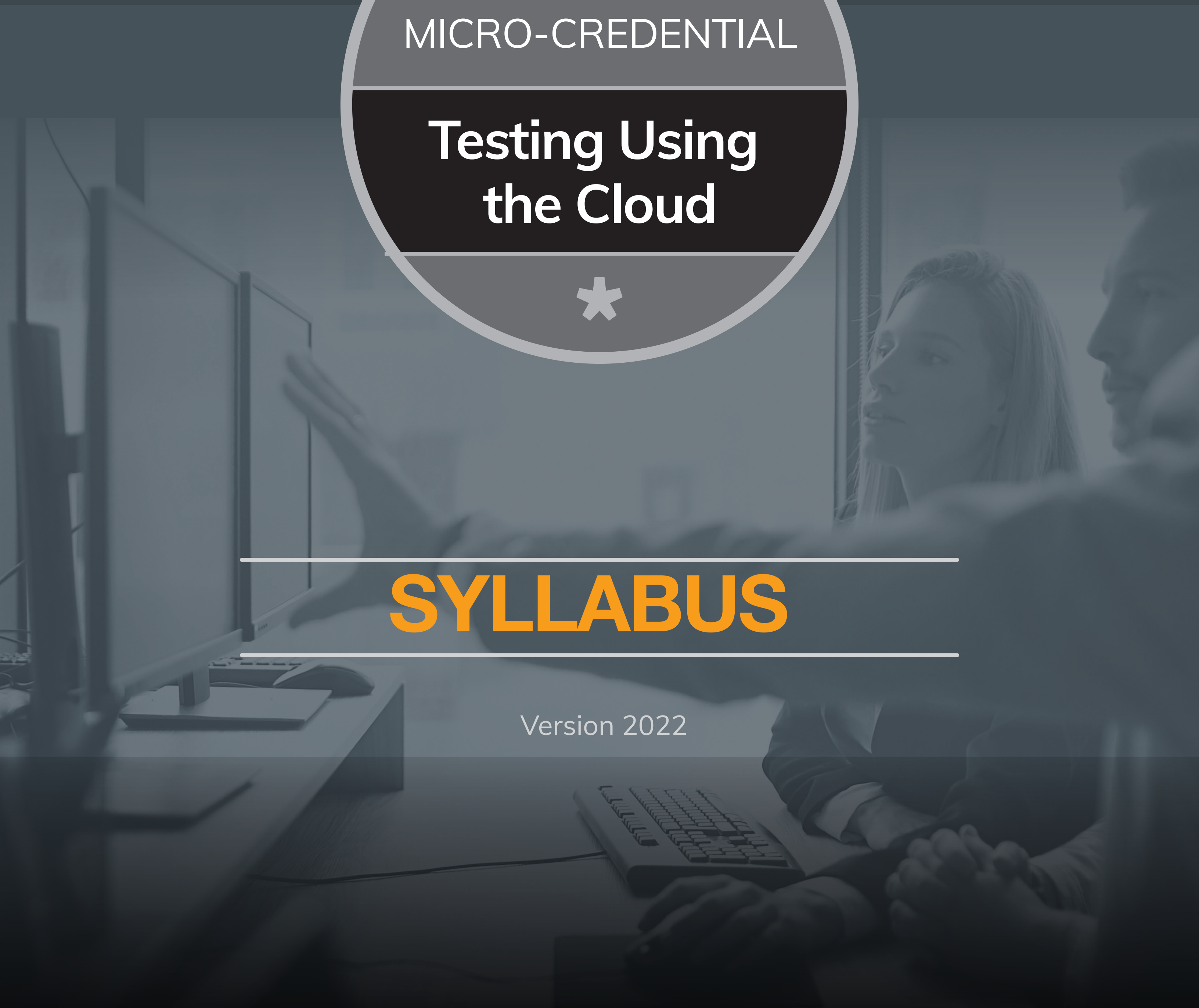
AT*SQA

MICRO-CREDENTIAL

Testing Using the Cloud



SYLLABUS

Version 2022

AT*SQA

ASSOCIATION FOR TESTING &
SOFTWARE QUALITY ASSURANCE

Global Certification Body for ISTQB and ASTQB

AT*SQA Syllabus

Testing Using the Cloud

Version 2022

Table of Contents

0. Introduction to this Syllabus

0.1. Purpose of this Document

This syllabus forms the body of knowledge the AT*SQA micro-credential Testing Using the Cloud.
AT*SQA is a non-profit organization for software testers that supports and disseminates new and relevant knowledge about research and practice in software testing. AT*SQA provides this syllabus as follows:

1. To training providers, to produce courseware and determine appropriate teaching methods.
2. To certification candidates, to prepare for the exam (as part of a training course or independently).
3. To the international software and systems engineering community, to advance the profession of software and systems testing and as a basis for books and articles.

AT*SQA may allow other entities to use this syllabus for other purposes, provided they seek and obtain prior written permission.

1. Introduction to Cloud Testing - 20 mins.

Keywords

Cloud, On-Premise, SaaS, PaaS, IaaS, TaaS, External Service Provider, Cloud Tools,

Learning Objectives

1.2 Definition of Testing in the Cloud
LO-1.2 (K2) Explain the features of testing in the cloud that make it unique

1.3 When is Testing in the Cloud Used?
LO-1.3 (K1) Recall when testing using a cloud is preferred to on-premise testing

1.4 Benefits of Testing in the Cloud
LO-1.4 (K1) Recall the benefits of testing in the cloud for business value
LO-1.5 (K1) Recall the benefits of testing in the cloud

1.5 Cloud Testing Tools
LO-1.4 (K1) Recall the benefits and options when using a Cloud testing tool

1.1 Introduction to Testing in the Cloud

This Testing in the Cloud course covers specific information on testing techniques, advantages, disadvantages, challenges, and tools available when executing testing in a Cloud. Testing in a Cloud may be performed on-site, or anywhere with an internet connection and the ability to authenticate into a service provider. Unlike testing on-premise, internally managed servers and databases, testing in the cloud offers flexibility and, often, overhead cost reductions for a software development business.

1.2 Definition of Testing in the Cloud

Cloud testing is using a remote server to access data and other services, or even a complete software system. Software development businesses can test internally on a cloud structure or externally using a separate cloud service provider.

Cloud providers manage and provide a service. There are four main types of cloud service bundles:

- SaaS – Software as a Service
- PaaS – Platform as a Service
- IaaS – Infrastructure as a Service
- TaaS – Testing or Tech as a Service

SaaS or “Software as a Service” are applications hosted and managed by a cloud service provider (also known as “third party” providers) that are accessed as members (customers). SaaS software is third party software used that is not hosted or managed internally. Most of the SaaS options offer a free version with basic services and a variety of paid versions priced based on what and how much the business uses.

For example, one can sign up for a free Zoom account but the time limit for a meeting or call is 40 minutes. One may sign up for several paid versions based on the individual or business need.

Some SaaS examples include:

- Zoom
- Microsoft Office 365
- Dropbox
- Salesforce
- Google Apps
- Slack

These are only a few of thousands of options available.

Similarly, PaaS or “Platform as a Service “provides a software platform to use for software development or business management functions. PaaS services are hardware, software, or a combination of both. For example, a business may use Microsoft Azure as its platform to develop software applications, APIs, and databases. A few examples of PaaS providers include:

- Microsoft Azure
- SAP Cloud
- AWS Lambda
- IBM Cloud Foundry

IaaS services provide the infrastructure tools for data storage, networking, and virtualization.
Like most cloud services, businesses pay only for what they need and use. All of the infrastructure needs are managed by the service provider. Examples of IaaS providers include:

- AWS EC2
- Rackspace
- GCE (Google Compute Engine)
- Digital Ocean

TaaS - Testing as a service means a provider provides the testers, and all the testing deliverables, manages test execution and outcome reporting. TaaS also refers to Crowdsourcing test providers as well as test services companies that provide the QA testing talent and manage testing as a project.

Additionally, TaaS providers give access to members to test on a variety of mobile devices, desktops, laptops, and tablets. The TaaS provider supplies all the devices testers need to conduct testing without all the tasks associated with managing the devices when physically present. This is a virtual implementation of a physical testing lab where a variety of release builds were made available on separate devices with varying platforms and configurations.

A few examples of TaaS providers include:

- Browserstack
- Applause
- Testlio
- QA Wolf
- Mindful QA
- TestCrew

TaaS is also used to refer to “Transportation as a Service”, to cover service providers like Uber and Lyft.

Additionally, TaaS may be referred to as “Tech as a Service” to cover businesses that provide user assistance for hardware and infrastructure items. This may take the form of personal help desk support for questions, installation, and device maintenance. Neither of these options fall within the scope of this syllabus.

1.3 When is Cloud Testing Used?

Software development businesses use cloud testing when they need to extend resources, or spend less time managing hardware, data, and server systems and more time developing and testing. Testing in the cloud typically reduces the overhead cost of managing internal servers, databases, and related systems. Additionally, testing in the cloud is used when ease of access is necessary as resources are available from anywhere with an internet connection.

Testing in the cloud increases the flexibility of work without the overhead of additional management costs. Cloud services provide adaptable cost menus to apply to nearly any business budget based on use so using a cloud service typically saves a business time and money.

Testing in the cloud for a QA team improves access to devices, platforms, and configurations to expand test coverage. Testing coverage expands in a variety of ways including being able to test compatibility, performance, and others. For example, when testing mobile and tablet devices using a cloud provider, testers can access a wide variety of devices of varying platforms without the need to maintain them. Compatibility and device platform testing for mobile apps expands based on the variety of devices available.

1.4 Benefits of Testing in the Cloud

The benefits of testing in the cloud include:

- Reduced cost
- Availability of scalable resources
- Faster setup for test execution

- Ability to customize environments and expand test coverage
- Increased test coverage based on access to a variety of hardware or devices

Testing in the cloud gives software testers unique opportunities to expand testing without having to spend testing time managing or configuring servers, devices, or databases.

Additionally, if testing service is used to compliment or perform testing, using the service is typically on a per-use, or as-needed basis so there is less testing cost overhead. When a business uses a testing service, both the testing, test development, and execution are managed by the service provider. The business only needs to await the reported results, analyze them, and fix any defects as desired. Finally, for the overall business, there is a reduced need for the IT group to support and manage a variety of testing platforms and devices.

When using a testing service, software testing becomes more like a scheduled project activity rather than an internally managed employee group. However, not having QA testers on an internal team can be a disadvantage to using a test service.

As an example, the need to communicate and collaborate with the development and product management teams is often beneficial. QA testers who work alongside the rest of the development team are more effective, especially for new feature development.

In the TaaS model, communication issues may be a concern when QA testers are external to the development team as it may require more time and effort to ask questions and get answers when needed.

Keep in mind when testing in the cloud, there are often internal testers testing applications using a cloud service provider's resources or applications located in the cloud. There are also TaaS service providers who provide testers and test project management. An organization may use one of these approaches or a combination of both.

1.5 Cloud Based Testing Tools

Testing in the cloud often involves working with a variety of cloud testing tools. As with any development task, there are an abundant number of useful and helpful tools to assist the team. Cloud testing tools provide a configurable platform for running manual and automated testing. Most cloud testing tools allow testers to change platforms, devices, and server load to cover nearly all types of testing.

An advantage of using a cloud testing tool is to free the testing team from the task of managing devices, increasing operating system (OS) coverage, increased coverage browser variations, and skipping the additional setup and maintenance managing all the testing parts implies. Instead, the testing team can focus on testing.

Most cloud testing tools are for load, performance, security, or provide a device farm (or multiple devices and browsers). The following is a short list of available cloud-based testing tools:

- Browserstack
- Xamarin Test Cloud
- testsigma
- AWS Appthwack
- watir
- tenable

2. Fundamentals - 60 mins.

Keywords:

Platform, Server, Database, On-Premise, Security, Load, Performance, Device Farm, Application, Infrastructure, Network, Data

Learning Objectives

2.1 Cloud Testing vs. On-Premise Testing
LO-2.1.a (K1) Recall the differences between on-premise testing platforms and cloud testing platforms

2.2 Core Cloud Testing Components
LO-2.2.a (K1) Recall the core cloud testing components involved in testing in the cloud

2.3 Cloud Testing Types
LO-2.3.a (K1) Recall the types of testing performed on a cloud base
LO-2.3.b (K2) Explain each type of testing and what it means when software testing

2.1. Cloud vs. On-Premise Testing

For **On-Premise** testing, the testing platform or servers, databases, and backend connections are:

- Servers in a known location, like a data center or housed and managed internally by the business.
- Database storage is in a known location and managed internally by the business.
- Application interaction and transactions occur between known servers and databases that are internally managed.

For **Cloud-based** testing, the testing platform, servers, database, and backend connections are accessible using an internet connection and:

- Cloud service provider houses the servers and databases. Databases and servers, as well as other platform components, may be located anywhere around the globe.
- The testing platform, configuration and maintenance are managed by the cloud provider.
- All application interactions and transactions occur between the cloud and calls coming from the application (business).
- Little internal management is required for any component within a cloud testing platform. The main responsibility of the tester is to ensure the correct versions are installed on virtual devices.

Testing in a cloud platform can be internal or used with on-site testing teams as well as external teams or TaaS service providers.

2.2. Core Cloud Testing Components

Testing in a cloud-based system or application allows for testing within the four core cloud-based system components:

- Application
- Infrastructure
- Network
- Data

Testing the application core component refers to functional and feature testing of an application that is hosted in the cloud.

The data component ties into application testing as most software applications require data access and the manipulation (add, change, delete and query) of data to function. The cloud testing platform may allow testers to create test data specific for testing the application. In other cases, test data may be external and supplied to the cloud-based application.

Testing infrastructure is a difficult task in most cases, but far less work is needed when done in a cloud platform. The reason for this is that most of the configuration or setup is done by the cloud provider.

Testing the network essentially means testing a combination of integration, performance, and load testing against the entire application including the infrastructure, hardware, and connections. Similar to testing cloud-based infrastructure, much of the responsibility of network configuration and maintenance is performed by the cloud provider.

2.3. Cloud Testing Types

There are several types of testing that benefit from a cloud testing structure. Software testers can perform any type of testing on a cloud system depending on the provided accessibility and configuration.

The following types of testing can be performed on a cloud system:

- Functional
- Integration
- Interoperability
- Performance (Load and Stress Testing)
- Compatibility
- Multi-tenancy
- Security

Testing multi-tenancy and compatibility capabilities increase when using the cloud. The cloud platform allows multiple users to log in anywhere with an internet connection and valid credentials.

Additionally, when performing compatibility testing, test efficiency improves as the testers can access a managed group of real or virtual devices and platforms without the need to have a device in hand. Cloud-based compatibility testing has significant benefits, especially considering the extremely high numbers of possible mobile computing devices and configurations.

3. Testing Types - 60 mins.

Keywords:

Functional, System, Load, Performance, Security, Multi-Tenancy, Compatibility

Learning Objectives

3.1 Functional Testing

LO-3.1.a (K1) Recall the definition of functional testing
LO-3.1.b (K2) Understand how functional testing is performed when testing in a cloud

3.2 Integration Testing

LO-3.2.a (K1) Recall the definition of integration testing
LO-3.2.b (K2) Understand how integration testing is performed when testing in a cloud

3.3 Interoperability Testing

LO-3.3.a (K1) Recall the definition of interoperability testing
LO-3.3.b (K2) Understand how interoperability testing is performed when testing in a cloud

3.4 Load & Performance Testing

LO-3.4.a (K1) Recall the definition of performance testing
LO-3.4.b (K2) Understand how load and performance testing are performed when testing in a cloud

3.5 Compatibility Testing

LO-3.5.a (K1) Recall the definition of compatibility testing
LO-3.5.b (K2) Understand how compatibility testing is performed when testing in a cloud

3.6 Multi-Tenancy Testing

LO-3.6.a (K1) Recall the definition of multi-tenancy testing
LO-3.6.b (K2) Understand how multi-tenancy testing is performed when testing in a cloud

3.1 Functional Testing

Functional testing ensures the application's features and functions work as expected or intended for the user. Functional testing of a cloud-based system or applications allows functional test execution to occur using a test environment managed by the cloud provider.

In other words, functional testing seeks to determine if the application perform functions as required. For example, if testing an application that creates tax forms and files annual taxes, functional testing ensures the calculations are accurate and current tax laws are applied correctly to allow the user to correctly file annual taxes.

In functional testing, a tester designs and executes functional test cases that involve creating an account or designating the tax filer's personal ID, filling out financial forms, and creating income statements. All of the data would be saved and used in calculations to tell the user how much they must pay or how much they get returned to them.

3.2 Integration Testing

Integration test execution is performed the same way within the cloud as it is in an on-premise approach. The only difference is where to access the integration pieces that form the application. For example, cloud-based integration testing involves finding the location of the APIs, databases, connections for messaging or data transfer, and any third-party software.

Similar to functional testing, the software tester determines what integrated pieces to test within the backend systems and any third-party applications. One difference for cloud-based integration testing may be accessibility to APIs, databases, or servers. Since cloud-based components are managed by the cloud provider, testers may need to plan a different approach for testing integration compared to on-premise integration testing.

3.3 Interoperability Testing

Interoperability testing is testing with partners, or third-party applications or services, including microservices. The objective of interoperability testing is to assess the correctness of processing after integration has been achieved. Interoperability testing is similar to integration testing in that it involves coordinating testing activities with testers, developers, or other development team members from partner companies.

For example, when testing a messaging system for a healthcare application that interoperates with a pharmacy application, testing sessions are scheduled when testers from all organizations verify their application's functionality with each other. Typically, interoperability testing is executed as a group activity.

Interoperability testing in a cloud-based approach offers several advantages including:

- Access and ability to change configuration settings for testing
- Access and ability to switch on and off configured systems
- Ability to add in, or have partner applications installed for testing by the cloud provider
- Access for different organizations to log in for testing

3.4 Performance Testing

Performance testing provides the ability to test against the network setup, the database, and the application itself. Performance tests are executed with a series of loads or performance baseline times to determine if the application processing working as expected in terms or response times and/or throughput levels.

The intent of performance testing is to determine where bottlenecks might occur when the application is in use. In load testing, testers can vary the number of users or load on the system to determine the ability of the system to handle certain levels of user and/or data load. In stress testing, the goal is to identify breaking points due to poor performance in the application, database, and the network itself.

Testing performance in a cloud-based system has points in common with performance testing of an on-premise system. For example, the design of the performance tests often involves the same workflow and user levels regardless of cloud or on-premise.

Cloud-based performance testing allows for the ability to generate and sustain much greater levels of simulated user load. Cloud-based test environment make this possible without the high cost of physical test environments.

Because of the need to generate and sustain high levels of simulated load on a system, tools are required for performance testing. One of the great advantages of cloud-based performance testing is the ability to scale and perform simulate user load quickly and easily. Cloud-based performance test tools can also simulate and measure load originating anyplace in the world.

3.5 Compatibility Testing

Compatibility testing is executing tests against different versions of the application in combination with varying versions of dependent software or hardware. For example, a common testing practice is to test on multiple browsers for a web or mobile application. The tester executes functional tests by using the application in different browsers looking for discrepancies in application functionality based on the browser version or type.

Compatibility testing in a cloud-based test environment is significantly advantageous. The ability to test by simply switching browsers or devices within a cloud saves time and cost. The ability to easily switch platforms, configurations, devices, or browsers for example allows for expanded test coverage based on saving time from not having to configure and manage separate devices, platforms and versions.

3.6 Multi-Tenancy Testing

Multi-tenancy testing is testing an application's response when more than one user logs in at a time. Using a cloud system, it's simpler to be able to log in and test with multiple users testing with various configurations, devices, and platforms all at one time. Multi-tenancy testing focuses on determining how the application as a system handles multiple logins as well as multiple users performing functions at the same time.

Similar to compatibility testing, using a cloud for multi-tenancy testing is significantly advantageous. Crowdsourcing is essentially mass multi-tenancy testing. It's an effective method of efficiently testing without having to organize planned group testing sessions.

3.7 Security Testing

Cloud computing opens many new opportunities for attackers to compromise applications, networks and data.

Security testing of cloud applications is a major way to know if the defenses of digital assets are working as designed and are effective in protecting those assets from being stolen or otherwise compromised.

One main concern of cloud security is knowing where the data is actually residing in the cloud. The uncertainty around this one concern as data center security becomes the responsibility of the cloud service provider, but if the data is lost or stolen, the corporate cloud customer may be legally responsible. See 4.1 for more about cloud data security and testing.

4. Cloud Testing Challenges - 10 mins.

Keywords:

Architecture, Application Integration, Network, Infrastructure, Configuration, Data Security, Data Privacy, Interface, Data Migration

Learning Objectives

4.1 Data Security & Privacy
LO-4.1.a (K1) Recall the reasons data security and data privacy may be challenging in a cloud

4.2 Interface Compatibility
LO-4.2.a (K1) Recall the reason interface compatibility may be challenging in a cloud

4.3 Data Migration
LO-4.3.a (K1) Recall why data migration may be challenging in a cloud

4.4 Application Integration
LO-4.4.a (K1) Recall why application integration may be challenging in a cloud

4.1. Data Security & Privacy

Data security and privacy mean keeping application and company data from being stolen or accessed without authorization. Data stored in non-cloud systems or locally is typically secured by server security, firewalls, monitoring, and security testing. At the beginning of the cloud era, data security and privacy were not as advanced as today. With data security a prime security interest for everyone both individual and businesses alike, cloud providers have increased data security and privacy methods to ensure customer data remains open to authorized users only.

Additionally, cloud service providers (CSPs) are at the forefront of the development of security tools to protect their clouds and keep up to date on the latest malware, intrusion protection, firewalls, monitoring, and event tracking. Many CSPs are employing the use of machine learning and artificial intelligence to enhance security practices.

CSPs provide more security coverage than storing data locally on protected servers. It’s the livelihood or the business of the CSP to secure customer data

Data security and privacy are a risk, always. As a software tester incorporate as much data security testing as possible into test execution suites. In some cases, security testing may be performed by specialists.

4.2. Interface Compatibility

Interface compatibility means testing to determine if different versions of browsers, hardware platforms, devices, and backend interface systems (messaging) function as expected when working together. Testing tasks may include switching backend interface systems to different products or versions and retesting. This kind of testing might be seen as creating a test menu. The tester configures a system that likely mimics a customer system and tests against it. Then, they switch out interface products and retest. Next, the tester may switch browsers or OS platforms.

In a cloud-managed system there may be limited options based on what is provided.

4.3. Data Migration

Migrating data from a local server to a cloud server database is a time-consuming task. The process of extracting, transforming, and loading data (ETL) takes time depending on the size of the database. Additionally, once the data is loaded then the database needs tested to ensure the accuracy of the transfer.

Time and resources are required to ensure an accurate data migration before the new database is ready for use. Typically, this requires the assistance of a database administrator.

4.4. Application Integration

Application integration may be a challenge in a cloud environment. Depending on the cloud configuration, testers may have not have access to all the application versions needed. The challenge in testing application integration is in planning and paying for all the applications, including third-party applications that need testing coverage. In order to help mitigate the challenge, ensure all required applications and versions are present for testing access.

5. Trademarks

The following registered trademarks and service marks are used in this document.
AT*SQA™ is a trademark of the Association for Testing and Software Quality Assurance.

6. References

1. https://www.techtarget.com/searchstorage/definition/cloud-testing

2. https://www.guru99.com/cloud-testing-tutorial-with-saas-testing-primer.html