identified for fixing that out.

B) Migration Strategy

1. A client has an SQL Server database that has SSIS jobs, and a Service Broker configured on it. The

database is approximately 10TB in size and grows about 10GB monthly. It currently uses 24-core

VCPU and 256GB of RAM under SQL Enterprise Edition 2016. This is a transactional database that

has a max downtime limit of 4 hours on special update events.

2. What will be the strategy to migrate such a database to PostGRE considering the size and

transactional volume? Mention any tooling (open-source or proprietary) that can ease out this

process.

3. What can be the issues being faced and possible mitigation plan?

4. What will be the roadmap for the transition and what factors will determine the timelines of such

a migration?

Please find below the Company Profile.

SourceFuse Inc. is transforming the way today’s most successful companies develop breakthrough

roadmaps leveraging cloud-based technologies.

A leading AWS Advanced Consulting Partner, SourceFuse has delivered over 600 AWS

implementations that boost efficiency, ensure compliance, deliver actionable insights, and lower

the total cost of ownership.

We have a global presence, with offices in Florida, the USA, the UK, Australia, and Noida, Mohali,

Hyderabad India. With more than 15 years of deep expertise, commitment to digital innovation,

service excellence, and customer success, we enable enterprises through application

modernization and Windows migration services of legacy workloads.

**Solution**

Solution for (A)

1. Use AWS DMS for schema conversion.

2. Any errors logged while should be manually fixed/rewritten.

3. Views/Procedures/Functions/Triggers needs to be re-written in postgresql.

Solution for (B)

1. Drop all primary/foreign key constraints, drop/disable all triggers

2. Enable CDC using AWS DMS

3. Use AWS DMS Full-load to migrate data from sql server to postgresql

4. Check for errors in log and fix them.

5. Validate migrated data in postgresql with data from sql server

6. Migrate pending data using AWS DMS CDC to postgresql

7. Apply scripts of primary/foreign key, enable triggers

8. Apply indexes

9. Timelines will depend on duration for converting Views/Procedures/Functions/Triggers and how CDC is implemented