

Digital Center

TCS - Cummins Confidential

Confidentiality Disclaimer: All information contained in this webpage is privileged and confidential and intended solely for the use of the individual or entity to whom it is addressed. Any unauthorized disclosure, copying, or distribution (physical or digital) of the contents is strictly prohibited.

App Modernization Framework

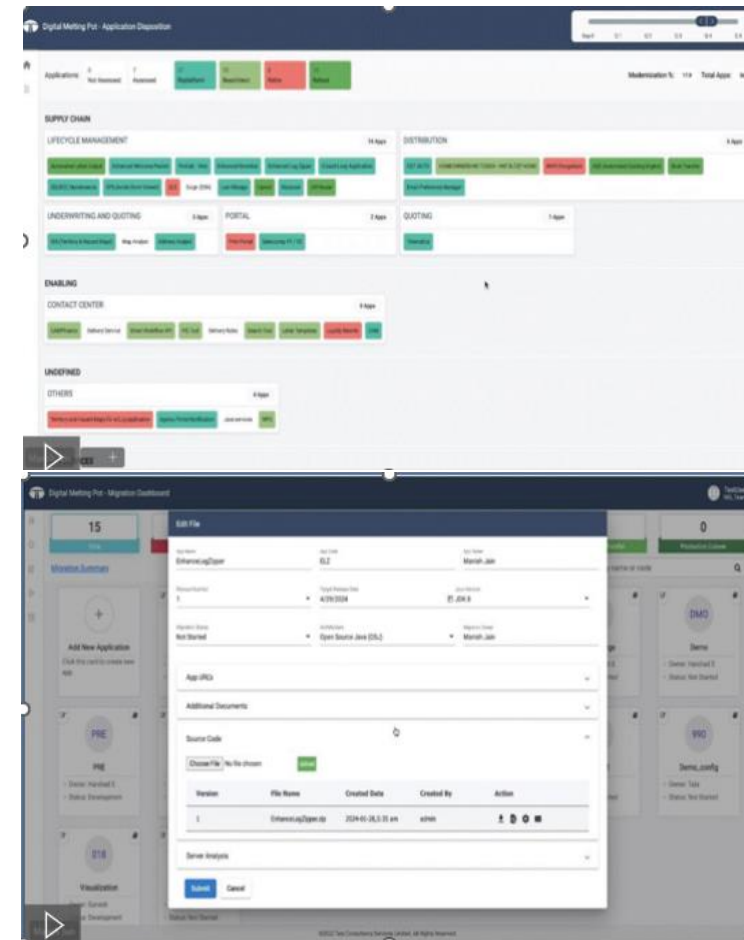
Framework for Modernizing legacy apps to serverless apps.

Key Features

- Initial tech upgrades via Mod 360
- Accelerator for Oracle to Postgres Migrations
- Version upgrades for back-end and front-end
- Cloud migrations with DevSecOps pipelines

Key Outcomes

- ↑ 50% Faster cloud migration
- ↑ License and data center cost savings
- ↓ Improved security
- ↑ Enhanced user experience



TCS App Modernization Framework

Confidentiality Disclaimer: All information contained in this webpage is privileged and confidential and intended solely for the use of the individual or entity to whom it is addressed. Any unauthorized disclosure, copying, or distribution (physical or digital) of the contents is strictly prohibited.

Micro Front-end Decomposer

Key Objective

A design approach where front-end app is decomposed into individual, semi-independent “microapps” that can be built by different teams using different technologies. This solution works on different types of server and serverless architectures.

Key Features

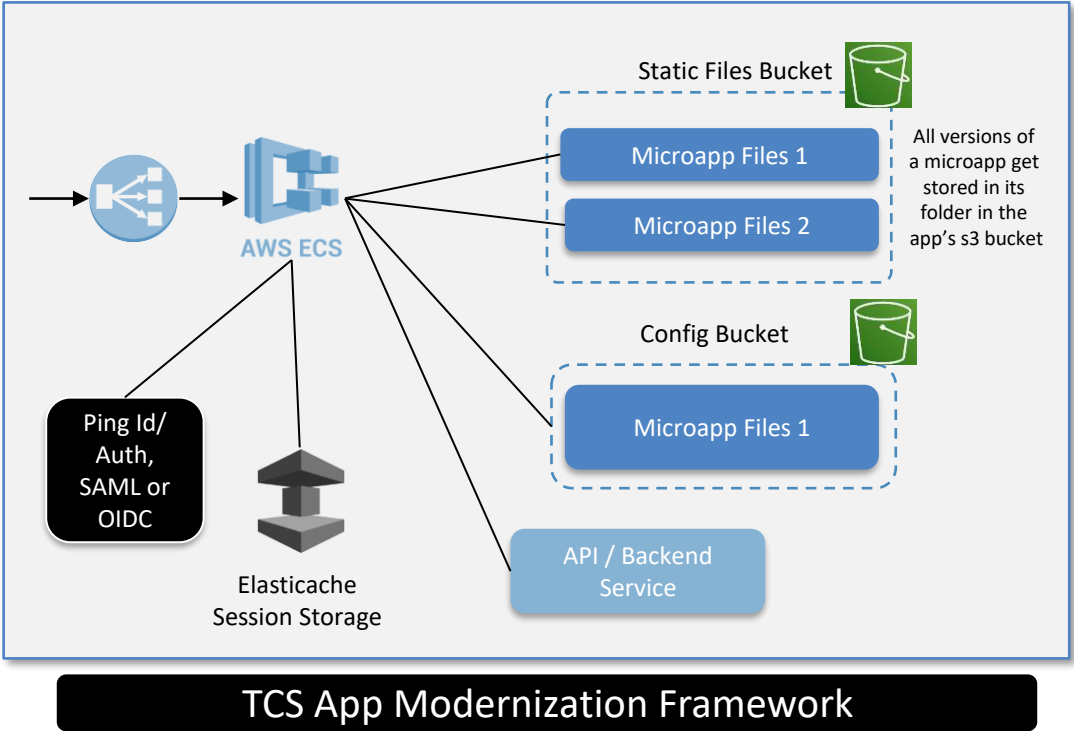
Accelerated Development with Autonomous Teams: Every component of a web app or site can be developed by an autonomous team. Each team is fully independent

Higher App Stability Due to Loose Coupling: Ensures the advanced stability of a web solution: the failure of one component has minimal or no impact on the operation of all the other components

Faster Feature Roll-out Due to Independent Deployments: Independent deployments mean that each micro frontend is deployed separately from the other components of a web solution

Key Outcomes

- **Independent** Implementations
- **Higher Performance**
- **Faster Development**
- **Reliability and Scalability**



Confidentiality Disclaimer: All information contained in this webpage is privileged and confidential and intended solely for the use of the individual or entity to whom it is addressed. Any unauthorized disclosure, copying, or distribution (physical or digital) of the contents is strictly prohibited.

Schema Conversion Tool

Key Objective

Managed migration and replication tool that helps move database schemas and analytics workloads to cloud / different databases quickly, securely, and with minimal downtime and zero data loss. Tool supports migration between many popular database and analytics engines, such as Oracle to Postgres, MySQL, MySQL to Amazon Relational Database (RDS) for MySQL.

Key Features

- Simplify application migrations by automating schema analysis, recommendations, and conversion at scale.
- Can also scan application source codes for embedded SQL statements and convert them as part of a database-schema-conversion
- Automatically assess and convert the source database schema and a majority of the database code objects, including views, stored procedures, and functions, to a format compatible with the target database.
- Compatible with popular databases and analytics services as source and target engines, including Oracle, SQL Server, PostgreSQL, and MySQL.
- Save weeks or months of manual time and resources.

Confidentiality Disclaimer: All information contained in this webpage is privileged and confidential and intended solely for the use of the individual or entity to whom it is addressed. Any unauthorized disclosure, copying, or distribution (physical or digital) of the contents is strictly prohibited.

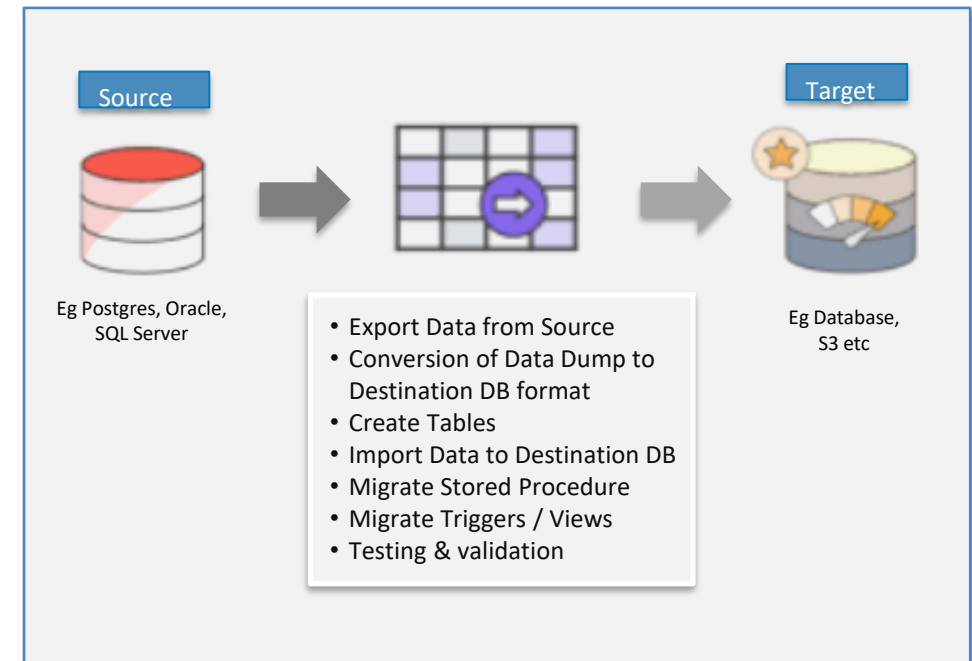
Key Outcomes

Multiple Database one Solution

OnPrem to Cloud Migration

OnPrem to AWS S3 bucket

Reliability and Scalability



TCS App Modernization Framework

CI/CD DevOps Accelerator using AWS CodeBuild Cache

Key Objective

Use AWS CodeBuild to reduce project build time

Key Features

- Have more control over caching. There is no extra cost involved local cache is used
- Configuring the retention period for S3 cache. This is useful when the build is not frequent

Key Outcomes

\$35k per year in build savings

About AWS CodeBuild Cache

AWS CodeBuild can use Cache to reduce the build time for the project. This will impact overall turn around time for deployment in the CI-CD pipeline and thus saving time for developers and ultimately Cost.

There are 2 types of Cache :

- Amazon S3 Cache
- Local Cache

Implementation (3 Ways):

- local Cache via CLI
- Cache via Console – S3 or Local
- Local Cache by AWS CloudFormation template

Confidentiality Disclaimer: All information contained in this webpage is privileged and confidential and intended solely for the use of the individual or entity to whom it is addressed. Any unauthorized disclosure, copying, or distribution (physical or digital) of the contents is strictly prohibited.

Amazon Q - Upgrade Plus

Key Objective

Accelerate application maintenance, upgrades, and migration in minutes

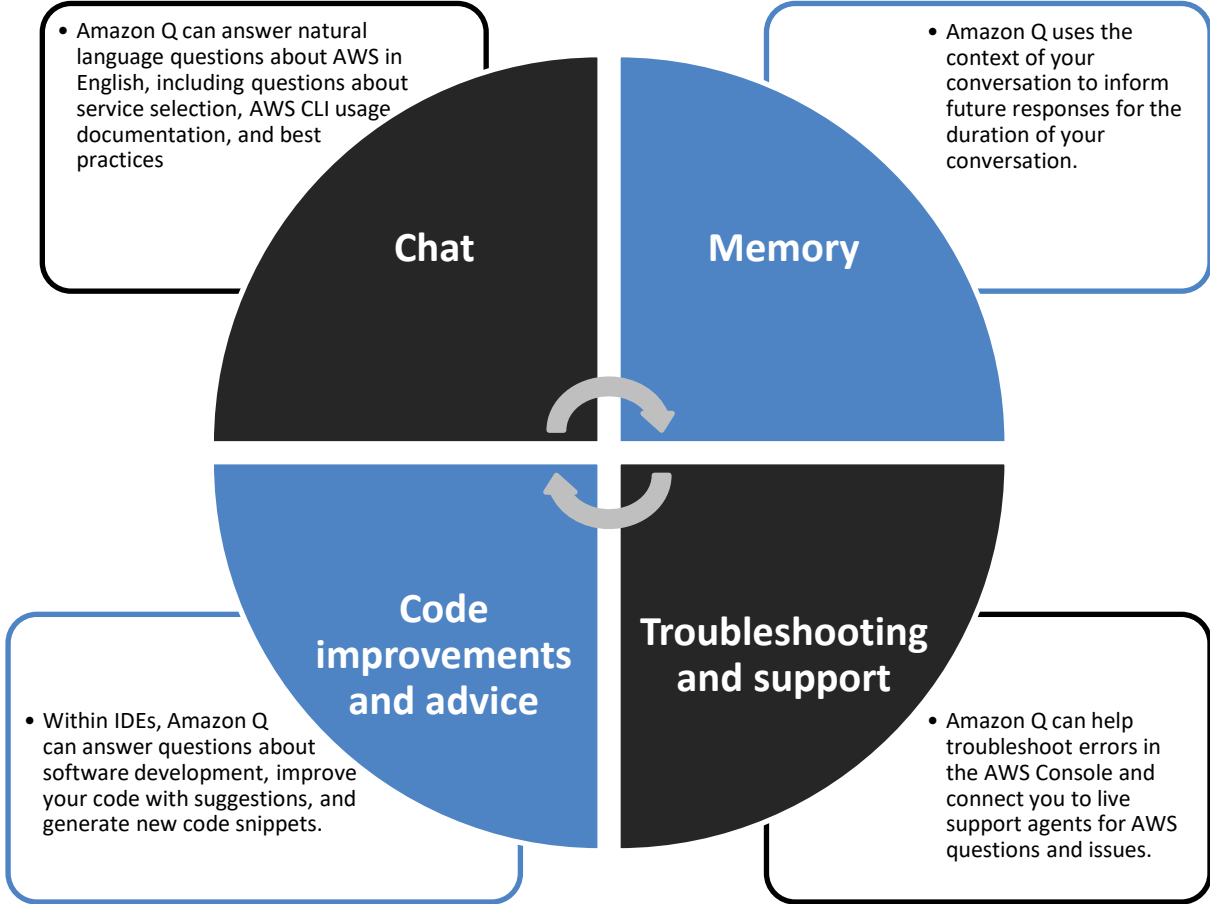
Key Features

- Complete Java language upgrades in a fraction of the time
- Enhance security postures and performance
- Accelerate migration from windows to Linux (awaited)

About Amazon Q

Amazon Q first builds your code in the source language version (either Java 8 or Java 11) to verify that it's able to transform it, and then it transforms your code. After Amazon Q completes the transformation, verify the changes made to your files. Choose View diff to view changes in a file diff view and accept or reject them. You can also view the transformation summary for next steps.

Key Features



Confidentiality Disclaimer: All information contained in this webpage is privileged and confidential and intended solely for the use of the individual or entity to whom it is addressed. Any unauthorized disclosure, copying, or distribution (physical or digital) of the contents is strictly prohibited.