kloia

SQL Server Transition

SQL Server Trends & Challenges

SQL Server has been improving itself on each new version. Although is it traditionally known as a Relational Database, latest versions also include platform capabilities like integration with Apache Spark and also HDFS. Beside those, as an underlying platform, SQL Server now supports also Linux and Docker. Another important news for SQL Server 2008, whose is approaching to its End-of-Life during 2019, which means that customers will not be able to get vendor support.

kloia solution: SQL Server Transition

SQL Server Transition can improve your current platform together with its new features.

Initially, kloia runs a session for the purpose of understanding your objectives and priorities for this transition. This helps to provide you the optimal solution for your transition.

Some of the possible transition objectives can be:

- Performance Optimization
- Cost Optimization
- End-of-life / Support

Based on your objective(s), there can be various solution alternatives. In order to determine appropriate scenarios for your transition, kloia runs an assessment session on your current Software and Database structure. This helps to understand the constraints of the software architecture which effects the proposed solution. This assessment includes deep-dive sessions about your approach to Database from Software perspective. By the end of this session, the dependency between the current Database structure and your software is identified. Based on this assessment, one of the following options is offered:

- SQL Server 2017 on Linux
- AWS RDS SQL Linux
- AWS Aurora

The next phase includes to identify your custom requirements which will affect the implementation phase. Those may include:

- I/O requirements
- Backup Policy
- Monitoring and Alarm

The last phase includes the implementation which includes two steps:

- Installation/Configuration: Based on the requirements, we develop "as-code" using Terraform
- Migration: The main responsible of this phase is determined based on the resources on customer

As a summary, the steps for the solution are:

1- Objectives & Priorities: Identifying your underlying motivation for this transition
2- Assessment: Deep assessment of the dependency between the current Database structure and your Software
3- Solution Offering: Based on your constraints and software architecture, offer a Database alternative
4- Installation/Configuration: Development of your solution with "as-code"
5- Migration: Migration of the Data and switch-over

Benefits

Cost Optimization(SQL Server Linux):
SQL Server on Linux does not depend on Windows Server Operating System which brings you additional cost optimization. Beside, if you are on AWS RDS, transition to SQL Server Linux brings you additional saving from RDS costs.

Performance:
Linux Operating System comes together with its robust and high performance low-footprint kernel

Cutting-edge Technologies:
SQL Server latest versions come with cutting-edge features

Key Solution Features

Infrastructure-as-code: kloia benefits from Terraform in order to keep the infrastructure as code. This helps to keep the infrastructure stable and consistent across the environments like Development, Staging and Production. All changes are historically a git commit which helps to keep track of what has been done. The side benefit of this approach of also to have a Disaster Recovery environment on demand, triggered by a simple command whenever needed.

AWS Public Cloud: Most preferred cloud alternative to host your Database

SQL Server: Improved feature-set and performance. The latest 2019 version comes with Apache Spark and HDFS.

SQL Server Cache: Additional cache layer using ScaleArc wherever is needed.

SQL Server Read Replicas: For reporting purposes or read-intensive queries, additional read-replicas help the application to scale with minor software refactoring effort.

Monitoring: kloia employs SQL Server monitoring with Grafana which comes with SQL Server focused metrics.

Bonus Feature

DBmaestro
DevOps for Database

DevOps-for-Database: In case the functionality that you deliver with your software depends on the development and changes on SQL, in such case this pipeline should also be managed same like as the Software pipeline including rollbacks, versioning, RBAC and restrictions. DBMaestro is used to automate the releases for the Database-level developments.

How it works

kloia begins their solution process with an audit, onboarding session, and Statement of Work (SOW). The onboarding process lays out a timeline of all the solution elements, from which your team can select the proper elements for your environment. These elements are then taken through kloia's five step Transition.

For collaboration with the development team, kloia team creates a Shared #slack channel for instant communication. Beside, the tasks are managed on Jira Board upon to the preference.

Getting Started

You can submit an inquiry by sending email to **sqltransition@kloia.com**