Redbooks

ibm.com/redbooks

Accelerate Mainframe Application Modernization with Hybrid Cloud

Skyla Loomis

Kyle Charlet

Suman Gopinath

Peter McCaffrey

Tim Brooks

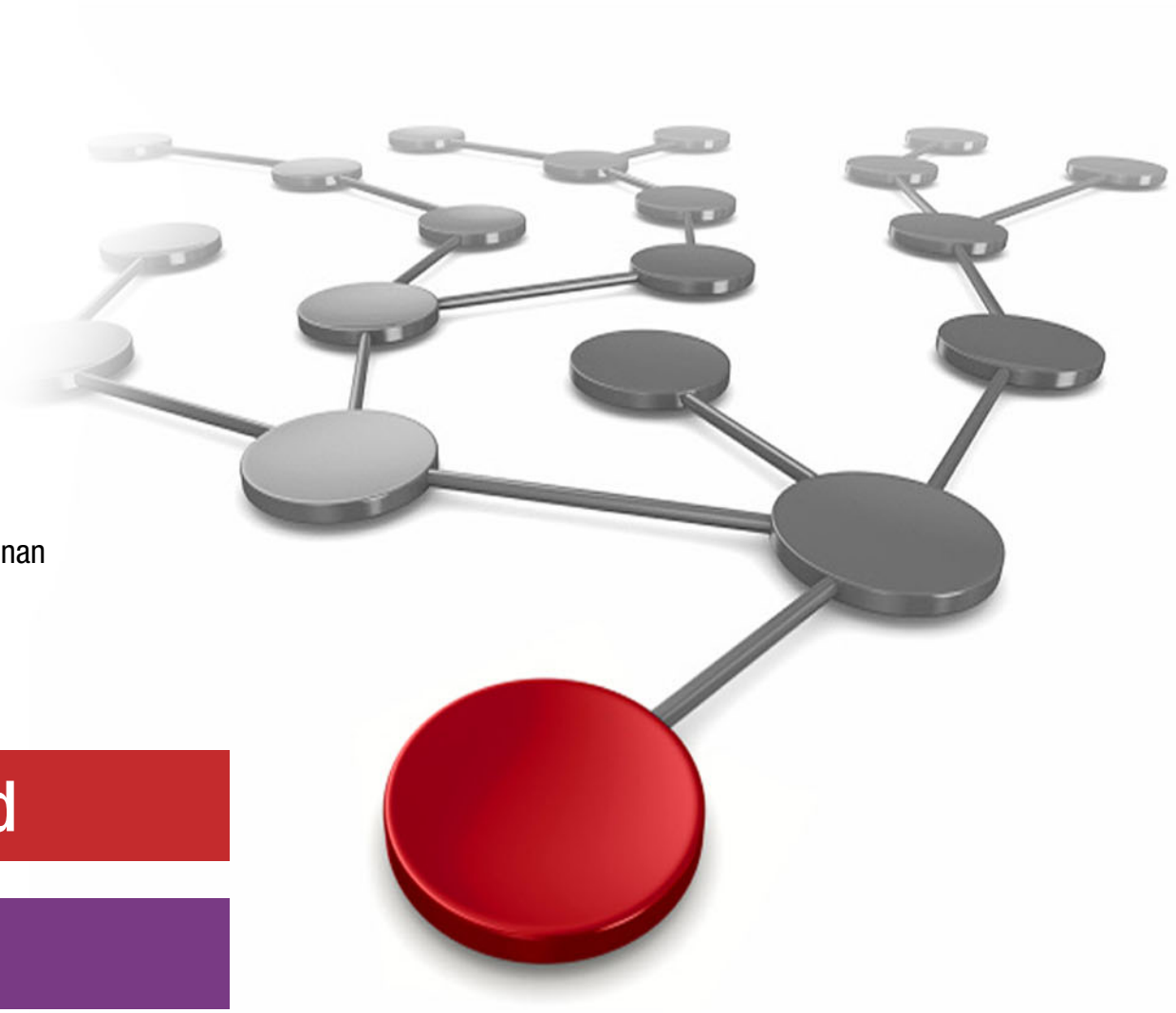
Juergen Holtz

Bryant Panyarachun

Purvi Patel

Mythili Venkatakrishnan

Yichong Yu



Cloud

IBM Z

IBM

Point-of-View

Step up your digital transformation, business agility, and productivity

Highlights

- **Better together:** Designed to modernize mainframe applications faster at lower cost and less risk by combining the strengths of the IBM Z platform and cloud.
- **Proven best practices:** Learn prescriptive approaches to accelerate your application modernization journey.
- **Reference architectures:** Study and deploy the solution that is right for your enterprise across IBM Z, Amazon Web Services (AWS), Microsoft Azure, and IBM Cloud.

What is the best strategy to an effective digital transformation of your business? How do you make your business more agile, productive, and less siloed so that you can provide engaging customer experiences rapidly and at lower costs?

> **Answer:** *Accelerate your mainframe application modernization and leverage hybrid cloud.*

This IBM Redbooks® publication provides an overview of the IBM® strategy to help you modernize applications faster, at lower cost and less risk, by using IBM Z® and public cloud solutions together in your modernization journey.[1] Included are prescriptive approaches, best practices with supporting technologies, reference architectures, and accelerators to help businesses achieve the right outcomes.

Challenges

To make this critical business transformation with mainframe applications and cloud a reality, you must achieve these goals:

- Accelerate digital transformation.
- Increase productivity and business agility.
- Address skills gaps.

Accelerating digital transformation

Organizations that leverage both IBM mainframes and the cloud must find new ways to provide innovative and engaging experiences that satisfy existing customers, attract new prospects, and gain a competitive edge. These organizations need effective tools and platforms that developers are familiar with that can make them more productive.

As businesses digitally transform, they can impose significant demands on existing applications and data. According to an IBM Institute for Business Value study, "*4 of 5 executives say their organizations need to rapidly transform to keep up with competition, which includes modernizing mainframe-based apps and adopting a more open approach.*"[2] The objective is to accelerate time to market, which is a key benefit when trying to gain a competitive advantage.

[1] https://www.ibm.com/cloud/blog/lower-costs-and-increase-roi-with-application-modernization-on-a-hybrid-cloud
[2] https://www.ibm.com/downloads/cas/7BJPNGND

Increasing productivity and business agility

Your organization's most valuable assets are its people. When it comes to gaining a competitive advantage through IT, you want to make sure that your application developers and IT administrators always have the right set of technologies (and the most effective tools) at their fingertips to unleash their creativity to build and manage amazing customer and employee experiences.

Building a culture around DevOps can make a significant impact on productivity and agility.

Addressing skills gaps

Across the IT industry, the availability of skills is a top-of-mind issue for many companies. This situation also is true in the mainframe space where businesses are worried about finding enough mainframe-specific skills. What fuels this concern is the belief that the mainframe requires unique skill sets that are distinct from other talent across the organization. For example, many enterprises lack the right developer skills; cannot easily access data on the mainframe; and use older tools and processes, all of which lead to higher overall business costs.

The reality is that the IBM Z platform now supports popular programming languages, common CI/CD toolchains and development practices, and standard approaches to IT automation, which enable companies to optimize their developer and site reliability engineers (SRE) across their enterprise and help to address skills requirements.

Solutions

In the recent Application modernization on the mainframe IBM Institute for Business Value study, among the surveyed, 71% of executives say that mainframe-based applications are central to their business strategy. As stated earlier, four out of five respondents say that their organizations must rapidly transform to keep up with competition, which includes modernizing mainframe-based apps and adopting an open approach that includes cloud.

IBM can help you meet these challenges by providing the tools and methodologies to help you transform your business.

To increase this productivity, agility, and close the skills gaps, you should modernize your mainframe applications and embrace hybrid cloud. Specifically, consider the following best practices:

- Embracing a *hybrid cloud* approach to mainframe application modernization.
- Leveraging a *continuous application modernization journey*.
- Taking advantage of the recommended *cloud reference architectures* to leverage IBM Cloud®, Amazon Web Services (AWS), and Microsoft Azure.
- Getting started fast with the IBM proven *co-creation methodology*.

Embracing hybrid cloud

Application modernization is the process of updating mainframe applications so they can be maintained, extended, accessed, and managed in ways that enable businesses to meet their current and future needs.

Many businesses support many different computing models ranging from traditional on-premises to multiple public clouds and software as a service (SaaS) offerings, and increasingly edge-based computing. The overall complexity of this heterogeneous model continues to increase, with inefficiencies occurring as development teams and processes become siloed. Converging everything to a single public cloud platform is impractical, expensive, and risky.

Our perspective on technology is clear and consistent: hybrid cloud and artificial intelligence (AI) are helping to usher in a new era of greater productivity, faster insights, better decision-making, and enhanced employee and customer experiences. Therefore, we designed our products and services to maximize the business value of hybrid cloud and AI for our clients.

Hybrid cloud has become the leading architecture for enterprises because it offers more value than relying on a single, public cloud. This value takes the form of scale, security, ease of use, flexibility, seamless experiences, and faster innovation cycles. Companies continue to adopt hybrid cloud to unify their data and applications across multiple clouds, on-premises, and at the edge.

Making the mainframe (the IBM Z platform) an essential part of your hybrid cloud strategy is the first step in realizing the mainframe's full value for your enterprise. For many clients, this situation means you do not need to think about the mainframe as an “island” that must be isolated or protected from the rest of your hybrid cloud environment. Instead, you should proactively work to integrate the mainframe into a hybrid cloud by leveraging open standards ranging from APIs to IT automation, and cloud-native container platforms like Red Hat OpenShift.

By integrating the mainframe, you get the best of both worlds: Maintaining an ultra-high resiliency and secure environment that is cost-efficient as it scales while experiencing the time-to-market value and innovation that comes with a common DevOps operating model with shared tools and compliance processes.

The IBM Z platform is the best-fit option for secure transaction processing

The IBM position is that the ideal approach to mainframe application modernization leverages the best of the IBM Z platform and the innovation of cloud.

With a hybrid cloud, you can leverage your existing applications and the best-fit platforms for application deployment, as shown in Figure 1.

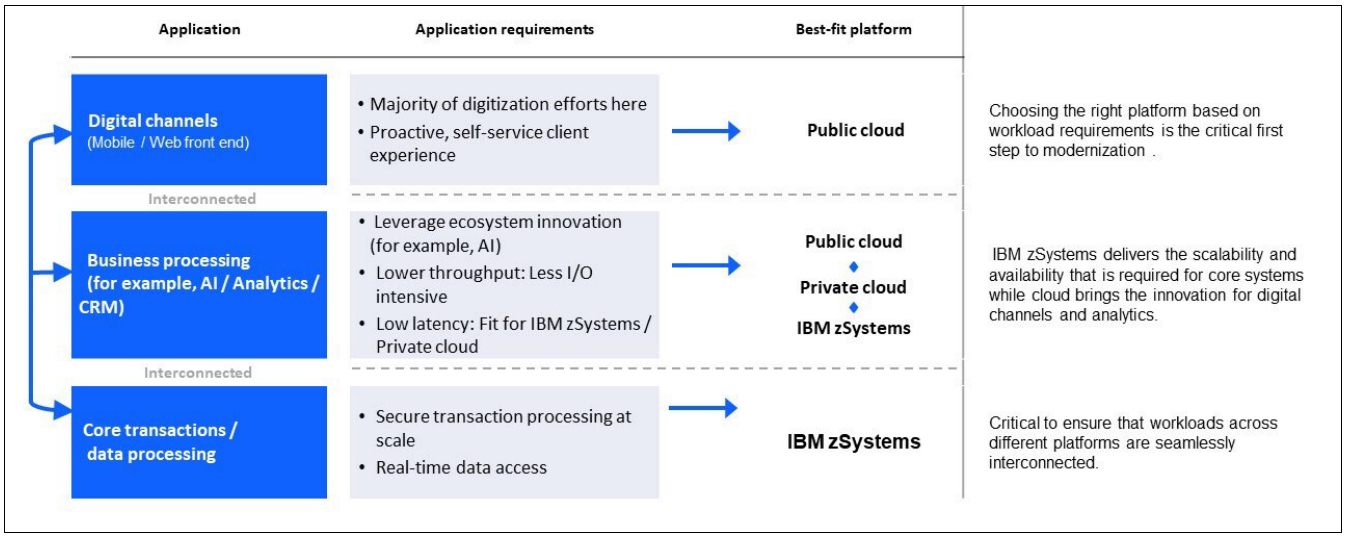


Figure 1 Selecting the best-fit platform for applications

The IT industry is moving from a general-purpose infrastructure to a specialized infrastructure to support application workload characteristics and scale efficiently. For example, in most public clouds, there are specialized or “best-fit” options for graphics-intensive workloads, AI and machine learning, and video transcoding. The IBM Z platform is considered “best-fit” in the industry for secure transaction processing because of the following reasons:

- **Data privacy, security, and compliance**

  Data is encrypted in all contexts: in flight, at rest, and in use. Data is protected from being compromised, either accidentally or deliberately, from inside and out. Demonstrating compliance is simplified with automated evidence-gathering and reporting capabilities, which do not require workload changes.

- **Transaction processing requires near-zero downtime**

  When data centers fail or security breaches expose private information, the cost is astronomical and grows year to year. With the IBM Z platform, you can achieve continuous availability, recover with near-zero data loss, and reduce recovery time from days to seconds.[3]

- **Transactional consistency and integrity**

  IBM z16™ is capable of processing up to 25 billion encrypted IBM z/OS® OLTP transactions per day. Businesses should be able to depend on their applications running consistently with full data integrity.[4]

- **Scale and performance**

  The IBM Z platform is a vertically integrated hardware and software stack that is optimized for performance and efficient scalability. The IBM Z platform supports low latency transactions at a massive scale.

[3] https://www.ibm.com/common/ssi/ShowDoc.wss?docURL=/common/ssi/rep_ca/6/897/ENUS222-106/index.html
[4] https://www.ibm.com/downloads/cas/MRPZPJPK

- **Cost optimization**

  Respond faster to customer demand and keep costs down while leveraging the efficiency of The IBM Z platform as you scale. Tailored Fit Pricing offers cost effective, usage-based pricing that can elastically scale and deliver capacity on demand when workload spikes. You also can leverage IBM Z® Integrated Information Processors (zIIPs), which are dedicated processors for select new z/OS applications that do not impact software usage charges of existing applications. zIIPs provide a cost-effective way to add new applications to IBM Z servers.

- **AI for real-time insights at scale**

  The IBM Z platform can help you create value across your business with accelerated AI insights that are derived from and applied to each customer interaction at a transaction level. Bring AI into your transactions to help detect fraud, for example, before a transaction completes.

Through decades of innovation across a vertically integrated stack, the IBM Z platform excels at mission-critical transactional workloads with unmatched throughput, availability, and security. It was designed to run secure transactional processing on the IBM Z platform or public clouds, often in as little as one half of the operating expense (OpEx) of running that same application on a public cloud.[5]

Digital transformation initiatives have accelerated transaction growth, which puts pressure on an IT organization's ability to meet service-level agreements (SLAs). Deploying applications on the best-fit platform across hybrid cloud provides the best approach to achieve non-functional requirements around scale and performance.

Overall, a hybrid cloud approach that implements the IBM Z platform is designed to deliver more than 2.5 times the value of a public-cloud-only approach.[6]

Collaborating with AWS, Microsoft Azure, and IBM Cloud on a hybrid approach to application modernization

The IBM Z organization is collaborating with AWS, Microsoft Azure, and IBM Cloud to support a hybrid cloud approach to mainframe application modernization that leverages the core strengths and attributes of the IBM Z platform together with the extensive cloud services that are available on Microsoft Azure, AWS, and IBM Cloud.

A hybrid strategy that includes the IBM Z platform and cloud allows for rapid development by leveraging an agile DevOps approach; provides for access to mainframe applications and data; and can address skills gaps with open tools and IT automation. Together, the IBM Z platform and Microsoft Azure, AWS, and IBM Cloud can help accelerate mainframe application modernization to improve agility, optimize costs, and lower risk.

To start, we identified several application modernization initiatives for an effective hybrid cloud approach, among which are the following ones:

- **Simplify access from digital channels.**

  Simplify access to mainframe applications and data from cloud services. As organizations enhance their customer experience through new digital channels, there often is a need to access core business applications and data that is on the mainframe.

- **Share real-time information between z/OS applications and cloud.**

  Providing customized offers that can attract customers and differentiate your business is a key objective for most businesses. Increasingly, this goal requires real-time exchange of information between core business applications running on the mainframe and digital front-end applications running on cloud.

[5] Tailored Fit Pricing for IBM Z Offers Cost Effective Workload Growth Compared to Three Public Cloud Examples, IBM, May 2021
[6] IBM z16: The platform designed to build the business of tomorrow, May 2022

- **Leverage DevOps for z/OS applications on the cloud.**

  Embrace an enterprise DevOps model that includes developing and modernizing z/OS applications for hybrid cloud to increase speed and agility for greater developer productivity.

- **Automate IT across the IBM Z platform and cloud.**

  IT automation has played a critical role in large-scale enterprises for decades, and it now is essential because businesses have adopted a hybrid cloud architecture to support their digital initiatives.

The IBM Z platform is committed to working with AWS, Microsoft Azure, and IBM Cloud to help identify and provide more solutions for our clients' needs and objectives.

Starting or continuing your application modernization journey

Because the IT landscape is continually changing, as a best practice, businesses should consider application modernization an ongoing journey, not a one-time implementation. It is an initiative that maximizes the potential of existing investments. With this situation in mind, businesses can evolve and grow to continuously align with current and future business needs. The ongoing journey includes the following tasks:

Building the right foundation	Optimize hardware and software costs while streamlining application management and performance. Businesses can integrate the IBM Z platform into a hybrid cloud solution to simplify access to mainframe applications. Implement a standardized approach to IT automation to help businesses support a hybrid cloud architecture in a consistent and productive way. Additionally, organizations can unlock more revenue by increasing access to data for analytics and AI through an application programming interface (API) and a data-modernization strategy while still ensuring security and compliance.
Increasing business agility and productivity	Move to an enterprise DevOps process and an automated CI/CD pipeline. You can fully align this approach to meet present priorities while being open and standard across the business.
Accelerating your journey	Reduce time to value with proven application modernization patterns, tools, and best practices. These resources include how-to guides and showcases for working with z/OS, Linux on IBM Z, IBM LinuxONE, and public cloud together. Taking a continuous approach to modernization with the IBM Z platform alongside public cloud can be the catalyst for lowering costs and increasing return on investment (ROI).
Co-creating with IBM	IBM experts with Client Engineering and IBM Consulting™ can help you generate innovative ideas and equip you with the resources, technologies, and expertise to rapidly turn those ideas into business value. The experts at IBM can help bring you and your customer challenges into focus. You can empower IT teams to take manageable risks while adopting leading technologies, speed up product development, and measure the value of everything that you do. You can accomplish your transformation journey with an iterative framework that guides success from ideation to build and scale.

So where do you start

In this IBM Redbooks publication, we describe several entry points where you can start or continue your mainframe application modernization journey depending on your business needs, environment, challenges, and current processes. You should select which entry points will drive the most immediate value for your business. As a best practice, you should do the following tasks:

I. Optimize existing applications

II. Enhance and extend applications

III. Integrate across hybrid cloud

IV. Simplify information sharing and data access

V. Adopt enterprise DevOps and observability

VI. Make AI-driven decisions at scale

VII. Automate and standardize IT

I. Optimize existing applications

IBM's continued investments in its mainframe hardware and software is designed to optimize the cost and performance of running and deploying applications on the IBM Z platform.

This entry point describes the Tailored Fit Pricing program, which is a commercial model that is suitable to most mainframe environments, but particularly useful for a modernized mainframe environment. This section also guides you through the different optimizations and capabilities that you can take advantage of to bring down operating costs, improve the efficiency of existing IBM Z applications, and increase resiliency.

Taking advantage of Tailored Fit Pricing

Tailored Fit Pricing is a flexible pricing model that dramatically simplifies the existing pricing landscape through cloud-like, software, and hardware consumption-based pricing options for on-premises computing. Tailored Fit Pricing for Software is about pricing that is based on the amount of processing, that is, consumption rather than the traditional approach of pricing based on the peaks that are reached.

This consumption-based model can be seen as better suited to modernized workloads and environments that are frequently spiky, volatile, and unpredictable in their profile. The IBM Z technology is exceptionally capable of processing workload profiles of these kinds, and Tailored Fit Pricing is equally suited. No longer are spikes and peaks in processing a real challenge for software pricing; rather, you can enjoy the confidence to modernize your mainframe applications knowing that both the technology and pricing model are suited to the processing in today's IT era.

Tailored Fit Pricing for Software, with consumption-based pricing at its core, also greatly improves pricing predictability and transparency as organizations manage their existing applications and introduce new ones on the IBM Z platform. The model removes the need for any capping, and enables businesses to leverage all machine resources at their disposal if and when they need it without the limiting peak price factor. This model encourages IT departments to design solutions based on best-fit technology choices with business outcomes influencing decisions rather than cost analysis and the fear of disproportionate software billing influencing technology choices as they continue their application modernization journey.

Tailored Fit Pricing for Hardware also is available, and can be used with its software counterpart. This model extends the suitability of the Tailored Fit Pricing program to modernized application workloads and environments.

Extra hardware can be provisioned to use on top of what a business already permanently owns. This model is suited to scenarios where a customer might receive unexpected but brief spikes in demand, but the extra hardware is “always on” and ready and available to be used as needed. The extra hardware is provisioned on a subscription basis with a granular usage fee for when a customer spikes into this extra hardware.

Figure 2 provides an example of Tailored Fit Pricing.

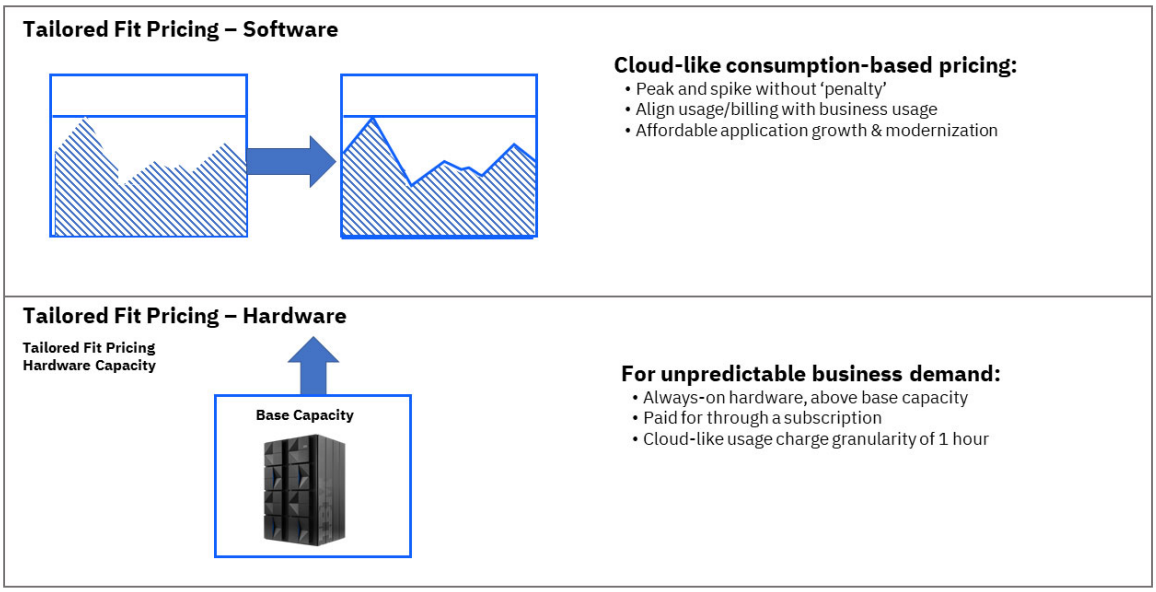


Figure 2 Tailored Fit Pricing

Using specialty processors

The IBM Z platform provides the following two specialty processors that help businesses lower the cost of running new applications on the platform while allowing them to take advantage of IBM Z platform strengths in security, availability, and scale:

- **IBM Z Integrated Information Processor (zIIP)**

  IBM zIIPs are dedicated processors for select new z/OS applications that do not impact software usage charges of existing applications. zIIPs provide a cost-effective way to add new applications to the IBM Z platform. A zIIP is designed for select data (database), and cloud and transaction (Java) processing workloads. zIIPs allow users to leverage extra processing power for new applications without affecting their total million service units (MSU) rating or machine model designation.

  The types of applications that can leverage zIIP processors include Java applications, and z/OS Container Extensions, including Red Hat OpenShift, data virtualization, machine learning, APIs, System XML parsing, ONNX models, IBM Z AI Data Embedding Library, System Recovery Boost, various IBM and ISV products as defined by those products, and IBM Db2® DRDA. The potential total cost of ownership (TCO) savings can be considerable compared to running the same application on comparable x86 servers and public cloud infrastructures.

- **IBM Integrated Facility for Linux (IFL)**

  The IBM IFL is a processor that is dedicated to Linux application workloads on IBM Z and IBM LinuxONE. It is supported by the popular Linux operating system for the IBM Z platform and LinuxONE (including Red Hat), IBM z/VM®, and Kernel-based Virtual Machine (KVM). IBM IFL allows Linux applications to take advantage of the IBM Z and LinuxONE massive processing capacity and on-chip compression acceleration; provides a foundation for hybrid cloud computing and industry-leading capabilities in security and data privacy; and cyber resiliency.

The benefits of IFLs allow clients to consolidate Linux application instances, which lower costs by reducing operational efforts, software licensing charges, and energy and floor space, helping business IT organizations achieve their sustainability objectives. On IBM z16, applications can leverage the latest hardware improvements or specialty engine usage, which are designed to increase throughput compared to running the same compute-intensive applications on previous IBM Z generations or reduce the cost and complexity of existing distributed systems while keeping applications close to the mission-critical core.

Leveraging platform improvements

By achieving software and hardware currency and leveraging new IBM Z platform innovations, you can optimize the cost and performance of your current mainframe environment. By using enhancements that are on the IBM z16, businesses can achieve a significant improvement in transaction processing and gain from newer capabilities that bring efficiency to the overall system. On IBM z16, your applications can leverage the latest hardware and software improvements to reduce cost and increase throughput compared to running the same compute-intensive applications on previous IBM Z generations.

For example, on IBM z16, compiling your compute-intensive applications with the latest software can reduce CPU usage compared to running the same compute-intensive applications on IBM z14®.[7]

A large percentage of IBM z/OS applications is written in COBOL spanning millions of lines of code and developed over decades, representing core business processes and institutional knowledge. A combined upgrade of COBOL with the latest hardware can bring in a significant performance improvement. One example of a possible benefit is on IBM z16: IBM Enterprise COBOL for z/OS 6.4 can reduce CPU usage by approximately 30% on average for compute-intensive applications that were built with Enterprise COBOL 5.2.

On IBM z16, optimizing your compute-intensive applications with IBM Automatic Binary Optimizer for z/OS 2.2 can reduce CPU usage by 60% on average compared to running the same applications on IBM z15®, which was built with IBM Enterprise COBOL for z/OS 4.2.

IBM continues to optimize and enhance COBOL to use the IBM Z architecture and provide tools to mitigate risk and allow for modern programming techniques to support modernization and interoperability with other programming languages, such as Java. For example, IBM has achieved the following tasks:

- Simplified COBOL and Java interoperability to extend the capabilities of your COBOL programs with Java without writing object-oriented COBOL to reduce the number of manual JNI calls that are required.
- Built-in interoperability between AMODE 31 (31-bit) and AMODE 64 (64-bit) COBOL programs to handle your growing COBOL program data without converting the entire application to 64-bit, which removes the need to convert the entire COBOL application to 64-bit. You can convert 31-bit applications to 64-bit over time.
- Support for user-defined functions to enable you to write your own functions and invoke them like intrinsic functions, which can improve code modularity and maintainability. COBOL now supports user-defined functions, which might give new COBOL programmers a familiar structure.
- Improved integration with IBM Automatic Binary Optimizer for z/OS so that modules that you compile today with COBOL 6.4 can take advantage of later IBM Z hardware enhancements without having to be recompiled. IBM Automatic Binary Optimizer for z/OS is designed to reduce operating costs, CPU usage, and processing time for critical business COBOL applications without a code change or recompilation.

Another example of the kind of performance improvements that are possible is moving from one release of middleware to a current release. With IMS 15.3 running on IBM z16™, the results of the IBM internal testing that was done in a controlled laboratory environment demonstrated improvements in the Internal Transaction Rate (ITR), ranging from 4% - 21% depending on the workload, when compared to IBM z15. For more information, see the *IMS Performance Evaluation on IBM z16* white paper.

[7] https://www.ibm.com/products/cobol-compiler-zos/details

Maintaining business resiliency as applications change

With growing application workloads, clients might face an increasing challenge to ensure their system's resiliency. To reduce the risk for outages in workloads that exceed a load threshold, load and scale testing in production-like setups are crucial. From a cost perspective, it often is not cost-efficient purchase extra capacity permanently. The IBM Z Business Resiliency Stress Test (zBuRST) solution allows organizations to use spare IBM Z physical resources to load and stress test at production scale before introducing changes to a production environment. It can increase business resiliency with testing capabilities at production scale and at a wanted price point.

As part of an overall business continuity plan, what is imperative is the ability to react and adopt to internal and external or planned and unplanned business demands and threats in a way that the operation continues to run. Because of the IBM commitment and investment in 99.99999% availability,[8] the IBM Z Flexible Capacity for Cyber Resiliency solution was introduced with IBM z16. This offering can shift capacity between participating IBM z16 systems at different sites in a disaster recovery situation. The target configuration on the target site can be used for up to 12 months. The shift can be fully automated by using IBM GDPS®. The flexibility can provide optimized costs compared to purchasing double capacity. In regulated industries, this flexibility often is a requirement. For more information, see IBM Z Flexible Capacity for Cyber Resiliency.

Modernizing by streamlining business practices

Today, businesses often share in fixed and variable costs according to their actual usage through a governed chargeback system.

As mainframe application modernization increases productivity and optimizes costs, organizations must adapt to how they reflect efficiencies or costs to their internal customers. The company can through its IT organization realize these cost reductions in terms of both capital expenditures (CapEx) and operational expenditures (OpEx). Because most organizations can distribute their IT costs to their constituent lines of business by using either chargeback or showback, the departmental operational budget also can see a reduction in cost because the modernized and efficient IT resources can be shared across more lines of business.

II. Enhance and extend applications

Enterprises need their applications to be maintained, extended, deployed, and managed in a way that meets market demand and allows their businesses to grow. With the success of Java applications interacting with COBOL applications and the ability to create open APIs for existing applications and to connect to data sources with data virtualization, mainframe applications can be extended to support new business requirements.

Based on the type of data processing and latency requirements, cloud-native applications running on public cloud or on Linux on IBM Z can access mainframe applications and data.

Today, applications operate across multiple IT platforms: the mainframe, distributed, public, or private cloud. Many core business applications have been running on the mainframe for decades. Growing business needs require that applications continuously churn out new capabilities as the market demands. How these enhancements are made and where the applications are deployed depends on the type of application and the nonfunctional requirements around quality of service, security, and accessibility to data and maintainability.

This section describes the next entry point to your continuous application modernization journey and the best practices for building new capabilities and where they are best deployed.

[8] https://www.ibm.com/z/resiliency

New ways of development

A way for an enterprise to build new capabilities is to harness the value of the existing applications and data and make them available for digital consumption without changing them. The IBM Z platform supports tools for development of APIs that are based on standard OpenAPI specifications from core applications on the IBM Z platform and a robust, comprehensive runtime environment for exposing APIs. For more information, see “III. Integrate across hybrid cloud” on page 17.

In many cases, the API that provides IBM Z services and data is extracted from the existing applications. In other cases, you might need to map the API to an existing business application that requires enhancements to code. As enterprises come more into the API economy, net new functions can be built.

It is also possible that over the decades, the mainframe IT system might have morphed into a monolith, and business areas are not represented by the scope of programs and transactions. By adopting DevOps processes, tools, on-demand test environments, and test automation, it is possible to enhance these tightly coupled applications. However, if this architecture prevents the creation of efficient granular APIs, then decomposing the architecture into discrete functions becomes a necessity. The logic and the rules that have been fine-tuned over decades can be used for development.

Rather than a “big bang rewrite” that can be risky and expensive, as a best practice, you can extend and enhance existing applications as business needs arise. This way, you are confident that you are updating the parts of the application that are active and likely to change.

Using an automated discovery tool like IBM Application Discovery and Delivery Intelligence (ADDI) can speed up the process of discovering existing functions that can be decomposed into functions where necessary.

In all these cases of new development, there is now a choice in the technology that is used for the new functions. The mixed-language support that is available in the IBM Z platform means that when developing new functions, the new code can either use the same programming language, or another language that might meet the needs of the development team or the characteristics of the function.

Using the right programming language in the right context

There are multiple choices when it comes to choosing a programming language for an enhancement or new function. The new business function can be made in the current native language of the z/OS, whether it is COBOL or PL/I. The changes can be efficiently managed and deployed by making these applications part of an automated enterprise CI/CD process. The latest compilers for these languages and the IBM Automatic Binary Optimizer for z/OS for COBOL use the latest hardware facilities for acceleration of computations, especially for the decimal types that are ubiquitous in financial applications.

Java developer skills often are ubiquitous in the marketplace, and they often are used for creating applications on the IBM Z platform. Java also has the benefit of being zIIP-off-loadable, which also can be beneficial for cost optimization. Writing a new function in Java can enable common developer skills to be leveraged, and over time significant and relevant portions of the applications can be built in Java.

According to a case study with IBM and Atruvia, Pascal Meyer, a Senior Enterprise Architect at Atruvia says, “We see Java on IBM Z as a key technology in driving competitive advantage for our clients.” Furthermore, this case study has the following statement:

“Java on IBM Z alongside COBOL would enable developers to enrich core banking functionality in a seamless, low-risk manner by replacing COBOL subroutines with Java without having to rewrite large programs. And it would be easier for software architects within the distributed environment to call core transaction services directly from IMS applications.”

The new business function can be made in the current native language of the z/OS, whether it is COBOL or PL/I. The changes might be efficiently managed and deployed by making these applications part of an automated enterprise CI/CD process. The latest compilers for these languages and the IBM Automatic Binary Optimizer for z/OS for COBOL use the latest hardware facilities for acceleration of computations, especially for the decimal types that are ubiquitous in financial applications.

Among the z/OS programming languages, Java is often used for writing new functions. Applications that must invoke existing z/OS resources in the same unit of work are often written or extended as Java applications in IBM CICS® or IMS. These transaction servers provide libraries that might make it easy to call native resources by using Java interfaces and call out to data sources like Db2, IMS Database, and IBM MQ based on JDBC/JMS specifications. The new function and the existing application can be run in the same global transaction scope, which can ensure integrity in database commits across language boundaries.

New functions that are written in Java also can interoperate with existing COBOL applications. The COBOL Java interoperability is managed under the same transactional context with the capability to handle 31-bit and 64-bit address spaces.

Figure 3 shows an example of this interoperability in IMS where the IMS dependent regions establish a persistent 31-bit address space for running COBOL and a 64-bit for running Java in a 64-bit JVM.

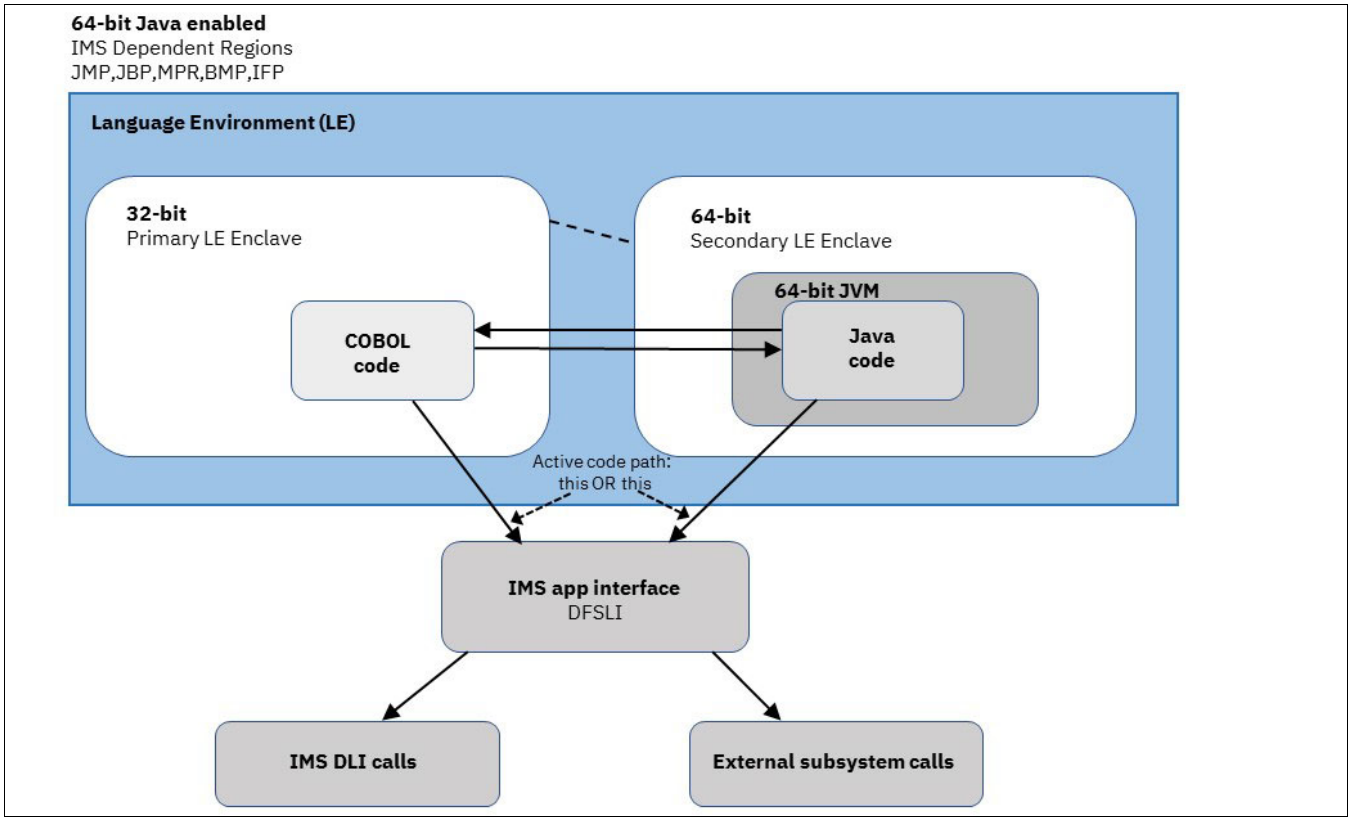


Figure 3 Java and COBOL interoperability

Based on the available skill sets, the new business function can be built or augmented by using other leading language choices, for example, Node.js, Go, and Python. Enterprises can choose the language that fits their needs and take advantage of package ecosystems to possibly speed up delivery. Any user interfaces should be moved to Java and modern languages. Granular, reusable services that require light coupling with other applications and services are often written as cloud-native applications.

Figure 4 on page 13 provides an example of the usage of Java virtual machine (JVM) and Liberty servers with CICS and IMS.

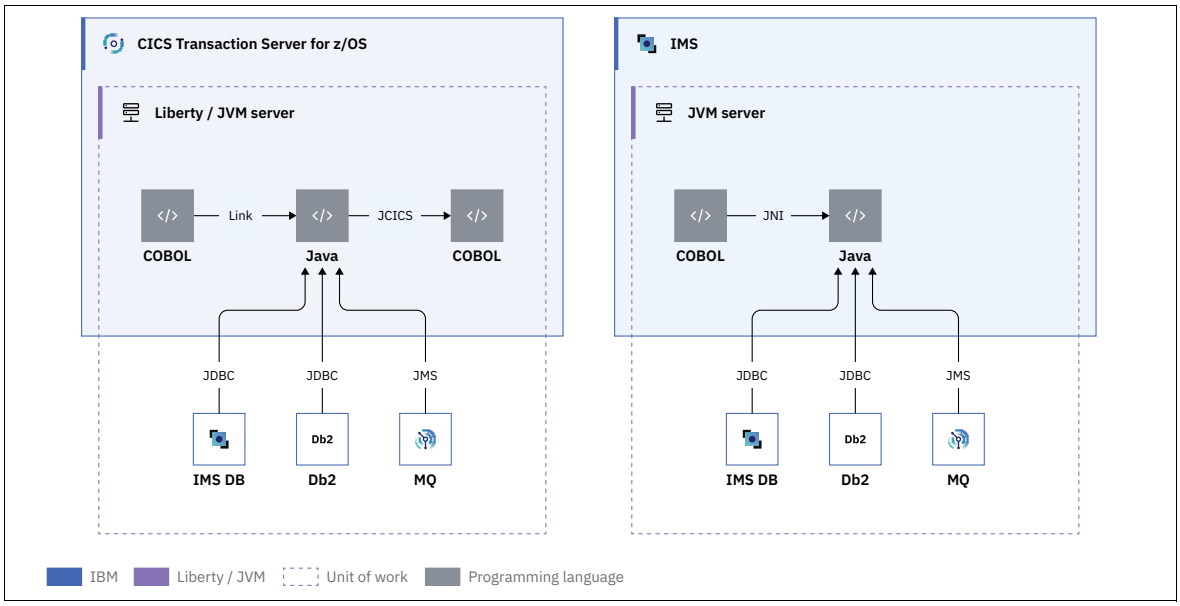


Figure 4 Using JVMs or Liberty with CICS and IMS

Transforming applications

The various transactional and batch applications that run on z/OS often were designed decades ago and tacit knowledge was lost. However, in response to ongoing business needs, they continue to evolve.

Whether a new business capability is derived as an API from an existing asset or a new function that is developed by extending existing assets or by using data stores on the IBM Z platform, application understanding and discovery play a key role. IBM ADDI builds a metadata repository of existing applications that is based on static analysis, which allows development teams to extract the right asset to expose as an API or explore the data lineage or impact of a change to existing applications in an automated fashion.

For example, according to a case study with Sun Life and IBM, Sun Life reduced application code search times by 50% compared with manual approaches:

"In addition to identifying where the connections were in the code, the developers needed to know whether the changes followed COBOL's rules-based restriction. Using ADDI enabled us to reduce analysis time by 35%."

Sun Life also stated the following information:

“Historically, none of our developers had the ability to generate job graphs”, he [Gary Lesage] notes. “The fact that we were able to inject our scheduling information into ADDI and generate graphs was a totally new. It added to the overall improvement in the efficiency of our team.”

Generative AI-assisted refactoring and language translations

With the advent of generative AI and Large Language Models, it is now possible to take this analysis one step further. As discussed earlier, there might be a need to refactor the existing applications into modular functionalities. This may be required if the architecture does not allow for exposing discrete services or inhibits maintainability. IBM recommends an iterative approach to this refactoring - IBM watsonx Code Assistant for Z enables a developer to quickly discover and refactor existing COBOL applications into discrete services, reducing the time required in decomposing and componentizing the monolith. Using graphical analysis, a developer is guided with the right context of business rules and presented with recommendations for refactoring. Once refactored, the business transactions become more maintainable and lend themselves to further modernization say being exposed as an API or converted to a reusable service.

Modular, organized code also provides the option to rewrite into a different language based on the best fit language for the right context. While some parts of the application are best retained in COBOL or PL/I, some application components that process XML, JSON, YAML etc. benefit from being written in languages like Java. Java not only provides a rich library of processing these artifacts but also improves maintainability by implementing these functions in a compact easy-to-understand manner versus being implemented in a structural language like COBOL. IBM watsonx Code Assistant for Z enables this transformation by utilizing generative AI and a trusted, well trained large language model to convert the COBOL service into discrete, object-oriented Java. The AI service analyzes the COBOL code semantically along with all its interfaces to generate an object-oriented class structure with appropriate method templates. It then invokes generative AI to implement these methods based on the COBOL code. The generated classes and methods are presented to the developer at each step to allow for standardization.

Whether deciding to stay on COBOL or move to Java, it is important to integrate the application into a modern CI/CD pipeline. Java and modern languages are maintained in Git and modern SCMs. With applications being moved partially to Java and the adoption of language interoperability, it becomes important that the existing application components be a part of the same integrated pipeline to ensure consistency in deployment. This is described in detail in section “V. Adopt enterprise DevOps and observability” on page 26.

Batch Modernization

Batch processing is still used for an organization's most critical and time-sensitive business operations. There are various operations that require bulk processing like complex time-based reporting that processes large amounts of data, or application of repetitive rules across extensive data sets. Information integration like collating data from internal and external systems and using it to gather business intelligence or run analytics is yet another very common use case. Automated batch processing is the best way to handle these operations. However, it is important that:

a. Organizations distinguish functions suitable for real-time, batch, micro-batch intra-day processing methods. There could be instances where a business functionality that typically does not require bulk processing is a part of the batch when it is better suited to a message driven or API architecture or an intra-day smaller batch cycle.

b. Organizations must optimize and modernize their batch systems based on their business goals to make maximum use of their IBM Z systems by appropriately placing the right workload in batch and OLTP processing.

IBM Z Workload Scheduler is a powerful workload automation solution that enables organizations to automate, plan, and control complex systems' workloads utilizing modern technologies to optimize processes. It provides centralized management of workflows across multiple platforms and business applications, offering a single point of control. In conjunction with IBM Workload Manager, it can manage the critical path execution, by allocating the right priority to critical path batch jobs while optimizing overall performance.

IBM recommends that batch processing is continuously optimized along with the rest of the real-time applications by doing the following:

1. Utilizing the right set of batch performance analysis tools like IBM Z Performance and Capacity Analytics can help operations managers and capacity planners make informed decisions about application batch performance. Analysis tools like IBM Application Discovery and Delivery Intelligence can help development teams plan their application change better by understanding dependencies between shared batch and online resources.
2. Being at currency with the latest releases of IMS and Db2 as well as COBOL and PL/I compilers help bring optimization in batch processing, utilizing the latest hardware features. When it is not always possible to recompile modules, Automatic Binary Optimizer for z/OS can be utilized to optimize COBOL modules to support the latest IBM hardware, reducing processing time and CPU usage. The use of technologies like VSAM Record Level Sharing or RLS allows for greater concurrency between batch and online processing.
3. As business processes have a need to become more real-time, enterprises look to increasing the batch frequency where possible. However, the right technologies need to be utilized to move from traditional to more

intra-day operations. In use cases where a large amount of read-only data is required from applications, alternate technologies like IBM Z Digital Integration Hub and IBM Db2 Analytics Accelerator (referenced in the earlier chapters) can be a solution to remove the contention caused by parallel access to and update of the same data in both OLTP and batch. Other modern technologies like IBM Data Virtualization Manager and IBM Data Gate can be used to access data directly from the mainframe for business intelligence, analytics or machine learning. This can replace existing ETL(Extract, Transform and load) processes that extract large amounts of data off the platform for the transformation.

4. As with the different languages that are now available on IBM Z to create new business applications, enterprises have an option to utilize JZOS Batch Launcher and run Java applications directly as batch jobs or started tasks. Java applications can be fully integrated as job steps to augment existing batch applications. Popular java batch frameworks like Jakarta Batch and Spring Batch are supported on the mainframe. Java batches can be used to initiate multiple threads in situations where serial processing is not a requirement.

Improving latency: collocating cloud-native applications with z/OS applications

There often is a choice in deploying new applications. Cloud-native applications running on Kubernetes often are used to build scalable applications that can be deployed in a multiple-cloud environment. The IBM Z platform provides a runtime environment for cloud-native applications, traditional z/OS workloads, and distributed applications.

The degree of latency and interoperability are key drivers in determining the environment. The applications that are off-platform (that is, distributed or public cloud) might encounter an order of magnitude higher latency while accessing core functions and data on the IBM Z platform, which also leads to lower throughput with frequent access to core functions and data.

Figure 5 shows how latency can be reduced or increased depending on where applications are.

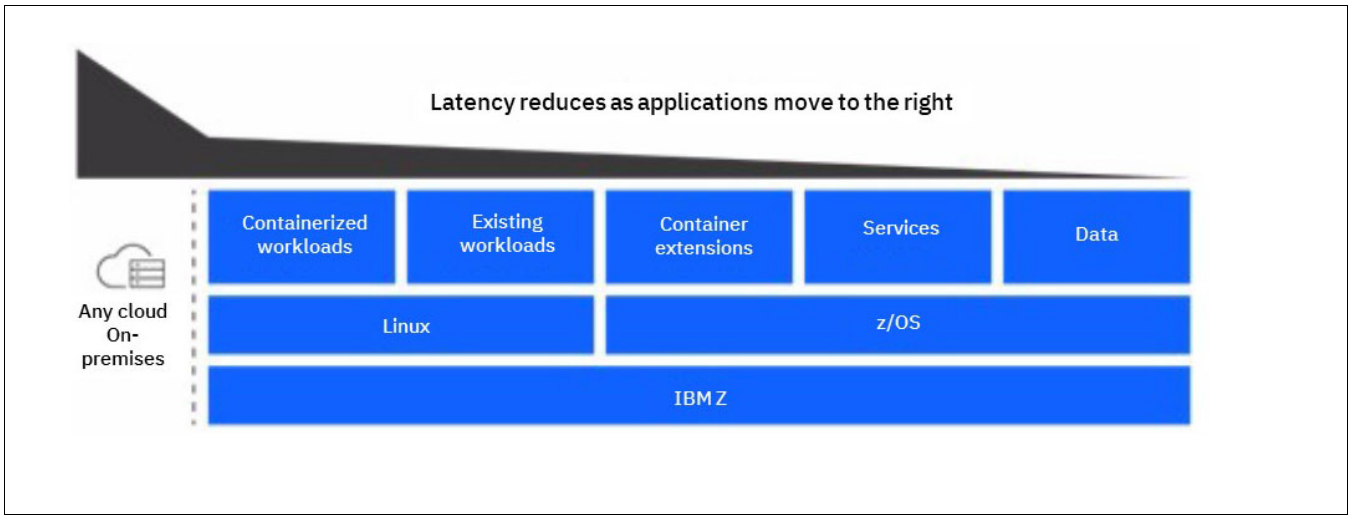


Figure 5 Reducing latency through colocation

Consider the following options:

- The public or private cloud across any cloud provider is a good option for applications that do not require low latency when accessing mainframe data.
- The applications that require low latency and have “data gravity” toward z/OS often are deployed within Linux on IBM Z or Red Hat OpenShift on Linux on IBM Z. IBM z/OS Container Extensions (zCX) is a new z/OS 2.4 feature that enables clients to deploy Linux applications as Docker containers on z/OS as part of a z/OS workload. This approach maintains operational control of the Linux environment within z/OS, brings z/OS qualities of service to the application deployment, and does not require the provisioning of separate LPARs or system images. When Linux on IBM Z is not a readily available option, these applications can also be deployed in z/OS Container Extensions, which help you integrate Linux on IBM Z applications with z/OS. For a sample architecture, see Figure 6 on page 17.

  Application developers can develop and data centers can operate popular open-source packages, Linux applications, IBM software, and third-party software together with z/OS applications and data.
- Existing enterprise applications on x86 that have a data affinity to mainframe applications are often on the IBM Z platform. In contrast to distributed systems where smaller components of distributed applications might be spread over multiple lower capacity hardware platforms, the IBM Z platform and LinuxONE provide significantly higher compute and I/O density in a small physical footprint, which leads to better vertical and horizontal scalability. Traditional distributed software typically has core-based pricing where higher cores lead to higher software licensing costs that lead to higher TCO. Thus, consolidation and colocation of application components through vertical and horizontal scaling improve both performance and the TCO.
- Java applications that deal with both data and transactional gravity requiring a global transaction scope with other z/OS applications are often deployed in IMS or CICS transaction systems in z/OS.

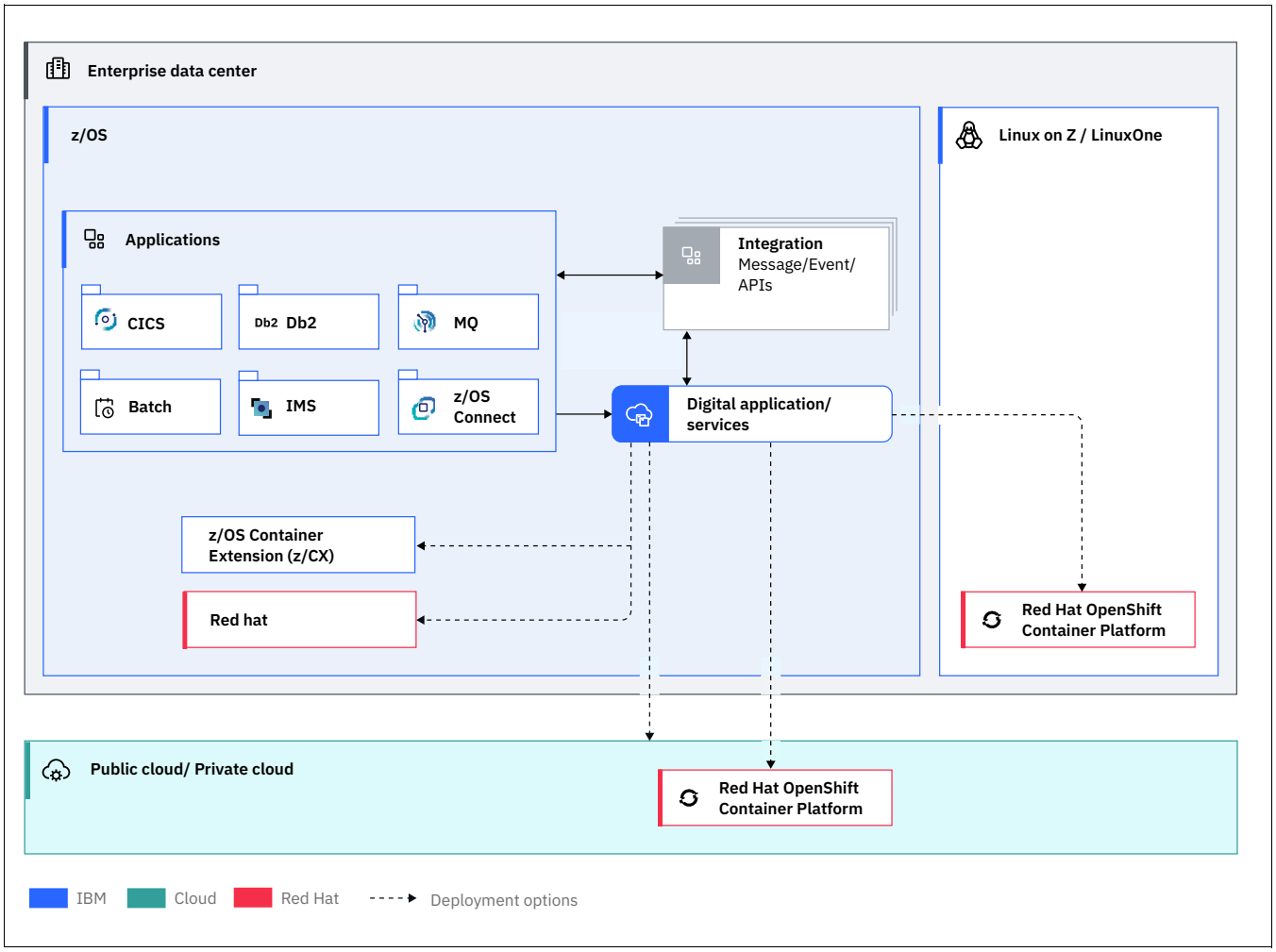


Figure 6 Enterprise data center with a mix of applications on z/OS, LinuxONE, and cloud

III. Integrate across hybrid cloud

When businesses embrace a hybrid cloud strategy, the strategy provides the option of selecting the “best fit” platform based on application characteristics and service-level requirements. This approach includes using and developing business applications on the IBM Z platform and the cloud to create hybrid applications.

Today, APIs are one of the most secure and efficient ways to expose application services and data to a consumer in a standard, controlled manner. A rich API ecosystem allows enterprises to deliver efficient user experiences and streamline business processes by collaborating with different stakeholders. The wealth of business-critical applications running on the IBM Z platform can and should be part of an API strategy so that enterprises can access their secure applications running on z/OS and integrate them into the digital services on cloud instead of rewriting functions.

APIs, cloud services, and trends across many industries are creating the need to transform business processes into real-time or near-real-time responsiveness, even those industry use cases that often are satisfied with latent information. Event-triggered fast-decision and response systems satisfy customers, whether they see the status of a payment, get food delivery, or track a shipment. This situation led to the rise of event-driven architectures and a subscription model of information processing.

When real-time data is needed at scale, the core mainframe systems must evolve continuously to handle the increasing volume and unpredictability of query traffic. It often is optimal to separate the models for data query from the model that is being used to update it. This method of caching data and information for various sources is detailed in “IV. Simplify information sharing and data access” on page 21.

Using APIs

APIs are a pervasive part of a digital ecosystem because they simplify access between applications. Open APIs are the standard for software to securely access and update low-volume business data on demand in real time.

The IBM Z platform is no different from any other platform when it comes to implementation and deployment of APIs. APIs might have the same function and the same interfaces regardless of the platform, but the reliability, scalability, throughput, and security of the APIs depend on the platform in which the APIs are services, and the IBM Z platform shines in this regard.

IBM z/OS Connect, which is available as part of the IBM Z and Cloud Modernization Stack, enables an enterprise to accomplish the following tasks:

- Create consumable APIs to make z/OS applications and data accessible to your enterprise and cloud applications.
- Call APIs from z/OS applications to extend them with data from internal enterprise APIs or public APIs from cloud services, for example, a weather or market data service.

IBM z/OS Connect includes a web-based designer that provides a low-code approach to creating APIs. The designer enables you to create APIs from existing code and use external APIs. The containerized deployment and tool model integrates fully with the enterprise DevOps approach, allowing developers to work in parallel to build the APIs that they need while system administrators maintain control through the container orchestration platform.

IBM z/OS Connect can be deployed either directly on z/OS or as a container-managed version, as shown in Figure 7 on page 19. Developers can create a secure API that is based on an existing Open-API 3.0 definition by using a “contract first’ approach that can support a company's API governance models and ensure that APIs are readily consumable by the business applications.

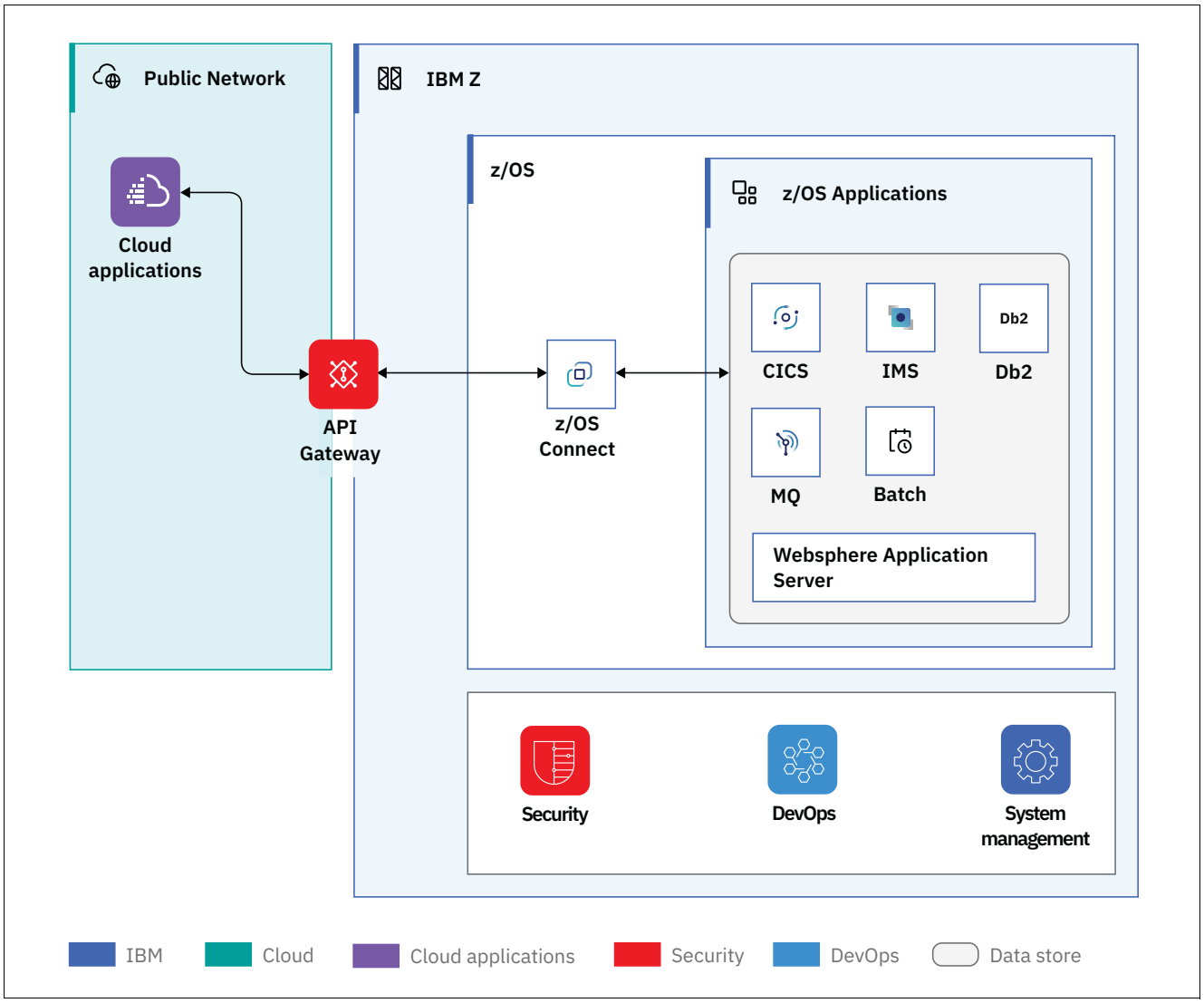


Figure 7 Using an API gateway application on cloud and applications on the IBM Z platform

The APIs can be orchestrated and governed by any OpenAPI 3.0 standard API gateway that is available in the enterprise hybrid cloud, which enables APIs on the IBM Z platform to be managed, secured, and made available through an enterprise API portal the same way that all other APIs in the enterprise are handled.

Mainframes as a part of an event-driven architecture

Every organization has critical business processes that require real-time data to make decisions or update consumers and become more responsive. The different events that comprise a process must be efficiently communicated, acted on, and processed to derive insights and intelligence. This information must be shared in a fast, efficient, and flexible way that supports business processes that help make decisions or even report the right status to the consumer.

To achieve the real-time delivery of relevant information to business processes, many organizations are implementing event-based architectures and backbones across their enterprises. Mainframe applications form an inherent part of this event-based architecture because critical business data is derived from the applications and system of records.

Apache Kafka often is used for event-driven applications. Kafka-based event backbones such as IBM Event Streams or Confluent Platform can run in the cloud and on the mainframe on Linux on IBM Z.

Here are some of the advantages of using Kafka and IBM Z software:

- z/OS Java applications can directly share event data to Kafka by using the Java Kafka Client library. For more information, see Setting up Apache Kafka for IBM Z.
- Existing business events that flow through IBM MQ or events that require transaction affinity can be reused with minimal impact to the existing assets. After the events are in IBM MQ, the IBM MQ Source Connector in Kafka Connect is used to produce the events to Kafka.
- When the volume of information to be inquired is large and aggregated from different sources, IBM Z Digital Integration Hub (IBM zDIH) can be used to optimize the current data.

A business application on the IBM Z platform also can act as a consumer of events by using the following similar mechanisms:

- A Java application on z/OS can use events by using the Kafka Client library and process them by using existing applications.
- Events can be streamed to an IBM MQ Queue through the IBM MQ Sink Connector in Kafka Connect where they can be used in a traditional way.

Figure 8 presents an example of an event-driven architecture.

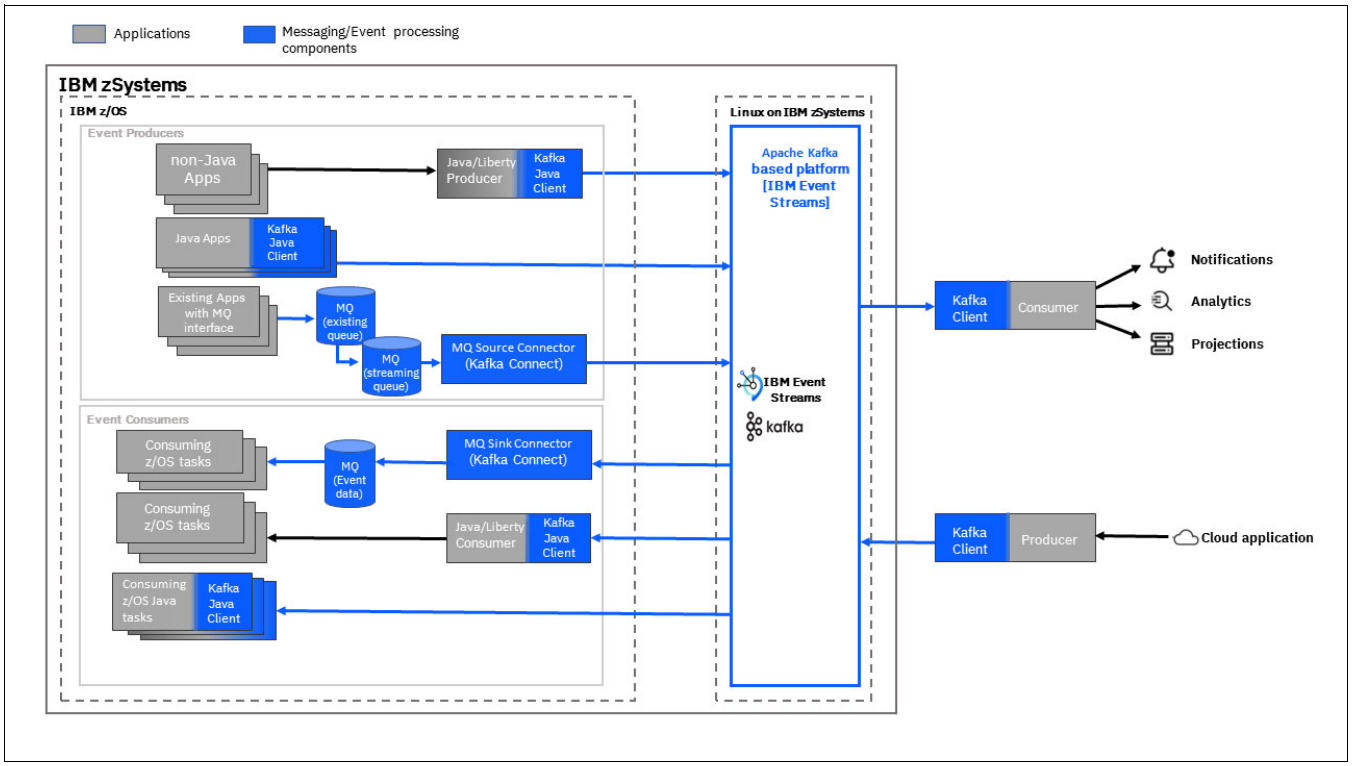


Figure 8 Event-driven architecture

Additionally, when event information is required at scale, an optimized Command Query Responsibility Separation (CQRS) pattern that uses IBM zDIH can feed into the Kafka queues.

The Kafka cluster is typically run outside of the IBM Z platform; however, IBM Event Streams can host the cluster on Linux on IBM Z under Red Hat OpenShift (in addition to cloud and distributed platforms). When the cluster runs on Linux on IBM Z, the adjacency of the cluster to mainframe workloads can provide resilience and scale for event flows along with the improved security and governance capabilities that the IBM Z platform provides.

After the events are in Kafka topics, they are available to be streamed across multiple cloud environments where the data can be used for notifications, analytics, or (projected, streamed, or used by) databases to provide local “in sync” copies of information. A few examples of using event processing through Kafka include getting real-time information for monitoring and analyzing trends, getting delivery notifications or processing notifications, or in business processes like ordering, claim handling, account debits, and credits.

IV. Simplify information sharing and data access

Data is the foundation for all businesses. This entry point is about the collection, gathering, and sanitization of data across the enterprise so that this data can be used for making business decisions, machine learning, and analytics.

Data is one of the most valuable assets that an enterprise possesses. As the needs of businesses grow, everyone in the organization needs access to data. For example, here are some of many examples of the need for data:

- Sales and marketing must respond quickly to changing markets and client behaviors.
- Teams need reliable financial forecasts.
- Operations must continuously improve automation with insight.
- IT systems must innovate and modernize applications.

The amount of data across the enterprise becomes untenable if there is no central governance and management mechanism. A data fabric helps to organize this data, access it in place with the right governance and compliance, and efficiently funnel it for the right purpose.

Much of the world's structured data originates on the IBM Z platform. Therefore, the data fabric solution that encompasses the IBM Z platform helps to drive better business decisions from the most current transactional data. It enhances customer-centricity by making structured and unstructured data from the mainframe available for decision making. It also ensures the right data security and governance.

The data modernization capabilities that exist on the IBM Z platform today allow the following goals to be met:

- The IBM Z platform becomes a part of a hybrid cloud data fabric through open access to both structured and unstructured mainframe data and data replication into the fabric for large-scale business intelligence and reporting.
- Information sharing for business software can respond to data that can be shared at scale without affecting the other business processes acting on it.

The IBM Z platform as a part of the data fabric

Data fabric is an architecture that intelligently and automatically connects the right data to the right people with appropriate governance with the right business context. This task is achieved through *metadata*, which is the description or annotation of data. The data fabric does not require a “rip and replace” of the existing data management infrastructure. Existing systems can passively participate in the fabric by sharing their metadata, as shown in Figure 9.

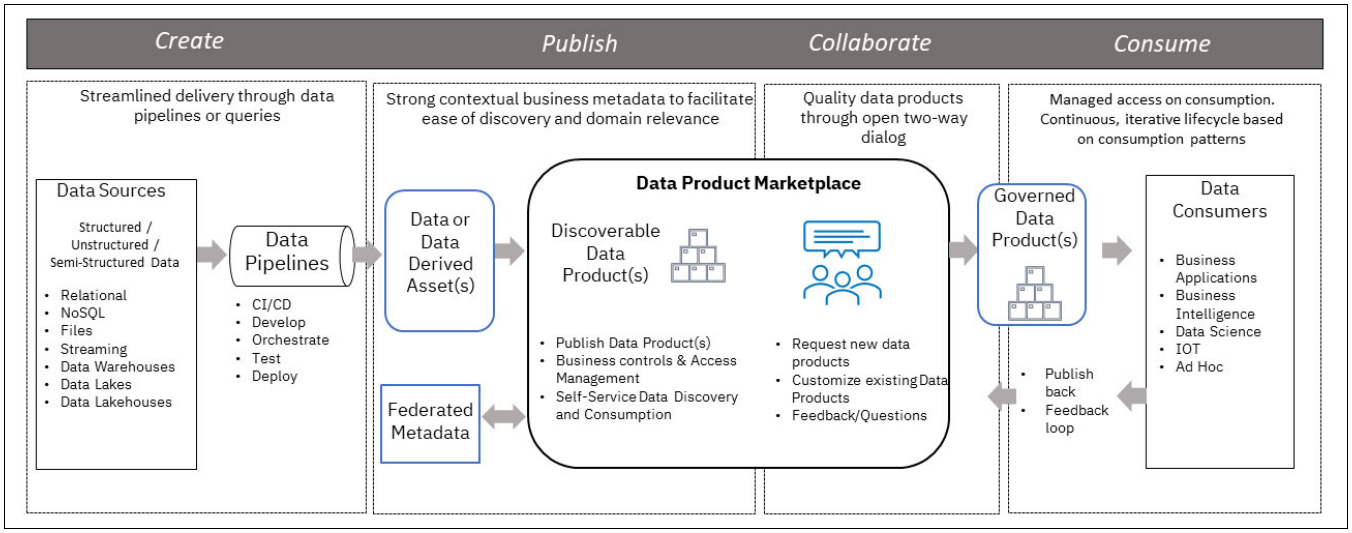


Figure 9 A create, publish, collaborate, and use data fabric architecture

The primary objective of the data fabric is to make the right data available to businesses for making the right decisions, whether it is for machine learning, business intelligence, or analytics.

A well-designed data fabric should achieve the following goals:

- Simplifies access to decentralized data across the hybrid cloud.
- Automates data governance and privacy.
- Integrates data across domains to provide a 360-degree view.
- Cleanses data to remove anomalies in real time.
- Provides for training models for AI.

When integrated with data from the IBM Z platform, the data fabric provides a full picture and augmented knowledge of IBM Z data to business users and provides the required governance and security when accessing replicated data from the IBM Z platform for data warehouses and business analytics.

In many of these processes, the data that is collected and organized is from the mainframe, so a comprehensive data management system involves robust mechanisms to digest data from disparate mainframe data sources.

Simplifying data collection

IBM Cloud Pak® for Data simplifies and automates data collection, data organization, and analysis, and accelerates the flow of AI throughout the enterprise.

The foundation for accessing data across disparate data sources is the IBM Watson® Knowledge Catalog in IBM Cloud Pak for Data.

Watson Knowledge Catalog allows for discovery and governance, and it catalogs all disparate sources of data in z/OS by using the metadata that is stored in Data Virtualization Manager for z/OS and Db2 for z/OS. Data Virtualization Manager for z/OS creates virtualized and integrated views of all data within z/OS, enabling applications read/write access to IBM Z data in place without having to move, replicate, or transform the data.

Relational and nonrelational data that is in Db2, VSAM, IMS, Adaptable Database System (ADABAS), or Integrated Database Management Systems (IDMS) can be accessed through HTTP, SQL, and REST (through z/OS Connect). Data Virtualization Manager for z/OS allows these disparate sources to be a part of the data fabric. Figure 10 shows how data collection can be simplified by using virtualization.

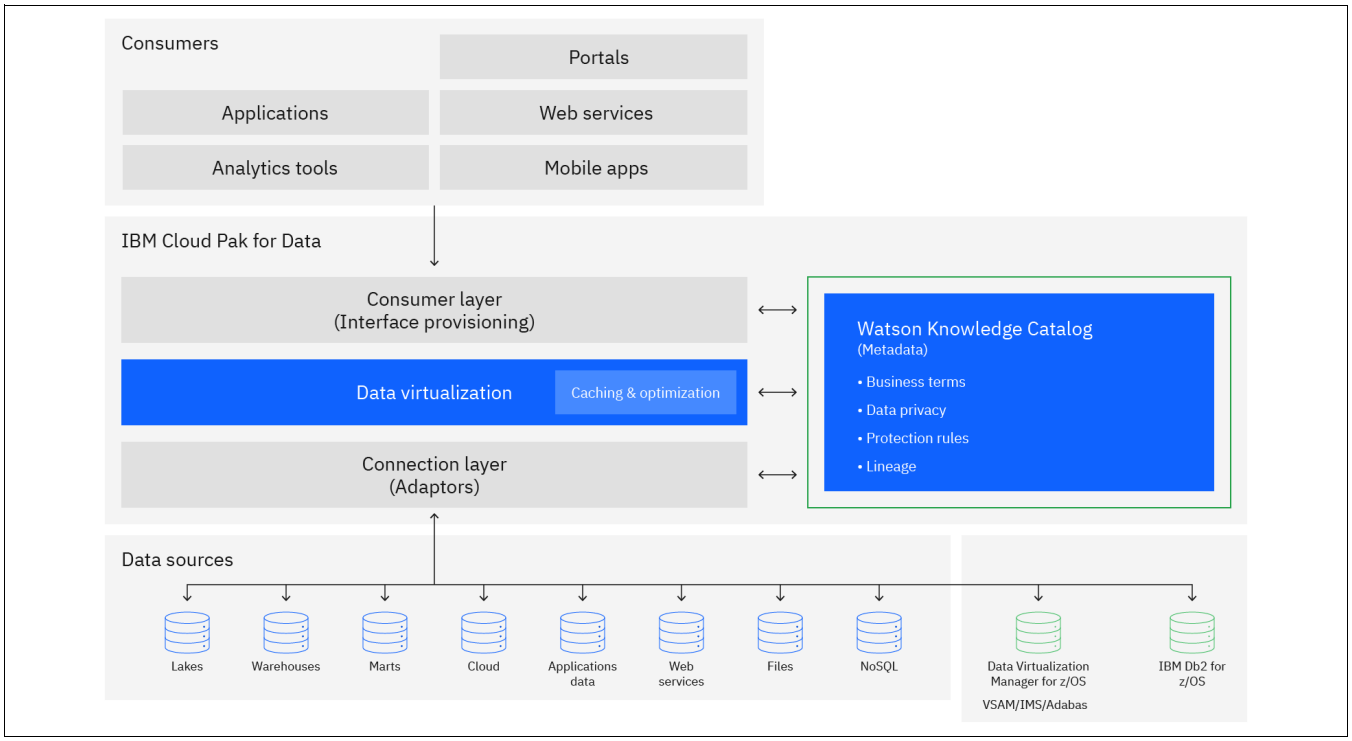


Figure 10 Simplifying data collection through virtualization

With a centralized view of data, including IBM Z data, within Watson Knowledge Catalog, you can build, test, and train machine learning (ML) models on the platform of your choice. Then, you can deploy AI models to Watson Machine Learning for z/OS to address complex information needs in business services that run on z/OS, as shown in Figure 11.

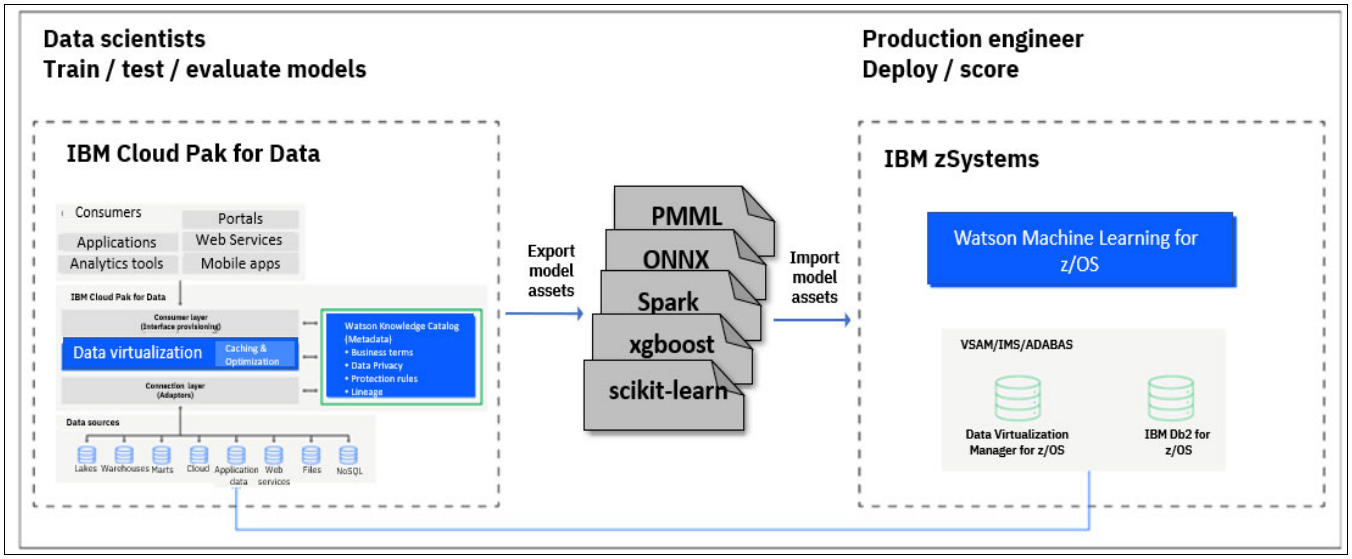


Figure 11 Using ML models to address complex information for business services

In both Figure 10 on page 23 and Figure 11, notice that the data sources include lakes, warehouses, marts, cloud, application data, web services, files, and NoSQL.

Optimized real-time information at scale

Digital transformation leverages both APIs and events to create meaningful interactions with core business processing and expand the ecosystem. Event processing enables efficient decisions that are based on the most current information, which can improve overall business outcomes.

When real-time information is required at scale, directly accessing data sources is not recommended. Core applications might be impacted; you might not get the most current or consistent data; you might have data models that are not easily consumable by hybrid cloud applications; or you might not see complete or compiled information, such as account balances.

These use cases are examples that show why it is preferable to decouple increasing and unpredictable queries from the critical application. This task requires businesses to implement an optimized CQRS pattern that separates the unpredictable large-scale queries to keep them from affecting day-to-day operations.

The goal is to access real-time data at scale for digital transformation, event processing, and cloud services.

IBM Z Digital Integration Hub

IBM zDIH provides a real-time flow of information between core business systems, which are hosted on the IBM Z platform (z/OS), and using applications, which are hosted on hybrid cloud. IBM zDIH offers optimized performance and minimal impact to the core applications when applications need real-time mainframe data at scale.

IBM zDIH includes highly performant caches that use Java based and runtime environment and in-memory databases, and an IBM zDIH Developer Kit that auto-generates the caches and applications for accelerated low-code adoption, templates, and samples to accelerate the integration with CICS and IMS applications through application exits, direct writes from the applications, or by using IBM MQ on z/OS.

Figure 12 shows how to apply IBM zDIH to implement an optimized CQRS pattern for accelerating mainframe core systems of record that are integrated with hybrid cloud.

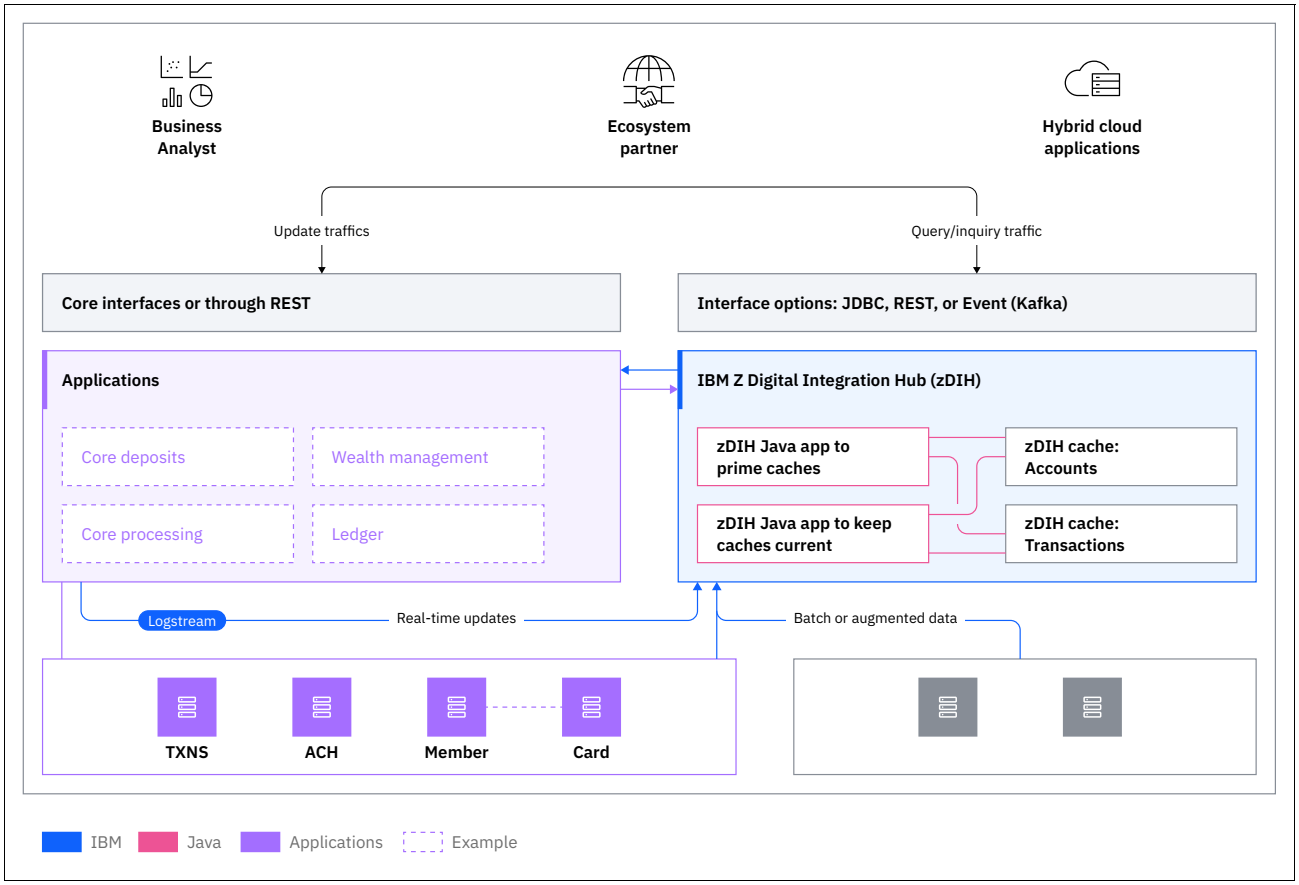


Figure 12 Implementing an optimized CQRS pattern by using IBM zDIH

The optimization is the result of running IBM zDIH on zIIP processors, not requiring systems of record to initiate transactions for all inquiry traffic, and not recomputing information that might not have changed.

IBM zDIH can directly flow cache information to enterprise-wide architectures for event processing (through Kafka), API management, or mediation on public cloud. Kafka topics can be configured to be updated after any information in the caches change. Those topics can be used by public cloud (SaaS) solutions that derive significant value from real-time characteristics, and potentially improve the time to value of cloud-native applications.

With the data modernization methods that are available from IBM, the valuable data that is contained in the mainframes can now be a part of the data and event processing ecosystem within an enterprise:

- For real-time information needs like business event processing, cloud digital services, enterprise applications and services, the mainframe data and information should remain within the platform and accessed directly from the IBM Z platform. As a best practice, use APIs when you need for digital and cloud services to access and update information at low volumes in near real time.
- A data fabric simplifies access to decentralized data across the hybrid cloud, automates data governance and privacy, and integrates data across domains. IBM Cloud Pak for Data can integrate data sources from the IBM Z platform into the fabric by accessing it in place. The metadata catalog itself can be within Red Hat OpenShift as part of IBM Cloud Pak for Data in any cloud provider.
- The IBM Z platform has the necessary data warehousing capabilities for analytics and business intelligence. As a best practice, place this data wherever the enterprise data lake or warehouse is. If the data lake or warehouse is not on the mainframe, IBM has the secure tools that allow data to be replicated with the right security in a read-only fashion for business analytics.

In summary, as described in Figure 13, the IBM Z platform lends itself to be a part of a data fabric in an enterprise and to modern data access. It provides open access to all data from applications regardless of the format by using Data Virtualization Manager for z/OS.

Transactions and analytics can be performed concurrently without affecting SLAs with rapid acceleration of queries by using IBM Db2 Analytics Accelerator. The IBM Z platform can also provide teams that require real-time information and event processing at scale access to data by implementing a CQRS pattern by using IBM zDIH as part of an event-driven architecture.

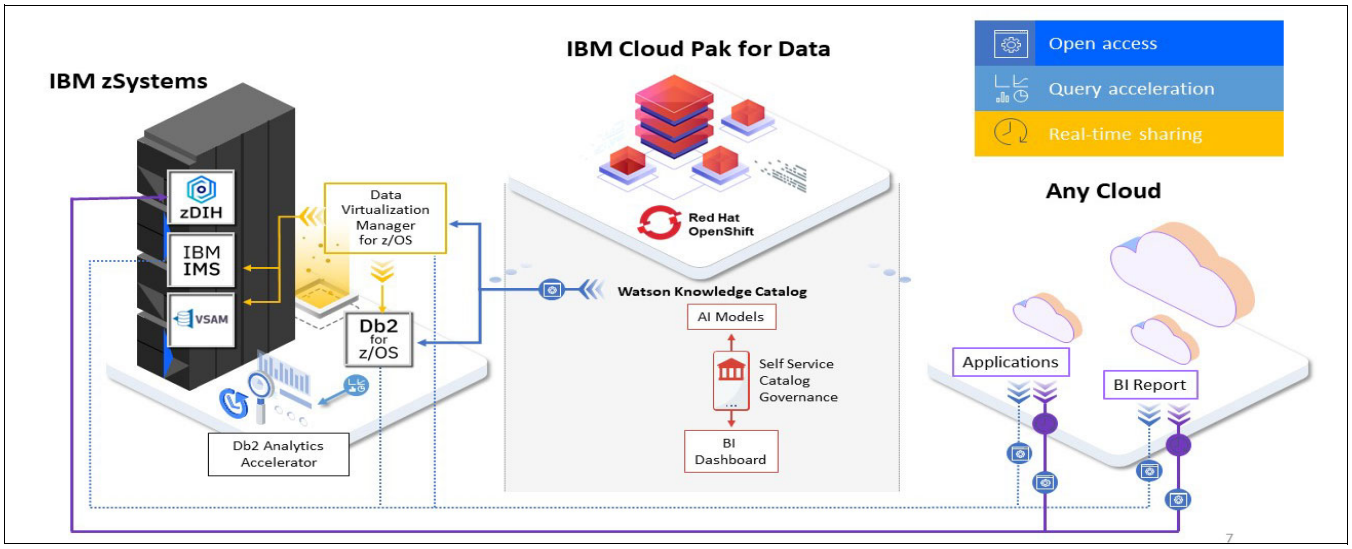


Figure 13 IBM Z integration with IBM Cloud Pak for Data and cloud

V. Adopt enterprise DevOps and observability

Adopting current DevOps practices and enabling a modern developer and operator experience helps remove any unnecessary differences in developing and managing applications on the mainframe, making it easier to attract new skills to the platform.

As an entry point for application modernization, the IBM approach to DevOps and its capabilities are built on top of a set of transformation principles and a modern and open source-based solution that is based on Git. This approach enables enterprises to have a platform-neutral, enterprise-wide common governance model for building enterprise applications.

DevOps practices in mainframe development can potentially achieve higher business agility by bringing down their release cycles from a few times a year to on-demand (multiple times a day) by automating test environment configurations and deployments while also safeguarding quality.

IBM DevOps tools bring mainframe applications into the enterprise CI/CD pipeline. When deployed, IBM can provide full application observability for hybrid applications, including those components running on the IBM Z platform.

DevOps as a culture

DevOps is a way of working that drives a culture of continuous learning, improvement, innovation, transparency, visibility, and trust. Large enterprises must adopt new practices and develop a new culture to truly transform so that they can constantly innovate and adapt to market demand. This approach requires software development teams to also adopt modern tools and implement automation. It is possible for the IBM Z platform to be a part of the enterprise DevOps strategy.

A DevOps culture helps organizations to build a resilient software delivery process and address any skill gaps. A single heterogeneous pipeline should be used to orchestrate the development, integration, and deployment of an application across multiple target platforms and environments with the right security, governance, and quality gates in place, such as the IBM Z platform, the cloud, or other parts of the enterprise.

As with development processes, IT operations should include the IBM Z platform as part of the enterprise application observability. Moreover, the integration of AI in IT operations and observability has become an important consideration for organizations looking to modernize their IT operations and stay ahead in an increasingly competitive business landscape.

Observability provides organizations with the visibility and understanding that they need to effectively manage the performance, reliability, and scalability of their hybrid applications. The deployment of AI-driven solutions can enable organizations to better manage the growing complexity and scale of their IT infrastructure, address skills gaps in their teams, and improve SLAs while reducing costs.

By automating routine tasks and synthesizing insights, IT teams can focus on high-value activities, freeing up time and resources to drive innovation and improve the overall quality of service delivery.

Furthermore, using AI can help identify and resolve issues in real time by addressing downtime and outages, which can lead to a better user experience and increased customer satisfaction.

DevOps and CI/CD for the IBM Z platform

The IBM vision and strategy around DevOps processes is to ensure enterprise standardization of tools and processes, and requiring the IBM Z platform-specific capabilities only where necessary. Here is the approach:

- Adopt or extend open-source and enterprise standard tools to z/OS usage in those areas where a platform-specific tool hinders a unified enterprise development process. This task includes adopting enterprise standard tools for source code managers (SCMs), editors, pipeline orchestrators, automation for integration tests, and artifact repositories.
- Develop specific IBM Z capabilities in areas that should be platform-specific, such as unit test automation, building, and deployment of traditional z/OS applications.

With this approach, mainframe development can use the same DevOps tools across the enterprise for on-premises or cloud.

The IBM Z platform supports many programming languages, toolchains, and development practices. An integrated development environment (IDE) is used for modern development. The IBM strategy is centered on the concept of “bring-your-own-IDE” for z/OS development. These IDEs, such as Eclipse, Visual Studio, and container-based web IDEs on Red Hat OpenShift Dev Spaces, allow developers to analyze and innovate for the mainframe by using the languages and tools with which they are comfortable so that they can focus on the business logic to implement.

Mainframe applications typically were developed over several decades, and as a result, the knowledge in these applications might not have been maintained at the same level as with recently developed applications. This analysis helps developers, architects, and business analysts to regain ownership of their code, understand how the different pieces interact, and understand the data model that is involved.

This analysis also provides current information with versions of the application that likely differ from what was documented. IBM ADDI helps close this skill and knowledge gap by providing up-to-date, consumable information about mainframe software assets and resources so that architects and developers can visualize application flow, and generate reports to act on their modernization strategy and plan increments with confidence.

According to the IBM commissioned study A Mainframe DevOps Journey Starts with IBM Application Discovery Tools-Sun Life accelerates digital transformation through modernization:

“Time to market is critical to going up against digital competitors and to meeting client expectations. What we've done shows that with DevOps tools like ADDI (Application Discovery and Delivery Intelligence), the mainframe can be just as quick as distributed environments in the long run.”

The article further states:

“… developers used ADDI to find complex connections between a variety of investment and insurance applications. It took us 50% less time than a line-by-line code search.”

“In addition to identifying where the connections were in the code, the developers needed to know whether the changes followed COBOL's rules-based restriction. Using ADDI enabled us to reduce analysis time by 35%.”

Continuous integration is a practice that enables teams to develop in isolation incrementally and continuously integrate code and test the increments. A modern SCM is essential for any continuous integration pipeline.

A modern SCM like Git also brings configuration, automation, and control of all technologies that comprise an enterprise application into the same SCM, enabling auditability and standardization of pipeline practices across the organization. Modern SCMs support parallel development and controlled code merges, unlike traditional z/OS based SCMs. This approach allows developers to code and unit-test their changes in isolation. z/OS applications often are written in COBOL, PL/I, assembly language, JCL, Java, Go, Python, and other languages. The databases and middleware configurations can be first-class participants in a Git-based pipeline with deployable binary files that are stored in an artifact repository.

By using an SCM, applications can be built once and deployed many times, which enable traceability of source code and the ability to deploy versions across multiple environments. In Java and distributed technologies, build systems like Maven can help you to connect all the dependencies that are required for a build. IBM Dependency Based Build understands the necessary dependencies and provides a modern build toolkit to build traditional mainframe applications in Git.

Continuous integration requires new code to be automatically tested as part of the pipeline to ensure that the built product is free from regressions. Building in test automation at various stages of development is key to CI/CD.

Pre-deployment test automation as part of the build phase allows developers to focus on updates to the code rather than any external impacts or changes. Modern languages such as Java or Python often have unit-testing frameworks that are available as either part of the core language or as an accessible framework. Traditional z/OS applications can be unit-tested in the same way. With IBM z/OS Automated Unit Testing Framework (zUnit) and IBM Z Virtual Test Platform, automated tests can be created and run in an environment-independent way as part of a pipeline, stubbing out run times like Customer Information Control System (CICS) or Information Management System Transaction Manager (IMS TM) and database calls. z/OS developers can unit-test COBOL and PL/I application changes within their branch in isolation before deploying them into a shared environment, which enables shift-left application testing. Because runtime environments are stubbed out, these tests can be run in a repeated fashion without resetting data. The unit tests are SCM-manageable, and they can automatically run as part of an enterprise CI/CD pipeline.

As application changes are integrated into a business function, the requirements for automation move away from being language-specific to integrating the external interfaces of the application together.

This testing must scale across platforms and address various levels of post-deployment testing, both driving the system of engagement while also validating those changes in the system of record. The IBM open-source automation framework Galasa provides seamless integration with mainframe applications. Most clients already have a set of post-deployment tests that they are running manually. Automating these tests speeds up application deployment and enables test teams to drive further quality improvements by deriving further tests.

For example, a hybrid application on Node.js that retrieves core data from a COBOL application through a REST API can be tested end to end with zUnit, IBM Z Virtual Test Platform, and Galasa. An enhancement to the COBOL program can be easily unit-tested in isolation with zUnit and IBM Z Virtual Test Platform. After the changes are merged and delivered, the post-deployment integration tests on the transactions, the API, and Node.js can be automated with Galasa.

Agile developer environments

On-demand environments are a part of continuous testing, so development teams can test their changes in isolation without breaking test systems or waiting for environments to be allocated to them. z/OS environments can now be spun up and down on demand in Red Hat OpenShift in IBM Cloud or on-premises, making z/OS development truly agile.

The idea of creating fit-for-purpose isolated test environments is a paradigm shift for mainframe development. This approach solves the problem of shared environments and brings the z/OS developer experience closer to any other compute platform.

These environments are available as part of the IBM Z Cloud and Modernization Stack as a service on IBM Cloud with IBM Wazi as a Service, and IBM Virtual Dev and Test for z/OS running on Linux on IBM Z. These products provide a way for development teams to ensure that their developer and feature branches are unit-tested successfully in isolation with an automated pipeline before business changes are merged for further testing.

Figure 14 presents a sample architecture that shows how z/OS applications can be active participants in an enterprise DevOps pipeline.

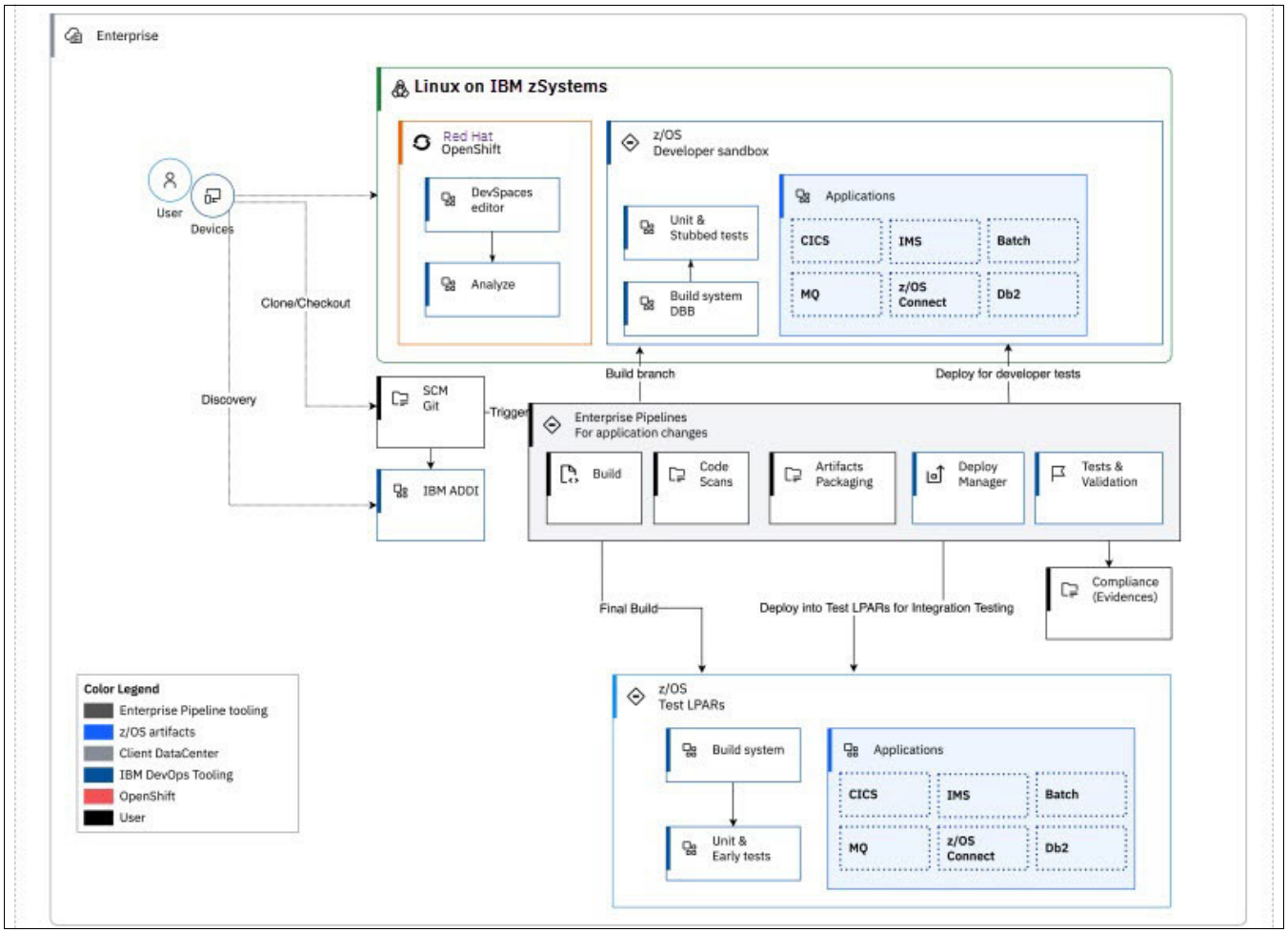


Figure 14 z/OS applications in an enterprise DevOps pipeline

The DevOps journey

The mainframe application development teams can start the DevOps transformation by breaking the silos that exist between the different platforms within the enterprise and sharing and adopting best practices. Starting a modern development experience should be done iteratively, that is, one step at a time. A value stream analysis approach can help determine what is the best place to start. The starting place could be new IDEs versus automated testing, depending on what brings the maximum value to the organization that is being assessed. An example of an iterative DevOps journey for most enterprises might include the following steps:

1. Adopt modern development tools like VS code, Red Hat Dev Spaces, or Eclipse so that new developers can edit, understand, and debug code. Introduce the concept of sandbox environments if a lack of test environments is a concern.
2. In parallel, identify opportunities for test automation for integration and system tests.
3. Design the development pipeline based on the current enterprise best practices and existing tools that already are adopted by the distributed systems.
4. On-board the mainframe applications iteratively into a Git-based modern development process where branches can be isolation. Create a basic pipeline that can build from a branch and deploy into existing environments.
5. Introduce development teams to the concept of unit testing. Add unit tests into the build step within the pipeline.
6. Expand the pipeline to include gating mechanisms, such as code scans, automated tests, and automated audit reports.

Industry-standard automation technologies and languages such as Red Hat Ansible and Python can be used consistently across the IBM Z platform and other platforms for automation and infrastructure provisioning within a pipeline. These tools can link into existing mainframe automation tools that are used within the enterprise. All these capabilities ensure that continuous integration is a reality for z/OS applications by aligning the z/OS development experience to the experience with any other cloud-native development experience.

Lastly, audited and controlled deployments are an important part of a CI/CD process. Scripted deployment frameworks that are based on Red Hat Ansible are equally applicable to z/OS applications. The deployments can be managed from cloud and Kubernetes based environments to bring the IBM Z platform into the modern deployment ecosystem.

Application observability

In addition to investing in your DevOps journey, you should ensure that you have the proper tools and processes in place to manage your hybrid application. The first step for realizing hybrid application observability, inclusive of the IBM Z platform, involves the following three primary tenets:

- **Improve visibility** for the IBM Z platform through hybrid cloud integrations to help with earlier issue resolution, faster decision making, and increased efficiency.[9] This task can be achieved by enabling mobile to mainframe application transaction tracing, streaming of golden signals for key infrastructure metrics and logs, and correlating monitored events and anomalies across the hybrid application estate.
- **Empower operation teams** with AI insights through proactive incident detection, automated correlations, and advanced log and data analytics. By leveraging machine learning to understand normal operations by using historical data, operations teams can automate thresholding settings for key performance indicators and accelerate diagnosis time by using anomaly correlations and modern analytics platforms to isolate the root cause.
- **Accelerate incident management** through workflow automation to increase the speed of routine and repetitive tasks and deliver AI-driven insights to where people work. Through chat-based collaborations and intelligent automation, known IT operational issues can use fully automated remediations while new issues can be transparently triaged to help avoid high-impact control room situations.

[9] https://www.ibm.com/z/hybrid-cloud

Faster issue resolution

The first tenet uses dynamic tracing technology that enables automatic tracing of transactions as they flow through the system, which provides a complete view of the application's behavior and performance across different systems that is inclusive of the underlying IBM Z infrastructure. SREs can identify issues and isolate hybrid application problems.

Through integrations with IBM Z infrastructure monitoring solutions, such as IBM OMEGAMON® products, key infrastructure metrics are collected, analyzed, and streamed to IBM Instana® so that the SRE can determine whether there is a broader infrastructure issue that might be impacting more than one of their hybrid applications.

When you take a hybrid application approach for proactive incident management, events and topologies from IBM Z infrastructure monitoring solutions can be integrated with the IBM hybrid AIOps platform, which is IBM Cloud Pak for Watson AIOps. IBM Cloud Pak for Watson AIOps applies multiple machine learning algorithms and advanced event correlation capabilities so that organizations can identify and resolve issues by aggregating and analyzing data from multiple sources to pinpoint the root cause across a hybrid application.

Figure 15 shows a sample architecture for anomaly detection.

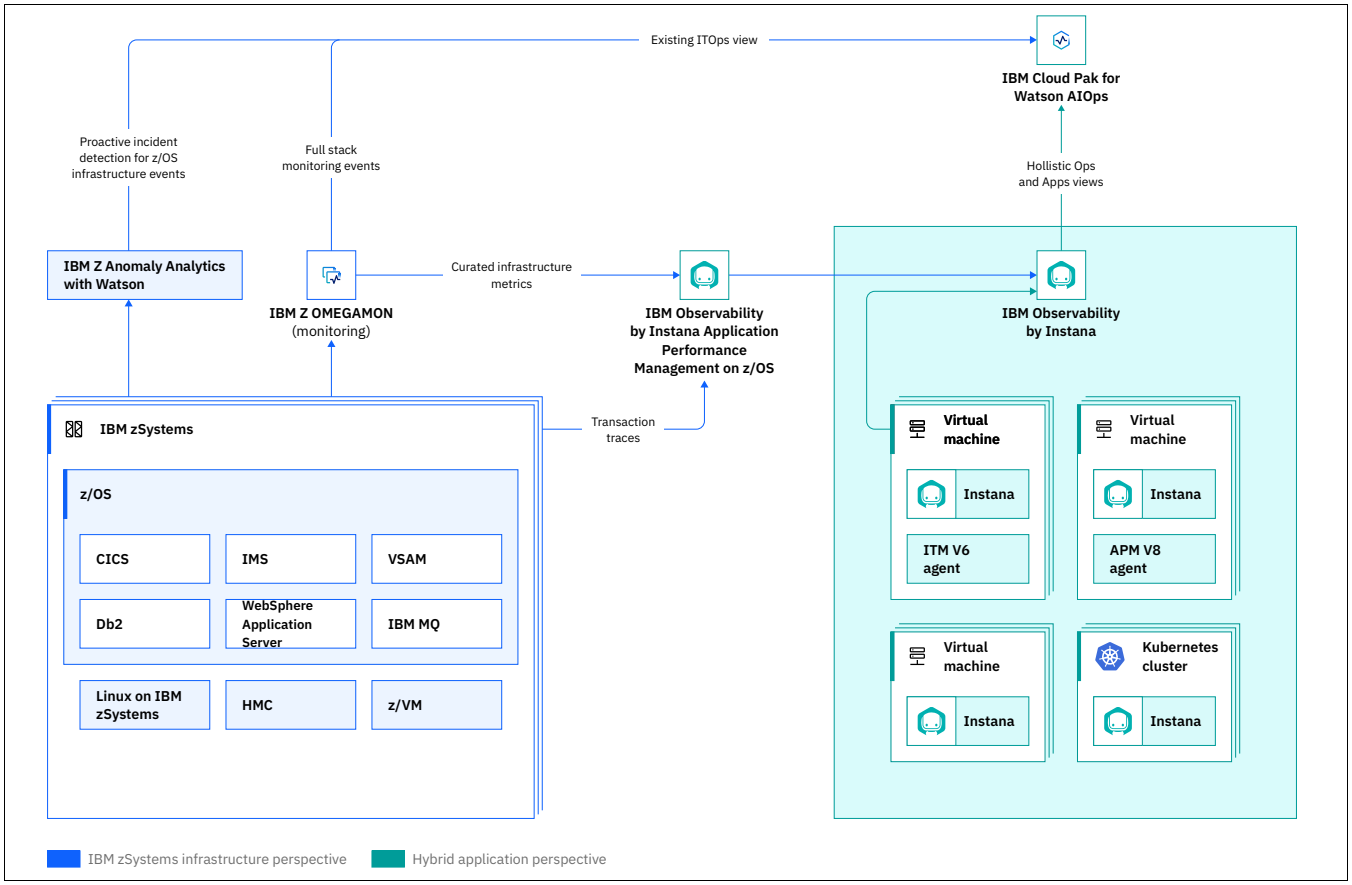


Figure 15 Sample architecture for anomaly detection

AI insights

The second tenet includes empowering IBM Z infrastructure teams with AI insights to help accelerate incident detection and time to action to help reduce diagnostic time. By leveraging machine learning and historical data to build models for what normal operations look like for your IBM Z infrastructure, IBM Z operation teams can get advanced warning of anomalous activity that could lead to a SLA-impacting event. By putting the work on the machine to automate threshold setting for alerts, these organizations can reduce their reliance on subject matter experts (SMEs) to set and manage thresholds in their traditional monitoring solutions.

When IBM Z infrastructure teams become aware of an operational issue, they can use extra AI and advanced statistical tools to accelerate the triaging and root cause analysis process.

Correlating distributed operational insights with insights from the IBM Z platform can accelerate hybrid incident identification and diagnosis time by streaming near real-time operational data to an enterprise analytics platform, such as Elastic or Splunk, to visualize and search hybrid operational data. Similarly, when triaging IBM Z infrastructure situations, analyzing historical data to identify, correlate, and prioritize anomalous problem signatures across z/OS subsystems might significantly reduce root cause diagnosis time for application workload impacts, critical infrastructure situations, and outages.

Automated workflow

The final tenet of AIOps involves accelerating incident management through automated workflows. There are several tools and techniques that can be leveraged, but one important hub for cross-team collaboration and cross-product integrations is IBM Z ChatOps.

IBM Z ChatOps provides a conversational interface for IT operations and application teams so that they can collaborate and break down organizational silos. In addition, IBM Z infrastructure teams can share mainframe insights with the application teams that rely on their services through integrations with chat tools like Slack, Microsoft Teams, Mattermost, and IBM Z management tools.

With this integration, IBM Z infrastructure operators and SMEs can query IBM Z domain tools and surface insights in chat so that everyone that is involved in the triaging process, including SREs and the application team, has full transparency, and knowledge sharing can be accelerated.

Overall, it is important for organizations that are embarking on mainframe modernization journeys to consider how the speed and complexity of their newly developed applications or features can be managed by the IT operations teams to ensure customer satisfaction.

When organizations gain visibility into how the IBM Z infrastructure supports their hybrid applications, the next step is to apply extra domain-specific tools to accelerate and automate infrastructure tasks. To learn more about how to apply AIOps solutions to your IBM Z infrastructure operations and management, download *Best practices for taking a hybrid approach to AIOps on IBM Z*.

VI. Make AI-driven decisions at scale

Each of the entry points that have been described help to drive businesses and optimize the processes in the mainframe. However, this next entry point is about how AI on the IBM Z platform can help bring intelligence into applications and business transactions, including use cases such as the following ones:

- Anomaly detection for operations
- Fraud detection modules within z/OS applications
- Intelligence in data analytics to bring efficient, cleaner data

AI is no longer an experiment that is limited to a select few organizations. Today, AI presents a huge opportunity to turn data into actionable insights and actions that can amplify human capabilities, decrease risk, improve operational efficiencies, and increase the return-on-assets by achieving break-through innovations. AI continues to have a profound impact on the way enterprises do business, and it is expected to revolutionize nearly every sector. Hence, the adoption of AI is not as much of a choice for an organization as it is a business imperative.

Data is the key currency of an information-based economy, and valuable insights can be drawn from this data. Every organization has its historical business data that can provide valuable insights in drawing conclusions. However, getting these insights where and when they are needed is a significant challenge. In other words, how can the mission-critical workloads of an enterprise use AI without impacting the SLAs of the applications?

Ideally, applying AI to every transaction can deliver a huge value to an enterprise, but an IBM sponsored paper with Celent (Operationalizing Fraud Prevention on IBM z16: Reducing Losses in Banking, Cards, and Payments) indicated that only 10% of transactions in high-volume enterprise workloads go through real-time AI screening. Hence, operationalizing AI has challenges.

AI inferencing on the IBM Z platform

The IBM strategy is to provide hardware and software capabilities on the IBM Z platform for clients to infuse AI in real time into their applications on the IBM Z platform. Therefore, the IBM z16 processor has an industry-first on-chip AI accelerator that is designed for high-speed and latency-optimized inferencing. It can accommodate 300 billion inference requests per day with a 1 ms response time. It provides consistent response times with optimized inference that can scale with IBM Z workloads and score every transaction while still meeting the most stringent application SLAs.

IBM has invested in open-source technologies such as Open Neural Network Exchange (ONNX) and IBM Deep Learning Compiler (DLC), and the AI ecosystem is built on top of the z16 accelerator by porting popular open-source data science technologies. With this infrastructure, enterprises can use their choice of software on the IBM Z platform.

The AI ecosystem is optimized to leverage many open-source AI frameworks, like TensorFlow, which enables seamless integration of open-source products into the IBM Z platform. This infrastructure promotes the flexibility of building and training AI models in any framework on any platform, such as on-premises, public cloud, private cloud, or hybrid cloud, and then deploying the AI models onto the IBM Z platform.

With tight integration of AI with data and core business applications that are on an IBM Z platform, you can leverage the qualities of service that you expect from the IBM Z platform, which ensures a design for applications and for AI resiliency, 99.99999% availability, portability of AI assets, and IBM Z flagship security for enterprise data.

Figure 16 shows how data can be organized, trained, and deployed on IBM z16.

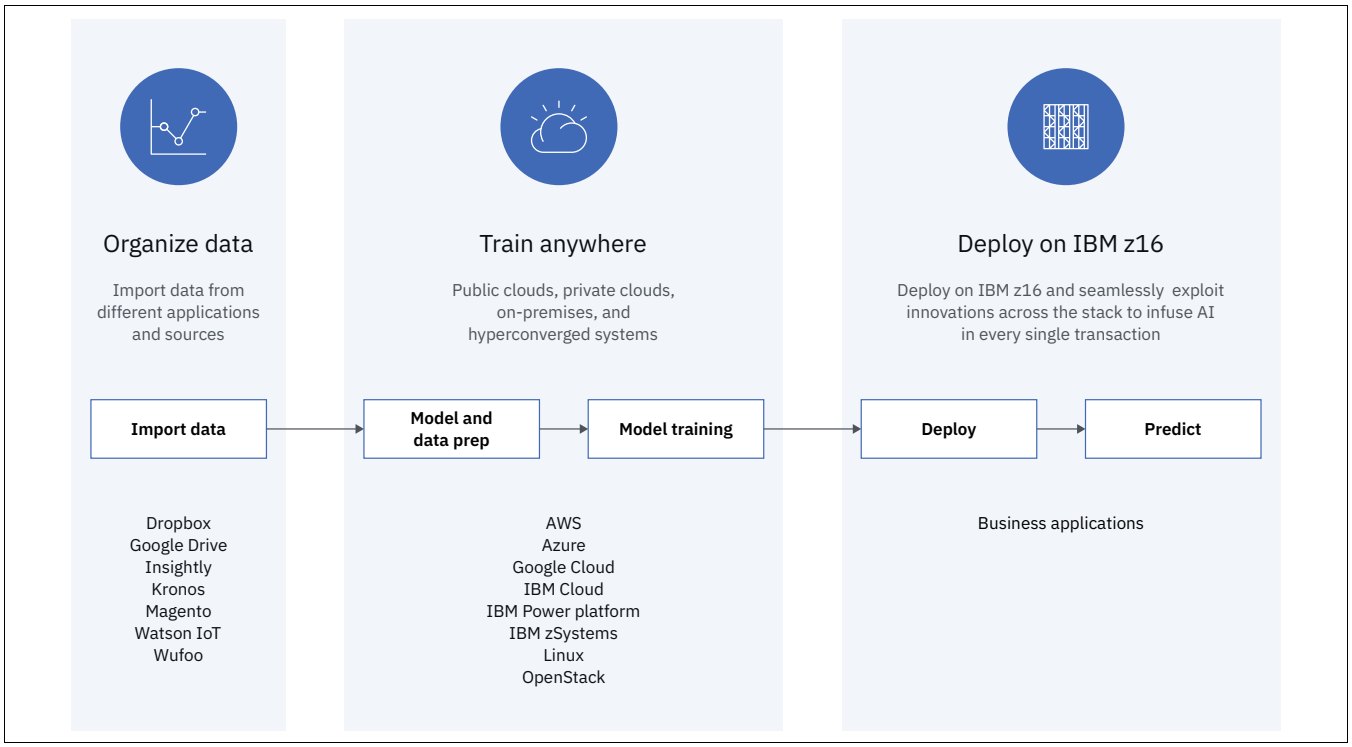


Figure 16 AI inferencing on the IBM Z platform

Enhancing z/OS applications to include AI inferencing

The following points explain some of the reasons why enterprises are now interested in enhancing their z/OS applications to include AI inferencing:

- **AI impacts many industries.**

  In the financial sector alone, it has been estimated that $298 billion dollars are lost today due to avoidable credit card transactions declines.[10] In addition, banks can score only 10% of transactions today by using the existing fraud detection algorithms, which result in a significant revenue loss.

- **On-premises fraud detection.**

  As an example, an American bank was unable to score all transactions in real time to detect fraud patterns and prevent fraudulent transactions. When using an off-platform AI inference solution, clients could not score 80% of transactions due to SLA impacts. However, with IBM z16 on-chip AI accelerator, this American bank now can score 100% of transactions in real time and on-platform, and get response times within their SLAs, which result in significant cost savings and improved customer satisfaction.

- **Clearing and settlement.**

  Preventing fraudulent transactions is paramount to maintaining trust and improving the customer experience. A credit card processor needed to predict which trades and transactions have high risk exposures and foreign currency involvement before settlement, with no impact to SLAs and the batch process window. This company was able to augment its existing rules-based approach and embed an AI solution that uses TensorFlow for high performance on IBM Integrated Accelerator for AI and low latency scoring, avoiding costly consequences and regulatory violations.

[10] https://www.ibm.com/downloads/cas/DOXY3Q94

The IBM approach to training models

The approach that IBM is taking for AI model development and training is to enable organizations to build and train models anywhere and enable deployment onto the IBM Z platform. Therefore, IBM is focusing on making assets that are created by using any popular framework to be portable on to the IBM Z platform. This goal is achieved by taking advantage of ONNX technology so that data scientists can build models in any framework and convert the models into a common, portable format. Then, ONNX models can be deployed on the IBM Z platform by using an ONNX model compiler that was developed by IBM Research®. This compiler builds on the ONNX-Multi-Level Intermediate Representation (MLIR) project to create a highly optimized inference program that can be embedded in applications.

The IBM Z platform also enabled other popular open-source data science packages like TensorFlow, Spark, and scikit-learn, and has optimized libraries and compliers of AI frameworks and run times that leverage the IBM Z architecture. With the cloud-native development ecosystem built around Red Hat OpenShift on IBM Z and the various open-source packages, enterprises can now build the AI ecosystem in the IBM Z platform.

IBM Watson Machine Learning for z/OS is the AI and machine learning platform for the IBM Z platform. You can train AI models in any framework on any platform, and then readily deploy those models onto the IBM Z platform to score and inference in-transaction in real time.

Figure 17 shows the AI services that you can leverage to build an AI ecosystem.

Figure 17 AI services, ecosystems, and business partners

Adoption of AI has shown promising results in various domains. In the industrial sector, deployment of AI can improve efficiency, increase customer retention rate, increase productivity, tremendously expand data processing capabilities, and augment human capabilities.

VII. Automate and standardize IT

IT automation and standardization, which is another IBM best practice entry point for a continuous application modernization journey, has been a critical part of large-scale enterprises for decades.

Now, this automation is necessary for those enterprises that embrace hybrid cloud. Codifying business processes and IT workflows with automation enables efficient functioning of enterprises. This approach in combination with automated operations and provisioning enables rapid fulfillment of everyday business and technical operations that enterprises demand. Even a small investment in automation can yield significant savings, freeing up highly skilled resources to accelerate more modernization.

Taking a narrow approach when applying automation often leads to needing expertise about multiple domains, duplicating efforts across teams and organizations, and building silos across departments. When dealing with application modernization, enterprises should implement robust and efficient IT automation to cope with growing complexity and meet the expectations of the consumers. This task requires a standardized approach that is consistent from the edge on the far end of the IT infrastructure up to the IBM Z platform as the central backbone of the enterprise.

Red Hat Ansible Automation Platform can run on multiple clouds, and it is a foundation for building and operating automation across an organization and IT environments. The platform includes the different capabilities that are needed to implement enterprise-wide automation across multiple clouds and the IBM Z platform. Red Hat Ansible Certified Content for IBM Z[11] includes collections that accelerate productivity and the automation of z/OS, z/OS middleware products, and other automation software by codifying IBM Z specific knowledge into interfaces.

Standardizing at scale

A standardized approach across the enterprise is needed to ensure the scalability and longevity of any IT automation solution that is required to meet the needs of a growing business. To standardize at scale, you need more than just intent, tools, or methodologies. Rather, you need a platform that can achieve the following tasks:

- Provides a broad foundation for automation that extends across domains (DevOps, networking, security, development, and others) and brings them together.
- Is open and integrates to meet the specific needs of each domain.
- Promotes the convergence and streamlining of processes.
- Can be quickly adopted, maintained, and scaled with ease and confidence.

Red Hat Ansible Automation Platform

Red Hat Ansible Automation Platform was designed to support hybrid cloud environments, both with deployment options and automation solutions. Orchestration across domains with Red Hat Ansible Automation Platform breaks down silos and enables integration of tools and processes across domains, which establish consistent governance of tools and processes. Clear and common interfaces provide a framework to leverage automation resources and extend or implement new automation.

Automation on the IBM Z platform can be roughly divided into three categories:

- **A system programmer's toolbox** with utilities for day-to-day routine tasks, such as building a new system, or on-boarding or off-boarding a user.
- **Automated operations** that deal with fully automated IPL and shutdown of the system, including all middleware and dependent applications. These operations also handle any kind of recovery, such as restart in place or fail over to another system, and, in combination with IBM GDPS, disaster recovery in a local or remote region.
- **Workload automation** that addresses the broad area of running batch workloads on the IBM Z platform that scale into millions of jobs a day.

[11] For more information, see https://www.ansible.com/integrations/infrastructure/ibm-zo.

Red Hat Ansible Automation Platform provides native programming interfaces for automation across various platforms and infrastructure, but there might be scenarios that require more automation and integration, for example:

- **Automation that is written by using other languages** can be driven by Red Hat Ansible or independently. Modern automation languages provide the utility, flexibility, and skills that are suited for a hybrid cloud automation strategy.

  Languages such as Python, Groovy, and Java are some of the languages that meet these criteria and specifically have IBM supported distributions that are developed for the mainframe, which widen the breadth of developers with the skills who can develop modern applications and automation on the mainframe. Python is an industry standard for automation today, and it can be used with support on the IBM Z platform with the IBM Z and Cloud Modernization Stack.

- **Integration with any existing mainframe automation leveraging native languages or tools**, such as JCL, REXX, CLIST, z/OSMF workflows, and IBM Z System Automation. Integration can be done through the capabilities that are provided by the Red Hat Ansible Certified Content for IBM Z collections.

  Red Hat Ansible Certified Content for IBM Z contains different collections that provide modules, roles, and plug-ins for automating against z/OS and other IBM Z middleware products, such as IMS, CICS, IBM Z System Automation, IBM z/OS Management Facility, and the IBM Z Hardware Management Console. With these collections, you can automate a wide range of use cases.

  Products such as IBM Z System Automation, which is optimized for automated operations, provide powerful policy-based, reactive, and specially designed z/OS automation capabilities that integrate with a dedicated Red Hat Ansible collection.

  Likewise, products such as IBM Z Workload Scheduler, which is optimized for batch job scheduling, can orchestrate millions of jobs, plan the daily schedules, monitor execution, recover from errors, and ensure that batch service levels are met. Such products can be integrated *as is* and augmented because they are brought into the Red Hat Ansible ecosystem with visibility across domains, teams, and use cases.

  Figure 18 on page 39 shows how this workflow and integration might work in a hybrid cloud environment.

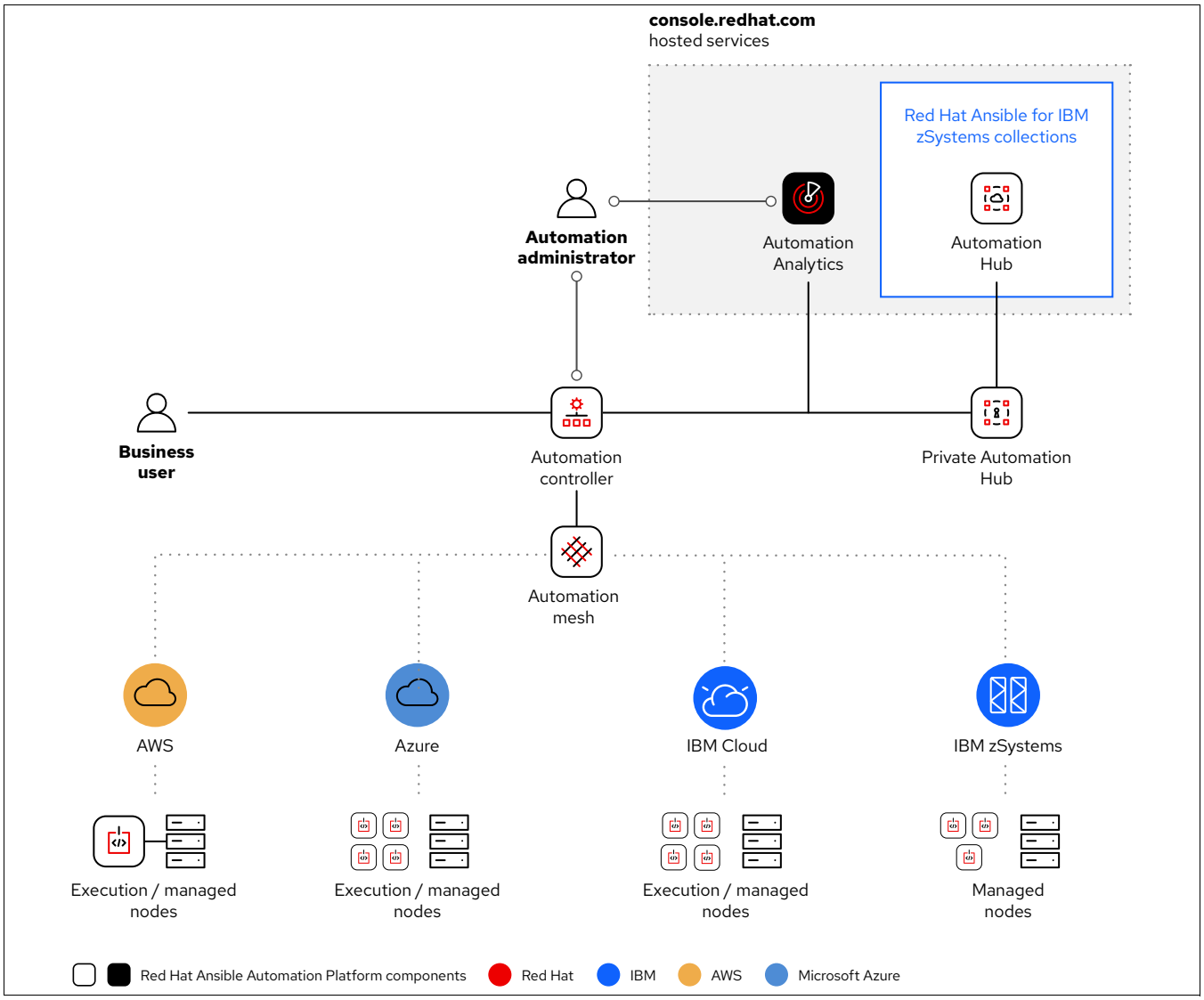


Figure 18 Reference automation workflow

By combining the capabilities of the Red Hat Ansible Automation Platform, modern automation languages, existing IT automation products, and Red Hat Ansible Certified Content for IBM Z, you have an integrated solution that is equipped for an enterprise hybrid cloud automation strategy, which is illustrated by the following examples:

- When a new IMS region is provisioned by using Red Hat Ansible, you can communicate this action to IBM Z System Automation to provide visibility, control, and automated operations by applying a role that is contributed by IBM Z System Automation to the Red Hat Ansible Certified Content for IBM Z. This action requires no human effort, takes effect immediately, and ensures that existing stakeholders can manage these resources.
- Like the capabilities in Red Hat Ansible Certified Content for IBM Z regarding access to z/OS job control language, z/OS, or TSO commands, you also can leverage IBM Z NetView or IBM Z System Automation REST APIs from playbooks to issue NetView commands or perform remote queries and health checks with automation that exists.
- Because IBM Z System Automation knows the state of every resource in a z/OS sysplex, you can integrate this stateful insight with Red Hat Ansible automation to reduce system state discovery tasks for precise and optimized automation processes, for example, to stop CICS and all dependent applications to allow Red Hat Ansible based maintenance.

Other use cases of Red Hat Ansible include the following ones:

- **Configuration management**

  The Red Hat Ansible declarative task approach and structural YAML syntax synergizes with configuration as code so that you can define your system configuration, such as an IPL, network, or security configuration in source-controlled managed files that Red Hat Ansible can employ in playbooks.

  Configuration artifacts for middleware such as server configuration or properties files can be managed the same way and deployed consistently and at scale to those middleware instances.

- **Security automation**

  Security automation for z/OS typically is a unique and isolated process from other platforms, such as SSL certificate renewal. However, with Red Hat Ansible, you can combine these disparate processes into a single streamlined renewal process that connects tasks on the target z/OS systems, and other interactions with off-platform certificate management software for a complete end-to-end cycle.

  Typical or daily procedures such as defining new users for a system and providing the necessary permissions and access to SAF resources or performing password resets can be streamlined, which can reduce the overhead for z/OS system programmers.[12]

- **CI/CD and application deployment**

  Red Hat Ansible can integrate into existing DevOps pipelines to drive deployment tasks. You can take advantage of rapid deployment of middleware applications in development and test environments and create new subsystem resources for those applications, such as defining new transactions, database updates, and queue resources.

- **Orchestration**

  Red Hat Ansible provides the flexibility to interact and orchestrate existing automation that uses JCL, REXX, z/OSMF workflows, Zowe CLI scripts, REST APIs, and other in-house automation.

  You can have better visibility across the enterprise to orchestrate existing automation across the hybrid cloud and the IBM Z platform.

  You can create automation or integrate by using existing automation. For example, the status of a system or particular run time may be state-managed by the z/OS automated operations solution, and a playbook can act on that state information to run where a particular workload is or is not running.

- **Gathering system information**

  You can collect system and security configurations for audit purposes, system status, or health checks. Pre- and post-process validation of the system or resource before performing a particular automated task also is available.

Enterprise IT automation should extend to include the IBM Z platform. The Red Hat Ansible Platform is suited to achieve this task across multiple platforms. It can create automation or leverage the existing automation capabilities in the enterprise like IBM Z System Automation to provide configuration management, system discovery for audits and health checks, and process automation.

Red Hat Ansible can be a part of a CI/CD infrastructure and application pipeline for provisioning and deployment, which ensures that z/OS middleware systems can be provisioned, managed, and maintained in a controlled way like any other platform that uses modern IT management practices like Infrastructure as Code (IaC). The configurations can be stored in YAML as with any other platform, and the automation scripts can be version-controlled to bring in governance and control mechanisms.

[12] System Automation for z/OS: Customizing and Programming

Self-service management of z/OS resources

IBM Z and Cloud Modernization Stack provides a common and consistent platform as a service (PaaS) experience, which includes self-service management of z/OS middleware environments and z/OS software installations. These software or middleware instances can be created and managed by developers of hybrid applications spanning z/OS, as shown in Figure 19.

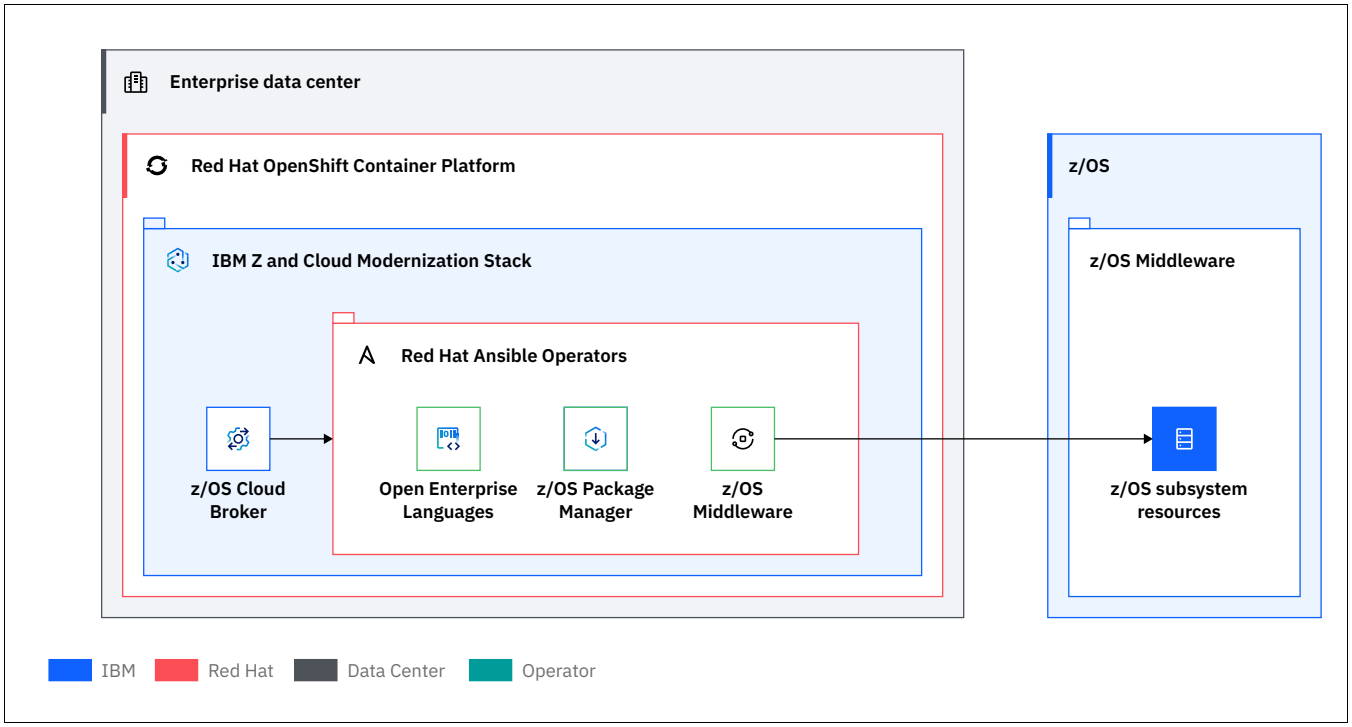


Figure 19 Self-service access to z/OS middleware

Software lifecycle management

IBM Z and Cloud Modernization Stack simplifies the installation of languages and tools for z/OS with Open Container Initiative (OCI) artifacts that are stored and retrieved on the IBM Cloud Container Registry. The following utilities and languages can be installed without SMP/e skills or unconventional file formats for UNIX based software installation on z/OS:

- IBM Z Open Automation Utilities
- Open Enterprise SDK for Python
- C/C++ for Open Enterprise Languages on z/OS
- 64-bit SDK for z/OS, Java Technology Edition
- Open Enterprise SDK for Go
- Open Enterprise SDK for Node.js

Middleware lifecycle management

IBM Z and Cloud Modernization Stack enables users to provision and manage z/OS middleware environments in a self-service manner. Users may request z/OS middleware environments through a Red Hat OpenShift catalog of available services. Provisioning of the middleware environments can occur against logical partitions (LPARs) on the IBM Z platform or virtualized environments like Wazi Sandbox.

Full lifecycle management of z/OS middleware environments can be achieved, and it can include the following items:

- New environment build and teardown
- Actions against provisioned environments (for example, start, stop, and bind)
- Optimized integrations between middleware environments that are deployed to z/OS
- Automated resource monitoring and reconciliation back to a wanted state

IBM Z and Cloud Modernization Stack provides a ubiquitous PaaS experience that ensures that deep z/OS skills are not necessary to be successful on the z/OS platform.

Reference architecture for modernization

A hybrid strategy that includes the IBM Z platform and cloud can significantly reduce skills gaps; enable rapid innovation with an agile DevOps approach; make it easier to access applications and data without significant changes; and optimize the costs of running or extending applications.

This approach can maximize business agility and ROI. The entry points in the previous sections explained the various ways to modernize mainframe applications with the hybrid cloud. This section describes some specific reference architectures that were developed in collaboration with AWS, Microsoft Azure, and IBM Cloud:

- Simplify access from digital channels with APIs.
- Share real-time information between z/OS applications and cloud.
- Leverage DevOps for z/OS applications on the cloud.
- Automate IT across the IBM Z platform and cloud.

IBM Cloud

In this section, we describe the solution architecture patterns for mainframe application modernization with IBM Cloud.

Simplifying access from digital channels with APIs

The REST APIs exposed from the IBM Z platform can be integrated into hybrid cloud applications that are running on IBM Cloud. The Financial Services validated IBM Cloud services provide the necessary orchestration, security, and operational management for the APIs and cloud-native business applications.

- IBM API Connect® can manage and expose OpenAPI standard APIs that are created from z/OS Connect, and it is available within IBM Z and Cloud Modernization Stack.
- Cloud applications can be run within cloud-native runtime services like IBM Cloud Code Engine and Red Hat OpenShift Kubernetes Services (ROKS) that are integrated in to the mainframe through REST APIs and Kafka based messaging with IBM Event Streams.
- The cloud-native applications can be deployed in a Secure Landing Zone with network isolation. Secure Landing Zone is a closed segmented network that complies with standards like NIST with private endpoints where the regulated workloads like Red Hat OpenShift on IBM Cloud (ROKS) are deployed.

The reference architecture diagram that is shown in Figure 20 provides a complete overview of the IBM Cloud for Financial Services® virtual private cloud (VPC) architecture pattern, which integrates with IBM Z applications in a hybrid cloud environment.

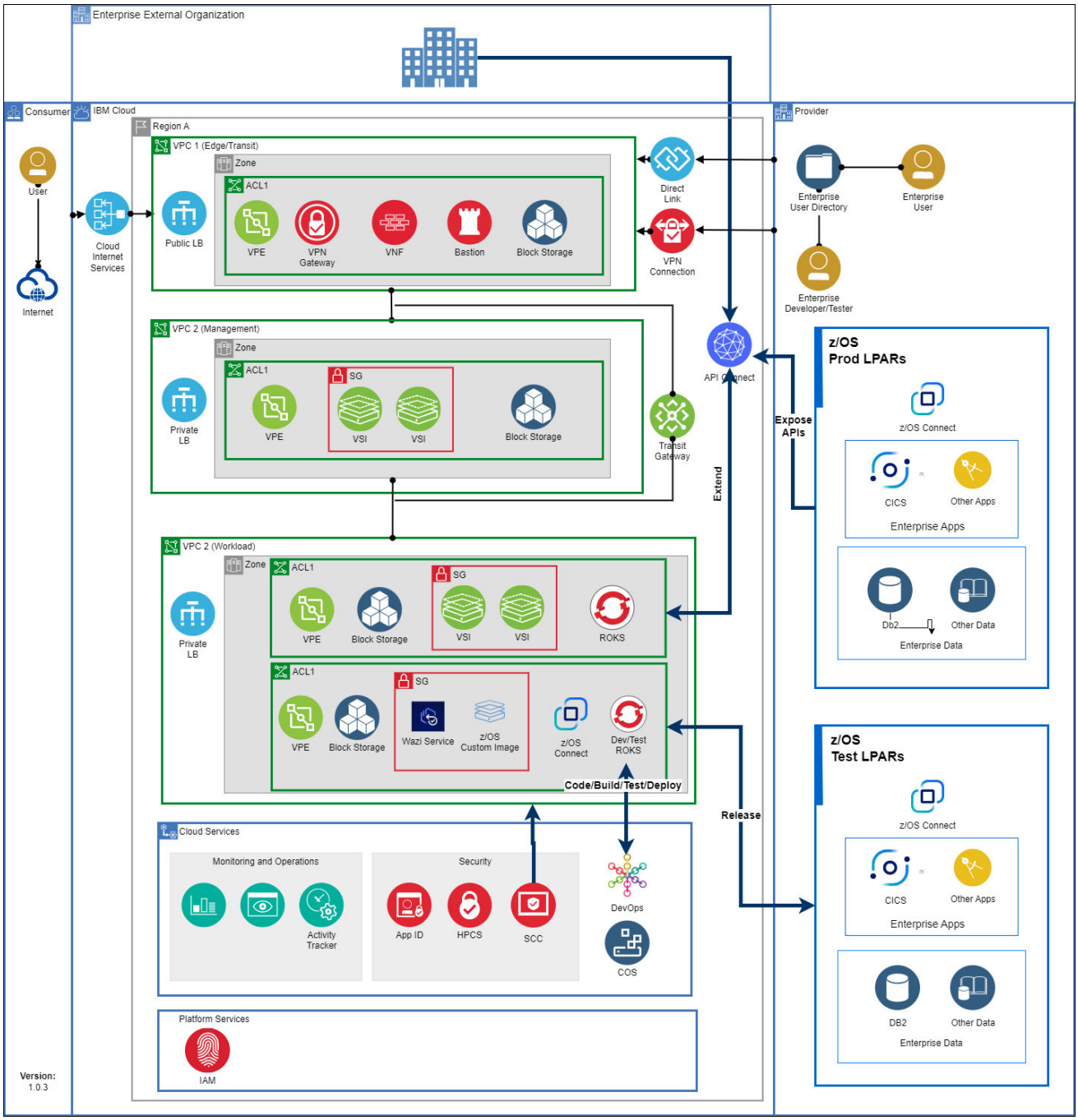


Figure 20 Accessing digital channels with IBM Z APIs through the IBM Cloud for Financial Services VPC architecture

This architecture is deployable through IaC by using IBM Cloud Tool Chain. IBM Z and Cloud Modernization Stack provides the development environment, and integrates with z/OS environments that are provisioned by using IBM Wazi as a Service. IBM Hyper Protect Crypto Services (HPCS) and IBM Security® and Compliance center provide for security, encryption, and compliance capabilities within IBM Cloud. All these resources are Financial Service Validated[13] and part of IBM Cloud Framework for Financial Services.

Figure 20 on page 43 provides a complete overview of the IBM Cloud for Financial Services VPC architecture pattern, which integrates with IBM Z applications in a hybrid cloud environment.

Sharing real-time information between z/OS applications and cloud

Enterprise business data is in the IBM Z platform. Depending on the use case, enterprises can adopt different data integration strategies to make mainframe data accessible to the applications in the cloud. IBM zDIH provides data for real-time information sharing at scale, adopting an on-premises caching strategy that is integrated directly with z/OS run times. The information that is exposed through IBM zDIH can be integrated with IBM Cloud through IBM Event Streams or accessed by cloud applications or the partner ecosystem in a secure manner. Information that is stored on IBM Cloud is strictly confidential and accessible to only the enterprise client.
HPCS supports Keep Your Own Key (KYOK), which ensures full control of the entire key hierarchy, where no IBM Cloud administrators have access to the keys.

Figure 21 shows the data flow with IBM zDIH and IBM Cloud.

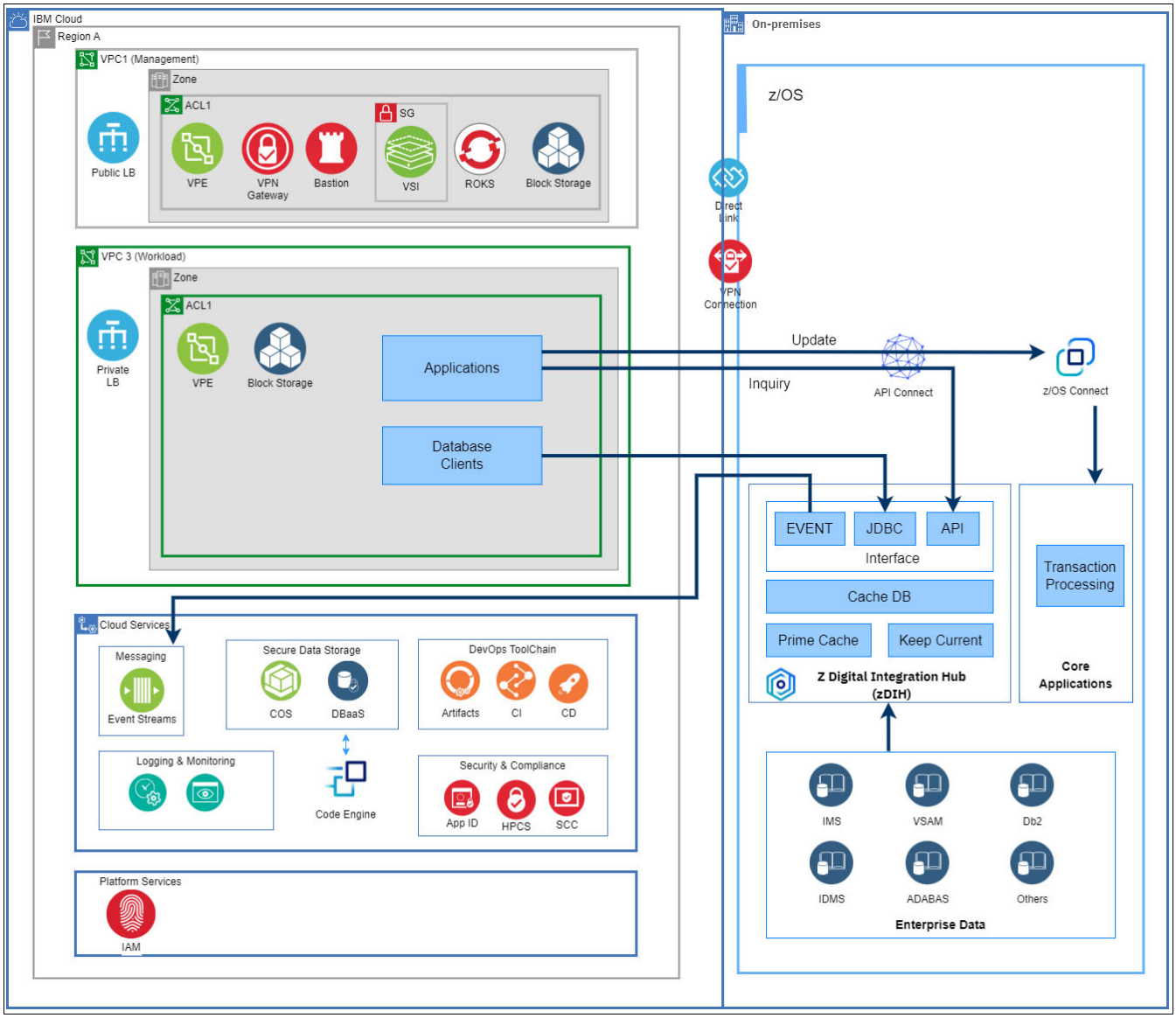


Figure 21 Real-time data sharing with IBM Z Digital Information Hub

[13] The Financial Services Validated designation signifies that you successfully complied with the control requirements of the IBM Cloud Framework for Financial Services, which might improve your ability to market to financial institutions.

Leveraging DevOps for z/OS applications on the cloud

IBM z/OS applications can be a part of the Enterprise DevOps toolchain that is hosted on IBM Cloud. IBM Cloud provides an on-demand Virtual Server Instance with z/OS, which is provisioned through the IBM Z platform and enables developers access to spin up z/OS development environments on-demand in a secure VPC infrastructure. IBM Toolchain and Continuous Delivery can integrate with the z/OS development environments on IBM Cloud and on-premises LPARs, enabling z/OS applications to be developed on sandboxes on IBM Cloud and securely deployed into IBM Z LPARs for automated testing. IBM Toolchain and Continuous Delivery provides capabilities for a secure, compliant pipeline with toolsets for code risk analysis, code scans, and deriving insights into the development cycle with DevOps Insights. The Secrets Manager, and Key Protect make the pipeline secure by providing the necessary encryption and security to build a zero trust pipeline.

Figure 22 provides an overview of z/OS application development in IBM Cloud through the Integrated DevOps toolchain.

For more information about the enterprise DevOps solution architecture on IBM Cloud, see Improve business agility and provide modern DevOps platform for mainframe applications with IBM Cloud.

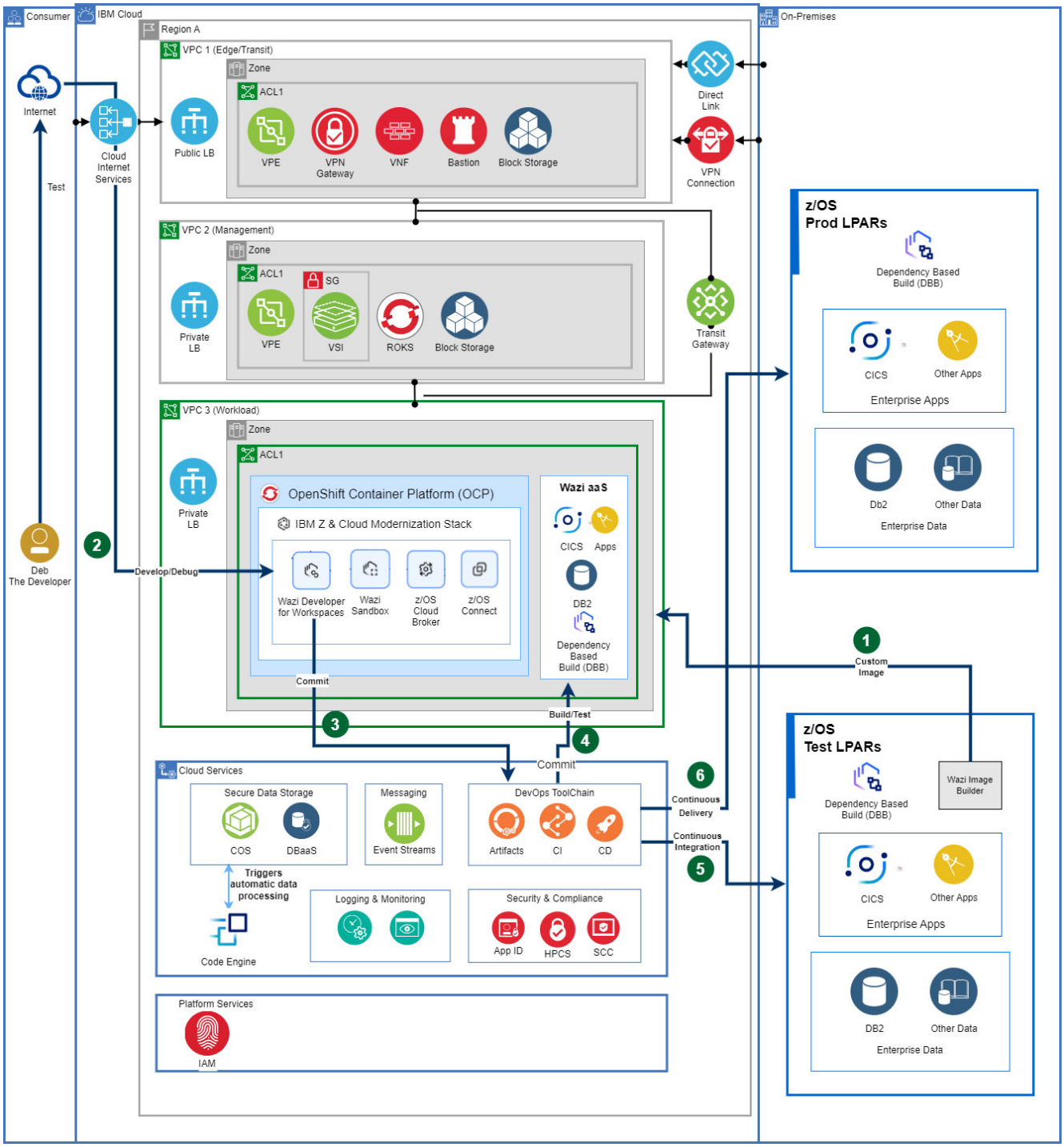


Figure 22 z/OS application development in the IBM Cloud through Integrated DevOps toolchain

Automating IT across the IBM Z platform and cloud

Automation is a top priority for organizations that want to reduce inefficiencies, lower production costs, reduce complexity in their IT environments, and discover and close gaps in security. IT automation scripts can be maintained as IaC by using IBM DevOps toolchains and the development environment capabilities in IBM Z and Cloud Modernization Stack. Red Hat Ansible Platform can be deployed on Red Hat OpenShift cluster in VPC in IBM Cloud, which helps to orchestrate IT automation scripts in a hybrid cloud environment.

The reference architecture that is shown in Figure 23 demonstrates the orchestration flow for IT task automation in hybrid cloud across IBM Cloud and the mainframe.

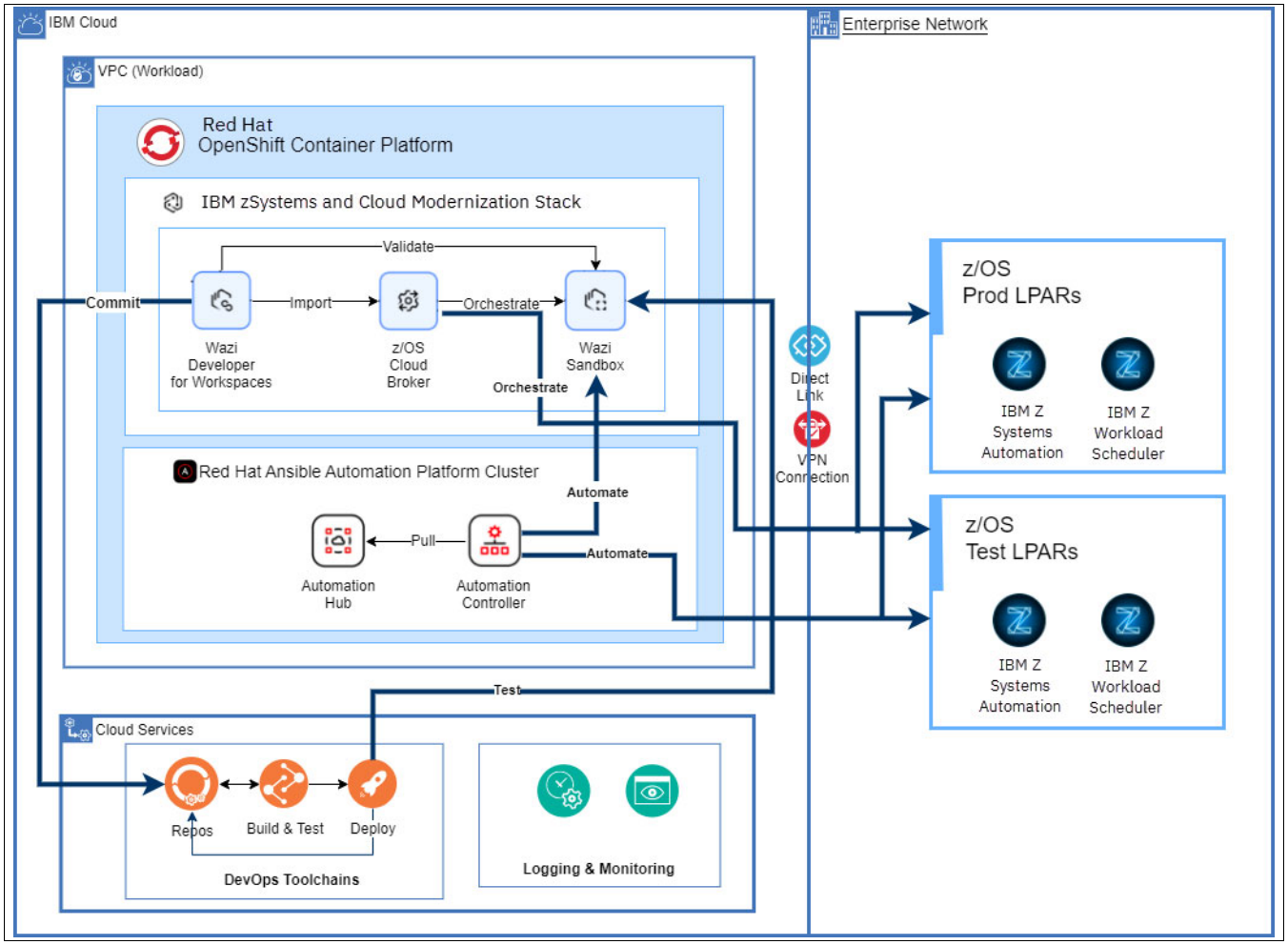


Figure 23 IT automation across the IBM Z platform and IBM Cloud

Amazon Web Services

This section describes the solution architecture patterns for mainframe application modernization with AWS.

Simplifying access from digital channels with APIs

The z/OS Connect capability, which is a low-code API solution component of IBM Z and Cloud Modernization Stack, provides scalable and secure z/OS APIs on Red Hat OpenShift on AWS. This configuration provides a simple way for cloud-native applications in AWS to access securely mainframe applications and data through RESTful APIs by using OpenAPI standards, as shown in Figure 24 on page 48.

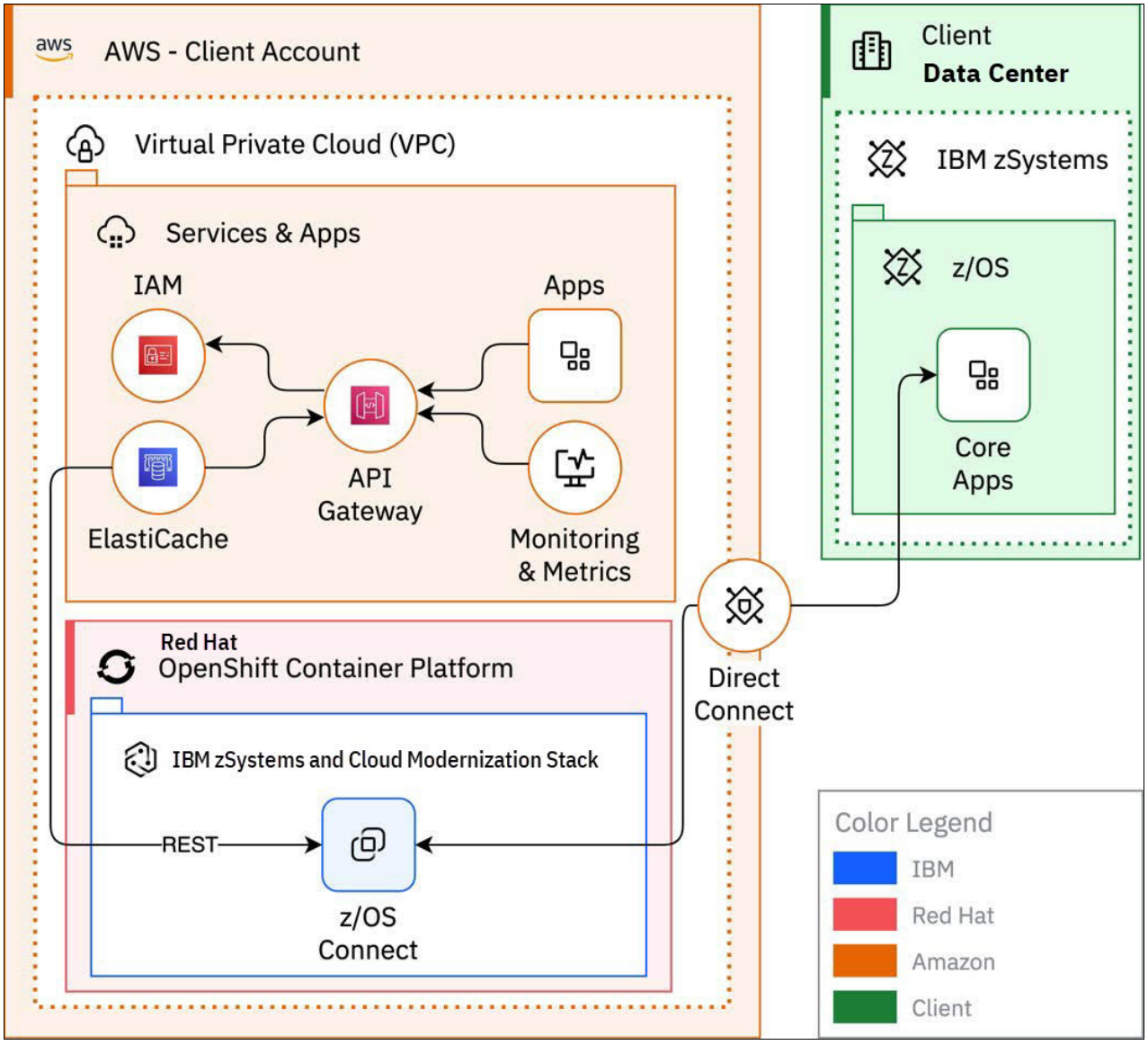


Figure 24 Reference architecture: Access through RESTful APIs

IBM Z and Cloud Modernization Stack, along with Amazon API Gateway, Amazon ElastiCache, and auto-scaling services provide a low-latency, horizontally scalable integration of on-premises mainframe applications and data. Platform connectivity between AWS and the IBM Z platform can be established with AWS Direct Connect and a virtual private network (VPN) link to connect securely the platforms with low-latency. APIs that are created with IBM Z and Cloud Modernization Stack can be integrated with enterprise API management solutions like Amazon API Gateway, which is a fully managed serverless service that makes it easy to create, publish, and secure APIs at any scale. As newer use cases and channels drive growth, Red Hat OpenShift on AWS can be scaled. Amazon Cognito can be used for web and mobile apps, where users can log in through Cognito user pools or federated identity providers (IdPs).

Sharing real-time information between z/OS applications and cloud

The real-time information that is required at scale from the IBM Z platform that is exposed through IBM zDIH can be integrated into AWS applications and services. The cloud-native applications in AWS can securely access this data through REST APIs or JDBC through AWS Direct Connect. The events that are exposed through IBM zDIH can be published into an event stream through Kafka Messaging Brokers that are installed on the AWS Cloud. The real-time update to this data in the core applications can be performed by using REST APIs, as shown in Figure 25.

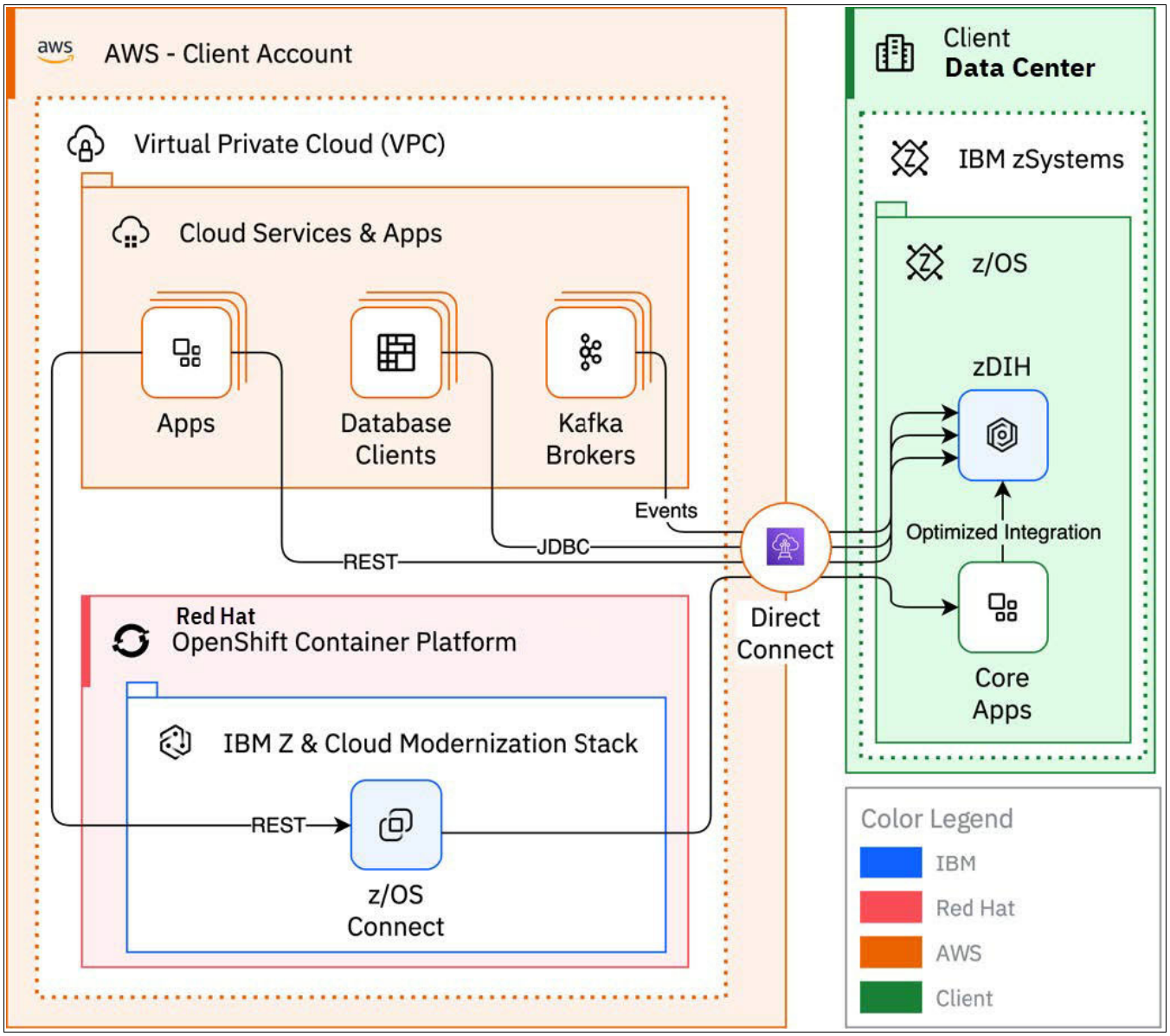


Figure 25 Reference architecture: Access through REST APIs

Leveraging DevOps for z/OS applications on the cloud

IBM Z and Cloud Modernization Stack in AWS provides a cloud-native developer experience for z/OS applications. The development and early testing can be performed in a z/OS sandbox in a Red Hat OpenShift environment, which enables teams to perform isolated development.

Figure 26 on page 50 provides an example of how an effective DevOps pipeline can be set up.

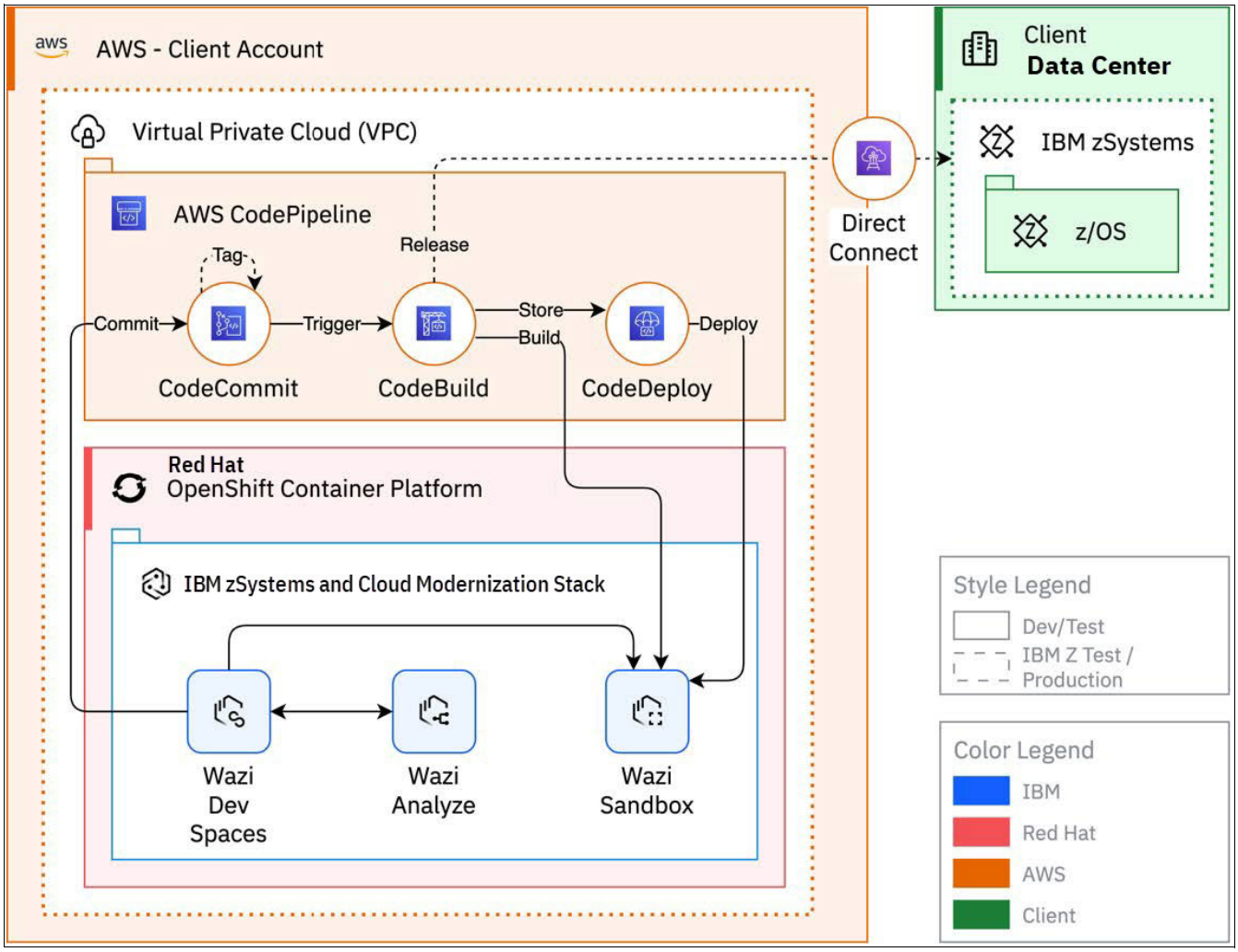


Figure 26 AWS Enterprise DevOps pipeline for IBM Z applications

Although the Wazi Dev Spaces provide an in-browser integrated developer environment (IDE) to code, build, and debug z/OS applications, Wazi Analyze and Wazi Dev Spaces can analyze and edit source code from a modern source-control management (SCM) tool like Git.

The DevOps tools are integrated with AWS CodeCommit and AWS CodePipeline. The build and deploy services on AWS, such as AWS CodeBuild and AWS CodeDeploy, invoke z/OS specific build and deploy services that are available as part of IBM Z Cloud and Modernization Stack.

Automating IT across the IBM Z platform and cloud

Red Hat Ansible supports IT automation for all architectures, including the IBM Z platform. Enterprises can benefit from automating and centrally managing their IT resources in a consistent way across hybrid multi-cloud environments. Figure 27 shows how Red Hat Ansible Automation Platform integrates with AWS cloud. Red Hat Ansible Automation Platform can be deployed in a VPC in AWS as a service or as a Kubernetes operator on Red Hat OpenShift on AWS. Red Hat Ansible Automation Platform can orchestrate automation scripts across AWS and the IBM Z platform through AWS Direct Connect. The automation scripts can be maintained as IaC by using AWS CodePipeline and the development environment capabilities in IBM Z and Cloud Modernization Stack.

Figure 27 presents an AWS Enterprise DevOps pipeline for IBM Z applications with Red Hat Ansible Automation Platform.

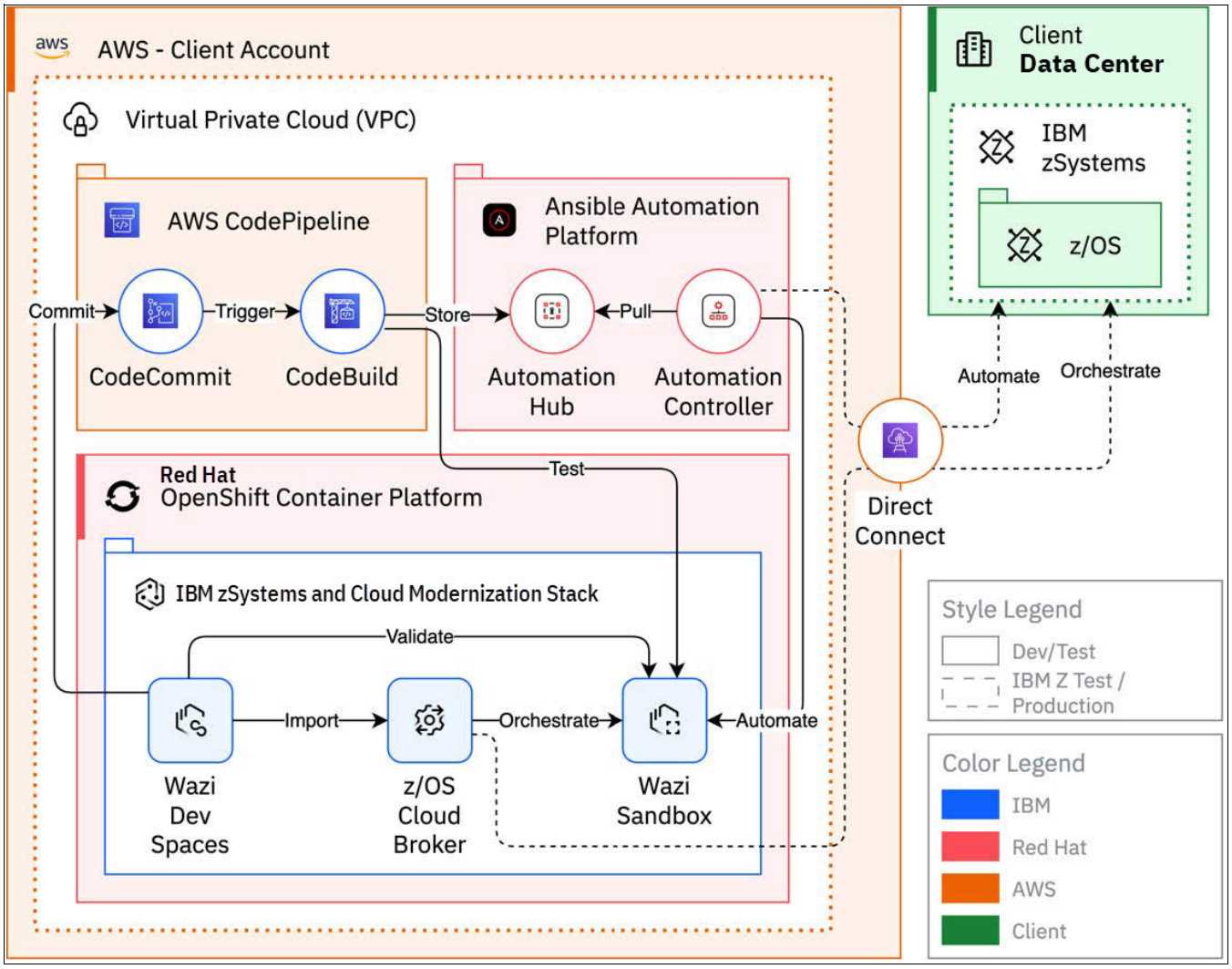


Figure 27 AWS Enterprise DevOps pipeline for IBM Z applications with Red Hat Ansible Automation Platform

Microsoft Azure

This section describes the solution architecture patterns for mainframe application modernization with Microsoft Azure.

Simplifying access from digital channels with APIs

IBM Z and Cloud Modernization Stack provides a low-latency, horizontally scalable integration of on-premises mainframe applications and data with Microsoft Azure Cloud services through APIs that are exposed through z/OS Connect (see Figure 28). Microsoft Azure ExpressRoute provides a secure direct connection between the cloud and the IBM Z platform, and auto-scaling services are provided with a Redis cache. Azure Virtual Network (VNet) is the fundamental building block for your private network in Azure.

Furthermore, APIs that are created with IBM Z and Cloud Modernization Stack can be integrated with enterprise API management solutions on Microsoft Azure and use the monitoring and metrics services that are provided as part of Microsoft Azure. As newer use cases and channels drive growth, Red Hat OpenShift on Microsoft Azure can be scaled.

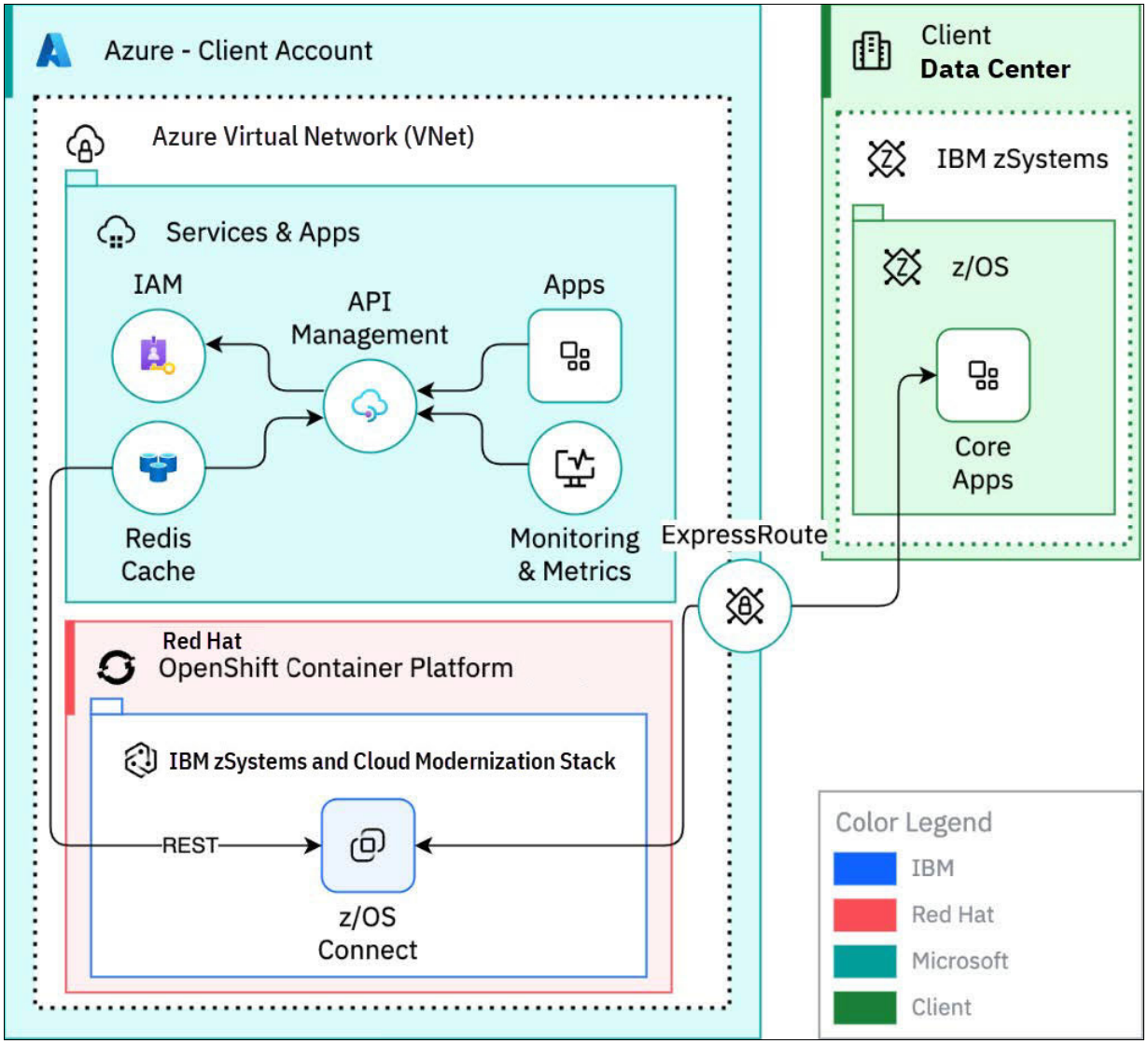


Figure 28 Reference architecture: Access through APIs

Sharing real-time information between z/OS applications and cloud

The real-time information that is required at scale from the IBM Z platform that is exposed through IBM zDIH also can be integrated into Microsoft Azure applications and partner services. The cloud-native applications in Microsoft Azure can securely access this data through REST APIs or JDBC through Microsoft Azure ExpressRoute (see Figure 29). The events that are exposed through IBM zDIH can be published in an event stream through Kafka Messaging Brokers that are installed within Azure Cloud. The real-time update to this data within the core apps can be performed through rest APIs that are exposed through IBM Z Cloud and Modernization Stack.

Figure 29 provides an example of this architecture.

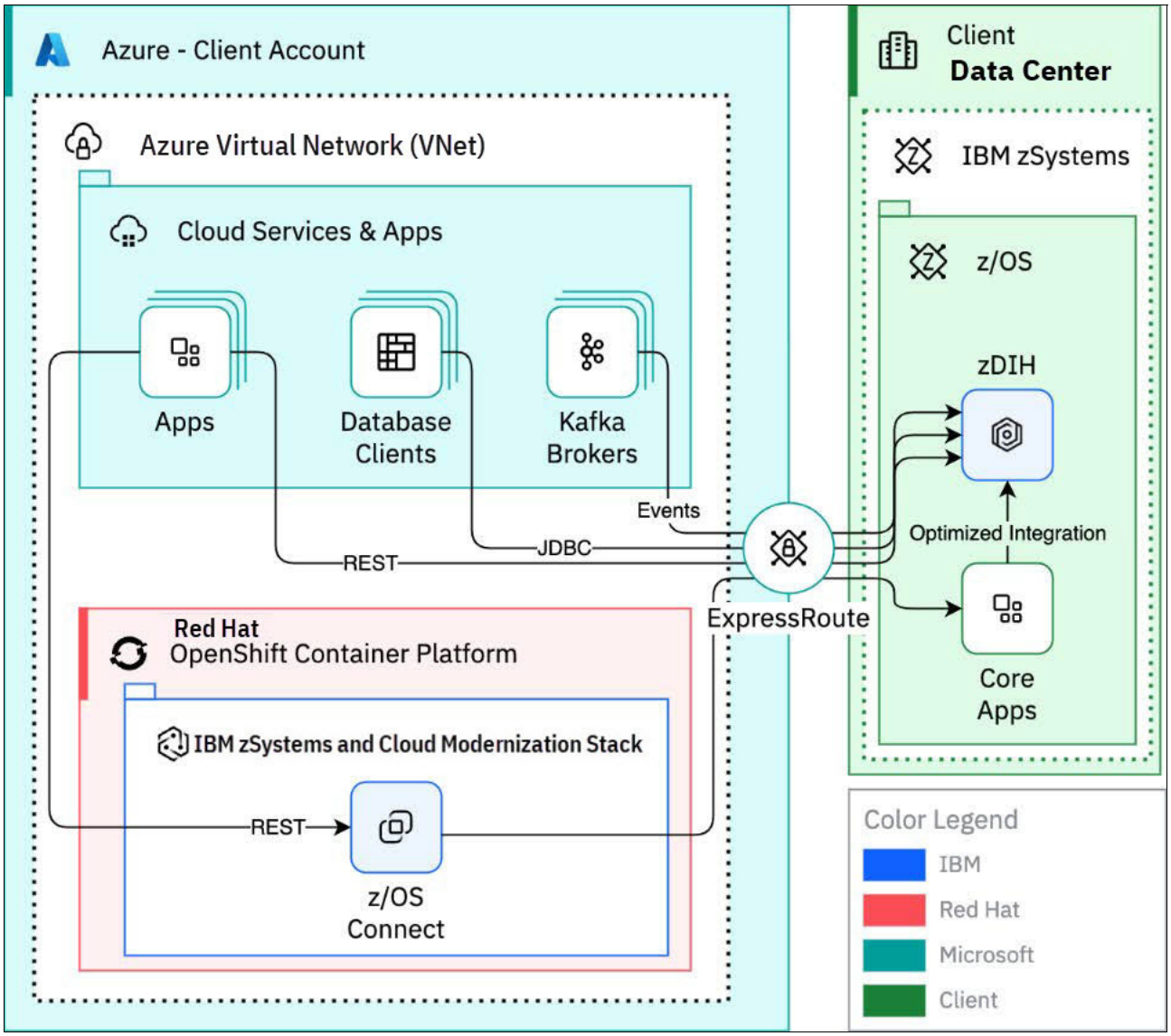


Figure 29 Reference architecture: Access through APIs

Leveraging DevOps for z/OS applications on the cloud

The IBM solutions to create a developer environment for code, build, deploy, and test can be integrated into the Microsoft Azure DevOps services that are set up in a VNet. The IBM Z application source can be stored in Microsoft Azure repositories that use the same SCM as the rest of the enterprise.

Wazi Dev Spaces and Wazi Analyze can work with Microsoft Azure repositories to enable analysis, editing, and debugging of code. Microsoft Azure pipelines can be run against a Wazi sandbox environment that is provisioned on Red Hat OpenShift in the VNet by running builds and storing build artifacts in Microsoft Azure Artifacts, and deploying changes to both a z/OS sandbox and the IBM Z LPARs, as shown in Figure 30.

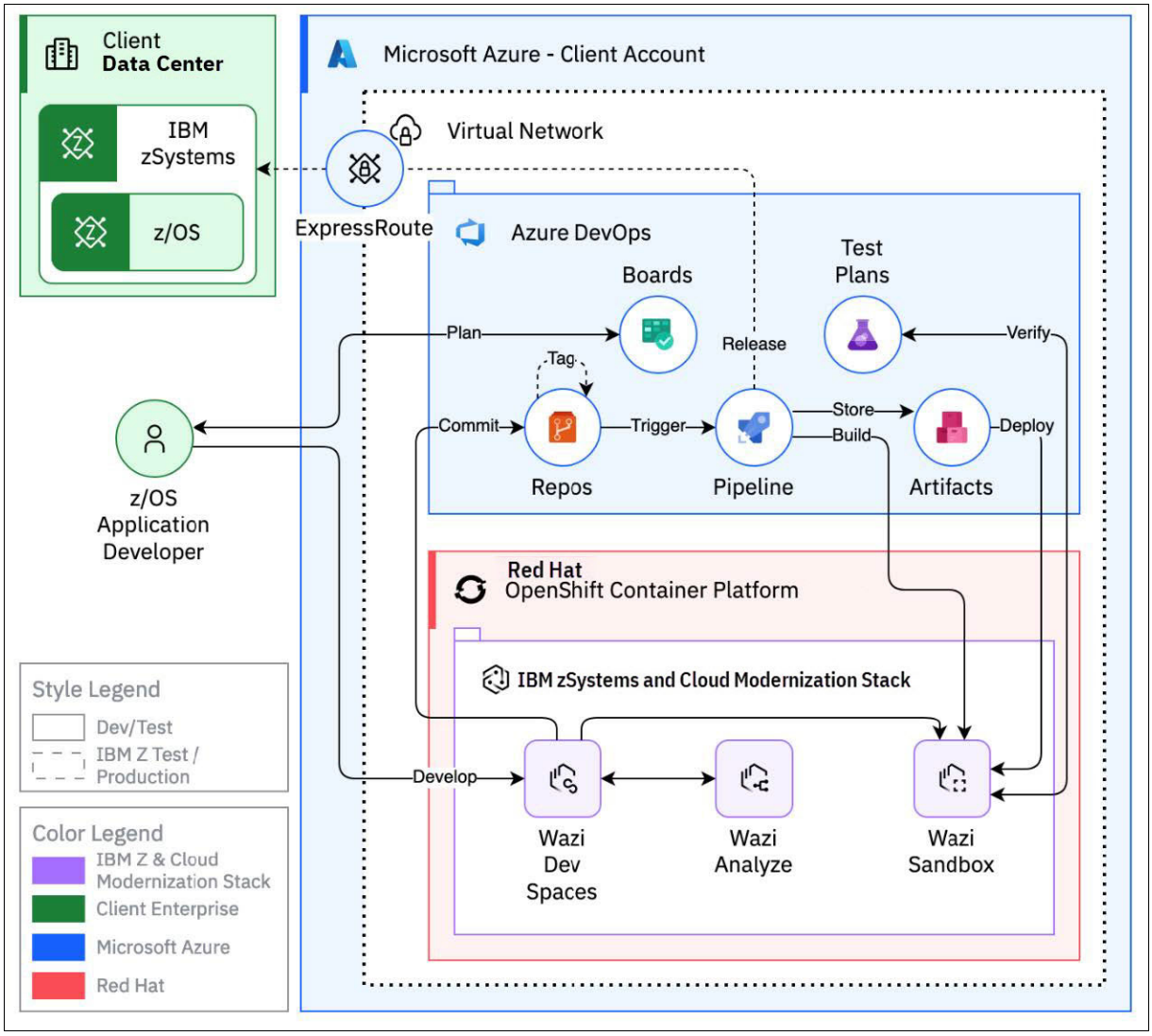


Figure 30 Microsoft Azure with Wazi Dev Spaces and Wazi Analyze

Automating IT across the IBM Z platform and cloud

Figure 31 shows how Red Hat Ansible Automation Platform integrates with Microsoft Azure. Red Hat Ansible Automation Platform can be deployed in a VNet in Microsoft Azure or as a Kubernetes operator on Red Hat OpenShift on Microsoft Azure. Red Hat Ansible Automation Platform can orchestrate automation scripts across Microsoft Azure Cloud and the IBM Z platform through Microsoft Azure ExpressRoute. The automation scripts can be maintained as IaC by using Microsoft Azure DevOps services and the development environment capabilities in IBM Z and Cloud Modernization Stack.

Figure 31 shows an example of this architecture.

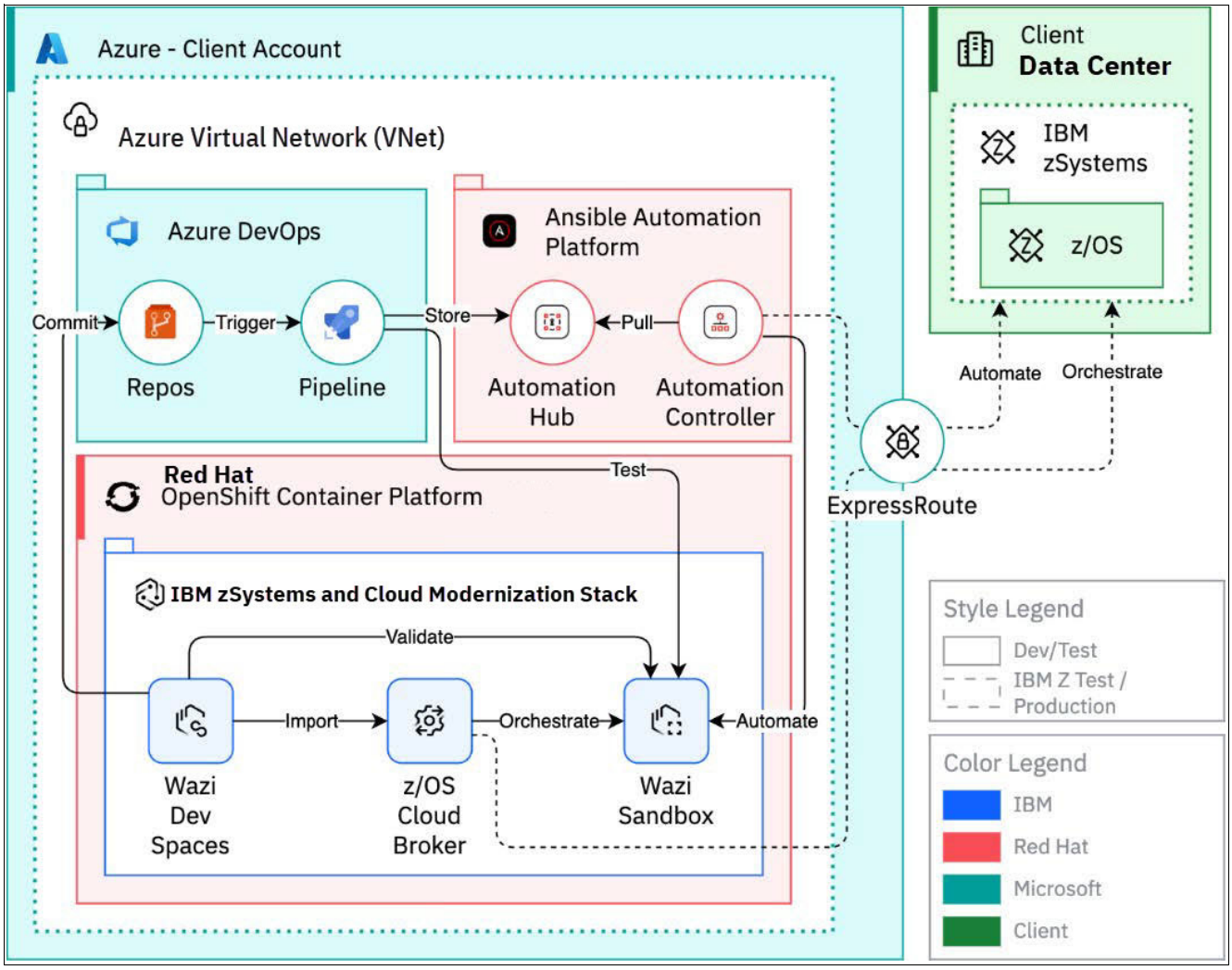


Figure 31 Microsoft Azure DevOps

Getting started with IBM

Innovate and co-create with IBM experts to solve your most complex business challenges:

- Accelerate your application modernization journey.
- Achieve meaningful and scalable business outcomes.

Co-creating with IBM

IBM Consulting and IBM Client Engineering deliver meaningful and scalable business outcomes across all industries. With our deeply skilled multi-disciplinary squad and human-centered approach, we provide value-based experiences and solutions catered to your organization's needs.

Whether a custom demonstration in your environment or an MVP to prove value, we meet you where you are and work with your organization at any stage of its application modernization journey. IBM Client Engineering is an investment in you to co-create and innovate by leveraging IBM technology and methodologies.

What you can expect

- **Solve complex business use cases:** Develop user-centric solutions to your most pressing challenges with measurable business outcomes.
- **Prove value in weeks**: Together, we define the proposed solution and deliver value in days to weeks.
- **Innovate to scale:** We co-create iteratively to grow alongside your organization, delivering enterprise scalability that is securely deployed on your platforms of choice.
- **Partner with global experts:** Leverage deep industry expertise with our agile teams of business technology leaders, technology engineers, solution architects, designers, and data scientists.
- **Modernize with speed:** Build rapidly with IBM hybrid cloud and AI technologies to stay ahead of an ever-changing market.
- **Access proven patterns:** Advance digital transformation with IBM proven technology, accelerators, and methodologies.

Conclusion

To accelerate your digital transformation, increase productivity and business agility, and close the skills gaps in a mainframe and hybrid cloud environment, you must build the right foundation. To that end, IBM recommends the following best practices:

- Embracing a *hybrid cloud* approach to mainframe application modernization
- Leveraging a continuous application modernization journey
- Taking advantage of the recommended *cloud reference architectures* to leverage IBM Cloud, Amazon Web Services (AWS), and Microsoft Azure
- Getting started fast with the IBM proven *co-creation methodology*

Review the different entry points and select the ones that will drive the most immediate value for your business depending on your needs, environment, challenges, and current business processes.

Partnering with IBM and following our recommended application modernization journey can help to you modernize mainframe applications faster at lower cost and less risk. The approaches and strategies that are presented in this publication, including best practices with supporting technologies, reference architectures, and accelerators, can help you to accelerate your digital transformation and increase productivity and business agility.

Authors

This publication was produced by a team of specialists from around the world working with IBM Redbooks, Poughkeepsie Center.

Skyla Loomis
Vice President, IBM® Z Software

Kyle Charlet
IBM Fellow, CTO, IBM® Z Software

Suman Gopinath
IBM, STSM & Chief Architect, DevOps for IBM Z Hybrid Cloud

Peter McCaffrey
IBM, Principle Product Management GTM Lead, IBM Z Software

Tim Brooks
IBM, IBM Z AIOps Sr. Product Manager

Juergen Holtz
IBM, STSM, IBM Z Management Software

Bryant Panyarachun
IBM, STSM - Ansible for z/OS Ecosystem Architect

Purvi Patel
IBM, AI on IBM Z

Mythili Venkatakrishnan
IBM US, Distinguished Engineer, IBM Z Financial Services Sector CTO

Yichong Yu
IBM, STSM, IBM Cloud for Financial Services

Thanks to the following people for their contributions to this project:

Lydia Parziale, Certified IT Specialist, PMI Certified Project Manager, IBM Redbooks® Project Leader
IBM Redbooks, Poughkeepsie Center

Will Burroughs, Visual Designer
IBM Hursley

Michelle Carey, Content Editor and Instructional Designer, IMS
IBM Silicon Valley Lab

Wade Wallace, Senior Content Specialist and Editing Team Lead, IBM Redbooks
IBM Austin

William Alexander, STSM & Lead Architect of the IBM Z Dev & Pipeline
IBM Austin

Carlos Alvarado, STSM - IMS Development
IBM US

Reid Copeland, STSM - Compilers and Performance
IBM US

Ivan Dovgan, Chief Architect - Cloud Enablement for z/OS Applications
IBM US

Surya Duggirala, IBM Cloud Platform Engineering Guild Leader
IBM US

Nicolas Dangeville, STSM and Chief Architect ADDI
IBM US

Jin Leem, UX & Visual Designer
IBM US

Skyla Loomis, Vice President, IBM Z Software
IBM US

Christian Miemiec, Manager IBM IT Economics, Americas and WW
IBM US

Anthony Papageorgiou, Technical Strategy Lead for Integration on IBM Z | Master Inventor
IBM US

Thanks to the following people for their assistance in reviewing this project:

DeWayne Hughes
Principal, Z Client Architect, Financial Services Market

Kyle Charlet
IBM Fellow, CTO, IBM Z Software

Joanie Gines
VP WW, Z Application & Platform Software Sales Global Sales

Notices

This information was developed for products and services offered in the US. This material might be available from IBM in other languages. However, you may be required to own a copy of the product or product version in that language in order to access it.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:
IBM Director of Licensing, IBM Corporation, North Castle Drive, MD-NC119, Armonk, NY 10504-1785, US

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some jurisdictions do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

IBM may use or distribute any of the information you provide in any way it believes appropriate without incurring any obligation to you.

The performance data and client examples cited are presented for illustrative purposes only. Actual performance results may vary depending on specific configurations and operating conditions.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

Statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to actual people or business enterprises is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corporation, registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the web at “Copyright and trademark information” at https://www.ibm.com/legal/copytrade.shtml

The following terms are trademarks or registered trademarks of International Business Machines Corporation, and might also be trademarks or registered trademarks in other countries.

CICS®	IBM Research®	Redbooks®
Db2®	IBM Security®	Redbooks (logo)  ®
GDPS®	IBM Watson®	zSystems®
IBM®	IBM Z®	z/OS®
IBM API Connect®	IBM Z®	z/VM®
IBM Cloud®	IBM z14®	z15®
IBM Cloud for Financial Services®	IBM z16™	z16™
IBM Cloud Pak®	Instana®	
IBM Consulting™	OMEGAMON®	

The following terms are trademarks of other companies:

The registered trademark Linux® is used pursuant to a sublicense from the Linux Foundation, the exclusive licensee of Linus Torvalds, owner of the mark on a worldwide basis.

Zowe, are trademarks of the Linux Foundation.

Microsoft, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Java, and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

Red Hat, Ansible, OpenShift, are trademarks or registered trademarks of Red Hat, Inc. or its subsidiaries in the United States and other countries.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Other company, product, or service names may be trademarks or service marks of others.

REDP-5705-00

ISBN 0738461091

Printed in U.S.A.