exactpro

EXITUS ACTA PROBAT

Enterprise Data Warehouse Cloud Migration

The Software Testing Perspective

BUILD SOFTWARE TO TEST SOFTWARE

Exactpro Systems Limited. Registered in England & Wales No 09485548

exactpro.com

Contents:

Introduction

Transitioning operations to a cloud infrastructure is a challenge that firms face on their digital transformation journey. Cloud-driven modernisation in the financial services space involves migrating core parts of the enterprise-grade IT infrastructure, such as the data repository. It entails fundamental changes in the system's business logic and architecture, data flows, the Data Base Management System (DBMS) functionality, warehouse operation, as well as the dependencies. The process demands alignment, coordination and careful planning within the organisation.

However, setting up the right migration strategy is only half the challenge: implementing it with no data loss or redundancies and with minimal performance issues in production is what a trading, clearing/settlement or core banking platform operator is looking to achieve. Developing end-to-end software testing expertise at an early stage of the cloud migration design and planning helps ensure the quality and reliability of the new cloud infrastructure.

Ahead of the migration, it is also crucial to thoroughly assess and understand the technology characteristics of your preferred cloud provider, as there are differences between operators. This step helps shape the underlying processes going forward and give an understanding of the level and extent of software testing able to support the transition.

This case study features examples of technology stacks for enterprise data warehouse migrations to the Amazon Web Services (AWS), Google Cloud Platform (GCP), Microsoft Azure and Oracle Cloud Infrastructure (OCI) environments. The testing approaches proposed are similar for these and other cloud services providers.

Forces Driving the Cloud Shift in Data Services

What are the benefits the financial space firms are trying to achieve by moving their data services and functions to the cloud?

Technology

- **Greater infrastructure flexibility** – eliminating the need to invest in self-hosted hardware-based data centre infrastructures, cloud computing resources enable efficient, pay-as-you-go resource management with an option to deploy new environments, scale the capacity up or down in a matter of minutes.

Regulatory Compliance

- **Advanced analytics and record keeping** – the option of unlimited cloud storage makes data control straightforward and the capacity for data retention infinite. The side benefit of the cloud setup is the ability to analyse the entire volume of data for patterns and introduce advanced market surveillance mechanisms.

- **Audit/certification documents** – cloud providers participate in various compliance programs and have the required documentation in place.

Product

- **More room for innovation** – new services and technologies can be experimented with, sandbox-tested, adopted or terminated, without having to set up additional hardware or software and eliminating the need for approval chains impeding the process of deploying extra resources.

Client

- **Lower latency** – a global distributed cloud infrastructure enables deployment of applications in regions close to customers and expanding to new regions as the business grows.
- **Data protection** (GDPR, CCPA, CRPA, etc.) can be ensured by using the appropriate cloud region.
- **Improved security** – cloud providers ensure usage of best practices in cyber security for its infrastructure and provide guidelines for their clients.
- **Better resiliency** – elasticity of the cloud will minimise chances of outages for clients, even at peak load.

Business & Sustainability

- **Resource and cost optimisation** – the principles of resource pooling and supply-side savings allow established technology firms and startups alike to reap the benefits of economies of scale in the cloud. Cloud providers' energy-efficient infrastructures have proven to lower clients' workload carbon footprints.
- **Setup-specific cost optimisation** – as one of the migration testing deliverables, the Exactpro team provides recommendations on the most optimal infrastructure setup.

However, these benefits should be assessed against the impediments intrinsic to a cloud-native infrastructure, please refer to the next section and the Cloud Limitations and Their Repercussions section for the most pertinent ones.

If you would like to start a conversation about your existing or future cloud infrastructure setup, drop us a line.

Discuss your cloud setup

Important Cloud Transformation Aspects

A data warehouse migration powering a firm's infrastructure modernisation is just one of the cogs of the multi-aspect shift. Any kind of technology transformation is a complex process involving not only Platforms, but also the underlying Processes and procedures, as well as the People following them. It is vital to not strive to revolutionise one individual aspect, but rather see an infrastructure modernisation as a step forward in the entirety of interconnected operational aspects. AWS's 2023 whitepaper An Overview of the AWS Cloud Adoption Framework visualises the 'transformation domains' as having a single starting point on the hypothetical timeline. The proposed overarching change implementation structure [seen in Fig. 1] is incremental and features 4 iterative phases: Envision, Align, Launch, Scale – encompassing all operational aspects on a continuous journey through the foreseen change.

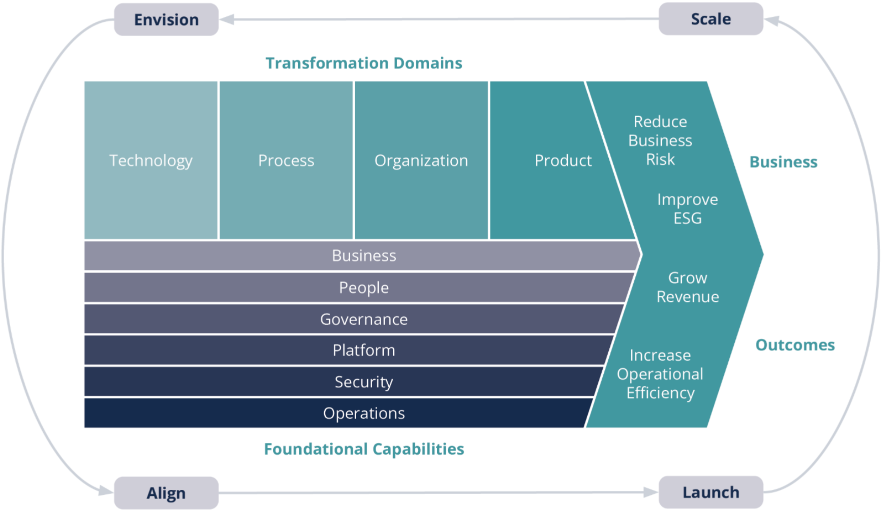


Fig.1 An overview of the AWS Cloud Adoption Framework (AWS Whitepaper, 2021)

It is essential to view a data warehouse cloud migration in the context of other changes, rather than as a standalone exercise in boosting data storage and recall efficiency. This explains why the anticipated benefits involve improved technology, as well as added value in products and business domains.

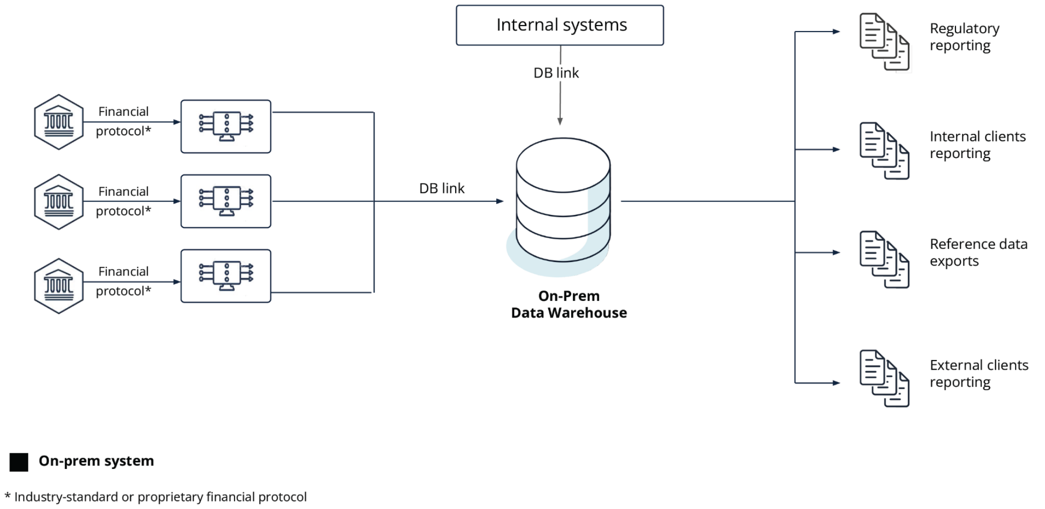
Enterprise Data Warehouse Cloud Migration Testing

From the technology perspective, the main objective of a data warehouse migration is an accurate transfer of years of historical data and all data processing jobs. Due to differences between the legacy system's formats/configurations and the new cloud setup, it is essential to carefully reconcile the two volumes of data. Engaging software testing expertise as early as possible in the migration (preferably, at the ideation phase) helps remediate architectural and documentation-related issues, as well as ensure a more optimal cloud setup.

In this paper, we illustrate our proposed cloud migration testing approach with several sample implementations, but the approach is equally applicable to any other cloud architecture. Penetration/security testing and billing testing are left out of the scope of this case study.

Test Objectives and Test Data

The traditional on-premise data warehouse infrastructure tends to rely on direct connectivity (or connecting via a dedicated component) to the on-premise database by way of financial protocols (industry-standard or proprietary) [see Fig. 2].



* Industry-standard or proprietary financial protocol

Fig. 2 *Pre-migration on-premise database architecture setup*

The migrated cloud setup involves eliminating the intermediary downstream component (passing the financial system's information over to the database) and replacing it with connectivity to the cloud infrastructure via an advanced Message Broker, e.g. Apache Kafka®, RabbitMQ or others. These are high-throughput high-availability event streaming platforms and message brokers for storing and processing historical and real-time data. This approach enables data consumers to read messages from the platform's message broker and its topics instead of the database or the financial systems' gateways (which requires proprietary software to be developed and maintained, due to the nature of the data protocol). Such an approach helps to increase the resiliency of the system, decrease the load on the database, and makes it possible to improve the data consumers' performance.

After being submitted to the Message Broker by the financial system, the data goes through several storage areas (Raw data, Processed data, Pre-generated reporting data), as seen in Fig. 3.

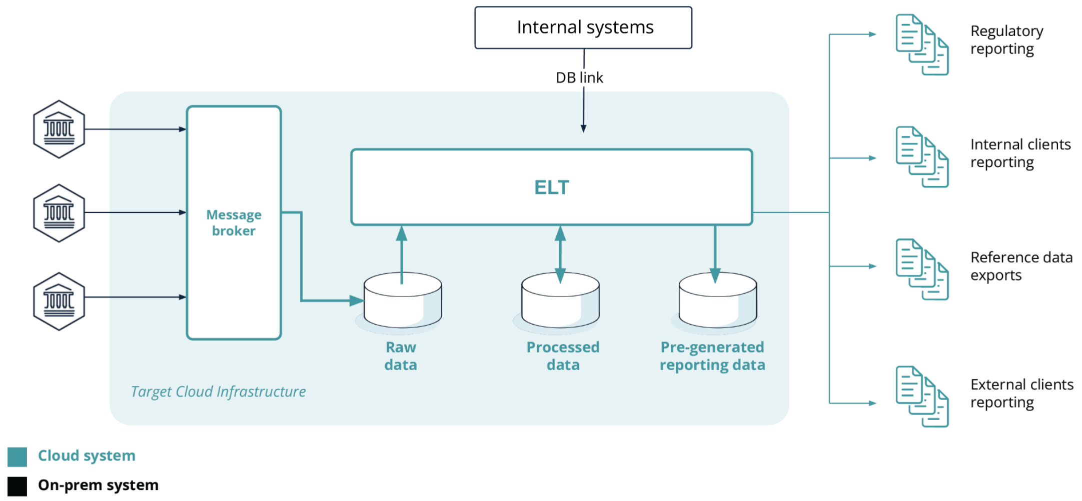


__Fig. 3__ Target cloud database architecture setup

The types of data in the testing scope are:

- Regulatory reporting data
- Internal client reporting data
- External client reporting data
- Reference data exports

Technology stack

These sample technology stacks feature sets of components for cloud migration implementations on Google Cloud Platform (GCP), Amazon Web Services (AWS), Microsoft Azure and Oracle Cloud Infrastructure (OCI) services. Components with similar functionality would be used in other implementations.

Cloud provider/ Component type	***Google Cloud Platform (GCP)***	***Amazon Web Services (AWS)***	***Microsoft Azure***	***Oracle Cloud Infrastructure (OCI)***
Data streaming/ message brokering	Cloud Pub/Sub	Amazon MSK	Event Hubs/Azure Kafka	Oracle Streaming Service
Unstructured data storage	Cloud Storage	Amazon Simple Storage Service (S3)	Azure Blob Storage	Oracle Cloud Infrastructure Object Storage

Structured data storage	Cloud Firestore/ Cloud Spanner	Amazon DynamoDB	Azure Cosmos DB	Oracle NoSQL Database/ Oracle Autonomous Transaction Processing (ATP)
Analytics	Cloud Dataflow (Apache Beam)	Amazon Kinesis Data Analytics (Apache Flink)	Stream Analytics	Oracle Stream Analytics
Computing	Cloud Functions	AWS Lambda	Azure Functions	Oracle Functions
Virtual machines	Compute Engine	Amazon EC2	Azure Virtual Machines	Oracle Compute Infrastructure

Data Storage:

A highly resilient, scalable and customisable database for storing the ‘Processed’ and ‘Pre-generated reporting data’ data is required in the new setup, as it will store a few dozen years worth of transactional data. The choice can be made between Snowflake, Amazon Redshift, Google BigQuery and similar systems from other providers.

Data Transformation and Processing:

ETL (extract, transform, load) tools such as Informatica or Matillion can be used for data transformation and report generation.

Environment Monitoring:

A highly customisable and versatile on-premise and cloud environment monitoring tool with a wide variety of alerting and dashboard options is required for enterprise environments monitoring – DataDog, New Relic, Dynatrace or others.

Environment Setup and Testing Strategy

The cloud environment setup in this given scenario is a combination of a Platform as a Service (PaaS) infrastructure and a number of Software as a Service (SaaS) solutions.

From the functional testing perspective, the proposed data warehouse migration strategy is demonstrated in Fig. 4 and features the full cycle of migration, from the on-premise setup to the cloud one, which are the 1st and last steps of the high-level workflow, respectively:

1. Legacy system
2. Verifying the completeness of data migrated to the cloud system
3. Reconciling the legacy system data with the cloud system data
4. Verification of data processing in the cloud
5. Verification of output (reporting, billing, analytics, client exports, etc.) data generation
6. Verification checks on both systems run in parallel
7. Full migration to the cloud system, decommission of the on-premise system

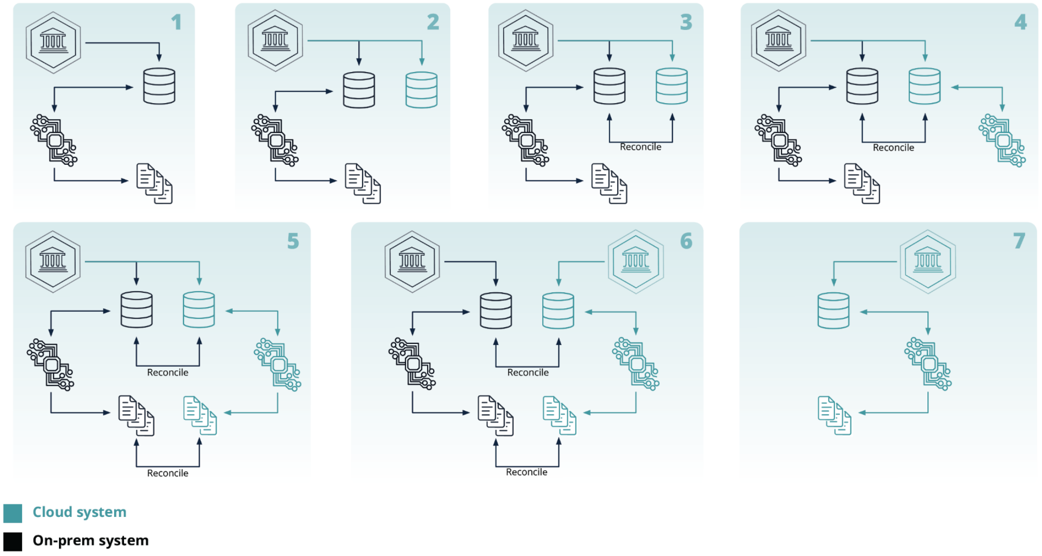


Fig. 4 *Data Warehouse Migration Workflow*

The testing workflow should be iterative. Processes are migrated and tested consecutively, e.g. each output is tested in non-production as well as production environments. Once a given functionality is cleared, the new service is launched, with users having to reconnect to the new service.

Exactpro’s Functional Testing Approach

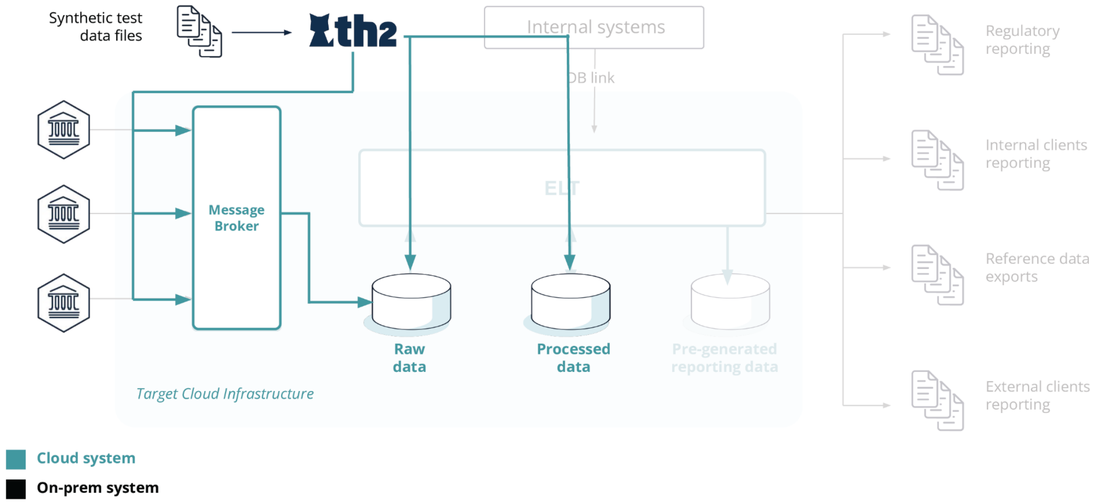


Fig. 5 Functional Testing with th2

The functional testing scope includes 3 main stages of verification checks:

Historical Data migration

- The list of checks performed for each table includes:
    - data definition label (DDL) checks – to verify table schemas before the start of the migration;
    - quick checks following the initial migration, to confirm the completeness of the process – row count, sum for numeric values.
- For large tables containing the financial system data: full data comparison, additional checks for winter/summer dates, data for different releases, etc.

Data Landing

- Processing of synthetic data to cover protocol-specific variations
- Processing of data generated as part of the execution of the full regression test library to cover possible business scenarios

Internal processes and reporting

- Execution of the on-premise corporate data warehouse regression test library
- Parallel run of the on-premise and cloud data warehouses and reconciliation of outputs generated by each system

Let’s review these steps in more detail.

Historical Data Migration

Historical data migration is a separate stage of the infrastructure transformation process, whereby billions of rows of historical data have to be picked up and seamlessly put on new rails. First, data definition-level (DDL) checks are run to verify whether the overall schema format is correct. Next verification checks include the number of rows, the sum of numeric values and the completeness of the migrated data, e.g. checks for the fractional part or cell overflow issues. After these checks are passed and any related defects fixed, full reconciliation is performed between the old and the new databases, using th2.

Data Landing

At the data landing stage, th2 and Python-enabled scripts are used to perform the functional testing of the financial system's data submission to the Message Broker storage. Test data is *randomly* generated to cover all data variations and edge-case scenarios and reconciled with the results that the system received for 'Raw data' and 'Processed data' areas, using the Exactpro reconciliation testing approaches. This step covers data consistency, completeness and conversion checks within the expected protocol-based scenarios.

The functional testing step is followed by end-to-end test runs based on the test scripts *pre-created* by th2. The financial system's environment is connected to the Message Broker, with all daily downstream data being captured as TCP dumps and parsed into a database. This enables us to create a reconciliation database. th2 is then used to recall the data from the database and compare it against the data processed by the system.

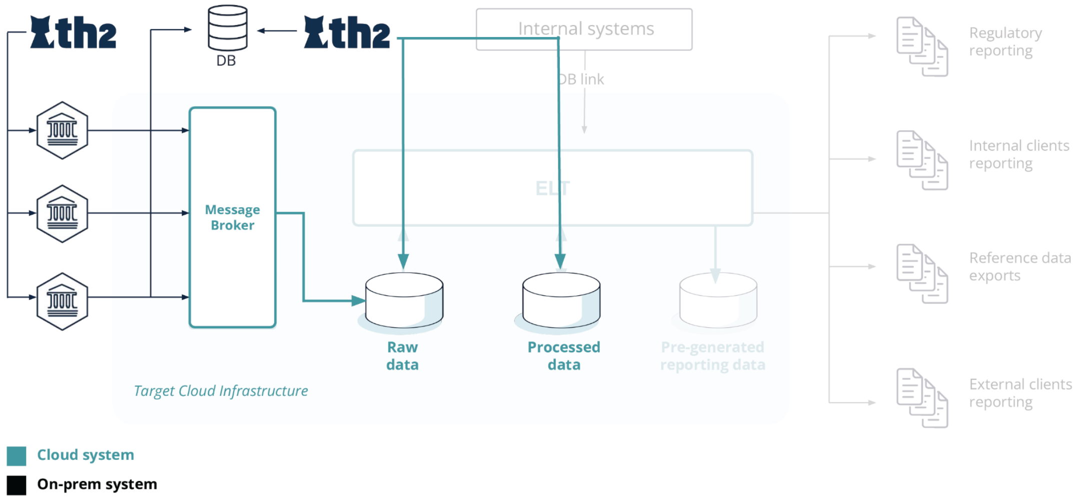


Fig. 6 Functional testing – Data Ingestion

Internal Processing and Reporting

Data processing and reporting testing is performed by way of parallel runs. With the same test library, the two test environments (one simulating the new system and one simulating the legacy system) are connected to the financial system simultaneously. The systems are working and producing test results in parallel, the resulting reports being compared to each other using th2. Tests are run in the test and pre-production environments, as well as in the production environment.

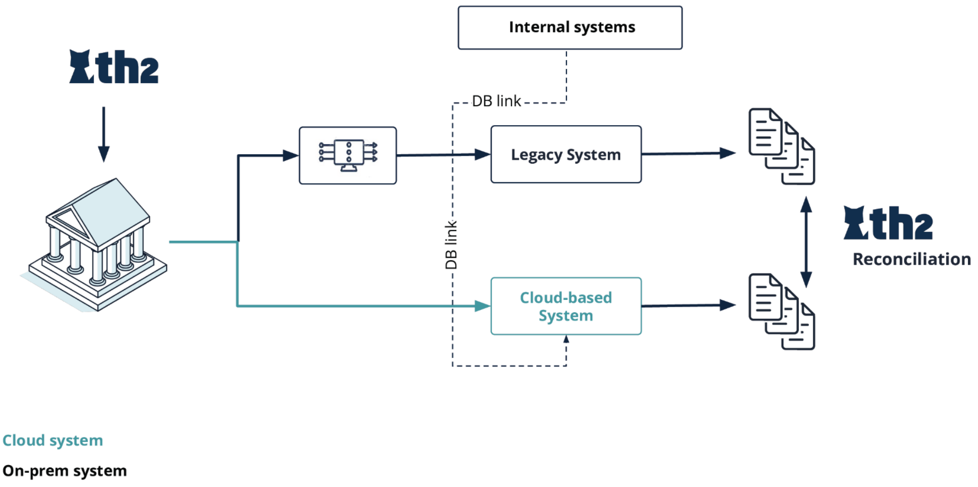


Fig. 7 Functional testing – Internal Processing and Reporting

As the new cloud-based infrastructure passes the non-production stage, it is released into the Production environment. Both systems (legacy on-prem and the new cloud-based one) are connected to the financial system's production environment, test run results are monitored live, both systems are observed and compared simultaneously (or by replaying production data in a non-production environment).

Non-Functional Testing Scope

If the nature and scope of functional testing between an on-premise and a cloud system are fairly alike, those of non-functional testing differ, at times, drastically, due to the technical characteristics of cloud-native systems.
Non-functional testing and validation checks include:

Performance & Latency tests

Latency KPIs used as a reference for the cloud setup are driven by the upstream system's KPIs for different load shapes (simulating peak demand in extreme market conditions). In the case of a downstream system (the system receiving data from the financial system), performance KPIs are identical to those of the main system. Latency tests measure the latency of a message from the moment of exiting the financial system to being recorded in the 'Pre-generated reporting data' area in the cloud data store.

Performance testing covers report generation. Compared to the on-premise configuration, the cloud setup tends to enable increased performance (from a next-day to a few-hours report generation).

Capacity tests

Warehouse Capacity KPIs should be in line with the financial system's metrics, in other words, the downstream system should be able to store and process all the data expected to come from the upstream system (the financial system itself).

The following non-functional testing checks are highly dependent on the particular cloud environment configuration – the challenge we address further in the Cloud Limitations and Their Repercussions section.

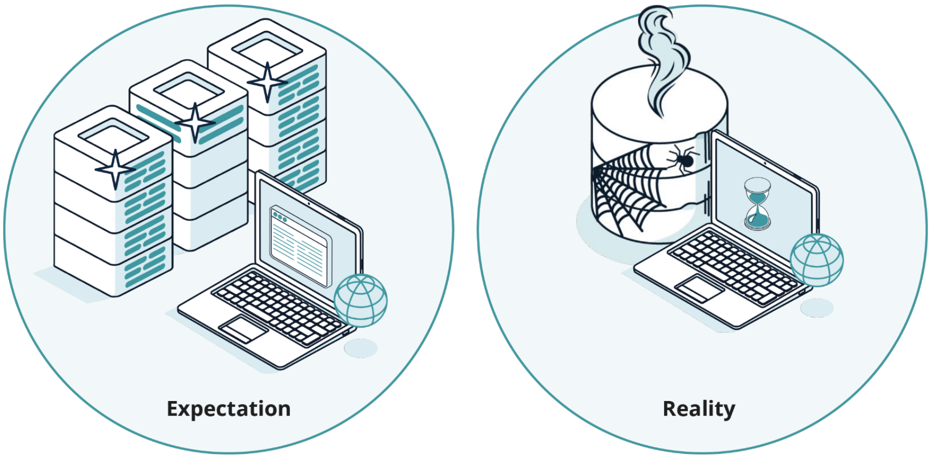


Fig. 8 *Expectation vs reality in cloud resource allocationInternal Processing and Reporting*

Resiliency tests (site failover – both financial system and warehouse – network failovers, etc.)

These ensure the system is resilient to failures and can recover on its own, when needed. It should also be able to efficiently and effectively operate from a secondary geographic region in case of failure of the primary system.

Resource provisioning and scalability

Cloud infrastructures provide the flexibility to scale up and down as required, to minimise idle resources/components. The downside, however, is that due to the fact that resources are shared among many participants, the service provider may run out of the available underlying hardware to use for deployment at any given time. So resource provisioning and scalability tests focus on the architectural configuration, while the provisioning at any given time is subject to the conditions of the subscription plan and/or the general availability of resources at that time.

Cloud components testing for:

- Data streaming/message brokering services
- Stream processing and batch processing framework (e.g. Apache Flink)
- Virtual machines
- Computing components

The checks include deactivation and restart of separate components while monitoring the entire system for failures (chaos simulation). These checks are limited by the access rights a technology provider can have in the cloud infrastructure. For example, the network between components cannot be down due to the cloud provider's restrictions, so verifying system behaviour in a situation of an inter-component network connection outage would be impossible, even using 3rd-party chaos toolkits.

Data storage testing (replication and tiering).

Daily Life Cycle – the timing and order of all schedule-dependent processes is verified.

Monitoring – checks for alerts and dashboard activity.

Cloud Limitations and Their Repercussions

Infrastructure limits

The underlying hardware is neither owned nor managed by the client firm, hence:

- After successful completion of resource provisioning testing, there is no guarantee that resources will always be available for the client's purposes: the infrastructure is not infinitely elastic and a request for additional resources may not be satisfied at once.
- There is little or no control over cloud components or the network bandwidth between them, which limits resiliency testing and confirmation testing for resiliency improvements.
- For various cloud services, the underlying infrastructure is assigned randomly and can have different performance or capacity characteristics.
- There is no visibility on the versions or release plans for the underlying infrastructure or software, which, from the regression testing perspective, implies a necessity of creating additional automated regression test suites factoring in such upgrades.
- Sporadic access or network issues may cause a failure with input/output (I/O) operations, hence, it is highly recommended to always include retry logic which, in its turn, is difficult to simulate and test.

Per-account limits and quotas

Activities in other environments within the same cloud account (or, in the worst case, within the entire cloud region) might affect performance figures. For example, a functional testing and a non-functional test environment deployed in one account will have to compete for resources.

Increased testing costs

Large volumes of transactions required for continuous end-to-end testing generate massive amounts of traffic, which can become costly. Some cloud components are charged for uptime, others – for the number of calls/runs. The configuration and number of test environments should be continuously assessed and optimised.

Pre-conditions for the Start of Testing Activities

- Access to the on-premise system under test and specialists capable of responding to technical and business questions about the system
- Access to the system’s Project/Defects tracker
- Allocated Dev/DevOps resources for support of test execution
- Dedicated cloud testing environments for:
    - Functional Testing
    - Non-Functional Testing
    - Integration Testing
- A virtual machine in the cloud provider’s environment for Exactpro test tools
- Connectivity between the cloud provider and on-premise test environments

FAQ

Why do cloud migrations not achieve the required results?

- **Underestimated costs of running a cloud infrastructure.** With a cloud migration, it is true that a significant part of CapEx (cost of building an on-prem data centre) is replaced with OpEx (monthly cost of running the cloud infrastructure) and less in-advance investments required, however, it is extremely important to carefully estimate the monthly expenses for the cloud services ahead of the migration.
- **Over-reliance on a single cloud services provider.** There is no perfect software or hardware, hence, while designing the system, it is vital to ensure proper monitoring and error/failure handling and prevention mechanisms. Cloud technology providers talk about ‘shared responsibility’ or ‘division of responsibility’ as a term signifying the necessity to mitigate such risks. It is essential to understand who owns responsibility for each aspect of the cloud system in the model your organisation is adopting (PaaS, IaaS, SaaS).
- **‘On-prem mindset’.** Running a system in the cloud effectively and efficiently requires a simultaneous comprehensive shift in various processes – Design, Dev, DevOps, QA.

What are ways to avoid data loss during a data warehouse migration?

- Before disabling the on-prem data store, comprehensive verification of the migrated data should be performed and sufficient (based on the criticality of data) data replication enabled.

If you have more questions about aspects of enterprise cloud migrations, talk to us via info@exactpro.com, we will be happy to set up a discussion.