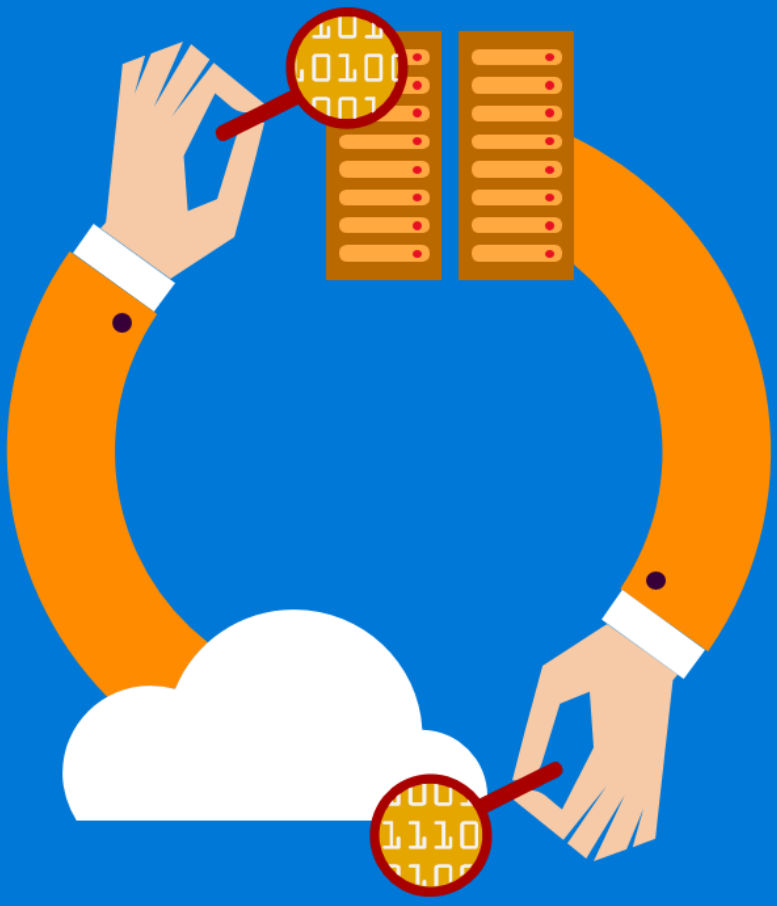
Microsoft



Demystifying mainframe to Azure migration

By Larry Mead
AzureCAT

October 2018

Contents

Authored by Larry Mead. Edited by John Kaiser. Reviewed by Benjamin Guinebertière.

Introduction

Organizations relying on mainframe systems face an increasing urgency to transition workloads to the cloud. Faced with a rapidly shrinking talent pool of mainframe expertise amid escalating costs, organizations stand to save substantial expenditures—if they can migrate equivalent functionality to Microsoft Azure.

Migrating legacy mainframe deployments that stretch back decades is a daunting task and many companies don't know where to start. Mainframe systems remain deeply entrenched across many organizations, thanks in part to an almost mythical aura spawned from their use in Cold War missile deployments and the Apollo moon landing or portrayal in cult movie classics.

Taking the myth out of the equation goes a long way to getting started. Think of it as a prerequisite. And knowing which systems are most easily migrated is the first step. This paper covers the major IBM z/OS mainframe components alongside their Azure equivalents and is designed to help mainframe customers complete this critical first step.

In many cases, mainframe workloads can be replicated in Azure with no loss of functionality and without users even noticing changes in their underlying systems. In other instances, customers may need to leave troublesome mainframe workloads unchanged while migrating all other workloads to the cloud.

Intended audience

Technical consultants, partners, or anyone interested in a high-level comparison of mainframe and cloud solutions.

What is a mainframe?

Mainframes were designed as scale-up servers to run high-volume online transactions and batch processing in the late 1950s. As such, mainframes have software for online transaction forms (sometimes called green screens) and high-performance I/0 systems for processing the batch runs.

Mainframes have a reputation for high reliability and availability, in addition to their ability to run online and batch jobs.

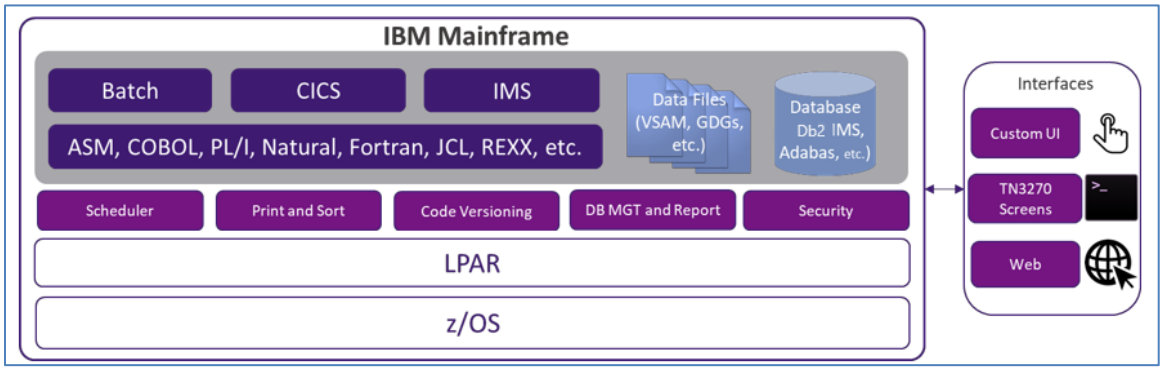


Figure 1. IBM mainframe components

Mainframe myths and facts

Over the years, a mystique has grown around mainframes related to their reliability and availability, in addition to their processing power.

Table 1. Mainframe myths

Myth	Reality
Mainframes never go down and have a minimum of five 9s of availability.	Mainframe hardware and z/OS are viewed as reliable and stable. However, mainframes must typically schedule downtime for maintenance and reboots, (referred to as initial program loads or IPLs). When these tasks are considered, a mainframe solution often has closer to two or three 9s of availability—on par with high-end, Intel-based servers. Finally, mainframes remain as vulnerable to disasters as Intel servers and require uninterruptible power supply (UPS) systems to handle these types of failures.
Mainframes have limitless scalability.	A mainframe's scalability depends on the capacity of its system software such as the Customer Information Control System (CICS) and the capacity for new instances of mainframe engines and storage. Some larger companies using mainframes have customized their CICS for performance and otherwise outgrown the capability of the largest available mainframes.
Intel-based servers are not as powerful as mainframes.	New core-dense, Intel-based systems have as much compute capacity as mainframes.
The cloud cannot accommodate mission-critical applications at places like large financial institutions.	Although there may be isolated instances where cloud solutions fall short (such as some algorithms that can't be distributed) these are exceptions rather than the rule.

Table 2. Mainframe facts

Fact	Reality
Mainframes using CICS can handle substantial numbers of users with minimal compute power.	While CICS and similar teleprocessing (TP) monitors use a stateless design allowing for many concurrent connections, they're typically limited to green screen terminals.
Mainframes have highly efficient I/O and can process large batch workloads.	The mainframe's I/O subsystem is highly optimized for throughput and minimal latency.
Mainframes are expensive to operate.	Everything about mainframes incurs excessive costs. They're expensive to acquire, require costly software, and consume copious amounts of power. Additionally, mainframe personnel command high salaries.

Mainframe strengths

Mainframes figure prominently in the history of computing and remain viable for highly specific workloads. Most agree mainframes have the following strengths:

- **Proven platform.** Mainframes scale well and provide reliable performance, assuming there are programs and developers available to design and run mainframe programs.
- **Reliable.** Long-established operating procedures, including mainframe job control language (JCL) and scheduling, make robust operating environments.
- **Usage-based.** Software can run based on usage, measured by million instructions per second (MIPS) and provide extensive usage reports for charge backs.

Mainframe weaknesses

Mainframe disadvantages result in excessive costs that are increasingly difficult to justify. Most agree mainframes have the following weaknesses:

- **Shrinking talent pool.** Traditionally expensive to operate, costs are escalating as experienced mainframe resources become increasingly scarce.
- **Outdated programming.** Computing languages used by most mainframes (COBOL, Fortran, PL/I, Natural) are considered obsolete. This makes finding or building innovative solutions for the mainframe difficult.
- **High costs for hardware/storage** Mainframes are very expensive to "scale out."

Why Azure makes sense for mainframe applications

Customers are clearly looking for an alternative platform capable of delivering equivalent mainframe functionality and features. As Azure has matured, its viability as a replacement for mainframe workloads has risen accordingly. Simply put, Azure provides the mainframe-like features that customers want, without the drawbacks often associated with mainframes. Azure benefits include:

- **High availability.** With commitment-based service-level agreements (SLAs), multiple 9s availability is the default, optimized with local or geo-based replication of services.
- **Customer-centric flexibility.** Virtual machines and services scale on-demand under a usage-based billing model.
- **Full mainframe support.** Built-in and partner-based solutions support existing mainframe applications. A wide range of independent-software-vendor (ISV) and system-integrator (SI) solutions enhance existing mainframe applications. And integration with existing mainframe environments enables support for hybrid solutions and migrations over time.

MIPS vs. vCPUs

MIPS provides a constant value of the number of cycles per second for a given machine. MIPS are used to measure the overall compute power of a mainframe. Mainframe vendors charge customers based on MIPS usage and customers can increase mainframe capacity to meet specific requirements. IBM maintains a processor capacity index showing the relative capacity across different mainframes.

Table 3 shows typical MIPS thresholds across small, medium, and large enterprise organizations (SORGs, MORGs, and LORGs).

Table 3. MIPS size across organizations

Customer size	Typical MIPS usage
SORG	Less than 500 MIPS
MORG	500 MIPS–5,000 MIPS
LORG	More than 5,000 MIPS

Although no universal mapping formula exists for determining the virtual central processing units (vCPUs) needed to run mainframe workloads, MIPS are often mapped to vCPUs on Azure. An accurate calculation depends on two factors—the type of vCPU and the exact workload being run. However, benchmark studies provide a good basis for estimating the number and type of vCPUs customers would need. A recent HPE zREF benchmark provided the following estimates:

- 288 MIPS per Intel-based core running on HP Proliant servers for online (CICS) jobs.
- 170 MIPS per Intel core for COBOL batch jobs.

This document uses 200 MIPS per vCPU for online processing and 100 MIPS per vCPU for batch processing

NOTE: These estimates are subject to change as new virtual machine series become available in Azure.

Mainframe costs vs. Azure

The annual cost estimate on Azure ($544,920 as shown in Table 4) reveals a near tenfold savings over an annual mainframe expenditure of $5,000,000 for a 5,000 MIP deployment based on $1,000 per MIP. Table 4 shows estimated annual costs on Azure after migrating a 5,000 MIPS mainframe deployment.

Table 4. Estimated Azure cost of migrated MIPS deployment

Workload	Azure type	Virtual machine number and type	Monthly cost
Database (DB)	F32 (32 cores)	One – primary virtual machine One – high availability (HA) virtual machine One – backup/disaster recovery (BU/DR) virtual machine	$19,396
Apps	G16 (16 cores)	One – primary virtual machine One – HA virtual machine One – BU/DR virtual machine	$10,892
Utilities	D8 (8 cores)	One – primary virtual machine One – HA virtual machine One – BU/DR virtual machine	$2,232
DB	D14 v2 (16 cores)	One – primary virtual machine One – HA virtual machine	$2,205
Apps	D14 v2 (16 cores)	One – primary virtual machine One – HA virtual machine	$2,205
Utilities	A7 (8 cores)	One – primary virtual machine One – HA virtual machine	$1,488
Development # 1	A6 (4 cores)	Two primary virtual machines	$248
Development # 2	A7 (8 cores)	Two – primary virtual machines	$496
Development # 3	A11 (16 cores)	Three – primary virtual machines	$1,161

Primary storage	P30 (SSD-page blob and disk)	Twenty - 1TB solid-state drives (SSDs)	$2,703
Virtual tape	HDD – page blob and disk	Twenty – 1TB hard disk drives (HDDs)	$1,384
		Support	$1,000
		Monthly cost	$45,410
		Annual cost	**$**544,920

Mainframe storage

Part of demystifying mainframes involves decoding various overlapping terms. For example, central storage, real memory, real storage, and main storage generally all refer to storage attached directly to the mainframe processor.

Mainframe hardware includes processors and many other devices, such as direct access storage devices (DASDs), magnetic tape drives, and several types of user consoles. Tapes and DASDs are used for system functions and by user programs.

Types of physical storage

- **Central storage.** Located directly on the mainframe processor, also known as processor storage or real storage.
- **Auxiliary storage.** Located separately from the mainframe and includes storage on DASDs, also known as paging storage.

Mainframe transaction processing

A transaction results from a piece of processing initiated by a single request, typically from a user at a terminal, but could also come from multiple other sources, including webpages, remote workstations, applications in another CICS system, or triggered automatically at a predefined time.

A transaction monitor tracks and manages all aspects of a business transaction. CICS ranks as the leading transaction management suite for the IBM z/OS and is typically used in conjunction with IBM Information Management System (IMS DC).

CICS manages the sharing of resources, the integrity of data, and prioritization of execution. CICS authorizes users, allocates resources, and passes database requests by the application to a database manager (such as Db2).

Comparing CICS and IMS DC

Created in 1969, IMS was implemented based on a hierarchical database model. By separating application programs from the physical structure and storage of the data, IMS was designed to reduce data redundancy by maintaining a single copy of the data. IMS DC complements CICS as a transaction monitor by maintaining state throughout the process, recording business functions to a data store.

Migrating mainframe workloads to Azure workloads

An expanding partner ecosystem now assists customers seeking to migrate mainframe systems to Azure. Most follow a pragmatic approach of reuse wherever possible before embarking on a phased deployment of rewriting or replacing applications.

Moving mainframe data to Azure

Migrating applications from mainframe environments typically involves one or more of the following strategies:

- **Remote host.** One of the principal benefits of the cloud is outsourcing infrastructure management.
- **Retire.** If there are any applications that are no longer needed, they can be retired before migration.
- **Reuse.** This "lift-and-shift" process reuses existing code, programs, and applications by moving them from the mainframe and recompiling the code to run in a mainframe emulator hosted in a cloud instance. It typically requires some engineering and refactoring along with data and file conversions for transitioning the database to the cloud.
- **Rewrite.** Although possible to completely rewrite programs, the added cost, complexity, and project duration precludes this option for most customers. It's more common to first move the applications to a cloud-based emulator, migrate the database to a cloud-based database, and then focus on replacing modules and code using code transformation engines.
- **Replace.** This swaps out mainframe functionality with a suite of programs such as a software-as-a-service (SaaS) application aimed at purpose-built solutions across such areas as finance, human resources, manufacturing, or enterprise resource planning. In addition, industry-specific apps may solve the problem that a custom mainframe solution was needed for decades ago.

Table 5. Transformation engine methodology[1]

Phase	Target
Discover	Code, data, software, processes, and requirements
Design	Architecture, databases, interfaces, and customization
Modernize	Code, static data, and application components
Test	Functions, data, interfaces performance, and load balance
Implement	Deploy, migrate dynamic data, system cutover, and monitor
Manage	Manage Azure environment, maintain and enhance apps

Planning the migration

Begin by planning which workloads should be initially migrated, and then determine the requirements for moving associated applications, legacy codebases, and databases.

Mapping mainframe workloads to Azure workloads

Figure 2 and Figure 3 show how workloads and applications migrate and map to Azure.

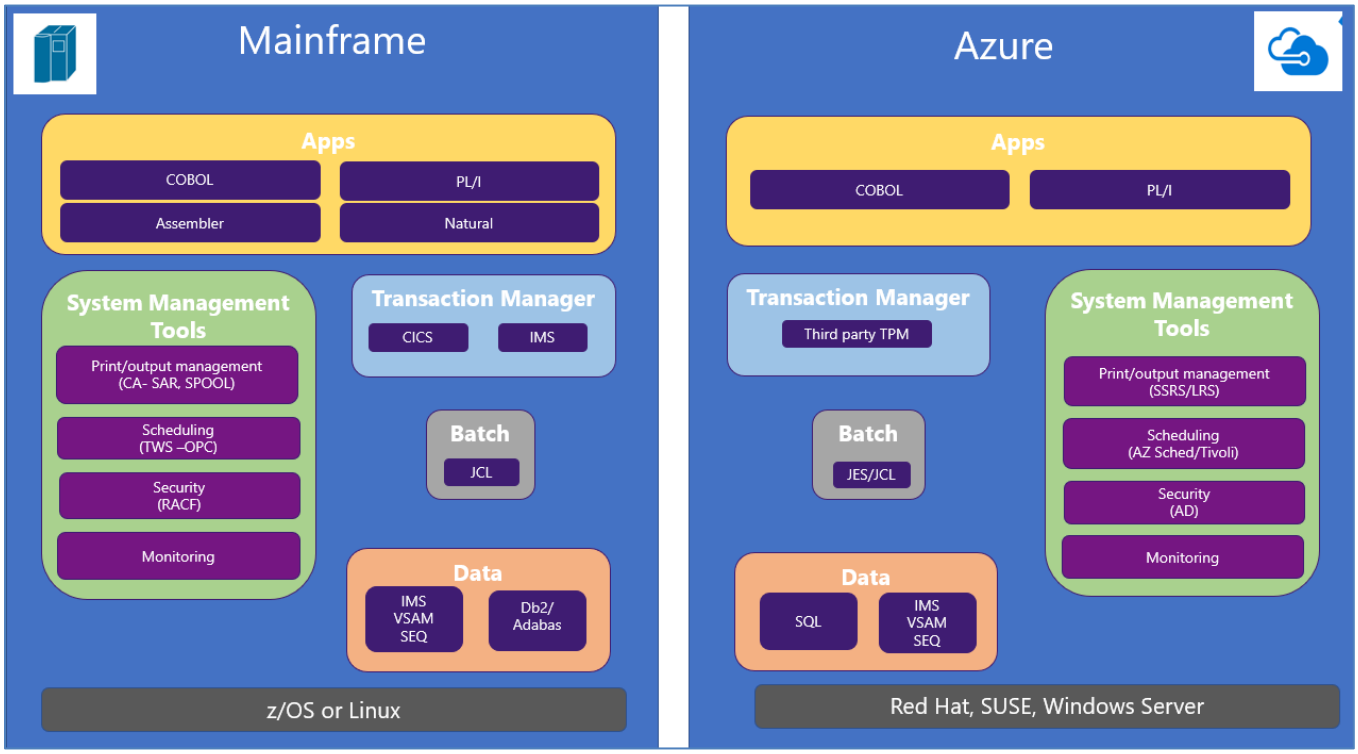


Figure 2. Rehosting workloads—before and after

[1] IBM Mainframe to Microsoft Azure Reference Architecture

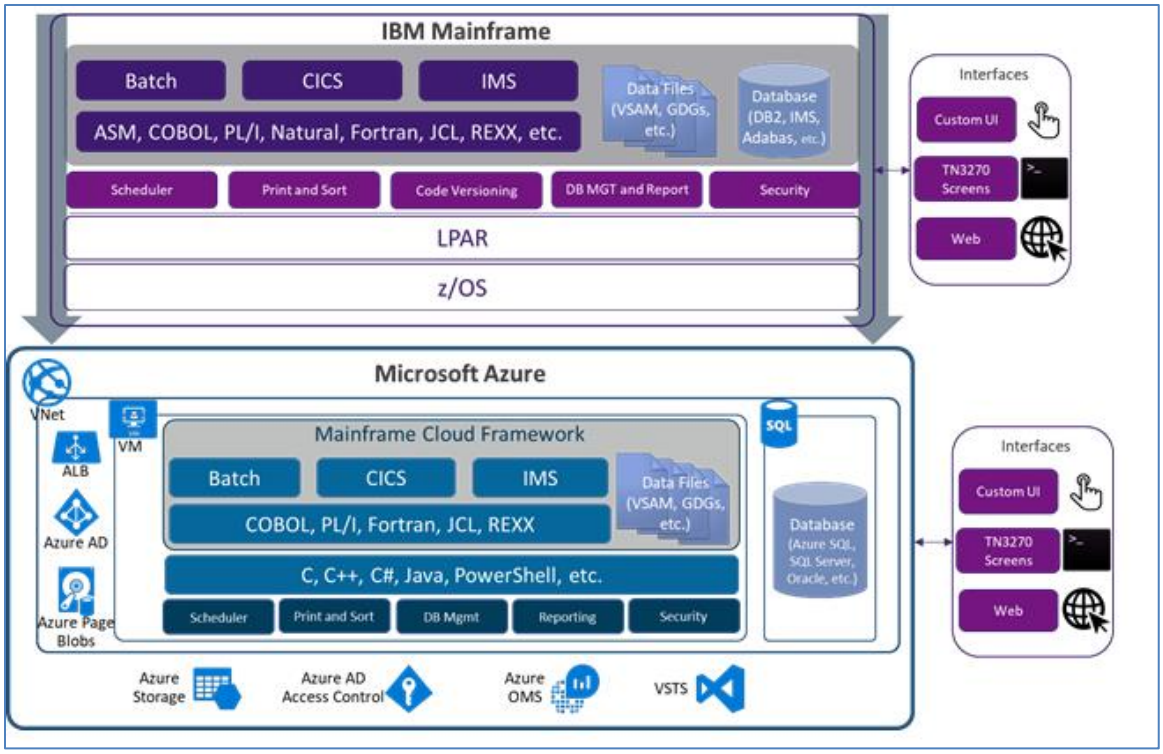


Figure 3. Migrating mainframe applications to Azure

Migrating development environments

Table 6 shows mappings for the most common programming languages.

Table 6. Code-modification mapping

Source	Target
z/OS	Azure (Windows, Linux, or UNIX)
CICS	Micro Focus, Oracle, GT Software (Fujitsu), TmaxSoft, Raincode, NTT Data, Rewrite using Kubernetes
IMS	Micro Focus, Oracle
Assembler	Raincode, TmaxSoft, COBOL, C, Java, or mapped to OS functions
JCL	JCL, PowerShell, other scripting
COBOL	COBOL, C, Java
Natural	Natural, COBOL, C, Java
FORTRAN, PL/I	FORTRAN, PL/I, COBOL, C, Java
REXX, PL/I	REXX, PowerShell, other scripting

Migrating databases

Table 7 shows targets for database mapping.

Table 7. Database-migration mapping

Source	Target
Db2	Azure SQL, SQL Server, Db2 LUW, Oracle
IMS DB	Azure SQL, SQL Server, Db2 LUW, Oracle
Virtual Storage Access Method (VSAM) and other flat files	Indexed Sequential Access Method (ISAM) flat files Azure SQL, SQL Server, Db2 LUW, Oracle
Generation Date Groups (GDGs)	Files on Azure using extensions in the naming conventions to provide similar functionality to GDGs

Understanding mainframe data: Db2, VSAM, IMS DB

This section provides a high-level overview of key mainframe components. IBM Db2, a relational database management system (DBMS), manages data structures for organizing user data and system structures. Dbspaces are structures containing one or more tables and are assigned to storage pools of physical data sets called dbextents. As shown in Figure 4 and described in Table 8, Db2 for z/OS uses VSAM datasets to store the data, which effectively makes each dbextent a VSAM data set.

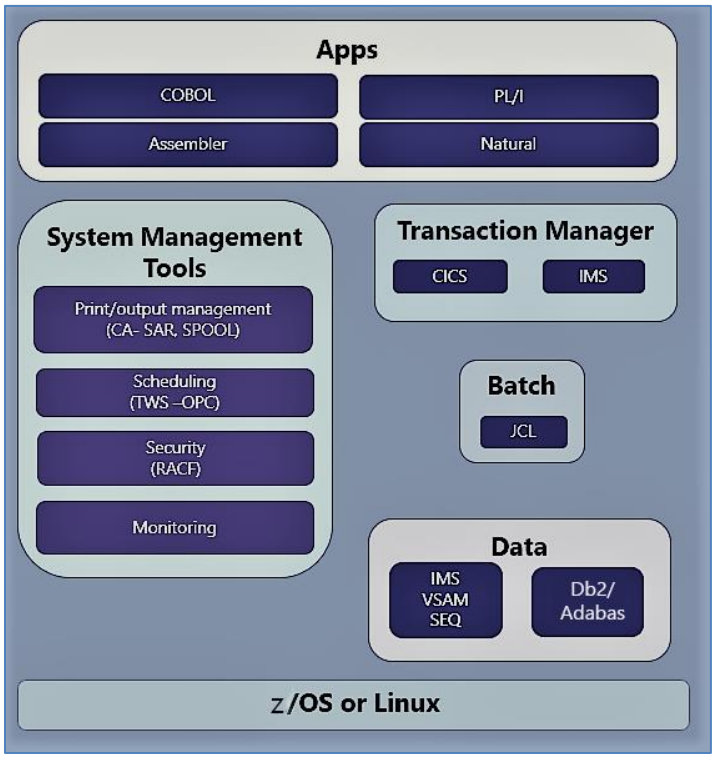


Figure 4. IBM z/OS components

Table 8. IBM database components

Component	Description
Database	Contains the following: • A collection of data contained in pools, each containing database extents (dbextents), which are z/OS Virtual Storage Access Method (VSAM) data sets. • A directory that identifies data locations in the storage pools. • A log that contains a record of operations performed on the database. A database can have one, two (dual or alternate), or four (dual and alternate) logs.
Database manager	Provides access to data in the database. The database manager runs in its own partition in a z/OS environment.
Application requester	Accepts requests from applications before passing them to an application server.
Online resource adapter	Includes application requester components for use in CICS transactions.
Batch resource adapter	Implements application requester components for z/OS batch applications.
Interactive SQL (ISQL)	Runs as a CICS application and interface enabling users to enter SQL statements or operator commands.
CICS application	Runs under control of CICS utilizing available resources and data sources in CICS.
Batch application	Runs process logic without interactive communication with users, such as producing bulk data updates or generating reports from the database.

Mainframe operations

Table 9 provides a summary of mainframe operations.

Table 9. Mainframe ops summary

Type	Description
Online	Online workloads include transaction processing, database management, and connections. Implemented via Db2, CICS, and z/OS connectors.
Batch	Batch jobs run without user interaction, typically on a regular schedule, such as every weekday morning. Batch jobs running on Windows or Linux-based systems can run using a third-party JCL emulator, such as Micro Focus Enterprise Server or BMC's Control-M.
JCL	The job control language (JCL) is used to specify resources needed to process batch jobs. JCL conveys this information to z/OS through a set of statements known as job control statements. Basic JCL contains six types of statements: JOB, ASSGN, DLBL, EXTENT, LIBDEF, and EXEC. A job can contain several EXEC statements (steps), and each step could have several LIBDEF, ASSGN, DLBL, and EXTENT statements.
IPL	An initial program load (IPL) refers to loading a copy of the operating system from disk into the processor's real storage and running it. IPLs are used to recover from downtime scenarios. An IPL is analogous to booting the OS on Windows or Linux.

Integration between the mainframe and Azure

Moving solutions from the mainframe to Azure may involve a "staged" migration, moving some applications first while leaving others on the mainframe (temporarily or permanently). This typically requires systems that allow applications and databases to interoperate between the mainframe and Azure. For example, an application may move to Azure, while the data used by the application remains on the mainframe. This scenario requires software to enable the application in Azure to access data from the mainframe.

Mainframe-to-Azure integration solutions include Microsoft Host Integration Server (HIS) along with solutions from IBM, Attunity, Codit, and Open Source solutions. The HIS solution provides the Distributed Relational Database Architecture (DRDA) required for applications in Azure to access data in Db2 that remains on the mainframe.

Mainframe high availability

How many 9s can a mainframe have?

In theory, mainframes claim five 9s availability. But as noted earlier, most system operators still need to schedule downtime for maintenance and IPLs. The actual availability approaches two or three 9s availability, on par with high-end, Intel-based servers.

Cloud systems, such as Azure, provide additional availability by replicating data from multiple storage devices, either locally or in other geographic regions. In the event of an Azure-based

failure, compute resources can access that replicated data on either the local or regional level. Azure PaaS resources, such as SQL DB or Cosmos DB, handle the failover automatically. In Azure IaaS, failover relies on specific system functionality, such as SQL Server AlwaysOn features, including failover clustering instances or availability groups.

Coupling Facility and Parallel Sysplex

Mainframes can use functionality called the Coupling Facility (CF) and Parallel Sysplex to failover to other compute and storage resources. A CF provides a shareable storage medium for multiple processers to access the same data by maintaining integrity and consistency of data.

The CF helps maintain the availability of the sysplex by allowing software on different systems in the sysplex to share data with integrity. The CF contains both processing capacity and storage with separate containers, known as structures. Each structure has a specific function in the role of the Parallel Sysplex. A Parallel Sysplex relies on one or more of the following CFs:

- Cache
- List
- Lock

Mainframe backup and recovery

Mainframe customers typically maintain disaster recovery sites or contract with a third-party mainframe for disaster contingencies. Synchronization with their disaster recovery site is usual done through offline copies of data. Both options incur high costs.

Likewise, automated geo-redundancy is also available through the mainframe coupling facility, albeit at great expense usually reserved for mission-critical systems.

In contrast, Azure, as an enterprise cloud provider, has easy-to-implement and cost-effective options for redundancy at local regional levels or via geo-redundancy.

Mainframe development and testing

Mainframes typically have separate logical partitions (LPARs) for development and testing, such as QA and staging LPARs. Mainframe development solutions include compilers (COBOL, PL/I, Assembler) and editors, such as Interactive System Productivity Facility (ISPF), which is the most common. Others include ROSCOE Programming Facility (RPF) and Computer Associates (CA) tools, such as CA Librarian and CA-Panvalet.

In addition, IBM and third-party tool vendors support mainframe emulation environments for development and testing of mainframe systems. This includes many of those listed in the Partner solutions on Azure section of this document, in addition to IBM Developer for System z (IDz), a mainframe emulation environment.

Emulation environments and compilers are available on x86 platforms, so development and testing are typically among the first workloads to migrate from the mainframe to Azure. The availability and widespread use of DevOps tools in Azure is accelerating development and testing migration.

When solutions are developed and tested on Azure and are ready for deployment to the mainframe, the source code is compiled on the mainframe.

Value of Azure

Organizations can take advantage of open commodity-based infrastructures, especially in environments using Azure. More than 90 percent of Fortune 500 companies are already running Azure for critical workloads. These are the same companies that were among the first to recognize the significant bottom-line incentives of moving to the cloud. Development and test workloads were among the first to move over to Azure, followed by DevOps, email, and disaster recovery as a service.

Today, in the third wave of cloud adoption, companies are looking to move legacy mainframe applications into the cloud where it makes sense.

For companies with a legacy mainframe environment, the value proposition from the cloud is clear with Azure delivering the following benefits:

- **Greatly reduced costs.** After earlier waves of on-premises to cloud migration and as cloud infrastructure continues to mature, companies are starting to take a hard look at their remaining legacy applications. The economics of cloud computing have grown too compelling for all but the most hardened mainframe devotees. The total cost of ownership (TCO) of the cloud's subscription-based, usage-driven cost model is simply far less expensive.
- **Skilled global workforce.** The cloud attracts a highly educated and skilled workforce of experienced and aspiring developers and IT pros. Mainframes no longer attract experts in large numbers and the talent pool continues to shrink every year.
- **Flexible open-systems environment.** At the heart of the cloud's competitive advantage is the ability to scale up or down to match exact customer specifications, enabling high productivity and rapid innovation.

Partner solutions on Azure

Microsoft Azure provides a proven, highly available and scalable infrastructure for systems that currently run on mainframes. Some workloads can be migrated with relative ease. Other workloads, depending on legacy system software, such as CICS and IMS, can be rehosted using partner solutions and migrated to Azure over time. Whatever the choice, a diverse group of partners are available to assist customers in optimizing for Azure while maintaining mainframe system software functionality. Table 10 summarizes Microsoft partner solutions for providing compatibility with mainframe system software.

NOTE: This list shows only a subset of partners within a larger group of officially recognized third-party solutions for mainframe emulation and services for Azure. For detailed guidance about choosing a partner solution, refer to the Platform Modernization Alliance.

Table 10. Partner tools for migration management

Solution/proof	Partner solutions	Description
Migrate mainframe closer to Azure	**Micro Focus**	
	• Visual COBOL	• COBOL development and integration tools.
	• PL/I	• Legacy compiler for the .NET platform, supporting mainframe PL/I syntax, data types, and behavior.
	Fujitsu	
	• NetCOBOL	• COBOL development and integration tools.
	NTT	
	• Data Enterprise COBOL	• COBOL development and integration tools.
	• Open PL/I	• Legacy compiler for the .NET platform, supporting mainframe PL/I syntax, data types, and behavior.
	Raincode	
	• COBOL compiler	• COBOL development and integration tools.
	• PL/I compiler	• Legacy compiler for the .NET platform, supporting mainframe PL/I syntax, data types, and behavior.
	• ASM370 compiler	• Compiler for the mainframe Assembler 370 and HLASM syntax.
	ASNA	
	• Visual RPG for .NET	• RPG compiler for the .NET Framework via Visual Studio plug-in.
	Modern Systems Natural	
	• CTU (COBOL-To-Universal)	• COBOL development and integration tools.
Migrate data to SQL Server	**Partner tools [Micro Focus, Raincode, and Modern Systems]**	
	• Microsoft SQL Server Migration Assistant (SSMA)	• Automated database migration tool to SQL Server from Microsoft Access, Db2, MySQL, Oracle, and SAP ASE.
	• Microsoft SQL Server Integration Services (SSIS)	• Platform for building enterprise data integration platform, managing SQL Server objects and data.

Deploy emulation environment [online and batch]	**Micro Focus**	
	• Enterprise Server	• Mainframe integration platform.
	NTT	
	• Data Transaction Processing Environment (TPE)	• Native transaction processing environment, running rehosted CICS transactions, IMS applications, IDMS DC programs, and other resources.
	• DATA Batch Processing Environment (BPE)	• Batch processing environment including JCL transaction capabilities.
	Raincode	
	• CICS	• Emulator for .NET and Azure platforms.
	• JCL	• Plug-compatible Job Control Language interpreter.
Instruction set conversion	**LzLabs**	
	• Software Defined Mainframe (SDM)	• Managed software container for migrating mainframe applications to Linux computers or private, public, and hybrid cloud environments.
Source code conversion	**Asysco, Averisource, Blu Age, Heirloom, and others**	
	• COBOL, C#, .NET, or Java,	• Several partner options.
	Accenture, Gemini, Cognizant, DXC, Fujitsu, HPE, Infosys, TCS, and Wipro	
Modernization services	• Global system integrators (GSIs)	• GSI partners design, build, and manage solutions for large public- and private-sector organizations.

Summary

Although mainframes may be appropriate for some niche purposes, in most cases their exorbitant cost doesn't justify continued use across most workloads.

A major driver for companies and organizations to move to Azure is the changing face of the application-development environment to be more agile and responsive to business needs. Migrating to the cloud allows customers to modernize their infrastructure by making mainframe applications—and the value they provide—available as a workload whenever the business needs it. Many workloads can be transferred to Azure with only minor code changes, such as updating names of databases. Other more complex workloads can be migrated in a phased approach.

The key to getting there is knowing where to begin. This paper provides a road map for starting the conversation with IT decision-makers who may still subscribe to outdated mainframe notions.

Mainframes may continue to have a place in highly specific scenarios that remain largely free of budgetary pressures. Most companies and organizations don't operate that way and can significantly benefit from moving some or all their mainframe workloads, applications, and databases to the cloud.

Learn more

For more information, see the following resources:

- Get started with Azure
- Platform Modernization Alliance: Mainframe migration
- Deploy IBM Db2 pureScale on Azure
- Host Integration Server (HIS) documentation