Redbooks
ibm.com/redbooks

Getting Started
Journey to Modernization with IBM Z

Ravinder Akula
Matthew Cousens
Makenzie Manna
Pabitra Mukhopadhyay
Anand Shukla

Cloud

IBM Z

IBM

Redpaper

IBM Redbooks

Getting Started: Journey to Modernization with IBM Z

March 2021

REDP-5627-00

Note: Before using this information and the product it supports, read the information in “Notices” on page 9.

First Edition (March 2021)

This edition applies to IBM z15 and IBM LinuxONE III.

This document was created or updated on March 11, 2021.

Contents

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 http://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®
Concert®
Db2®
DB2®
FICON®
GDPS®
IBM®
IBM Cloud®
IBM Cloud Pak®
IBM Garage™
IBM Services®
IBM Watson®
IBM Z®
MQSeries®
Parallel Sysplex®
Rational®
Redbooks®
Redbooks (logo) ®
UrbanCode®
WebSphere®
z/Architecture®
z/OS®
z/VM®
z/VSE®
z15™

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.

Open Mainframe Project, Zowe, are trademarks of the Linux Foundation.

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.

Ansible, OpenShift, Red Hat, 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.

Preface

Modernization of enterprise IT applications and infrastructure is key to the survival of organizations. It is no longer a matter of choice. The cost of missing out on business opportunities in an intensely competitive market can be enormous.

To aid in their success, organizations are facing increased encouragement to embrace change. They are pushed to think of new and innovative ways to counter, or offer, a response to threats that are posed by competitors who are equally as aggressive in adopting newer methods and technologies.

The term *modernization* often varies in meaning based on perspective. This IBM® Redbooks® publication focuses on the technological advancements that unlock computing environments that are hosted on IBM Z® to enable secure processing at the core of hybrid. This publication is intended for IT executives, IT managers, IT architects, System Programmers, and Application Developer professionals.

Authors

This paper was produced by a team of specialists from around the world.

Ravinder Akula is an Infrastructure Architect in IBM India. He is an IBM and Open Group certified IT Specialist at Expert level. He has over 14 years of experience in the field of IBM Z. He holds a B.Tech degree in Information Technology from Kakatiya Institute of Technology and Science, Telangana, India. His areas of expertise include IBM z/OS®, IBM Z Hardware, IBM Z modernization, zVM, Linux on Z, Disaster Recovery, Parallel Sysplex®, and IBM GDPS®. He has been working with IBM® since 2006. He is the technical lead for the z/OS platform support team, which handles multiple IBM Z client infrastructures across different geographies. He is an active participant in IBM Academy of Technology and has published several Tech docs. He is an IBM recognized speaker and presenter and teacher and educator who has been teaching IBM Z classes for IBM Z System Programmers since 2007.

Matthew Cousens is Lead Developer Advocate for IBM Z and IBM LinuxONE, North America. After receiving a Bachelor of Science in Computer Science, he spent 16 years testing on the IBM Z platform. Matt has worked in every major phase of the software testing lifecycle and has experienced first-hand the transition from waterfall to agile techniques. In his current role as a Developer Advocate, Matt is responsible for speaking about the business value of the mainframe and the technology that enables that value.

Makenzie Manna is an IBM Redbooks Project Leader in the United States. She has four years of experience in the Computer Science Software Development field and one year of experience in developing technical content. She holds a Master of Science degree in Computer Science Software Development from Marist College. Her areas of expertise include mathematics, IBM Z, and cloud computing, and has written about topics regarding modernization with IBM Z and COBOL programming in a modern IDE.

Pabitra Mukhopadhyay is an IBM and Open Group certified Infrastructure Architect in IBM Services®, India. He has over 14 years of experience in the field of IBM Z®. He is also an IBM Certified Specialist for Z Systems Technical Support. He holds a B.Tech degree in Electronics and Communication Engineering from West Bengal University of Technology, WB, India. He has worked at IBM for 8 years. His areas of expertise include IBM z/OS, IBM Z hardware, IBM Z modernization, Parallel Sysplex®, middleware, Disaster Recovery, resiliency, availability, monitoring, and automation. He has participated in industry-level events and written extensively about z/OS and middleware products in IBM Systems magazine and IBM Developer.

Anand Shukla is an Infrastructure Architect in IBM India. He has over 14 years of experience in IBM Z. He holds a M.Tech degree in Information Technology from Latrobe University, Melbourne, Australia. In his 3 years with IBM, he has worked as a Technical lead for the z/OS platform, handling multiple IBM Z client infrastructures for multiple geographies. Being an IBM recognized Educator and Presenter, he has conducted various skills enhancement training for IBM India team and for IBM Australia, Brazil, Thailand, and Singapore regions. His expertise includes, IBM z/OS, zVM, Linux on Z, IBM Z Hardware, IBM Z modernization, Disaster Recovery, Parallel Sysplex, GDPS, and IBM Z DC Migration. Prior to IBM, for over 11 years he worked as a z/OS Technical Specialist with CSC (now DXC) Australia.

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

Mark Bylok
Jordan Cartwright
Jim Crowley
Daniel Jast
Alex Lieberman
Luisa Martinez
Ryan Parece
Torin Reilly
James Roca
Xiaoyu (Allen) Zhou
Stephen Warren
IBM

Now you can become a published author, too!

Here's an opportunity to spotlight your skills, grow your career, and become a published author—all at the same time! Join an IBM Redbooks® residency project and help write a book in your area of expertise, while honing your experience using leading-edge technologies. Your efforts will help to increase product acceptance and customer satisfaction, as you expand your network of technical contacts and relationships. Residencies run from two to six weeks in length, and you can participate either in person or as a remote resident working from your home base.

Find out more about the residency program, browse the residency index, and apply online at:

ibm.com/redbooks/residencies.html

Comments welcome

Your comments are important to us!

We want our papers to be as helpful as possible. Send us your comments about this paper or other IBM Redbooks publications in one of the following ways:

- Use the online **Contact us** review Redbooks form found at:

  ibm.com/redbooks

- Send your comments in an email to:

  redbooks@us.ibm.com

- Mail your comments to:

  IBM Corporation, IBM Redbooks
  Dept. HYTD Mail Station P099
  2455 South Road
  Poughkeepsie, NY 12601-5400

Stay connected to IBM Redbooks

- Find us on LinkedIn:

  http://www.linkedin.com/groups?home=&gid=2130806

- Explore new Redbooks publications, residencies, and workshops with the IBM Redbooks weekly newsletter:

  https://www.redbooks.ibm.com/Redbooks.nsf/subscribe?OpenForm

- Stay current on recent Redbooks publications with RSS Feeds:

  http://www.redbooks.ibm.com/rss.html

1

Introduction to modernization

At the heart of many large enterprise hybrid cloud environments sits IBM Z, which churns out thousands of secure transactions every second. This computing platform is uniquely built to enable massive vertical scaling, which is perfect for today's most demanding peaks. Combined with built-for-purpose hardware with a keen focus on data privacy and security, IBM Z is often said to host not just enterprise data, but the world's data. If you swipe a credit card or book a trip on an airline, chances are you use an IBM Z system.

Cloud computing shattered the notion of closed systems operating within the confines of a data center. Today's enterprise computing is hosted on-premises in traditional data centers, and as-a-service on private and public cloud platforms. In fact, it is not unusual for an enterprise to assemble their offerings that use multiple cloud platforms, thus the birth of hybrid multi-cloud. As you might imagine, this shift in computing paradigms is enabled by significant improvements in technology.

In this chapter, we discuss why IBM Z is the proper environment for your modernization journey, what is meant by modernization, why there is a need for modernization, and what that journey can look like within an enterprise.

This chapter includes the following topics:

1.1 What is IBM Z?

IBM Z is a family of computers that are designed for enterprise computing with focus around three core principles:

- Reliability: The entire IBM Z platform (including the hardware, firmware, and operating systems) is designed with self-checking and self-recovery capabilities to ensure smooth operation.
- Availability: The “Z” in IBM Z represents zero downtime. The latest systems are designed for 99.99999% uptime.[1]
- Security and data privacy: Various powerful technologies are featured in the platform that is designed to help keep data on IBM Z safe.

The latest addition to the IBM Z family is the z15™, which first became available in December 2019. This system is designed to host one trillion secure web transactions per day. To accomplish this rate, the z15 boasts up to 40 TB of RAM, which is fully redundant and known as redundant array of independent memory (RAIM), which accommodates hardware failures in the memory modules. The sheer amount of RAM is put to use by the amount of data that is hosted in databases and the volume of work that occurs simultaneously.

The compute power of z15 comes from over a hundred cores, which are configured by system administrators as needed for their specific use case. Feeding the compute cores are multiple levels of cache and specialty processors. Throughput is achieved by dedicating hardware to common operations, which frees up the main processors for other work. For example, I/O and cryptographic operations each feature dedicated processors in the z15. The systems in the IBM Z family run various operating systems, including multiple Linux distributions, z/OS, z/TPF, IBM z/VM®, and others. In fact, it is common for multiple operating systems, and even multiple logical instances of the same operating system, to run on a single system.

Table 1-1 lists some of the operating systems that run on IBM Z.

Table 1-1 Selection of Operating Systems supported on IBM Z

Operating system	Description
z/OS	Designed to keep applications and data available, system resources secure, and server utilization high. z/OS maintains compatibility for applications and runs Linux on IBM Z containers on-premises and in hybrid multi-cloud.
Linux on Z	Provides an enterprise Linux platform that benefits from IBM Z qualities and by proximity to other systems hosted on Z, such as z/OS.
z/Virtual Machine (z/VM)	A hypervisor that can run thousands of Linux on Z virtual machines on one system and run z/OS, IBM z/VSE®, and z/TPF.
z/Transaction Processing Facility (z/TPF)	A special-purpose operating system that is used to process high transaction volumes, such as credit card transactions and airline reservations.

IBM Z systems are uniquely engineered to support large volumes of simultaneous transactions at higher throughput than other types of computers. It delivers the highest levels of data privacy and resiliency.

[1] https://www.ibm.com/downloads/cas/67Q9DDOR

Systems are even combined in such a way that work is automatically transferred to alternative sites if something such as a natural disaster takes out a data center.

For more information about IBM Z, see this web page.

1.2 Why IBM Z?

IBM Z is a trusted platform for government, large companies, and even startups. They rely on these systems to run the applications that are critical to their businesses and to maintain system-of-record data. These transactional and batch applications evolved organically, reflecting the accumulated business changes and requirements of decades.

The amount of business intelligence that is coded in these core applications is vast, and the value that is attached to these time-tested assets is huge. The value stems from more than just the applications; it is also based on the business data that is on or originates on IBM Z.

IBM Z earns this trust through multiple avenues. For example, IBM Z hardware, firmware, and operating systems always conform to a guiding set of architecture rules to ensure support of current and future workloads and services. Whenever new capabilities and features are implemented, the IBM z/Architecture® is extended rather than replaced, which enables compatibility with an earlier version. Therefore, applications that are running on IBM Z continue to function as capabilities are added. This feature is one way the platform helps organizations protect their investment; that is, by ensuring applications continue working in the future without requiring any application changes.

Many long-time users depend on the reliability and availability that IBM Z provides. By using the technologies in IBM Z, these clients can achieve years or even decades of uninterrupted service. Among the backdrop of laptops, smartphones, and even watches that must be restarted every so often, years without interruption is unthinkable. This inherent robustness enables trust.

1.2.1 Shift in computing paradigm

The IT industry is in the midst of a fundamental shift in the way people think and design new enterprise applications; that is, a shift in the way application developers code, test, deploy, and maintain applications. This change is a direct result of a shift in the way users interact with these applications and the way services are offered. For example, consider this shift through the lens of Company A.

At one time, enterprise computing was performed by using closed systems that were housed in data centers that were owned by the organization that was performing the computing. This meant that Company A owned the end-to-end processing, the data that was created by that processing, and they had physical custody of their data. Company A protected themselves by securing physical access to their data centers and implementing authentication and authorization best practices.

The era of cloud computing was born with the proliferation of inexpensive commodity hardware and operating systems, and sufficiently fast internet connections. Company A suddenly had many more options when it came to implementing their enterprise computing environments. They embraced multiple public clouds as they built out their client-facing sites and even some parts of their core applications. This move had some unintended consequences for Company A's Z-powerhouse, at the core of their business.

Aside from the technical impact that comes from driving the same systems in different ways, business impacts are felt as strongly. Faced with where to focus their resources, Company A directed their developers, architects, and budgets to the cloud. With so much focus on building out their cloud presence, other aspects of their computing landscape were not top of mind.

After a few years of focused efforts, Company A built a strong infrastructure across multiple clouds. Because they built it from scratch, they used all of the latest techniques for application design, build, deployment, and orchestration. These techniques were perfect for enabling quick application changes, and Company A found it rolled out updates to its clients more rapidly.

This brings us to the problem that Company A faces today. Although Company A's cloud systems are nimble, Company A never invested their resources to enable similar benefits with their environments that are hosted on IBM Z. Therefore, when a change is made that necessitates updates to their cloud systems and their Z systems, one side is often waiting for the other because of the difference in how they are set up.

Does this mean that Company A's Z system is old and outdated? Not at all. Company A's cloud systems use newer techniques and technologies because they were created more recently. In fact, the cloud systems use many of the same techniques and technologies that a company new to IBM Z get started with if they create an environment today.

Because Company A has had environments on Z for many years and did not modernize that environment with the addition of the cloud systems, they use older technology. All that they need to do is invest in updating their environment to newer techniques, tools, and technologies. This updating effort is what we mean by modernization.

1.3 What is modernization?

Modernization in the context of IT can mean different things to different people. For example, the Table 1-2 lists different roles and their respective potential understanding of modernization.

Table 1-2 Different meanings of modernization

Professional	Meaning of modernization
Customer Experience Officer (CXO)	Open up new experiences for clients in a way that fuels business growth, expands market share, adds value to their clients, and increases customer loyalty.
Chief Technology Officer (CTO)	Respond to business requirements faster by delivering new features and incremental updates to existing features more quickly.
Chief Financial Officer (CFO)	Maximize the return on investment (ROI) in the use of IBM Z and the surrounding system by increasing value without increasing expense.
Enterprise Architect	Integrate various technologies, platforms, and applications across the enterprise in a seamless and transparent manner.
Infrastructure Architect	Use new technologies and features to ensure the highest levels of reliability, availability, and simplification.
Application Architect	Make it easier to use products and services on the platforms of choice and simplify the applications' designs.

Professional	Meaning of modernization
Security Architect	Ensure the highest levels of security and data privacy by using new ways of protecting enterprise data and monitoring access.
Application Developer	Accelerate and grow quality in every step of the software delivery lifecycle by using modern tools and processes.
Business Analyst	Unleash the power of business data through increased insights that can be realized only by using the latest technologies with the most current data.

Every individual has their own perspective on this topic, one thing they have in common is a focus on the use of innovation to deliver better business value for their organization. No one wants to change technology for the sake of being called modern. It is the value that stems from modernization that makes the journey worthwhile.

Figure 1-1 shows how transformation is occurring in different areas of the IT system and what these areas are transitioning toward, including cultural changes.

Note: The terms *modernization* and *digital transformation* are used interchangeably throughout this publication. More specifically, *modernization* refers to the process or the practice of upgrading the IT landscape of an organization to address their market needs. Whereas *digital transformation* refers to the adoption and use of new features and digital technology to deliver value. Digital transformation also involves a fundamental change in the way people and processes work to enhance efficiency and increase the organization's competitiveness in the market.

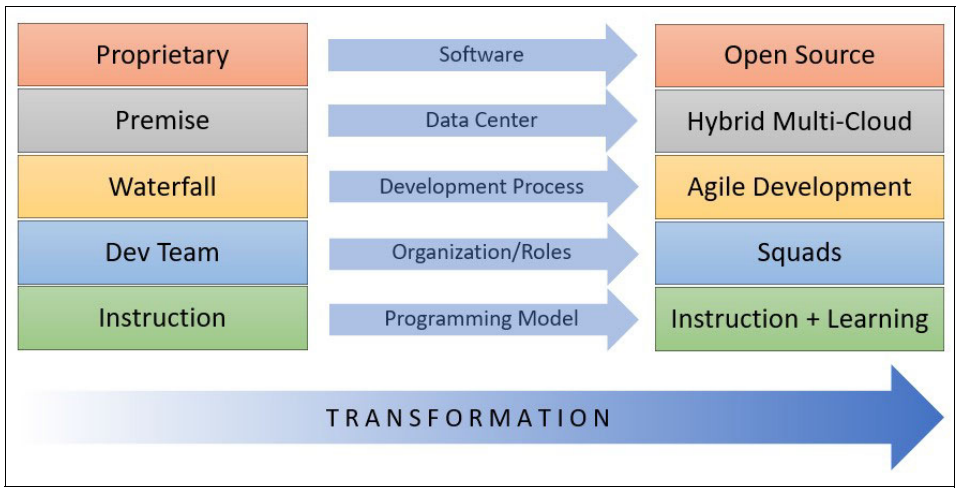

Figure 1-1 Transformation of IT system

In the era of disruptive technology, organizations continue to face unparalleled challenges. Companies need their applications, and the hosting platforms these applications run on, to be increasingly reliable, flexible, robust, and secure.

Modernization can be broadly classified into the following categories:

- Infrastructure or platform modernization

  Upgrade and use features that your IT infrastructure, such as the IBM Z platform, offers that you are not currently using.

- Application modernization

  Unlock the value of the business logic that is coded in core programs by externalizing them using industry standard APIs.

- Process modernization

  Ensure that your business can respond to the needs of the clients with speed and accuracy, focusing on the process of application development and the efficiency of the process.

1.4 What drives modernization?

Technology is shifting at a profound pace and significant risks exist to continue to rely on traditional methods of managing your IT environment and business applications. In our example, Company A faces some of these risks, including the following examples:

- Slower business growth
- Loss of customer loyalty
- Loss of business opportunities
- Inability to stay relevant in the market amidst global competition

In this section, we describe some of the key drivers of modernization today, including the following examples:

- “Business agility and speed-to-market”
- “Integration and interoperability of enterprise systems”
- “Security, trust, compliance, and regulatory requirements”
- “Resource optimization ”
- “Managing enormous amount of data”

1.4.1 Business agility and speed-to-market

Traditional methods of software development, such as the waterfall methodology, are based on knowing a project’s full design before getting started, which is extremely challenging to accomplish. The processes are inflexible; therefore, by the time the product is developed, making any changes or course correction is too expensive. Business leaders recognize this issue as a setback and pushed modernization to be a top priority because it directly relates to business outcomes. This driver is one of the main drivers for modernizing your software development methodologies and philosophy.

Implementation of process modernization approaches, such agile, DevOps, automation, and the continuous integration and deployment (CI/CD) pipeline helped organizations significantly reduce the time-to-market of products and services. For our example, one of Company A’s biggest competitors reduced application builds from weeks to hours by using DevOps with IBM Z.

DevOps helps streamline infrastructure and accelerate the delivery of high-quality software, simultaneously lowering overall cost through optimized development, testing, and deployment. The IBM Z platform supports common tools for product development and automation, so these approaches can be performed across platforms.

Application developers can concentrate on coding rather than worrying about where the application is going to be deployed. As a result, minimum viable products can now be released at a faster rate, which leads to more timely feedback from users. For more information about DevOps, see Chapter 4, “Introduction to DevOps” on page 39.

1.4.2 Integration and interoperability of enterprise systems

Most organizations do not run in a homogeneous environment. They use a diverse set of platforms and technologies, within different divisions, based on many factors, including criticality of the systems, data, and business needs. Applications were traditionally developed in silos, which resulted in duplication of functions. Maintaining these redundant systems and application code comes at a cost.

To gain more efficiency and increase reuse, business processes must be streamlined across the organization. In addition, diverse systems and applications must be interconnected and integrated so that they can work in tandem to achieve business goals. \

As we move toward the standardization of protocols, modern platforms, such as IBM Z, enable integration of diverse systems, which makes applications easier to develop, maintain, and modify quickly without disrupting the business. For more information about integration and interoperability of enterprise systems, see Chapter 3, “Modernization technologies on IBM Z” on page 25.

1.4.3 Security, trust, compliance, and regulatory requirements

Although our highly connected digital world made our lives easier than anyone imagined a decade ago, it also gave rise to new, unforeseen challenges.

The threat of cyberattacks is constant and without physical or political boundaries. Government and regulatory requirements around data privacy and protection laws are increasingly strict, and penalties are becoming more substantial to ensure compliance. By not modernizing your traditional IT systems and application security setups, you become more vulnerable to such threats.

The IBM Z platform offers highly secure ways to help deal with compliance and regulatory security requirements. For example, multi-factor authentication, pervasive encryption, centralized key management, and in-flight data protection allow you to implement security at the speed of business.

1.4.4 Resource optimization

New technology and methods are implemented to increase agility, simplicity, and maintainability. They also allow organizations to take advantage of common skills and tools across platforms. Modernization also permits organizations to make their critical and strategic applications and data more easily consumable, both inside and outside the enterprise. This, in turn, enables the feasibility of organizations that run similar setups across business units to achieve significant cost reduction by eliminating duplication of resources.

Organizations are also adopting open source software and standards to achieve higher levels of cost-effectiveness, speed, and agility. Total cost of ownership (TCO) is one of the most important factors for finalizing the platform for digital transformation. According to a study conducted by IDC, the transformational capabilities of IBM Z paired with today’s tools are helping clients realize signification business value. It is estimated that:

- The benefits might outweigh the investment cost for this transformation by an estimated factor of 6.2x.[2]
  - Cost and operational efficiencies: Estimated 2.5x benefits to costs
  - Efficiencies and higher revenue: Estimated 3.8x benefits to costs
  - Efficiencies, higher revenue, and protected revenue: Estimated 6.2x benefits to costs

- IBM Z clients can reduce the cost of operating their platforms by an average of 19% over the course of five years, which is worth almost $3 million per IBM Z system.[2]
- Organizations reduced already low levels of unplanned outages by an average of 43%, which not only reduced time that is lost to unplanned outages, but also minimized business and reputational risks that are associated with outages.[2]

It is also estimated that because of this transformation, organizations now have 64% more code releases and 44% less time per code release.[2]

1.4.5 Managing enormous amount of data

A huge amount of data is being produced and used on enterprise systems. This data is of great value and is expected only to increase in size and scale. Organizations must be prepared to efficiently and securely store and handle these massive sets of data. However, raw data has little value; therefore, it is critical that organizations also be equipped to process this data of varying types and forms for it to extract meaningful insights. Modern systems and tools were developed to complete these tasks more effectively. Hence, modernization is considered inescapable to meet these requirements.

1.5 How does modernization relate to IBM Z?

In our example, Company A has shown us how it is important for modernization to extend across platforms because even a single business transaction likely transcends multiple key platforms. A few notable methods are available that modernization can be implemented with IBM Z, as described next.

1.5.1 Containerization and hybrid multi-cloud

Enterprise workloads became dependent on more than one cloud provider. One technology that fuels application portability is the use of containers. By using containers, developers can build, deploy, manage, and orchestrate their applications across platforms or systems. This use is ideal for typical development scenarios where a developer might run on a public cloud for their sandbox, deploy to a z/OS test system that is running on Z hardware for performance and stress tests, and finally end up on another z/OS system for production.

Through an integrated and automated pipeline, the developer does not need to worry about system intricacies. Recent additions in operating system and middleware support expanded container support on IBM Z and more is planned for the future.

1.5.2 Securing data on IBM Z

Protecting data is a long-time focus of IBM Z. Traditionally, limitations to the amount of data that can be encrypted existed because of the computational overhead of cryptographic operations. Selective data encryption, the traditional method of encrypting data, requires teams to identify the data that they must encrypt. Enabled by advances in new Z hardware, a shift from selective encryption to encrypting all data, everywhere, removed this requirement.

2 IDC white paper, sponsored by IBM and Broadcom, The Business Value of the Transformative Mainframe, August 2019.

With pervasive encryption on IBM z15™, you can run up to 19 billion fully encrypted transactions daily without affecting service level agreements (SLA) and no application changes[3]. Pervasive encryption enables encryption of data in-flight and data at-rest. Within a trusted network shared by Z, all your data is encrypted when in-storage and in-flight through the network.

Data privacy is perhaps the most critically important issue facing digital business today. As regulations and compliance requirements increase, it is vital that you ensure that your client data remains encrypted, even as it moves off the system of record within your enterprise. IBM Z has several technologies available to permit data privacy at-rest, in-flight, and even in-use.

1.5.3 Higher availability, resiliency, and performance demands

Many organizations realized that traditional business resiliency metrics are no longer meeting their business requirements. The expectation is 24x7x365 availability and organizations are gradually moving toward near-continuous availability (near-CA) requirements. However, even that is not enough because having poor performance for several seconds can be considered an outage. This issue is a lot to ask in an environment where it is no longer safe to assume that peak online workloads can be expected only during business hours. Benefits in tuning systems and applications to cater to varying performance requirements and to help ensure they are always available can be achieved with modernization.

1.5.4 Skill development, lower learning curve, and productivity benefits

New features on IBM Z and the modernization of applications running on Z can empower and promote organizations to offer new services. It also helps normalize skills by bridging the gap between new and experienced professionals, enhancing productivity, and ultimately reducing costs.

Modernization also allows application developers on distributed platforms to develop applications for z/OS; for example, by using modern integrated development environments (IDEs), discussed in Chapter 4, “Introduction to DevOps” on page 39, without having to learn environment-specific tools and interfaces. New common tools increase developer effectiveness and efficiency, which permits the movement of highly skilled personnel to roles that add the most value to the organization.

The learning curve of a new professional or an individual with experience on a different platform can be steep. Organizations are increasingly focusing on simplification and modernization of their traditional IT environments to flatten the learning curve for increased productivity. This simplification also paves the way for IT personnel from other platforms and recent graduates to better understand the mainframe system, and the code that runs on these systems, without spending years mastering the skills.

1.5.5 Vendor lock-in

Vendor lock-in is a relevant concern with public cloud. This issue evolved from the traditional concern of vendor lock-in coming from a platform’s software and hardware vendor to a concern about a lack of standard APIs and data structures, interoperability, and portability issues as some of the major inhibitors for moving applications to cloud.

3 https://www.ibm.com/it-infrastructure/z/technologies/pervasive-encryption

Many organizations moved some critical workloads to the public cloud in an effort to modernize their IT enterprise. Such organizations benefit from the public cloud in terms of responsiveness to market changes and productivity. They also ran into a paradox: more of their data falls captive to that single cloud provider. This issue is known as *vendor lock-in*. Vendor lock-in is about more than not being able to migrate to a different vendor. Each cloud vendor is unique in terms of offerings, surrounding everything from the developer tools to the application capabilities.

With each cloud choice an organization makes comes a set of assumptions and features. By over-relying on one public cloud, companies can place themselves on a set path and find it difficult to migrate to another cloud service provider in the future.

Organizations have unique needs around their data and workloads that the public cloud alone might not satisfy. This issue is even more relevant for organizations operating in highly regulated industries that must comply with various global and local regulations that are related to data protection. As a result, hybrid multi-cloud as a solution, complemented by open source software and open standards, is becoming more widely adopted. Figure 1-2 shows the journey from single vendor public cloud to hybrid multi-cloud.

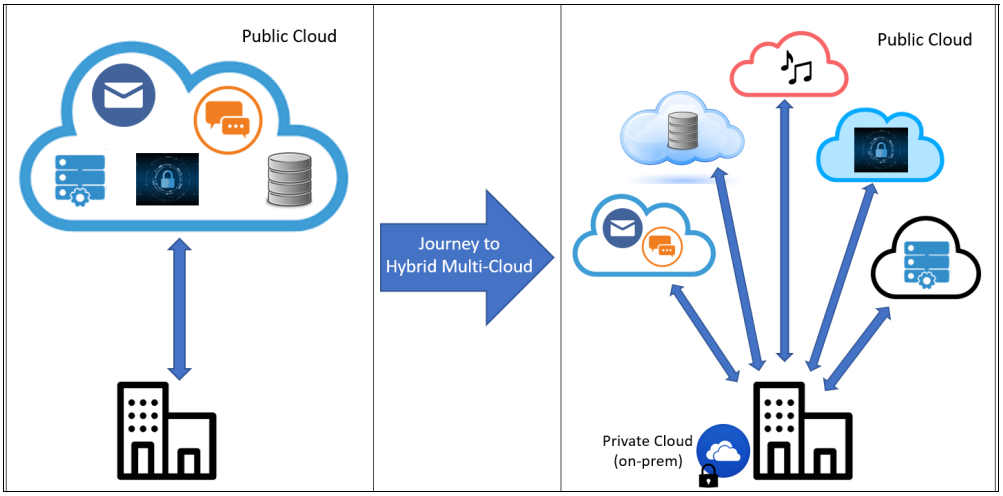

Figure 1-2 Journey to hybrid multi-cloud

1.5.6 Open-source software adoption and integration

From quantum and blockchain to containers, AI, cloud stack, application development, machine learning, and operating systems, IBM is actively leading many open source projects. These projects thrive upon the guiding principles of the open source software movement, such as transparent, open, no-cost, and collaborative participation among community members. Support for open source products on IBM Z demonstrates sustained commitment by IBM to open source innovation, delivering a broad portfolio of offerings that matter to clients.

These products help organizations minimize software cost and simplify license management to a great extent without vendor lock-in restrictions. Because most of the product source code is easily available, developers can customize the code to suit their requirements, fix any bugs, or enhance product features. These reasons are some example of why open source software gained so much traction. Support for open source products and integration of open source systems with existing systems on IBM Z has enormous potential for clients.

1.5.7 Mindset change

The fact that IBM Z ensures that compatibility with an earlier version created an unintended challenge. Programs that were coded years ago continue to run unchanged on the latest IBM Z platform, even after Z hardware was updated. Some organizations became used to leaving their applications alone unless they need to be altered for new features. Though these applications continue to function, they cannot use new hardware features that require recompiles. This issue can result in suboptimal performance of these programs.

In some organizations, DevOps teams work only on systems that are hosted in the cloud. Some might think that IBM Z and DevOps cannot coexist, when in reality things are quite the opposite. Organizations that embraced DevOps practices and open source software for agility on IBM Z take advantage of years of trusted reliability, performance, availability, and security capabilities, and reap the benefits of this digital transformation.

Some time ago, having the words “cloud” and “on-premises systems” in the same sentence was considered implausible, or even impossible. IBM Z evolved significantly and writing programs on Z and other cloud-oriented technologies together is now common.

To summarize, modernization requires a cultural shift and a change in mindset to create value by using new tools and processes. For more information, see Chapter 5, “Modernization best practices” on page 53.

1.6 Modernization journey

Although an organization’s IT systems are meant to act as an enabler, those same IT systems might become inhibiting for many organizations because of a lack of modernization. These systems are often considered to be responsible for hindering growth in the era of digital revolution. It is important to understand that modernization is viewed as a journey toward enablement. This modernization is not something that can be implemented overnight or with a single step. Most of us acknowledge that this IT modernization journey is inevitable. Many questions exist, such as the following examples:

- How do we embark on this journey?
- What is the goal of modernization?
- How do we make the journey sustainable and nondisruptive to the business?
- Should we approach modernization in a phased manner?

One of the foremost requirements for a successful digital transformation is ensuring that it is done at a pace you can control. A phased or iterative approach to the IT modernization journey offers more flexibility and being practical for large, complex projects.

Today’s transformation initiatives are complex, and technology is evolving rapidly, so the probability of running into something unknown is high. For this reason, you must consider a phased approach to modernization. A preferred practice is to limit your plan to only what is visible during the planning stage of a phase or sprint. This limit reduces the risk and gives you better control in terms of managing the risk and steering the project through necessary changes.

An organization’s modernization journey can be broadly categorized into the following phases:

- Phase 1: Learn
- Phase 2: Adopt
- Phase 3: Deploy

Figure 1-3 shows these phases, which are iterated continuously throughout the modernization journey.

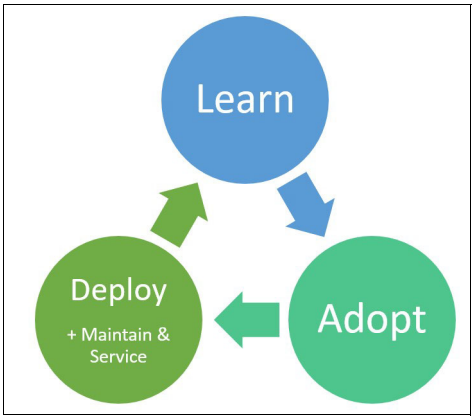

Figure 1-3 Continuous modernization journey

1.6.1 Phase 1: Learn

The first phase of an IT modernization journey is learn. At the beginning of this phase, you must identify a highly motivated, dynamic, and forward-thinking core team, and a leader or leadership team. Look for sponsors for the modernization initiative who can establish the vision and lead the organization's modernization journey.

The core team must include subject matter experts from different areas of technology and job roles, including architects, application developers, integration specialists, process specialists, security experts, and domain experts. The leaders, with the help of this core team, must ensure that the organization embraces change, which is critical to the success of modernization initiatives.

This core team is typically responsible for ensuring that the rest of the organization is not only on-board, but also suitably aligned with the organization's vision and direction for modernization. Consider putting the core team members through training that surrounds the modernization journey on IBM Z to better prepare them for the tasks ahead.

In this phase, the core team must understand the concepts and aspects of modernization on Z. They also must understand how new features of the hardware, operating system, middleware, automation, monitoring, security, and other product suites can help your organization digitally transform the IT landscape.

Understanding different approaches and options for transforming applications from being cloud-ready, to cloud-enabled, to cloud-native in this journey to hybrid multi-cloud environment, is equally important. The core team must also understand how IT and business analytics can help the organization get useful insight that can lead to further growth.

Parallel to understanding the new features and capabilities that are offered by the IBM Z platform, the team must also study their current IT setup in depth to understand what is working for them, what is not, and how much work is required to modernize the IT setup. They are encouraged to identify key areas that require urgent attention and prioritization in the modernization journey. They must also look at the current operational model to see whether it is in line with future needs.

The team must clearly identify and articulate the business value that is associated with IBM Z for their mission critical workloads. Further, they must understand and visualize how in-place modernization adds value and revolutionize their business and drive growth. The team might consider reviewing digital transformation success stories and use cases that closely resemble their organization and current IT setup to get an idea of the benefits they can expect by modernizing their IT network.

After you can understand and quantify all the benefits of modernization and how it can transform your business and maximize value, you can more easily define the goals, justify the spends, and secure funding for modernization projects and initiatives.

1.6.2 Phase 2: Adopt

The second phase in the modernization journey is adopt. It is in this phase where you evaluate the feasibility of the modernization solutions and technology in the context of your organization's IT network. In large enterprises that include multiple divisions that are working parallel in silos, it might be a challenging task. Arriving at a consensus with all of the groups and teams within the organization is easier said than done; therefore, the role of the core and leadership team is crucial in this phase.

The core team can consider formulating a broad framework and provide necessary guidelines for enterprise modernization. The rest of the teams and groups within the organization can work by conforming to this framework and following these guidelines to evaluate the feasibility of the possible options for modernization.

As part of this example, consider devising options for how the different IBM Z technologies, features, enhancements, DevOps solutions, products, and open-source software solutions, (researched in the learn phase) can be tied together. By doing so, you can arrive at a unique and customized modernization solution that is best-fit for your enterprise.

After the feasibility of the possible modernization solutions is evaluated, you can assess the options that are available for implementation and further narrow them down. Consider reviewing the current setup to determine the organization's readiness in terms of implementation of a modernization solution. You can prepare lists of various items and activities that can help you drive the initiative effectively, as shown in the following examples:

- Hardware, operating systems, and middleware that must be upgraded.
- Products and software that must be installed.
- Required security enhancements.
- Components or applications that are available and can be reused.
- Items that require simplification.
- How the monitoring, alerting, and automation solutions must be configured for effective, efficient, predictive, and insightful monitoring of the entire enterprise.
- How to set up or modify DevOps processes and CI/CD pipeline to meet the requirements, such as automated build, and automated and continuous testing.
- How to define the systems and applications to enable seamless integration and migration to a hybrid multi-cloud environment in the future.
- How to encourage an agile way of working across the organization and roll out collaboration tools that facilitate agile best practices, such as integration of agile tool sets and feedback channels with a CI/CD pipeline.
- List of technology and technical best practices to be implemented when deploying the solution, and so on.

Similarly, every team in the organization is often encouraged to have their own list of necessary implementations as part of the modernization initiatives. It is important to understand that it is possible that different teams are at different levels of maturity in terms of modernization and therefore, the entry point for every team might differ. All this information helps you to determine the amount of change that is required when implementing a modernization solution. It also helps you to plan, prioritize the tasks, arrange the initiatives or sprints in proper sequence, and in the process provide direction to the teams that are involved in implementation of the solution.

1.6.3 Phase 3: Deploy

The third and final phase in this journey is deploy. Although this phase is the final phase, it is not the end of this journey by any means. On the contrary, it is a continuous journey and each phase must be repeated in an iterative fashion.

Consider setting up samples or prototypes for complex parts of the solution to get early feedback. After you complete the deployment pre-work and testing, the next step is to deploy the solution. In this phase, you also maintain the solution and provide support for future enhancements. Suggestions, comments, and feedback that are gathered from users, support personnel, and test teams must be fed into the iteration planning system for the inclusion of defect fixes, and solution optimization in the subsequent iterations.

Consider conducting periodic reviews of the iterations to ensure that everything is on track. Some of these considerations can include the following examples:

- Features that were initially planned versus features finally deployed.
- What went wrong and what was done correct?
- Are there any new backlogs?
- Lessons that are learned from the iteration are documented and are used as inputs for the next iteration planning.
- Did the team achieve all the Key Performance Indicators (KPIs) they set?
- Do team members require more training and skills to support the modernization journey? Is a skill road map available for all of the teams?
- Do we need to revisit the modernization goals, targets, and road map?
- Is the modernization journey on track across the enterprise, or does any team require a course correction?

Such reviews must occur at every level to track the progress to ensure that you meet targets and deadlines for every iteration. After you complete the reviews and gather all of the necessary data points, you can start the next iteration, which begins with the learn phase as shown in Figure 1-3 on page 12.

Returning to our example, Company A decides it must invest in modernizing the way they use IBM Z so that it better fits into their hybrid multi-cloud strategy. Their vision is to release application changes with the same ease and cadence, regardless of which platforms the applications is on.

In Chapter 2, “Modernization goals and approaches” on page 15, we look at how you can set specific modernization goals and general approaches to get started, in addition to examples of the specific goals Company A can set and the approaches they might use to get started.

2

Modernization goals and approaches

So, you want to modernize your IT environment. Where do you get started? The first thing to consider is your goals for modernization. These goals need to be SMART:

- Specific

  It might take some time to get the wording correct, but it is important each goal is clearly defined.

- Measurable

  This metric enables you to gauge progress and impact as you implement each goal.

- Assignable

  You might not set specific roles when setting your goals, but you must think broadly about which teams are to be assigned to each goal.

- Realistic

  You might want to set some stretch goals, but most goals must be achievable within their set scope.

- Time-bound

  Having goals that are measured or restricted by time helps to focus the team and prevent a goal from growing to encompass more than originally intended.

This chapter discusses how to set modernization goals, different modernization approaches by using IBM Z, and where to begin your modernization journey. It includes the following topics:

2.1 Setting modernization goals

Every company is unique, with varying requirements and IT configurations. Even within a company, what works for one group might not work for another. As an organization, whether you are about to embark on your IT modernization journey or you already began, it is encouraged that some self-assessment is conducted periodically. This effort provides a clearer picture of the state of your IT setup and serves as valuable input for your short-term and long-term goals.

For example, you can self-assess by using questions similar to the following examples:

- What works for you today and will it remain strategic in the future?
- What does not work for you?
- Does your IT setup allow agility so requirements can be made quickly?
- Where is the industry heading in terms of the IT system?
- Is your investment in IT modernization generating the kind of return on investment (ROI) that you expected?
- How does your IT setup fare in comparison to that of your competitors?
- What are the technical skills in your IT organization today, and do the same skills continue to be relevant in your wanted future state?
- What is the extent of technical debt in your organization?
- What kind of platforms are used today? Are these platforms suitable to enable your modernization journey?

After you acquire the answers to these questions, prioritize the problems that you want solved. With this information, you are in a much better position to make an informed decision regarding the modernization journey of your IT setup.

Try not to jump from the problems to be solved directly to their solutions; that step in the process comes later. (It is no coincidence that not a single product or offering was discussed thus far in this publication.)

Figure 2-1 expands on the three phases of a modernization journey as discussed in Chapter 1, “Introduction to modernization” on page 1.

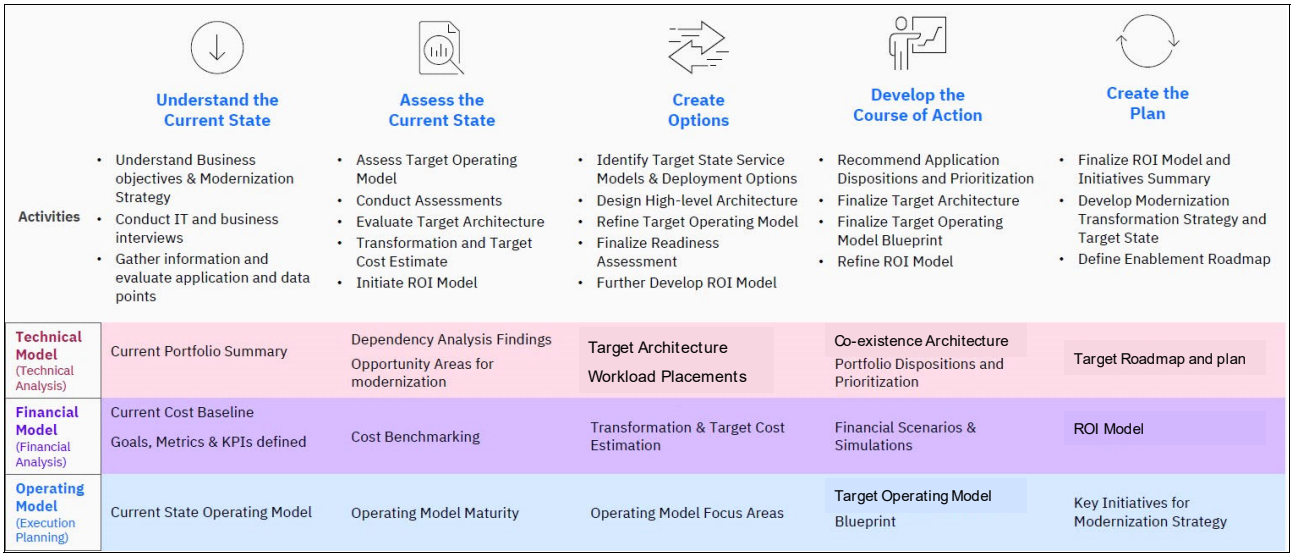

Figure 2-1 Phases of IT modernization journey

From our example that is presented in Chapter 1, “Introduction to modernization” on page 1, after understanding their current state, Company A begins thinking about their wanted future state. Some of the goals Company A decides on setting for their modernization project include the following examples:

- Deliver code changes more frequently with higher quality.
- Reduce the number of manual steps that is needed to promote code from a development sandbox to the test environment.
- Enable connections to data and transactions on IBM Z from other platforms by using industry-standard APIs.
- Improve performance of core applications running on IBM Z.
- Encrypt all client data without requiring application changes.
- Leverage machine learning technology to increase the effectiveness of security monitoring.

As we discussed in Chapter 1, “Introduction to modernization” on page 1, modernization goals can be classified into three categories: infrastructure, application, or process. We discuss these categories next.

2.1.1 Infrastructure modernization

Every new generation of IBM Z includes innovative features; however, you cannot benefit from some of these features if you do not activate them. In the case of infrastructure modernization, it is not merely a matter of keeping your IBM Z systems current; rather, it is updating how you configure them to ensure that you realize the utmost benefit.

Figure 2-2 shows a high-level view of the different layers in an infrastructure stack.

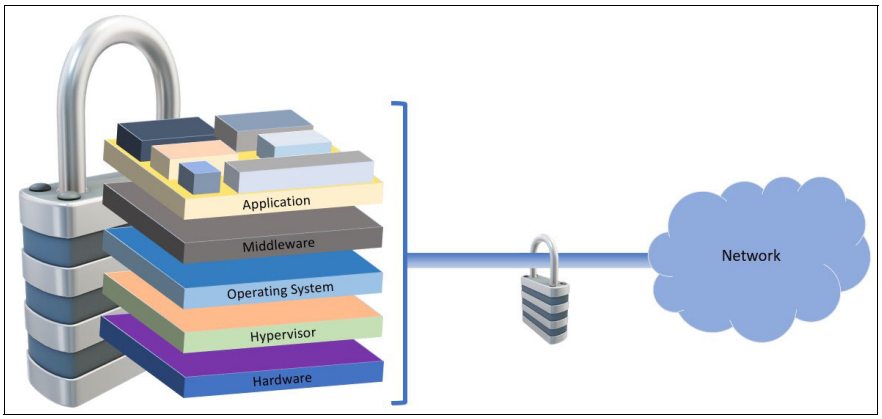

Figure 2-2 Typical infrastructure stack with end-to-end encryption

Comprehensive modernization plans take each layer of the infrastructure into consideration. For example, if you want to modernize applications and how applications are developed on IBM Z, it is important that your middleware products and operating systems are also modernized.

Similarly, to modernize the operating systems, you must have the proper features available and enabled on your hardware. This availability ensures that you use all of the capabilities of the IBM Z platform and applications that are running on it. Consider asking questions to assess what works for your organization and what does not, including the following examples:

- How often does your hardware team consider implementing new features?
- What is your upgrade strategy for your IBM Z systems?
- Is your IT infrastructure resilient to failures and cyberattacks?
- Is your IT infrastructure designed for near-continuous availability?
- Can your IT infrastructure continuously deliver optimal performance with varying workload?
- Can your IT infrastructure handle an increased workload as a result of a planned outage, and quickly catch up with the lost time after everything is back up and running?
- Is your IT security team highly stressed and concerned about data breaches?
- Is your data security encryption algorithm and key quantum-safe?

2.1.2 Application modernization

The focus of application modernization is to build off your applications with minimal changes. Three approaches are available for this type of modernization, each with multiple entry points.

Each well-scoped business requirement can be an entry point to application modernization and can be addressed with a specific solution. These modernization approaches are independent and can be run in parallel. Some approaches and entry points are less complex and can deliver immediate value, such as exposing assets with REST APIs to make them easily accessible and consumable by cloud and mobile applications. Other approaches might be a bit more involved.

In this section, we briefly describe the following approaches and their entry points:

- Approach 1: Expose core mainframe assets
- Approach 2: Containerize and deploy new cloud workloads
- Approach 3: Transform core applications and data assets

Approach 1: Expose core mainframe assets

This approach includes the reuse of long-term investments that are made on business-critical applications that are running on IBM Z by exposing them as OpenAPIs to ease interoperability with other applications. A key business benefit to this approach is that no new code development or code changes are required to access core applications and associated data.

Also, it helps to avoid the need for complicated interfaces to access mainframe assets, time-consuming development to allow integration, or multiple runtime environments.

This approach delivers a key capability for enabling digital transformation and hybrid cloud solutions, which demand new ways of delivering services through new channels and enhanced interactions with cloud-based solutions.

For more information about this approach, see 3.2, “Enabling data accessibility on IBM Z” on page 30.

Approach 2: Containerize and deploy new cloud workloads

This approach focuses on cloud-native application development, with a focus on containerizing existing applications and building new applications with cloud-native technologies. These cloud-native applications can integrate with data and are optimally deployed across, and managed within, the hybrid multi-cloud because of their containerization.

You can deploy cloud-native applications on IBM Z without compromising application development approaches. Some of these architectures and methodologies include microservices, container-based deployment, and Kubernetes or open standards-based service management.

Cloud-native development, with its “develop once and deploy anywhere” philosophy, and the myriad of open source tools that can be integrated to streamline an automated process, revolutionized the DevOps strategy of developing and managing applications. For more information about cloud-native development, see 4.5, “Code” on page 44.

Approach 3: Transform core applications and data assets

Applications that run on IBM Z often were used for many years and underwent numerous evolutions in their lifespans. Application developers altered these programs to meet an endless number of evolving business requirements. An unintended consequence of this long history is that the code for these existing applications can be complex.

To make matters more challenging, home grown business logic is not always fully documented. Although new requirements emerge for accessing data, it is a challenge to use data from your traditional applications’ databases without developing complex application logic as the data model grows along with the application. It is difficult to re-create any of this logic without a deep understanding of the applications.

Because you must keep the business running, the transformation of core applications requires careful, incremental approaches. This approach is critical to address these concerns as it focuses on incremental transformation of mainframe applications through refactoring and modern development approaches.

To modernize such processes, you might consider breaking monolithic applications into several application components. Each component implements a higher-level business task. Then, one or more application components are exposed as APIs that the new business process application can use to compose an agile business process.

You might also consider refactoring the code incrementally, which is aimed at simplifying the code without changing its behavior, which improves the maintainability, flexibility, time-to-market, and above all, the lifespan of these proven programs. Consider externalizing business rules and policies that are embedded in mainframe assets so that these rules and policies are easily accessible and usable across the organization over standard protocols.

When refactoring, you might also consider supporting development in the latest programming languages that complement programs that are written in other languages. This approach not only helps in improving agility, but also to extend existing, or develop new, applications that manage systems of record data.

2.1.3 Process modernization

DevOps practices can help you transform the processes that you use to deliver your mission-critical applications without sacrificing stability or security. Combining a common developer experience with integrated and open source tools, and a streamlined process for developing and maintaining existing and new assets across platforms provides many benefits. Modern development software act as enablement tools making it easier for development and operations to perform the following tasks:

- Analyze and plan: More accurate planning through identifying the effect of a change and its associated risks by identifying dependencies between different IBM Z applications across the enterprise.
- Code and build: Faster application development, testing, and problem determination through integrating IDEs and various product suites to aid developers, and by automating the process of compiling and linking incremental changes.
- Test: Run tests earlier in the lifecycle, or in parallel with other tests, through automation, both of which decrease time to deployment.
- Provision, deploy, and release: Simplify deployment and release management while enabling audit trails and automated tracking of feedback.

In addition to DevOps, modernizing other processes also realizes many other benefits. For example, tracking transactions and monitoring the correlation of application, transaction, and system resource performance data on IBM Z can provide valuable insights, predict failures, and alert support staff for necessary corrective action. User-friendly monitoring products can be customized to suit your requirements to empower the operations teams.

For more information about DevOps on IBM Z, see Chapter 4, “Introduction to DevOps” on page 39.

2.2 Where to start

Modernization is not about putting your applications on a cloud offering. Rather, it is about the use of the significant business value of your investment in IBM Z and enabling it to become the center of your hybrid cloud, which is unlocked through modernization.

Figure 2-3 shows some entry points into a modernization journey. Each of these entry points can naturally pull in the adjacent areas as the journey progresses.

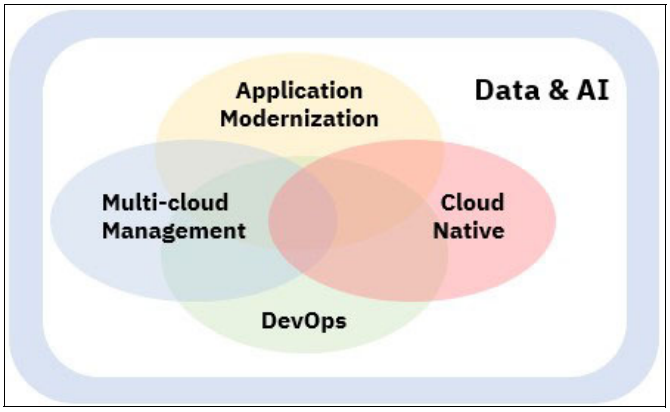

Figure 2-3 Modernization entry points

Modernization comes in different categories and approaches. You can approach modernization in various incremental ways with, or without, the need to change the underlying code.

Where you start depends on the level of modernization that is required based on your needs and priorities. In this section, we discuss approaches on how to decide where to start your modernization journey based around the following topics:

- Disruption to your business
- Business risk
- Specific drivers to modernize
- Teams affected
- Business continuity plan
- Questions for IT team
- Questions for application developers
- Role of an IT leader in the journey to modernization

Because you need to maintain availability always during modernization, the transformation of core applications requires cautious, incremental approaches. We discuss later in this section conversation starters to consider that can aid in planning your modernization project.

Disruption to your business

Most teams consider a production outage that is caused by a project to be a significant failure. Some questions to ask when considering a potential disruption to your business, include the following examples:

- How much time is needed for project completion?
- Does this project require education to build the team's foundational knowledge before implementation?
- How long after completion will it take to see a positive ROI?

- How can this project be implemented without taking down existing systems?
- Is there a fallback plan if problems arise when deploying to production?

In general, you want to start a modernization journey with a project that has a low likelihood of causing major disruptions. After a few projects are successfully completed, you can apply the lessons from those projects to a project with a higher risk of disruption.

Business risk

Part (but not all) of the risk that is assigned to a modernization project comes from the potential disruption to the business. In the case of a modernization fall out, it is important to make specific considerations, such as the following examples:

- What are the worst- and best-case scenarios for this modernization project?
- Is the amount of risk that is introduced by this modernization project acceptable?
- What steps can be implemented to reduce the risk that is associated with the modernization project, and how long does it take to implement these steps?

The suggested practice is to start a modernization journey with a project that has no more than a moderate amount of business risk. You want to select something that demonstrates considerable benefits with an acceptable amount of risk.

Specific drivers to modernize

Modernization can be done for many reasons and to meet many objectives. You can ask yourself several important questions, such as the following examples:

- Are you implementing modernization projects to improve user experience?
- Do you have the latest technology and are you fully harnessing the new technology?
- Are you addressing issues of skill-gap with this project?

It is important to periodically reflect back on your modernization drivers to ensure that they are being addressed during planning and execution. This review prevents the project from drifting off course.

Teams affected

It is not only the IT department that feels the effects of modernization. It is crucial to understand how all areas of your workforce might be affected, even if they are not working with IT infrastructure directly.

IT departments are directly affected by modernization because they are tasked with the implementation of the selected modernization solution. They are also expected to be well versed with the modern technology and to confidently implement the modernization steps in a near flawless manner. This staff not only is affected during implementation, but also as they use the new solutions going forward.

Other departments might also feel the effects of modernization, even though they were not tasked with the implementation. Therefore, it is important to consider how the modernization affects other teams.

It also is imperative to the success of your modernization journey that you be prepared for extreme scenarios through performing a thorough risk assessment with all affected teams. The suggested practice is to start your journey with projects that are simplified by affecting only a small group of people.

Business continuity plan

To help you be better prepared in the case of modernization tasks having major effects, you can ask many questions, including the following examples:

- If an unforeseen issue occurs, is a fall-back plan in place?
- Is an escalation matrix available for you to contact the required personnel if you need help during the back out plan?
- Is your expert team equipped for what needs to be done and how to fall back successfully, if need be?

Questions for IT team

As a part of the planning process, several important conversation starters are available to begin discussions with your team, including the following examples:

- Do you have a list of key business areas or business processes that run on IBM Z?

  This list helps you understand the potential for modernization with IBM Z across your business.
- When was the last time you took an inventory of the features that were added with your latest IBM Z hardware to make sure that you squeeze every last drop out of your investment?

  This assessment can help you identify a hole in your upgrade process, and find ways to improve your implementation.
- Are all the key features of your current IBM Z environment used?

  This question helps in identifying the key focus areas of modernization.
- Are there any factors, such as back-leveled infrastructure, skill availability, and subject matter expert readiness, inhibiting the journey to modernization?

Questions for application developers

As a part of the planning process to modernization, some of the following conversation starters can be used with your developers and identify appropriate DevOps solutions:

- Which, if any, agile methodologies do you use in development, testing, and production deployment process?

  This question helps in identifying the type of DevOps solutions from which to choose.
- How quickly can you release a new application or update?

  This question can help prompt a conversation about which tasks take the longest amounts of time and how that process can be improved.
- Are you aware of any tools for application development that can make your job easier?

  This question helps you identify potential ways to enable developers that are using new tools.

Role of an IT leader in the journey to modernization

IT operations teams are becoming strained as applications are changing at a more frequent pace, and application architecture is becoming complex and heterogeneous. Skilled personnel are retiring, creating skill gaps while budgets are seldom unlimited.

IT leaders play a key role in the digital transformation process, as leaders are responsible for making informed business decisions to set their plans and strategies in relation to modernization.

The details that the IT leader at Company A implemented to help with collaboration include the following examples:

- IT leader: Sets initial strategy and make key, high-level decisions.
- Core team: The key personnel from the technical team and their immediate managers that are responsible for implementing the modernization project.
- Stakeholders: The key personnel from a business perspective for decision making in the modernization project.
- Phases: The high-level steps of modernization, with target dates.
- Governance: How the team tracks and monitors the progress of the modernization project. This issue must include key metrics and the tools that are used to gather them.

Now that we reviewed at the key technical and business considerations that are part of a modernization strategy, we bring everything together and briefly define the strategy of Company A in Chapter 3, “Modernization technologies on IBM Z” on page 25.

3

Modernization technologies on IBM Z

According to a survey done by Wipro, a company that provides information technology, consulting, and business process services, and Ensono, which provides managed services for hybrid IT environments, 50% of businesses in Europe and the US are looking to expand their IBM Z capabilities, with only 5% thinking of reducing it.[1]

Also, the International Data Company (IDC) conducted research that determined that organizations can derive significant value from the platform's ability to serve as part of a hybrid cloud environment, present easy to-view graphical interfaces, run open source application development languages, and be highly analytics-driven and automated.[2]

Modernization initiatives, common to many clients, include the use of API to connect their IBM Z environment to both internal and external networks, simplify operations that use web-based interfaces, use Linux on the platform, integrate Z into DevOps deployment pipelines, and support agile application development on the platform.

In this chapter, we focus on the technologies and solutions that are available on IBM Z to accelerate your digital transformation, except for DevOps. For more information about DevOps, see Chapter 4, "Introduction to DevOps" on page 39.

This chapter includes the following topics:

1 https://www.ensono.com/about-us/press-room/research-ensono-and-wipro-finds-50-businesses-are-looking-expand-mainframe

2 IDC white paper, sponsored by IBM and Broadcom, The Business Value of the Transformative Mainframe, August 2019.

3.1 Hybrid cloud

A hybrid cloud is an IT infrastructure that connects a public cloud and at least one private cloud. It provides orchestration, management, and application portability between the two to create a single, flexible cloud infrastructure for running an organization's computing workloads. A hybrid multi-cloud is a hybrid cloud that includes more than one public cloud from more than one cloud service provider.

Broadly speaking, two cloud service models are available: Infrastructure as a Service (IaaS) and Platform as a Service (PaaS). Before finalizing the model you choose, you must understand and decide on the roles and responsibilities that you envision the cloud vendor, or your internal cloud support team, to manage versus the roles and responsibilities that you envision your application team to manage.

Figure 3-1 shows the separation of duties in IaaS versus PaaS models.

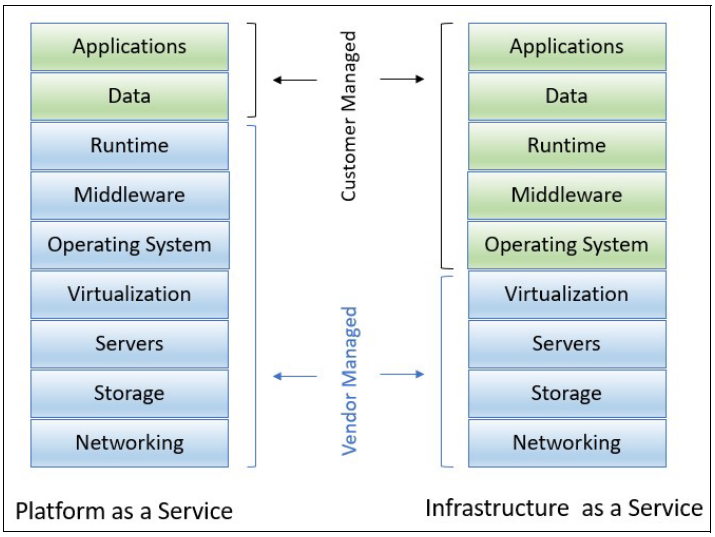

Figure 3-1 Separation of duties in IaaS and PaaS models

In the IaaS model, the cloud vendor manages the underlying infrastructure and provides you the capability to manage the virtualized infrastructure. The installation, patching, and management of the operating system, middleware, DevOps products, delivery pipeline setup, data management, and application remain the responsibility of the application delivery team.

In the PaaS model, the capabilities and management of the underlying layers of the infrastructure is provided as a service by the cloud vendor. The application development and testing tools are provided as services and the application delivery team is not responsible for managing the development pipeline.

IBM Managed Extended Cloud IaaS for IBM Z

IBM Managed Extended Cloud IaaS for IBM Z delivers a highly available IaaS platform for z/OS or Red Hat Enterprise Linux (RHEL) running on Z. This offering uses a cloud delivery model to provide access to a scalable, multi-tenant infrastructure that is flexible and adaptable.

On top of the wanted operating system, the latest versions of standardized software stacks are available to facilitate flexibility. IBM also supports independent software vendors (ISVs) and other IBM software on a custom basis.

Compute, storage, and tape capacity can all scale to meet your needs. Figure 3-2 shows IBM Managed Extended Cloud IaaS for IBM Z architecture with multiple tenant LPARs, which are hosted in IBM data centers.

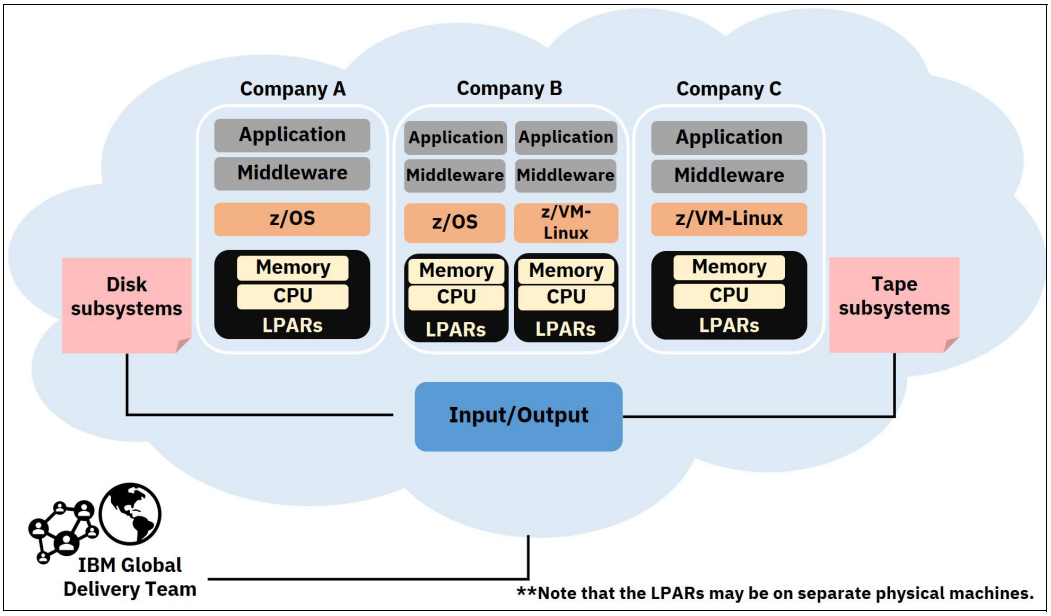

Figure 3-2 IBM-managed Mainframe IaaS Environment with LPARs and storage subsystems

IBM Z systems are housed in purpose-built, data centers around the world. This LPAR-based model is designed to offer high levels of security with the system achieving Evaluation Assurance Level 5 (EAL5) accreditation. For more information about IBM Managed Extended Cloud for IBM Z, see this web page.

From our ongoing example, the key to Company A's modernization project is to bring their IBM Z environment into their hybrid cloud. This is scenario is common for many organizations that require the freedom to securely deploy, run, and manage their data and applications on the cloud platform of their choice without running the risk of vendor lock-in.

A hybrid multi-cloud approach brings the flexibility to host your own software one day, move that same setup to a cloud provider the next, and still have the freedom to change cloud providers in the future. You can run sensitive, highly regulated, mission-critical applications on private cloud infrastructure.

You can run less sensitive or even temporary workloads, such as development and test environments, on the public cloud. With the proper integration and orchestration between the two, you can take advantage of both, when needed, for the same workload.

In addition, a hybrid multi-cloud approach enables organizations to adopt common management and software development capabilities that expand across all locations, whether in a public cloud, a private cloud, or on-premises. A hybrid multi-cloud approach brings all of the benefits of a public cloud to other facets of an organization's IT environment. It also allows companies to gain visibility and control over their entire infrastructure and in turn, release their innovations into the world in a more secure and efficient manner.

Figure 3-3 shows the hybrid cloud ecosystem that is built around IBM Z hardware in a complex system of interdependent components that work together to enable cloud services.

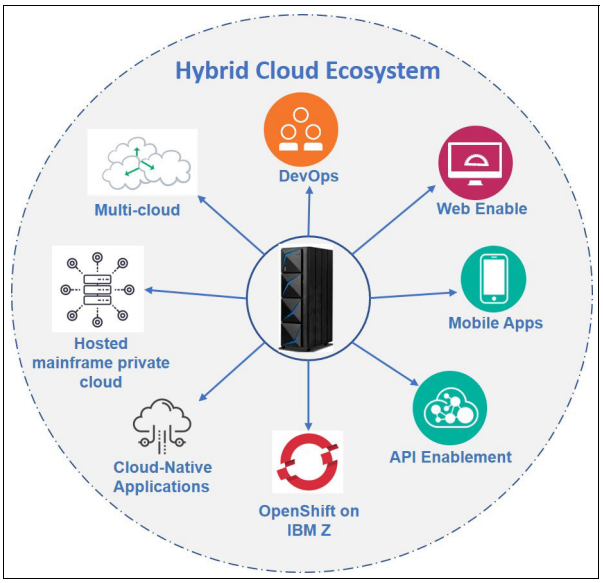

Figure 3-3 Hybrid cloud ecosystem

3.1.1 Hybrid cloud common use cases

Some common hybrid cloud use cases that effectively break through established silos, which might be relevant to your business, include the following examples:

- Software-as-a-Service (SaaS) integration: Connecting SaaS applications on the public cloud to applications on public/private cloud and traditional IT, which results in new solutions and fast innovation.
- Data and AI integration: Combining new data sources and analytics, machine learning, and AI capabilities on the public cloud with on-premises data.
- Enhancing traditional applications: The use of public cloud services to upgrade the user experience and deploy globally to new devices.
- DevOps: Embracing public cloud IaaS and PaaS for speed of testing and deploying on-premises to meet security, compliance, and business production needs.
- Composite multi-cloud: Building composite applications by using microservices from multiple vendors across on- and off-premises cloud environments to take advantage of the increase in new technology that is available from multiple cloud sources.
- Edge computing: Managing application performance for applications that rely on mass scale, distributed data by using edge computing to process and return less data in a hybrid model.

3.1.2 Hybrid cloud benefits

Many benefits are realized that come with the flexibility of the use of a hybrid cloud. In general, benefits that are achieved through hybrid cloud implementation include, but are not limited to, the following examples:

- Security and compliance: Deploy sensitive workloads in the safety of a private cloud, and still deploy less-sensitive workloads to public cloud services.
- Flexibility: Obtain increased options and flexibility when it comes to deploying workloads, which makes best use of your on-premises investments and infrastructure budget. You can also easily alter that deployment in response to changing workloads and new opportunities.

3.1.3 Modernization with hybrid cloud

Many methods are available to bring IBM Z environments into your hybrid cloud. Figure 3-4 compares three modernization scenarios that include IBM Z and hybrid cloud: cloud-native, hybrid, and in-place (on-premises), and emphasizes how the hybrid cloud setup stands out. It is the hybrid scenario that Company A pursued.

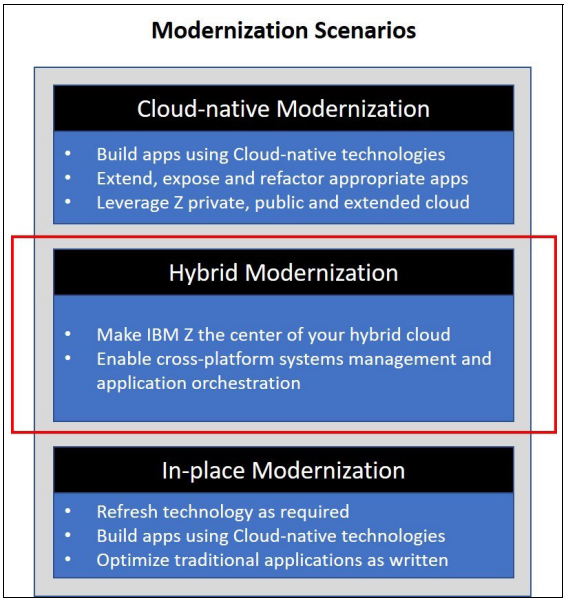

Figure 3-4 Hybrid cloud scenario on IBM Z

Cloud-native development is possible on IBM Z, as it is on other platforms. Developers can test their applications in a containerized, virtual environment.

Common tools, such as Microsoft Visual Studio Code for code editing, and Git for source control, work great with IBM Z. Developers can even self-provision middleware instances on z/OS without needing system programmer skills.

For more information about modernizing the application development process, see Chapter 4, “Introduction to DevOps” on page 39.

3.1.4 Colocation with IBM Z

Many organizations use the benefits of locating their Linux environments alongside their other production environments on IBM Z. These Linux environments benefit from the strengths of IBM Z while also reducing the latency between environments because they are hosted in the same data center, or perhaps even the same physical system. This colocation also allows for easy location switch over in the event of planned or unplanned outages.

Starting with z/OS Version 2 Release 4, an even closer location became available with z/OS Container Extensions (zCX). For Linux on Z components in direct support of z/OS workloads, zCX enables the deployment of a Linux on Z container in a z/OS system, where the Linux on Z container runs as its own address space in z/OS. Applications resemble any other Docker application to the developer.

Figure 3-5 shows how the zCX address space operates on z/OS.

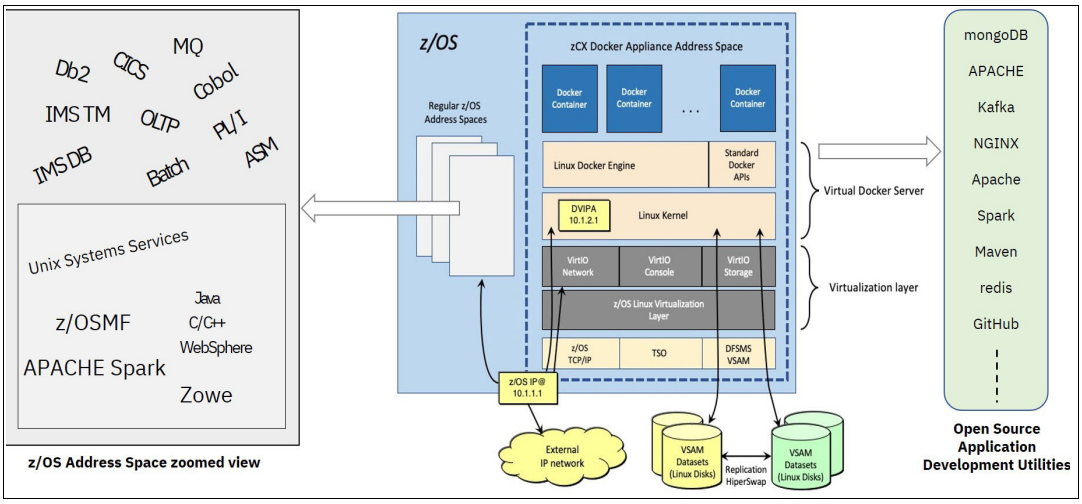

Figure 3-5 Expanding z/OS software ecosystem

Most applications that available to run on Linux only can run on z/OS with zCX. zCX runs Linux on Z applications on z/OS by using z/OS operations staff and the existing z/OS environment. For more information about z/CX, see *IBM z/OS Container Extensions (zCX) Use Cases*, SG24-8471.

3.2 Enabling data accessibility on IBM Z

To enable work across platforms and applications, data must be easily accessible in a common way. This, more than anything, has fueled the API economy as standards were implemented from cross organization communication. A key piece to modernization is enabling access to data that lives on IBM Z.

IBM z/OS Connect Enterprise Edition

IBM z/OS Connect Enterprise Edition allows you to serve APIs from common z/OS middleware, such as IBM CICS®, IBM IMS, and IBM Db2®. Essentially, z/OS Connect translates incoming REST calls to the format and codepage that is understood by the configured middleware. For example, you might have an incoming REST call in UTF-8 transformed to a JMS call in EBCDIC going to IBM MQ.

In addition to this basic support, z/OS Connect provides many powerful functions, including:

- Auditing
- Transaction logging
- API discoverability
- Request and response mapping by using simple Eclipse-based tooling
- Easy integration with API Management that uses OpenAPI descriptors

z/OS Connect also gives you the ability to easily use APIs from applications that are written on z/OS. In many cases, these applications were written before REST was conceived. z/OS Connect provides your COBOL application; for example, a control block with the results from a REST call so the application can use and act on them.

IBM Data Virtualization Manager

IBM Data Virtualization Manager (DVM) can be used to provide read/write access to IBM Z data, and other sources of data. It optimizes access to IBM Z transactional data, in-place, through standard APIs, such as REST, SOAP, Java database connectivity, and open database connectivity.

When you include DVM as a part of your modernization strategy, you decrease the negative impact that traditional data movement approaches possess. This opportunity to benefit from data, where and when it is needed, is obtained through DVM's ability to use these popular APIs. API usage also helps with lowering the need for mainframe skills when modernizing applications.

With DVM, enterprise applications can effectively and cost-efficiently access and update live transactional IBM Z data that is stored in IMS, IDMS, ADABAS, and VSAM. It supports a federation of IBM Z data with a myriad of structured and unstructured data sources, such as Oracle, Apache Hadoop, MongoDB, and web services data.

Structured, unstructured, internal, and external data (regardless of platform of residence) must be incorporated into analytics applications to provide a broader, more holistic view of data for improved insight. Bringing enterprise data and analytics together on a single platform enables fast, accurate responses to analytic queries and generations of predictive insights.

By using enterprise data in-place where it originates, organizations can use a security-rich environment, minimize latency, and improve the accuracy of data that is used in analytics. The insights that are gained from the analytical capabilities of DVM are valuable in recognizing compliance issues and system readiness, which prepares you to preemptively address operational threats. For more information about DVM, see this web page.

3.3 Protecting data with IBM Z

As organizations modernize their applications and infrastructure through shifting to a hybrid cloud architecture that blends on-premises infrastructure with private and public cloud models, it is important that security and privacy are considered through every step of that journey. Despite cloud offering clear agility benefits, it also runs the risk of exposing data in the process.

IBM Z can modernize your security and data privacy approaches to fit the hybrid cloud architecture. These approaches are designed to not only protect on-premises data, but also provide data privacy and security by extending it across the hybrid multi-cloud. In this section, we review some of the tools and features that are available for data privacy, safeguarding hybrid cloud, and identification and prevention.

3.3.1 Data privacy

One of the tenets of data privacy is ensuring that unauthorized parties cannot access sensitive data, even if they have a copy of the data. With a high number of users simultaneously running various applications with differing performance profiles within a single environment, a requirement exists for a multi-layered security approach to ensure data privacy. These necessary approaches can include user identification and authentication, access control, data encryption, and other advanced security measures.

To help enterprises achieve this security approach, IBM offers key tools and technologies, such as the following:

- Pervasive encryption
- IBM Data Privacy Passports
- IBM Z Data Privacy for Diagnostics
- IBM Fibre Channel Endpoint Security
- IBM Enterprise Key Management Foundation
- IBM Hyper Protect Virtual Servers and Hyper Protect Services
- IBM Secure Execution for Linux

Pervasive encryption

Encrypting data has long been a solution to preventing unauthorized parties from reading sensitive data. Traditionally, sensitive data must be identified manually and encrypted by policy.

With z14, IBM introduced pervasive encryption that allowed all data to be encrypted, both in-flight and at-rest. This major advancement was from traditional, selective encryption that was meant to protect data within the IBM Z platform.

Pervasive encryption on IBM Z is an approach that is aimed to enable extensive encryption of in-flight and at-rest data, simplify encryption, reduce associated costs with data protection, and achieve compliance mandates. For example, with z/OS, three capabilities are available for pervasive encryption: z/OS data set encryption, z/OS Coupling Facility encryption, and z/OS Encryption Readiness Technology.

For more information about pervasive encryption on IBM Z, see this web page.

IBM Data Privacy Passports

Data Privacy Passports are designed to control protected data after it leaves the IBM Z environment. The ultimate goal is to extend data protection and privacy from your Z environment across hybrid multi-clouds.

Data Privacy Passports protect your data through packaging the data into trusted data objects. Access to the data can be revoked regardless of where the data is located, a capability that can be applied to IBM Z data, and data from other platforms.

The benefits of the use of Data Privacy Passports include safeguarding sensitive data, simplifying how you maintain compliance under industry mandates and regulations, and managing access to shared data on a need-to-know basis with user access policies.

For more information about Data Privacy Passports, see this web page.

IBM Z Data Privacy for Diagnostics

Data can be encrypted in-flight and at-rest, but what protects data in-memory, particularly in memory dumps? For example, when a software component ends abnormally and a memory dump of its memory contents is taken, is there a way to avoid capturing sensitive data in its dump?

A common scenario is needing to share the dump with support for analysis and problem resolution but being unable to do so because sending sensitive data is prohibited. As a solution, IBM provides Data Privacy for Diagnostics, which allows enabled applications to tag sensitive data as critical for problem diagnostics. In this case, when the memory dump is captured the sensitive attribute of data is also captured in the dump. This capability protects data by redacting anything that is tagged as sensitive and creating a second diagnostic memory dump to be shared externally.

For more information about Data Privacy for Diagnostics, see this IBM Solution Brief.

IBM Fibre Channel Endpoint Security

IBM Fibre Channel Endpoint Security is another data security technology that contributes to the IBM Z approach of encryption everywhere. Fibre Channel is the premier data transport medium for Storage Area Networks (SAN). Although SAN provides highly available data access and optimal performance, the security measures for SAN must be in place to reduce and eliminate insider threats of unauthorized access of data at all times, even within the walls of the data center.

As you continue on your modernization journey, IBM Fibre Channel Endpoint Security can be used as an end-to-end solution that ensures all data in-flight on Fibre Connection (IBM FICON®) and Fibre Channel Protocol (FCP) links from IBM Z to DS8900F, or between IBM Z platforms over FICON channel-to-channel connections, is encrypted and protected. This feature enables encryption of all in-flight storage data, regardless of the operating system. For more information about Fibre Channel Endpoint Security, see *IBM Fibre Channel Endpoint Security for IBM DS8900F and IBM Z*, SG24-8455.

IBM Enterprise Key Management Foundation

As encryption becomes more widely deployed, enterprises are confronted with the challenge of managing a growing number of encryption keys. To effectively do so, IBM introduced Enterprise Key Management Foundation (EKMF) as a solution. It is a flexible and highly secure central management system for keys.

EKMF provides multi-platform, multi-site, and multivendor support, a central repository for stored all keys, certificates and metadata, monitoring of keys and certificates along with secure key generation, role-based access control, dual control, and audit logging. EKMF also offers a web edition to help simplify encryption, which includes a dashboard to view which data sets are encrypted, manage encryption deployment, and advanced and simplified auditability with consolidated key management logs. For more information about EKMF and EKMF web, see this web page.

IBM Hyper Protect Virtual Servers & Hyper Protect Services

IBM Hyper Protect Services are hosted within secure enclaves on Z systems in IBM Cloud®. These solutions are designed around the Bring Your Own Key (BYOK) principle, which means that you (and only you) have the key that is needed to decrypt the data in these environments.

IBM Cloud administrators are responsible for running the systems that serve the services, but do not have the key. Therefore, these administrators cannot read your data.

IBM Cloud hosts two database-as-a-service services, a cryptographic service that is a hardware security module (HSM) in the cloud, and a virtual server offering, all within the Hyper Protect family. Hyper Protect Virtual Servers can also be deployed on-premises. For more information about Hyper Protect Services, see this web page.

IBM Secure Execution for Linux

Secure Execution for Linux is designed to provide scalable isolation for individual workloads at a granular level to help protect them from internal and external threats and attacks. Current data protection approaches are designed for data at-rest and in-flight, but few address data in-use. Secure Execution addresses this dilemma through the implementation of a hardware-based Trusted Execution Environment (TEE). Integrating Secure Execution for Linux into your modernization strategy around security provides the following benefits:

- Isolation between a KVM hypervisor host and guests in virtual environments to provide protection and safeguards against insider threats, such as malicious administrators.
- Helps enterprises provide isolation between individual multi-tenant workloads that are running on a shared LPAR.
- Support for enterprise DevSecOps solutions by providing protection to the pipeline of code development from start to finish.

For more information about Secure Execution for Linux, see IBM Knowledge Center.

3.4 Artificial intelligence

Now that we secured our data and protected its privacy with IBM Z, we can work to get better insights from the data by applying artificial intelligence (AI) and analytics. Putting AI models into production within your core high-volume business applications can generate key benefits, including the following examples:

- Flexible model development
- Improved productivity
- Enterprise-ready AI model deployment
- Enhanced model accuracy
- Production-ready machine learning
- Quick-start solution templates

IBM offers various suites that run AI in your z/OS environments and are designed to help you gain actionable insights by making use of open source machine learning. The following AI tools that are available on IBM Z are discussed next:

- IBM Watson® Machine Learning for z/OS
- IBM Open Data Analytics for z/OS
- IBM Db2 AI for z/OS

IBM Watson Machine Learning for z/OS

IBM Watson Machine Learning for z/OS (WMLz) brings AI to your transactional applications on IBM Z. It offers an end-to-end machine learning platform that operationalizes predictive models. It also benefits from both proximity to your data and core IBM Z qualities of service. Organizations can keep their sensitive data in-place on IBM Z, at the same time adding external data to build complete insights. This approach allows users to maximize the value of mission-critical data.

For more information about WMLz, see this web page.

IBM Open Data Analytics for z/OS

IBM Open Data Analytics for z/OS (IzODA) consists of three components that are designed to improve your data processing experience on z/OS: z/OS IzODA Spark, z/OS Mainframe Data Service, and z/OS IzODA Anaconda. These components work together to help developers and data scientists gain real-time insights from data sources with reduced latency, and flexibility to access diverse data by using modern AI capabilities.

The use of IzODA in your journey to modernization on IBM Z include the following key features:

- Its ability to drastically improve iterative process performance by using caching intermediate results in memory versus writing them to disk.
- The integration facilities that it delivers for both on- and off-platform data source.
- Its ability to serve as a comprehensive solution for integrating computations to your data.
- Allows enterprises to experience open source run times and libraries.

It also can analyze data at its source. This feature removes the risk that is created when data is replicated and moved and increases the value of insights that are produced by leveraging the most current data available. With the power of Apache Spark, IzODA integrates key open source analytic technologies with optimized data access and abstraction. For more information about IzODA, see this web page.

IBM Db2 AI for z/OS

Db2 AI for z/OS (Db2ZAI) is designed to learn the behavior of connections and other types of threads in Db2 for z/OS and recommend Db2 profile controls to optimize performance and reduce any negative cross-application performance. It also detects Db2 for z/OS system performance exceptions and provides recommendations for fine-tuning that is based on your environment.

Db2ZAI uses machine learning to build its recommendations and is built on the services that are provided by WMLz. Db2ZAI empowers the Db2 for z/OS optimizer to determine the highest-performing query access paths that are based on workload characteristics.

Figure 3-6 shows the architecture of Db2ZAI. It is designed to reduce CPU consumption, improve Db2 application performance, and enable rapid model learning that is specific to the data and application behavior per subsystem, without requiring data science skills.

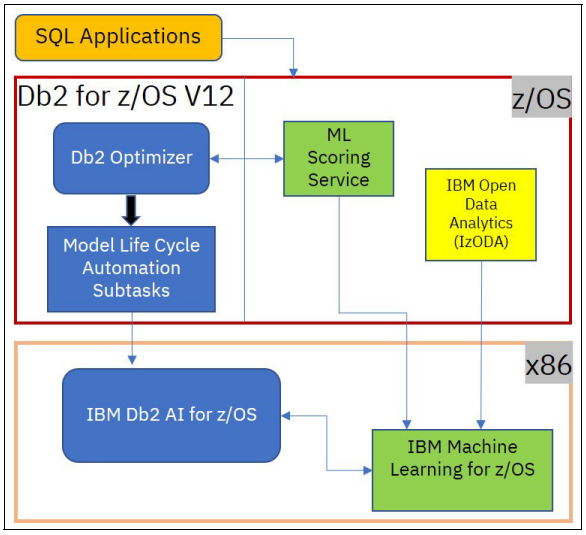

Figure 3-6 Architecture of IBM Db2ZAI

For more information about Db2ZAI, see this web page.

3.5 Big data analytics

Big data analytics is the use of advanced analytic techniques against large and diverse data, ranging up to zettabytes. The data ranges from structured, semi-structured, and unstructured, and can be from multiple sources.

Big data can have high volume, velocity, or variety, and is beyond the ability of traditional relational databases to capture, manage, and process with low latency. Analysis of big data by using advanced analytics techniques, such as text analytics, machine learning, predictive analytics, data mining, statistics, and natural language processing, allows for more informed decision making at a faster rate to gain new insights from previously untapped data sources, independently or together with existing enterprise data.

Today's businesses need real-time insight that is derived from transactional data. Multiple copies of the same data often exist throughout the enterprise, as shown in Figure 3-7, and each copy of data has an associated cost, latency, and risk.

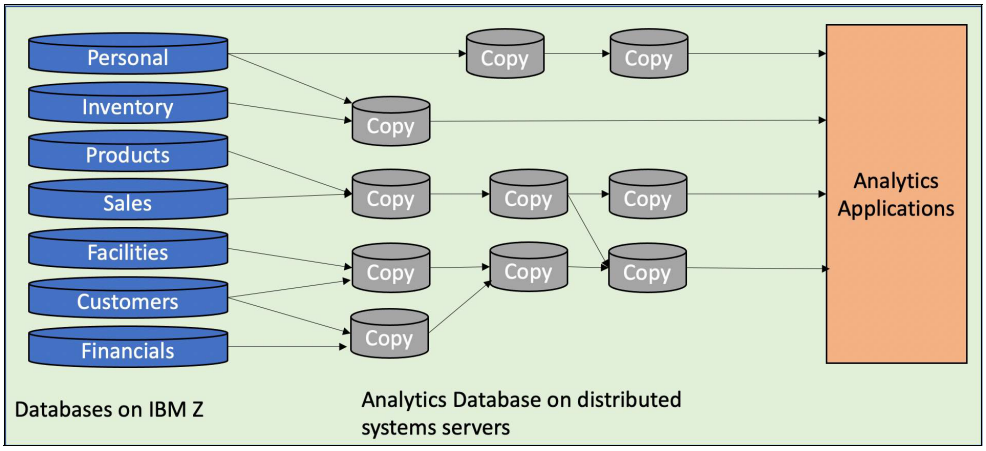

Figure 3-7 Data and its associated cost, latency, and risk

Security, data privacy, and data governance within this environment are inherently challenging. Because of this inflexible and complex approach to data distribution, new, live enterprise data that is created is not readily available for analysis. This configuration is unable to support modern analytics requirements and real-time insight.

Enterprise data, both current and historical, continues to be the primary critical data source for most analytics initiatives. If most of an organization's enterprise data is flowing through IBM Z, it has a distinct advantage in the pursuit of modern analytics. Organizations can modernize their environment and practices to cost-effectively combine enterprise data and analytics processing on a single platform, integrate that data with non-relational data sources, and enable powerful, real-time analytics and cognitive insights. This on-premises approach provides an analytics foundation that can then be extended to the cloud. Depending on what best addresses business needs, organizations can implement a cloud-based approach to analytics that is built on an on-premises private cloud, public cloud, or hybrid cloud environment.

Organizations are looking to gain advantage by modernizing and incorporating live enterprise data in analytics; that is, data within the course of a transaction's execution. Modern analytics applications must have access to current transactional data and the infrastructure must be designed with the same level of security, availability, scalability, and performance as transactional systems.

IBM Db2 Analytics Accelerator

IBM Db2 Analytics Accelerator (IDAA), which is tightly integrated with Db2 for z/OS, accelerates database queries. IDAA can be deployed in two ways: in a dedicated z/OS logical partition (LPAR) or as a hardware appliance. Either deployment model is designed to significantly reduce query response times.

By putting rich enterprise data to work in analytics, insight can be derived from live transactional data, in real time, and at the optimal moment. This approach transforms traditional tactics to analytics to provide an essential foundation for real-time analytics, delivering high-speed processing for complex Db2 queries to support business-critical reporting and analytic workloads.

For more information about Db2 Analytics Accelerator and HTAP, see this web page.

Many modernization technologies can be used with IBM Z, and hybrid cloud gives a choice of where to host your infrastructure. Table 3-1 lists the solutions based on where they are hosted.

Table 3-1 Summary of solutions by host location

Location	Operating system	Solution to consider
On-premise	z/OS	Traditional LPAR
	Linux	Linux on IBM Z
		z/OS Container Extensions running in z/OS
“In the Cloud”	z/OS	IBM Managed Extended Cloud IaaS for IBM Z
	Linux	IBM Managed Extended Cloud IaaS for IBM Z
		Hyper Protect Virtual Servers

Perhaps most important for developers is process modernization because it affects every step of the software lifecycle. For more information about DevOps and popular techniques for process modernization, see Chapter 4, “Introduction to DevOps” on page 39.

4

Introduction to DevOps

DevOps is what you form when you break down the barriers that traditionally separated developers (“Dev”) from IT operations (“Ops”). When these teams work in isolation from one another, lack of communication and coordination between them can lead to inefficiencies and delays because the people who write code are out of sync with the people who deploy and manage it.

A successful DevOps implementation typically relies on the integration of various tools or solutions, sometimes referred to as *toolchain* or *toolset*, to eliminate manual steps. This integration helps to improve visibility and efficiency, reduce errors, and provides capability to scale the setup from a small team to enterprise level for cross-functional collaboration.

DevOps emphasizes agility, meaning the ability to implement changes easily. DevOps teams focus on standardizing development environments and automating delivery processes to improve delivery predictability, efficiency, security, and maintainability. DevOps also encourages empowering teams with the autonomy to build, validate, deliver, and support their own applications.

This chapter discusses the phases of the DevOps cycle along with various solutions available for each phase to adopt DevOps practices with IBM Z. It includes the following topics:

4.1 DevOps culture

A successful DevOps strategy attempts to address the previously mentioned challenges by establishing a culture that incorporates some, or all, of the following key values:

- Collective accountability: Everyone on the team must take responsibility for ensuring that software is delivered on time and meets expectations. Software delivery is everyone's responsibility in DevOps.
- Transparency: In a DevOps culture, all team members must have constant visibility into what other members of the team are doing.
- Automation: DevOps places a priority on the use of automation to build quality into every step, ensure consistent results, and speed up delivery.
- Shared roles: Because DevOps brings together multiple teams, it might become more common for shared responsibilities. One example might be developers doing certain operational tasks in their test environments that were previously done by the operations team.

Figure 4-1 shows how developers and operations can come together by using DevOps culture to fulfill business requirements.

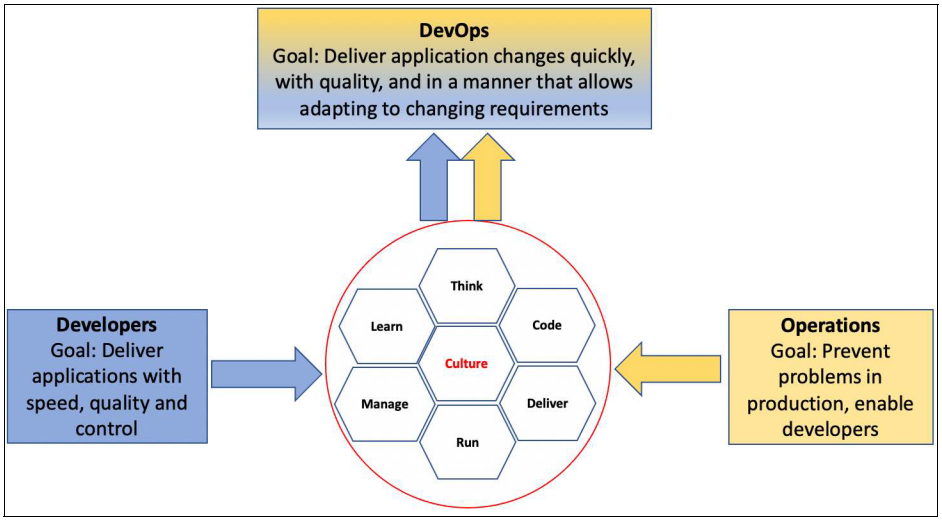

Figure 4-1 DevOps culture

4.2 DevOps on IBM Z

The flexibility, resilience, and agility a cloud platform brings to their hosted applications allows for streamlining an application delivery pipeline. Environments from development through testing and all the way to production can be provisioned and configured as needed, and when needed. This process minimizes the environment-related bottlenecks in the delivery process and makes for a compelling business case for cloud adoption with DevOps.

While integrating IBM Z into your hybrid cloud, it is imperative that developers and IT operations understand that the same, agile processes can also be performed on IBM Z, by using the same DevOps tools and having the same DevOps experience as other platforms. A range of solutions helps integrate systems, empowering developers with an open and familiar development environment with enterprise-wide, platform agnostic standardization, in turn helping developers build, test, and deploy code faster.

4.2.1 Continuous delivery and deployment with IBM Z

DevOps shifts the way developers deliver code. There are three key techniques in this regard:

- Continuous Integration: Developers frequently deliver their code to an integrated, common repository.
- Continuous Delivery: Code is frequently built and tested, all in an automated fashion. In this case, manual action is needed to deploy the code.
- Continuous Deployment: Code is automatically deployed to a live environment.

The goal of these techniques is to enable developers to rapidly deploy changes to an environment, simultaneously ensuring that the deployed code was rigorously tested such that quality not be compromised. The need to maintain quality at velocity is why test automation is an essential component of the process. The process that begins when developers commit their code changes to a repository and ends with deployment is often represented visually and is called a *pipeline*.

Figure 4-2 shows the difference between continuous delivery and continuous deployment pipelines.

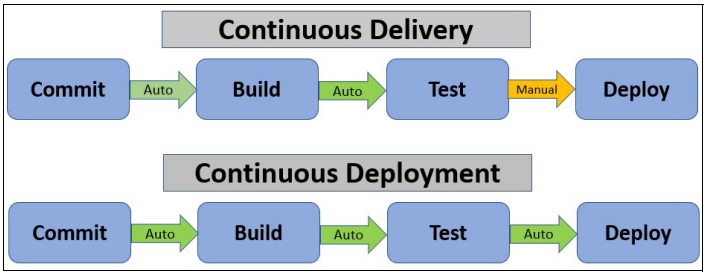

Figure 4-2 Continuous delivery versus continuous deployment

Large enterprises sometimes use a combination of continuous deployment and continuous delivery. For example, developers might be encouraged to deliver their code at least once a day because a nightly build is automatically performed, followed by automatic functional tests, and finally automatic deployment into a test environment. This process describes a continuous deployment scenario.

However, the pipeline into a production environment might be gated by a manual approval so that portion is continuous delivery. Many benefits can be drawn without the need to enable automatic deployment into production.

Many tasks are involved in each step of the code delivery process. Figure 4-3 shows the steps that are performed during continuous integration and continuous delivery (CI/CD).

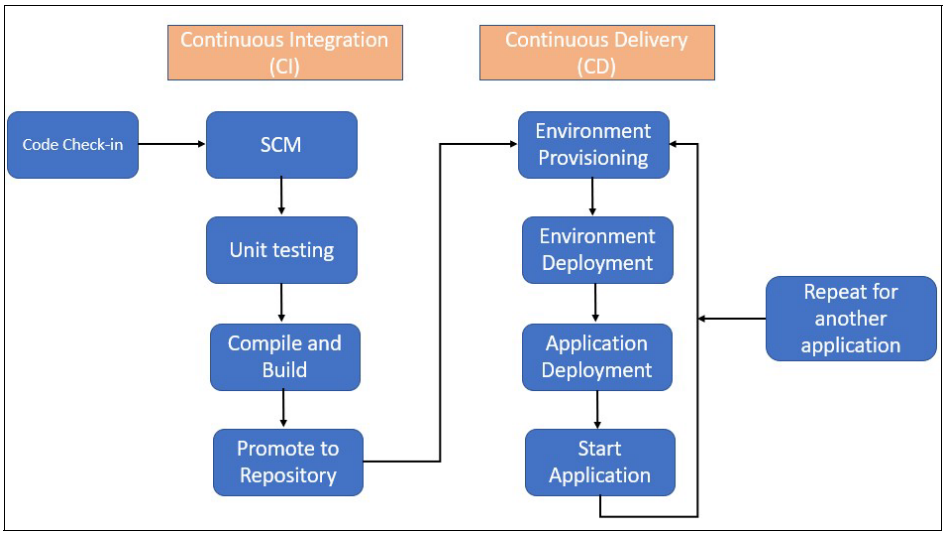

Figure 4-3 CI/CD flow chart

Fortunately, tools are available for each step of this process. We begin reviewing these tools by focusing on the beginning with planning. Though we highlight their capabilities individually, many of these tools are bundled to optimize their benefits.

4.3 Analyze and plan

Before you can safely alter an application, you must understand how the application is composed to get a sense of where changes must be made and how changes affect other components of the application. Because enterprise applications are complex, this process can take time and energy. Fortunately, the IBM Application Discovery and Delivery Intelligence (ADDI) tool is available that makes it much easier to gather these insights.

IBM Application Discovery and Delivery Intelligence

IBM ADDI analyzes applications that are designed for IBM Z to quickly discover interdependencies. It breaks down a complex application into its various pieces and represents the pieces in a way that is approachable for developers. Developers can then easily see where their changes must be made in source code, and the effects those changes will have on other areas of the application. By using this platform, you can identify business logic within the application, and determine the extent of its implementation.

For example, ADDI can break down a complex application that was written in COBOL into its various source code modules, and the CICS transactions that start each module. Any data files or database transactions also are shown, along with whether the operations are performed by a module are reads, writes, updates, or the like. Almost immediately, you can correlate transactions to COBOL code, to reads of VSAM data or updates of databases.

For more information about ADDI, see this web page.

4.4 Source code managers

To fully realize agile development with multiple developers working simultaneously, it is imperative to use a source code manager (SCM). An SCM supports parallel development where each developer can easily maintain their own code streams and merge their code when it is ready to be promoted to the next phase.

SCMs also provide backup and version control of source code, which creates a safety net if something goes wrong. Although some SCMs can work with binary files, most often SCMs are designed to work with text files because source code is stored as text files. We discuss two widely used SCMs next.

Git

The de facto standard for SCMs these days is Git; it is commonly used for cloud-native development and can be used equally for developing code on IBM Z. Git is an open source code manager with several popular client/server implementations, such as GitHub, Bitbucket, and GitLab.

For on-premises configurations, the Git server runs on Linux, which makes it perfect for Linux on Z, or on z/OS through z/OS Container Extensions as described in 3.1.4, “Colocation with IBM Z”. A Git client is available for z/OS from Rocket Software[1], which makes it easy to include Git-based CI/CD pipelines on z/OS. The real power lies in the mass integration of Git clients with seemingly every integrated development environment (IDE) you want to use. For more information about IDEs, see 4.5.2, “Integrated development environments ”.

IBM Engineering Workflow Management

IBM Engineering Workflow Management (EWM), formerly known as IBM Rational® Team Concert® (RTC), is a robust solution that is designed to act as a software development tool for your teams, regardless of on which platform the applications they are working. The SCM server component of EWM can run on z/OS, which might be convenient for z/OS-based applications. Also included in the solution are the tools agile development teams need to manage a backlog, track defects, and much more.

For more information about EWM, see this web page.

1 https://www.rocketsoftware.com/product-categories/mainframe/git-for-zos

4.5 Code

Several terms are used to describe the process of writing software around cloud, and it is important to understand the differences between them to choose the best offering for your business needs. Table 4-1 lists the different cloud services and strategies. It is followed by a deeper discussion into cloud-native.

Table 4-1 Cloud services and strategies

Cloud service/strategy	Definition
Cloud-native	Applications developed from the outset to operate in the cloud and take advantage of the characteristics of cloud architecture, or an application that was refactored and reconfigured to do so. Developers design cloud-native applications to be scalable, platform agnostic, and consisting of microservices.
Cloud-ready/ Cloud-enabled	An application that works in a cloud environment or a traditional application that was reconfigured for a cloud environment.
Cloud-based	A service or application being delivered over the internet “in the cloud”. It is a general term that is applied liberally to any number of cloud offerings.
Cloud-first	Cloud-first describes a business strategy in which organizations commit to using cloud resources first when starting new IT services, refreshing existing services, or replacing traditional technology.

4.5.1 Cloud-native

By using cloud-native applications, you can create an open ecosystem on IBM Z for access and use by administrators, developers, and architects, with no special skills required. With an open and connected environment, developers and administrators can more seamlessly build today’s business applications. These cloud-native applications can integrate with data and are optimally deployed across and managed within the hybrid multi-cloud.

Based on individual needs per workload (resources, time, cost, and so on), you have the choice to develop cloud-native applications in the private and public cloud, or a combination of both. Cloud-native development features the following attributes:

- Architecture: The architecture is microservice-based and the microservices run in dedicated containers.
- Automation: Everything is automated, including CI/CD, APIs, and configuration management.
- DevOps: Applications are driven by DevOps practices. The individuals that build the applications also run them; therefore, there is less of “throwing applications over the wall.”

A cloud-native application includes the following foundational elements:

- DevOps
- Continuous Delivery
- Microservices
- Containers

The deployment of applications is about empowering developers through supplying them with familiar tools and encouraging them to participate in an enterprise wide, fully automated, and continuous software delivery pipeline. Across the hybrid cloud ecosystem, IBM Z is designed to provide the flexibility to deploy workloads on- and off-premises, with the security, availability, and reliability you expect from Z.

4.5.2 Integrated development environments

IDEs are the digital home of your development teams. These suites of integrated tools allow developers to check out source code from an SCM, edit, and even run and debug their code in an intuitive experience. The goal is to enable developers to write code in parallel, receive rapid feedback on their work, deliver updates continuously, and maintain stable deployment environments.

IBM Developer for z/OS

IBM Developer for z/OS (IDz) is the premier IDE for z/OS. In addition to providing COBOL, PL/I, HLASM, Java, and C/C++ support, it includes a fully integrated debugger. IDz also provides the flexibility of editing style for new mainframe developers who might prefer the graphical flare, and experienced z/OS professionals who prefer command style.

Microsoft Visual Studio Code

Microsoft Visual Studio Code (VS Code) is a popular IDE that is built on open source. It is used for cloud-native development, and can be extended to work with z/OS by using the IBM Z Open Editor extension. This extension is at no cost and can be quickly added to VS Code with just a few clicks.

It provides support for IBM Enterprise COBOL, PL/I, HLASM, and JCL, including syntax highlighting, real-time error checking, code completion, and more features. The Zowe Explorer extension can also be added to VS Code to enable interaction with z/OS datasets, UNIX files, JES job output, and even MVS Commands.

This option is excellent for “hybrid” developers who are working across platforms and are familiar with VS Code from their projects on other platforms. VS Code is also widely used in school settings, which serves as a great way for an early professional who is new to IBM Z to quickly become productive.

IBM Wazi Code

IBM Wazi for Red Hat CodeReady Workspaces consists of two components: Code and Sandbox. Wazi Code enables developers who are working with z/OS applications a choice of VSCode or Eclipse-based IDEs.

Although you can obtain the Z Open Editor extension for VSCode for no cost, Wazi Code extends those capabilities to include debugging by using the IBM z/OS Debugger. This feature allows developers to perform visual debugging with variable inspection, breakpoints, and the like, in the same VSCode experience where they might choose to write their code.

This entire experience can be hosted in a Red Hat CodeReady Workspace, which provides the experience in-browser rather than locally installed. For more information about the Wazi Sandbox, see 4.7, “Test” on page 47.

4.6 Build

A build is a process in which source code is combined with any required libraries, and then is compiled, packaged, and made ready for deployment. This process might sound simple, but large applications can amount to millions of lines of code that might need hours to build.

Automated build is a repeatable build process that can be performed at any time and requires no human intervention. The build utility tools that are described in this section are designed to work with IBM Z.

IBM Dependency Based Build

IBM Dependency Based Build (DBB) provides the capabilities to build z/OS applications by using scripting languages that are commonly found in CI/CD pipelines. DBB APIs are written in Java and can be called by Java applications and Java-based scripting languages, such as Groovy and Maven. DBB includes an installation of Apache Groovy that was modified to run on z/OS UNIX System Services so you can run z/OS, TSO, or ISPF commands as part of the Groovy scripts.

With DBB, engineers can use Groovy scripts that were written for other platforms in the z/OS pipeline. DBB is ideal for compiling and link-editing programs. For this purpose, DBB provides a dependency scanner to analyze the relationship between source files, and a web application to store the dependency information and build reports.

DBB can also be used to add automated testing to the pipeline, or any other system administration task that you might write JCL for. DBB also works well with Git and Jenkins, as an example of an SCM and pipeline automation tool.

Figure 4-4 shows an example toolchain that Company A (from our example) is considering to implement DevOps principles on z/OS.

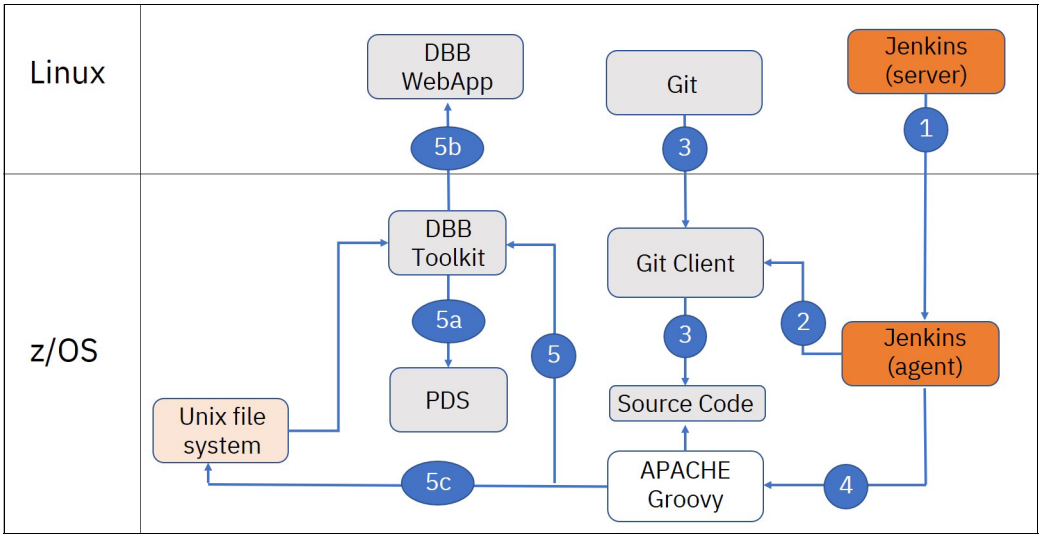

Figure 4-4 Git-based IBM Z open development open DevOps toolchain

Here, the application source code is stored in Git. DBB extracts the code from Git before it handles the compilation and generation of deployable artifacts, while Jenkins handles the pipeline and controls when DBB is run.

The following pathway is shown in Figure 4-4 on page 46:

1. The Jenkins server sends build commands to remote agent.
2. The Jenkins agent issues the Git pull command to update Git repository on z/OS.
3. The Git client automatically converts source from UTF-8 to EBCDIC during pull.
4. The Jenkins agent starts build scripts that contain DBB APIs to build code from the Git repository on z/OS.
5. The DBB Toolkit provides Java APIs to perform the following tasks:
   - Create datasets, copy source from z/OS file system (zFS) to Partitioned Dataset (PDS), start z/OS programs, and issue ISPF and TSO commands.
   - Scan and store dependency data from source files, perform dependency and impact analysis, and store build results.
   - Copy logs from PDS to zFS and generate build reports, which can be saved in the build result.

4.7 Test

Users who are new to DevOps sometimes think DevOps is about doing less testing because code moves more quickly through the development phases and into production. The reality is that DevOps might mean conducting more testing because every code change is tested.

The key is that this testing is automated. Performing every test manually on every code change is impossible; therefore, we rely on test automation to ensure quality into code delivery.

In an agile environment, a significant need exists to move testing earlier in the development lifecycle, which often is referred to as a “shift left.” This shift drives a need for isolated test environments, which might conflict with the availability of development and test systems.

Traditionally, these “dev/test” environments run alongside production systems on IBM Z hardware and they are shared by teams of people. Having a limited number of environments, especially if tests for one function cannot be performed when another function is installed, can pose a challenge to software delivery. A few solutions are available to address this challenge, including the following examples that are described next:

- IBM Z Development and Test Environment
- IBM Wazi Sandbox
- IBM Wazi Virtual Test Platform

IBM Z Development and Test Environment

IBM Z Development and Test Environment (ZD&T) allows any z/OS software to run on a x86-compatible, on-premises system, or cloud instance. This availability is possible through an emulation layer that translates the IBM Z instruction sets and devices.

Moving dev/test to x86 frees up capacity for production workloads simultaneously while allowing greater flexibility and availability for dev/test. ZD&T provides a dev/test platform for IBM z/OS middleware, such as CICS, IMS, Db2, and other z/OS software, to run on Intel-compatible platforms without the need for IBM Z hardware. Because the environment that is provided by ZD&T is emulated, performance is acceptable only for test and not production workloads.

IBM Wazi Sandbox

The other component of IBM Wazi for Red Hat CodeReady Workspaces, Wazi Sandbox, provides a containerized dev/test environment that is optimized to run on Red Hat OpenShift. This OpenShift support means you can run emulated z/OS for test purposes on your public cloud of choice.

As you might expect from a cloud-based solution, a dashboard provides your developers the ability to provision and deprovision their individual sandboxes as needed. Similar to ZD&T, performance for Wazi Sandbox is acceptable only for test and not production workloads.

IBM Wazi Virtual Test Platform

IBM Wazi Virtual Test Platform (VTP) provides the ability to conduct a full transaction level test without needing middleware. Therefore, you can perform integration testing during the build process, which is a huge step that is left in development. It provides full stubbing capability for the middleware, starting with CICS with calls to Db2, DL/I, IBM MQSeries®, and Batch (including Db2 and DL/I).

4.8 Provision, deploy, and release

Deploy is where tested code is moved into execution. The full power of DevOps comes together in this phase in the form of an automated pipeline. This pipeline facilitates the steady movement of code changes from source code repository to a test environment, where automated tests are routinely performed to validate the changes.

Deployment tools do more than perform orchestrated deployments, they also track which version is deployed at any stage of the build and delivery pipeline. They can also manage the configurations of the environments of all the stages to which the application components must be deployed.

IBM z/OS Cloud Broker

z/OS Cloud Broker provides access to z/OS services within private cloud platforms where they can be used by the development community. It promotes a modern cloud-native experience by combining z/OS-based services and resources with your hybrid cloud environments. Developers can quickly create, modernize, deploy, and manage applications within the security of their firewall, themselves, without the need for intervention from system administrators.

z/OS Cloud Broker includes the following key features:

- Integrate z/OS resources with Red Hat OpenShift platform and simultaneously maintaining control over how these resources are used.
- Use the experience and trust in existing IBM Z investments.

IBM Cloud Provisioning and Management for z/OS

The IBM Cloud Provisioning and Management (CP&M) tool is used to provision z/OS software subsystems rapidly increasing the agility of the DevOps team. It helps transform IT infrastructure by integrating z/OS into your hybrid cloud.

Software service templates that provision IBM middleware, such as CICS, Db2, IMS, IBM MQ, and WebSphere® Application Server can be created and tracked by using CP&M. It simplifies the provisioning and de-provisioning of an environment, as needed, through a self-service marketplace.

It is available through IBM z/OS Management Facility (z/OSMF) for tasks that fall under the cloud provisioning category, such as resource management and software services, through Representational State Transfer (REST) APIs. Applications can use these public APIs to work with system resources and extract data.

Figure 4-5 shows an overview of CP&M for z/OS as a solution.

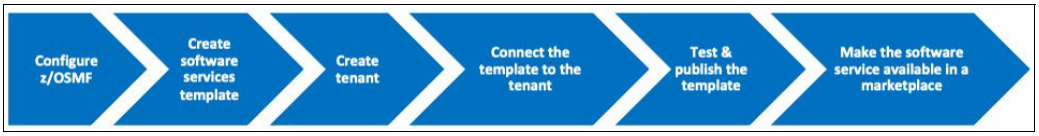

Figure 4-5 Cloud Provisioning and Management for z/OS workflow

For more information about CP&M, see this web page.

IBM UrbanCode Deploy

IBM UrbanCode® Deploy is a leading cross-platform deployment automation tool that is characterized as an application release automation solution with visibility, traceability, and auditing capabilities.

One of the many benefits of UrbanCode Deploy is its ability to help eliminate manual processes that are subject to human error, which in turn, leads to the enablement of continuous delivery for any combination of on-premises, cloud, and mainframe applications. The need for custom scripting is removed with UrbanCode Deploy by using tested integrations with many tools and technologies, such as Jenkins (which we discuss in the next section), Jira, Kubernetes, Microsoft, ServiceNow, and IBM WebSphere.

Quality checks are performed against every application before deployment. These checks are referred to as *deployment approval gates*. This process provides greater visibility and transparency into deployments for audit trails.

With UrbanCode® Deploy, you can set specific, required conditions that must be met before an application is promoted to an environment by establishing gates. These gates are defined at the environment level and each environment can have a solitary gate or conditional gates.

UrbanCode Deploy also aids in version control, which makes it easy to release only tested versions. It manages such control by providing application models and snapshots, where snapshots are manifests of versioned components and configuration and can be promoted as single items versus multiple components.

If you decide to use the IBM UrbanCode Deploy product suite, you receive access to Blueprint Designer. This component provides services, such as cloud orchestration, a graphical editor, IaC, and Cloud Automation Manager. These services in Blueprint Design collectively help to establish a CI/CD pipeline to generate and destroy short-term test environments to swiftly test application changes.

Some of the more critical benefits of UrbanCode Deploy are its ability to automate and increase the velocity of software deployment through different environments. It is designed to support the DevOps approach (a critical component of modernization), which enables the rapid release of incremental changes reliably and repeatedly.

You can use UrbanCode Deploy as a container and it is certified to work with Red Hat OpenShift and IBM Cloud Pak® for Applications. Support for the z/OS platform is available in the form of deploying it directly with the z/OS agent or through integration with existing z/OS deployment processes, such as IBM Rational® Team Concert® Enterprise Extensions.

It also includes build and test tools that are used to automate application deployment to mainframe production environments. Figure 4-6 shows the IBM UrbanCode for deployment automation flow.

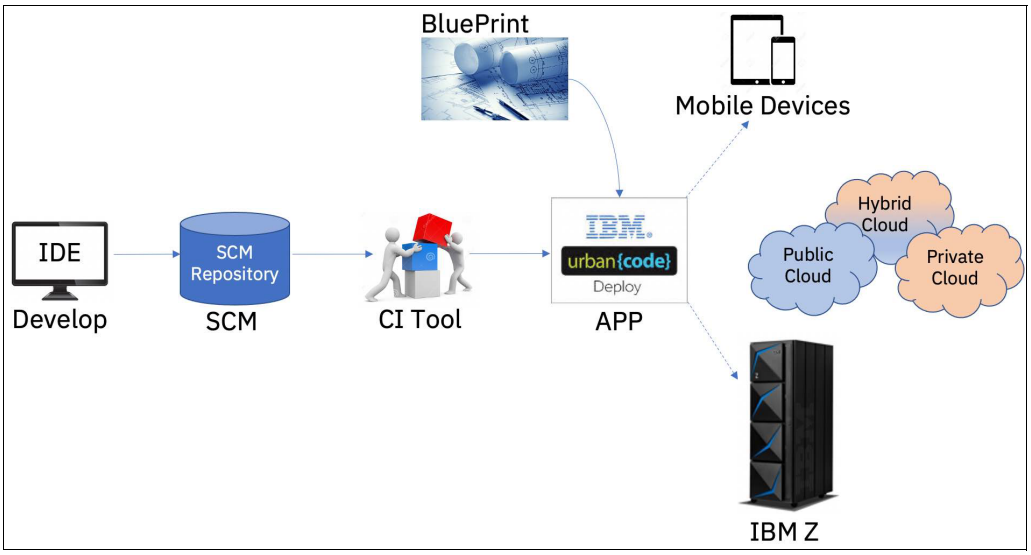

Figure 4-6 IBM UrbanCode Deploy

4.8.1 Continuous delivery with Jenkins and UrbanCode Deploy

UrbanCode also provides plug-ins to Jenkins continuous integration (CI). Jenkins CI server supports interactions with other technologies by using a plug-in model. Installed on a Jenkins server, the Jenkins Pipeline plug-in orchestrates UrbanCode Deploy deployments as part of a CI/CD pipeline in Jenkins (see Figure 4-7).

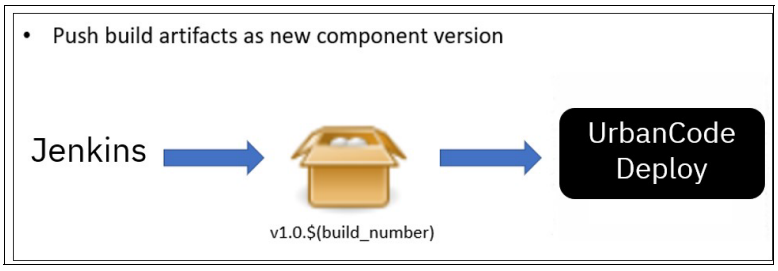

Figure 4-7 Jenkins plug-in for IBM UrbanCode Deploy

After running a build in your existing Jenkins CI infrastructure, the UrbanCode Deploy/Jenkins plug-in enables you to publish the build result to UrbanCode Deploy and trigger an application process for deployment.

Another aspect of deployment that is covered here is the deployment of IT Infra services. Red Hat Ansible Engine is the component within Ansible Automation Platform that uses hundreds of modules to automate all aspects of IT environments and processes. It helps developers and IT operations teams to quickly deploy IT services, applications, and environments to automate routine activities.

Red Hat Ansible

Ansible is an automation platform that uses a simple, English-like, widely used Open Source language that is called YAML for playbooks that automate application and IT infrastructure.

Ansible unites workflow orchestration with configuration management, provisioning, and application deployment in one easy-to-use platform. It uses code building blocks that are called *playbooks*, which are scripts with a group of commands or instructions and written in YAML to accomplish a task.

A significant step in enabling z/OS to participate in an Ansible-based enterprise automation strategy in the same way that the rest of your environments do, Red Hat Ansible Certified Content for IBM Z was made available to use Ansible on IBM Z. The use of Ansible to automate z/OS improves consistency across hybrid multi-cloud environments and allows z/OS to transparently participate in your infrastructure.

An initial collection of Ansible playbooks are designed to handle many tasks, such as working with datasets, retrieving job output, and submitting jobs on the system. More collections that are related to various use cases are being added to z/OS and the IBM Z broader community.

Implementing Ansible-based provisioning on z/OS can bring added value to your organization, especially so when you include Red Hat OpenShift, which was recently announced for Linux on IBM Z. With Ansible on IBM Z, you can seamlessly integrate workflow orchestration within a DevOps CI/CD pipeline of capabilities, including configuration management, provisioning, and application deployment in a single user-friendly platform, on any operating system.

Figure 4-8 shows a basic Ansible architecture.

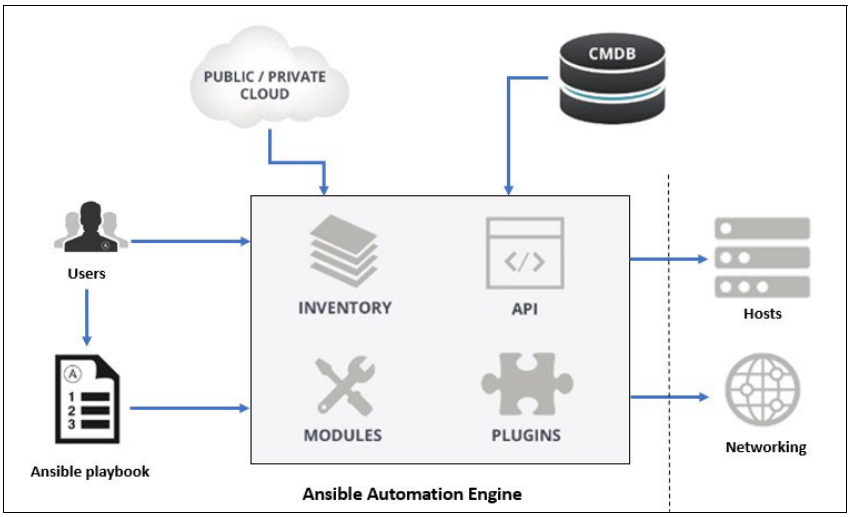

Figure 4-8 Ansible architecture

Red Hat Ansible Certified Content for IBM Z helps you connect IBM Z to your wider enterprise automation strategy through the Ansible Automation Platform ecosystem. The IBM z/OS core collection is part of this certified content that focuses on z/OS infrastructure deployment and management. This automation content enables you to start using Ansible Automation Platform with z/OS to unite workflow orchestration in one easy-to-use platform with configuration management, provisioning, and application deployment.

IBM z/OS core collection provides sample playbooks, modules, and plug-ins in the form of Ansible Content Collections that can help accelerate your use of Ansible against z/OS inventories.

The following z/OS core modules are used in the IBM z/OS core collection:

- `zos_job_submit`
- `zos_job_query`
- `zos_job_output`
- `zos_data_set`

For more information about how to use the playbook, see this web page.

5

Modernization best practices

Every organization has a unique experience navigating their modernization journey. However, some common best practices are available that can help any organization.

In this chapter, we discuss best practices for infrastructure, application, and process modernization with IBM Z.

This chapter includes the following topics:

5.1 Using a Source Code Manager

If one suggestion sits at the heart of application modernization, it is to use a full-featured Source Code Manager (SCM) to organize all of your source code. As development processes transform to support multiple developers simultaneously working in the same area of code, it becomes critical to use a tool to manage everything. It also airs the possibility that the existing tool that sufficed for single-stream development is no longer sufficient.

This issue becomes clear with teams who use library managers to promote their applications across environments. Library managers work well for writing applications in a single stream because their stages are tied to each environment between development and production. However, they do not handle multi-stream development well.

Instead of having a code “branch” representing a particular environment, modern SCMs use each branch for a specific feature or deliverable. Therefore, you can merge the branches together in different ways according to which features you need in a specific environment. Meanwhile, each branch in the SCM can be updated independently of the others.

Figure 5-1 shows these differences and emphasizes the increase in productivity that can be achieved by using a modern SCM-enabling, multi-stream development.

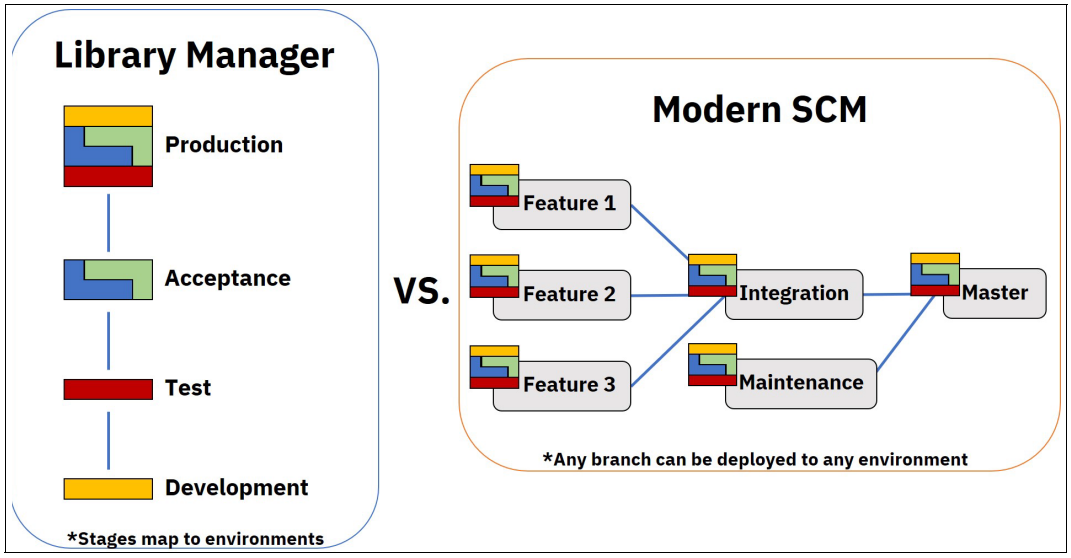

Figure 5-1 Library manager versus source code manager

One important detail to remember is that source code is not limited to application code only. Benefits can be realized by storing your test cases in an SCM, and especially in colocating those tests alongside the application code that they test.

This process also can be extended to include automation scripts and any other text files that are part of coding, building, or deploying an application. Having everything that you need in the same SCM enables CI/CD pipelines to include automated building, deploying, and testing for application changes.

5.2 Deploying common tools across platforms

The use of the same tools across platforms might include cost benefits if you can reuse something that you licensed. However, other reasons exist that might outweigh cost considerations.

Consistent tools across environments means that your teams can use shared skills and expand the number of people who might work in an environment. Historically, teams that work in one environment are unfamiliar with the tools that used in other environments and vice versa.

By using universal tools regardless of hardware platform, all teams become familiar with the same tools and might work in any environment. If the tools you use are common and used in a school curriculum, you are positioned for an even better advantage because of the immediate increase in productivity from new hires that join your ranks. All of these factors are beneficial to your team's velocity.

Common tools also allow for sharing digital artifacts across environments. For example, a common DevOps toolchain allows for the definition of one pipeline that performs some operations on IBM Z to deploy the back end, and some operations on public cloud to deploy the front end, with a centralized dashboard for reporting. In a general sense, scripts or common routines might be shared across environments, which increases shared skills and knowledge among teams and in effect boost productivity.

5.3 Embracing new methods and means

Numerous examples exist of how technology changed our lives; for example, the use of GPS instead of reading a paper map or finding directions online, or searching for a phone number online instead of the use of a phone book. Similar stories also exist regarding code writing or performing system operations.

It is common for developers to use a GUI-based editor when working on their code. Thus, it makes sense that we want to enable their use for application code that runs on IBM Z. This enablement might be a change for experienced programmers, and some tasks might be easier to complete without the GUI. However, the focus must be on enabling as many people as possible to increase productivity.

A similar story is true of operations. Experienced system programmers likely have their own customized sets of jobs, scripts, and so on, to perform daily tasks. Because they understand the details of what they are doing, they can quickly respond to requests. Those programmers with less experience and smaller libraries of scripts might be more productive if they can use a standard workflow to accomplish their task requests.

Modernization means doing things differently. The emphasis must be on doing things that enable future productivity, which means offering IDEs for coding and cloud provisioning and management for system programmer actions.

For more information, see 4.8, "Provision, deploy, and release" on page 48.

5.4 Allowing self-provisioned testing

Because environments that are hosted on IBM Z are expected to always be available, changes to these production systems are tightly managed to control the risk of breaking production. That expectation is something that is not likely to change with modernization. In fact, modernization might increase the importance of these already-crucial systems because they extend across an organization's hybrid cloud.

Multiple phases of testing are one of the components that contributes to this stability because nothing is promoted into production before it is thoroughly vetted for quality. The test environments that are used for this vetting must still be controlled by skilled system programmers, but an opportunity might exist to allow greater flexibility because the effect of downtime is not as critical as it is in production. Empowering developers by allowing them to self-provision temporary resources for their application testing can greatly speed up development.

For example, one of Company A's COBOL developers is changing an application that is hosted in CICS on IBM Z. In past configurations, one shared CICS environment was used for multiple testers, which meant that testers had to coordinate who can use it at a specific time so that no one affected anyone else's testing.

Through modernization with CP&M, Company A enabled developers who are not skilled as system programmers to easily provision a private, temporary CICS region to use for their testing and then deprovision it when they are done.

5.5 Consider emulated environments for early testing

A great way to scale and extend the scope of the self-provisioned model is to emulate the entire environment. Whereas allowing self-provisioned testing enables self-provisioning resources within a shared system, such as a z/OS LPAR, emulated environments allow developers to have a private system. These environments are emulated because they run on x86 hardware (on-premises or in-cloud).

Because developers get their own emulated z/OS environment, they have much more flexibility for customization as needed for their testing. Better yet, each developer can have their own environment, which means that developers do affect each other with their customizations or if their testing causes failures. This model gives the benefits of each developer having their own z/OS system without affecting IBM Z hardware usage.

At first, Company A relied on a shared z/OS LPAR for their developers to unit test their application changes. Eventually, they outgrew this model and had many developers working on changes that must be tested independently of one another.

At the time, commodity Linux servers were sitting unused in their data center, so they installed IBM Z Development and Test (ZD&T) and turned their developers loose to work in parallel, which increased productivity. Company A recently chosen Red Hat OpenShift as their container platform, and they are considering moving these emulated environments to IBM Wazi for CodeReady Workspaces.

For more information about ZD&T and Wazi, see 4.7, "Test" on page 47.

5.6 Updating older COBOL applications

IBM Z processors are continually advancing to include new instructions that are designed to perform specific operations at a faster rate. Unfortunately, compiled applications cannot use these new instructions until they are updated by something or someone that is aware of the new instructions.

Traditionally, this update meant recompiling application source code with an updated compiler; the newer compiler knows about the new instructions, so it uses them when it translates source code to machine instructions. For COBOL applications, another alternative is available: IBM Automatic Binary Optimizer for z/OS (ABO).

IBM ABO takes the executables that you use and optimizes them with the latest machine instructions. Even if you lack the source code for some of your applications, you can still optimize them with ABO. Although it is still necessary to test the optimized executables before deploying to production, you can be assured that no source code was changed and less required testing is needed than you typically associate with application source changes.

COBOL applications that are changed often are recompiled with each change, so they are more likely to benefit from compiler improvements. ABO is a great approach for those applications that are not changed frequently or cannot be changed frequently because of lost source code. These applications are no longer relegated to old machine instructions.

For more information about IBM Automatic Binary Optimizer for z/OS, see this web page.

5.7 Performing regular assessments of new features and functions

Hardware and software updates typically are planned with the cadence of IT budgets, so chances are good that your organization stays within supported releases. Although staying within supported releases is helpful to ensure that vendors act if something fails, it does not mean you use everything that is accompanied by your updates.

A best practice is to add an assessment step to each update that focuses on new capabilities. These capabilities can be features of new IBM Z hardware, such as compression cards, or it can be functions that come with upgrading a middleware subsystem, such as a new REST-enablement capability.

It is important for the assessment to identify any new capabilities and which teams must be contacted for awareness. One all-to-common scenario occurs when system programmers are aware of new hardware features, but database administrators are not aware. If the database administrators knew about the new hardware, they update their subsystems to use it.

Consider the following questions for an update assessment:

- What new features or capabilities are included with the update?
- What is the benefit of implementing the new features?

  If the benefit does not warrant implementation, nothing needs to be done.

The following questions assume that the benefits warrant implementation:

- Does a specific action exist that must be followed to enable the new features?
  - If new features are enabled by default, interested teams must be made aware of the update so they can watch for any potential fallout.
  - If action is required, the assessment must lead to a follow-on exercise to create a plan for implementation.
- Does a specific driver exist for the update, or is it a regularly scheduled update?

  If a specific team is requesting the update, often they are notified when it is complete. However, the assessment must consider whether related teams exist who might benefit from awareness.
- Which subsystems can benefit from the new feature?

  Knowing this information helps identify which teams must be made aware of the update.

5.8 Watching for latency

Developers might write code without worrying about hardware dependencies and some low level of integration, and architects might ignore topography concerns by using the cloud computing model. One pitfall of embracing cloud is to fall into the trap of thinking that you do not need to consider how things fit together because everything works “in the cloud.”

When running production across hybrid platforms, the latency between the platforms must be a point of concern and planning. Even for (or perhaps especially for) batch transactions that occur after the close of business hours, adding a fraction of a second can mean that work is not completed before the start of the next business day. This issue is seen even for on-premises computing that starts on IBM Z, involves calculations on a server rack in the same data center, and then ultimately finishes back on Z. The latency is increased by cloud because of network paths between systems.

For more information about the benefits of colocating work, see 3.1.4, “Colocation with IBM Z” on page 30. Consider the latency introduced between platforms as you plan modernization.

5.9 Making your talent future-proof

Modern environments that are supported by agile processes and tools provide a great foundation for IT that enables organizations to grow, although it is the people behind these environments who are its greatest assets. One way to future-proof your teams is to recruit talent that has enterprise computing skills.

The IBM Academic Initiative schools foster and progress education that is related to enterprise computing. Across the high school and university levels, schools with course curricula that include IBM Z also feature computer science societies and other clubs, and everything in between. IBM often partners with clients to create an event with students from their geographic area. This endeavor is a worthwhile event for all who participate because students need jobs, clients need new hires, and IBM wants both to succeed.

Another opportunity to find enterprise computing talent is through a program that is called Master the Mainframe. It is a coding competition that teaches enterprise computing concepts with the look and feel with which students are familiar.

The competition consists of three levels, with a digital badge that is awarded to participants who completed several hands-on exercises. Master the Mainframe is creating generations of skilled developers and system programmers; all you need to look for is their digital badge.

For more information about Master the Mainframe, see this website.

You also can contact the IBM Z Ecosystem team at: zskills@us.ibm.com.

5.10 Getting help from experts

Many software vendors offer one or more collaborative methods for designing a modernization journey. Even better, some of these offerings are available at no cost. Regardless of the cost, it is worth bringing in subject matter experts to get a sense of what solutions other organizations used for their modernization journeys to be successful.

IBM Garage

IBM uses the garage method for collaborating with clients to examine a challenge, design a solution, and foster it into production. The process starts with a Design Thinking session to disassemble the problem, brainstorm ideas, and test concepts.

Afterward, IBM Garage™ experts work to create a minimum viable product (MVP) with their counterparts on the client's team. The process can help accelerate value and reduce risk, in addition to providing guidance to deliver immediate business value.

The IBM Garage™ experience blends business strategy, design, and technology into a single end-to-end journey. The IBM Garage methodology is built around co-creating, co-executing, and cooperating a solution by applying technology to a business problem.

For more information about IBM Garage, see this web page.

Developer Advocates

If you want to hear technical information about a technology from experienced technical users, IBM Developer advocate teams might be what you are looking for.

Developer Advocates present on technical topics that are based on personal experience, without any sales pressure. If technically feasible, Developer Advocates can even hold hands-on workshops so you can get practical experience without affecting your own systems.

For more information about offerings from the IBM Developer Advocacy teams for IBM Z or IBM Cloud, contact your IBM account team.

IBM Developer

IBM Developer is a community where you can find a plethora of technical learning materials and engage with other professionals through blogs and events. Great technical content is available on IBM Developer, including step-by-step tutorials and code patterns that include some sample code that is published under open source guidelines to get you started. Podcasts where SMEs discuss technical topics in detail and videos that include demos also are available.

IBM Developer is a great resource for technical topics about application modernization and modern tools for operations. For more information about IBM Developer, see this web page.

By now, the picture is a bit clearer about how your IBM Z environments can play a role in your hybrid cloud.

In this chapter, we explored some of the ways to get started and discussed some of the technologies and products that can help you modernize.

Putting it all together

In this publication, we discussed what modernization is, how to get started on your modernization journey, some technologies that enable modernization, and best practices to consider for your journey toward modernization with IBM Z.

In this chapter, we introduce a technical showcase that exemplifies modernization.

This chapter includes the following topics:

6.1 What is a technical showcase?

A technical showcase is a collection of demonstrations that is built into a single environment. The showcase we highlight was built by the IBM Garage for Systems team to demonstrate technical capabilities across an intertwined topology of use cases.

This showcase is built on actual use-cases and is flexible so that it can be customized to better represent more real-world systems.

For more information about engaging with the technical showcase team, see this web page.

6.2 Topology overview

Our technical showcase is built with some elements that are running on Linux, others that are running on z/OS, and still others in IBM Cloud. Thus, some elements are deployed on-premises and others are deployed in the public cloud.

Figure 6-1 shows a visualization of the topology of this showcase, which is a stock trading application.

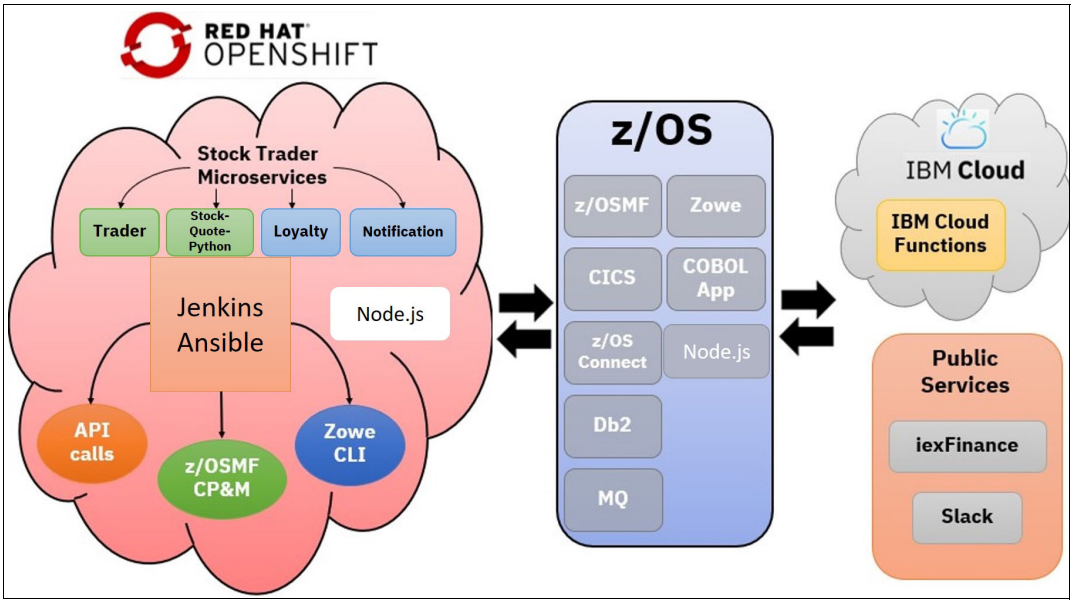

Figure 6-1 Technical showcase topology

The portion of our solution that is running on Linux is shown on the left side of Figure 6-1, where containers are running on Red Hat OpenShift. It is in this are where we use the open source container-orchestration system, Kubernetes. Kubernetes provides many useful functions, such as load-balancing, high availability, and scalability.

Within this environment, we deploy Jenkins as our orchestrator, through which we organize and schedule work to be performed. This deployment allows us to use the flexibility and high availability of OpenShift for our applications and our DevOps tools.

Jenkins uses the Ansible plug-in to run Ansible playbooks that we wrote to deploy our applications and middleware, primarily making use of APIs to perform actions in z/OS.

We can take APIs, plug them into Ansible, and Ansible orchestrates the API calls for us. Ansible and Jenkins contain most of our high-level automation logic. Also, we use the following technologies to interact directly with z/OS and its middleware:

- Zowe CLI
- z/OSMF Cloud Provisioning and Management
- Direct API calls against the middleware products

A portion of our private cloud is shown in the center of Figure 6-1 on page 62, where we have our z/OS system. Our middleware stack on Z consists of Customer Information Control System (CICS), Db2, IBM MQ, and z/OS Connect, all of which are deployed dynamically from templates that are defined in Cloud Provisioning and Management (CP&M). We also have Zowe, Node.js, and our COBOL application in our z/OS system.

Lastly, in the third environment that is shown on the far right side of Figure 6-1 on page 62, we have some public API services that are hosted in IBM Cloud. These services work in tandem with the stock trading application in our private cloud to provide certain functions.

We use the following public services:

- iexFinance: Used to retrieve real-time stock price data.
- IBM Cloud Functions: Used to notify a user of a change in their loyalty status, which occurs as certain portfolio thresholds are met. These messages can be sent to many different communication platforms, such as Slack and SMS messages.

6.2.1 Showcase technologies

This section briefly describes the key technologies that are used in this technical showcase and how they fit into application modernization with IBM Z. In the subsequent sections, they are discussed in greater detail.

The following technologies are used on z/OS in this showcase:

- Db2**:** A relational database that delivers advanced data management and analytics capabilities for transactional workloads.
- CICS: A general-purpose transaction processing subsystem for the z/OS operating system that manages sharing resources, the integrity of data, and prioritizing execution, with fast response.
- z/OS Connect: A Liberty feature that encapsulates calling z/OS target applications by using Representation State Transfer (REST) calls.
- IBM MQ: Provides a communications layer for visibility and control of the flow of messages and data inside and outside your organization.
- z/OSMF: Provides system management functions in a task-oriented, web browser-based user interface (UI) with integrated user assistance so that you can more easily manage the daily operations and administration of your z/OS systems. It also provides APIs for most of the manual tasks you can perform in the UI, which allows submitting work to z/OS and deploying middleware programmatically.
- Zowe: A technology from the Open Mainframe Project of the Linux Foundation that delivers new ways to integrate z/OS-based services into the enterprise.

Some of the technologies we implement in other pieces of our showcase include the following examples:

- Jenkins: An open source, server-based tool that builds and tests software continuously. It supports continuous integration and continuous delivery.
- Ansible: An open source software provisioning, configuration management, and application deployment tool.
- Red Hat OpenShift: A fully managed Kubernetes service that can be run at the enterprise scale and security of IBM Cloud, or on-premises on your choice of hardware, including IBM Z.
- IBM Cloud: The most open and secure public cloud for business, it is a next-generation hybrid multi-cloud and private cloud platform that features advanced data and AI capabilities and deep enterprise expertise.
- Docker: An open source platform for building, deploying, and managing containerized applications.

6.3 ISTPOMA: Stock trader application

Our stock trading application is name ISTPOMA. It is written in COBOL and run in a CICS region on z/OS. To fit our modern infrastructure, the stock trading application's core functions were exposed through z/OS Connect. After becoming exposed, these endpoints can be used by the containerized microservices that are used by our front-end application.

In our case, we have calls from the OpenShift environment going to our z/OS environment, ultimately reaching ISTPOMA through z/OS Connect in CICS.

A user can interact with the stock trader application in two ways: directly through REST APIs, or indirectly through the web-based UI.

Figure 6-2 shows the APIs that are exposed with z/OS Connect and then redirected to ISTPOMA on z/OS. The web-based UI also uses these APIs.

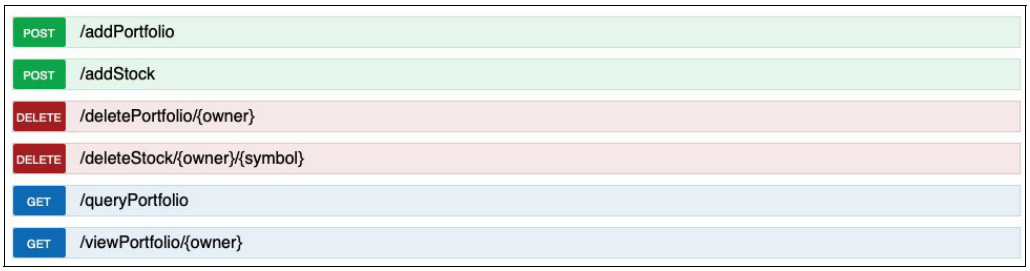

Figure 6-2 APIs on z/OS Connect in Swagger

When a user interacts with the stock trader application, six actions are available, as listed Table 6-1.

Table 6-1 Stock Trader application user actions

Command	Action
`/addPortfolio`	Add a stock trading portfolio to allow for further portfolio development.
`/addStock`	Add shares of a stock to a stock trading portfolio that was created in a previous action.
`/deletePortfolio/{owner}`	Delete a stock trading portfolio that is qualified by the owner of that portfolio.
`/deleteStock/{owner}/{symbol}`	Delete a stock from a stock trading portfolio that is qualified by the owner of that portfolio, and the stock you are deleting from that portfolio.
`/queryPortfolio`	Query the list of all stock trading portfolios that were added by users to view all loyalty and stock information.
`/viewPortfolio/{owner}`	View the stock trading portfolio of the owner that is specified to view loyalty and stock information.

Microservices are a method for development teams to modernize their COBOL programs and applications. Creating microservices allows development teams to use their applications to create a collection of API-enabled microservices that split the application up into logical parts. These microservices still drive the same core business logic through their calls through z/OS Connect while enabling more features and logic necessary for complex front-end applications without changing our COBOL code.

We created the following microservices out of our ISTPOMA COBOL application, which we chose to deploy within Docker containers:

- Stock-Quote-Python: A microservice that uses iexFinance to get current stock information.
- Loyalty: A microservice that receives notifications about new loyalty level updates and posts them to IBM MQ on z/OS. Loyalty level updates are determined by the value of the stock that the user adds to their portfolio. The following thresholds for the loyalty levels are used:
  - Loyalty level default: “Basic”
  - Loyalty level “Bronze”: If total is greater than $10,000.00
  - Loyalty level “Silver”: If total is greater than $50,000.00
  - Loyalty level “Gold”: If total is greater than $100,000.00
  - Loyalty level “Platinum”: If total is greater than $1,000,000.00)
- Notification: A microservice that watches for new messages on IBM MQ and sends information to IBM Cloud Functions to perform more notification processing, such as updating Slack or sending SMS messages. This service is the most optional service in this stack; no other microservices rely on this setup to function.
- Trader: A front-end microservice that bridges together all other services.

6.4 Jenkins

In the context of this showcase, Jenkins holds the role of orchestrator. In this role, Jenkins organizes and schedules when the automation must be performed but does not contain the logic to perform the automation.

Because most of our automation logic lives in Ansible playbooks, it is easy to switch to another orchestration platform if necessary.

In this technical showcase, Jenkins is running in a Kubernetes container within our Red Hat OpenShift environment. Jenkins performs tasks that are found in what are known as *pipelines*. In most cases, the tasks that are defined in our Jenkins pipelines specify different Ansible playbooks to run. These Ansible playbooks contain the logic to provision and configure the different parts of our application.

When a Jenkins build is triggered, the playbooks run and report their status back to Jenkins, which indicates the success or failure of the playbook.

For more information about Jenkins, see the Jenkins User Documentation web page.

6.5 Ansible

Ansible is an automation engine that automates provisioning, configuration management, application deployment, intra-service orchestration, and many other IT requirements.

Designed for multitier deployments, Ansible models your IT infrastructure by describing how all your systems inter-relate, rather than managing only one system at a time. Because it uses no agents and no other custom security infrastructure, it is easy to deploy, and most importantly, it uses a simple language (YAML) in the form of files that are called Ansible playbooks. these playbooks allow you to describe your automation jobs in a way that resembles plain English.

Ansible is extensible by using modules. Modules are discrete units of code that accomplish a specific task. Some common modules include the ability to perform regular expression queries or to update a systems' packages. During the automated configuration and deconfiguration of CICS, IBM DB2®, IBM MQ, and z/OS Connect, we make heavy use of the URI module, which calls REST APIs.

We use Jenkins to orchestrate the Ansible playbooks in this showcase. Jenkins allows us to trigger our automation in various ways, ranging from a change in a GitHub repository to a message that is sent to a Slack bot.

When a build is started in Jenkins, the pipelines run a series of Ansible playbooks, which use APIs that are running in our z/OS system to configure our environment. These playbooks start by provisioning our middleware, a process that is started through a z/OSMF CP&M API.

Subsequent playbooks call middleware APIs directly, such as the CICS CMCI APIs, which we use to dynamically define CICS constructs, such as Db2 connections and programs (for example, our COBOL program). Driving these APIs through Ansible is advantageous because it provides a common framework across our z/Architecture.

For more information about the use of Ansible with IBM Z, see this web page.

6.6 z/OS Management Facility

IBM z/OS Management Facility (z/OSMF) provides system management functions in a task-oriented, web browser-based UI with integrated user assistance. This tool allows you to more easily manage the daily operations and administration of your mainframe z/OS systems.

By streamlining some traditional tasks and automating others, z/OSMF can help to simplify some areas of z/OS system management. It also provides a framework for managing various aspects of a z/OS system through a web browser interface. This feature in turn allows you to access and manage your z/OS system from anywhere.

Multiple users can log in to z/OSMF by using different computers or browsers, or multiple instances of the same browser. All of the templates for z/OSMF CP&M are loaded into z/OSMF and then API calls are made to z/OSMF to begin starting the workflows and the templates.

z/OSMF provides you with a single point of control for the following tasks:

- Viewing, defining, and updating policies that affect system behavior.
- Monitoring the performance of the systems in your enterprise.
- Managing software that runs on z/OS.
- Performing problem data management tasks.
- Consolidating your z/OS management tools.

For more information about z/OSMF, see IBM Knowledge Center.

Note: z/OSMF is the only piece that must be configured on z/OS before any automation is run because of the necessity for specific APIs and CP&M of Software Services.

6.6.1 Cloud Provisioning and Management overview

CP&M is a suite of z/OSMF plug-ins that enables you to rapidly deploy z/OS middleware, such as Db2, IBM MQ, CICS, and z/OS Connect. After a template is created for any subsystem, you can deploy as many instances as your system can handle through a single click or API call.

To provision the middleware, you define templates in Software Services. Templates are provided by IBM for all the z/OS middleware we provision in the showcase, including CICS, Db2, IBM MQ, and z/OS Connect. After these templates are defined, you can provision unique instances of the middleware automatically, which creates all necessary data sets, startup procs, and file systems to run the middleware.

With the addition of the Network Configuration Assistant, all network resources that are required for the middleware instance also are allocated dynamically.

6.6.2 Workflows overview

Workflows are the basis of templates and define what work is to be completed and how that work ought to be completed. Workflows can use many z/OS functions, such as submitting JCL, issuing consoles commands, and using REST APIs.

Typically, a CP&M template at least includes a workflow for provisioning and deprovisioning. Each step of a workflow can be browsed as it runs and allows you to see the output of JCL, shell scripts, and REST responses. Workflows are easily editable if they must be custom tailored to your environment by the Workflow Editor plug-in.

6.6.3 Provisioning middleware with CP&M

CP&M is the primary method that we use to provision our middleware. As part of this provisioning process, we can perform some basic configuration, such as creating a default queue in IBM MQ, including our pre-built services for z/OS Connect, or enabling CICS features and definitions.

After our templates are defined and tested, we use the CP&M APIs to drive provisioning from Ansible by using the URI module. While the middleware instance provisions, we query the instance for status until the provisioning is complete. This process allows us to start a Db2, CICS, IBM MQ, or z/OS Connect instance through a simple API call.

6.7 Zowe

Zowe is an open source project that was created to host technologies that benefit the Z platform from all members of the Z community (Integrated Software Vendors, System Integrators, and z/OS consumers). Zowe comes with a set of APIs and operating system capabilities that applications build on, and includes some ready-for-use applications.

Zowe offers interfaces to interact with z/OS in a way that resembles what you expect of other cloud platforms. You can use these interfaces as delivered or through plug-ins and extensions that are created by clients or third-party vendors.

For this showcase, Zowe hosts APIs behind a single port, behind the API mediation layer gateway, and allows for easier interaction with our APIs on the z/OS system.

6.7.1 Zowe CLI overview

Zowe CLI provides a command-line interface with which you can interact with z/OS by using common tools, such as IDEs, shell commands, bash scripts, or build tools. The CLI provides a core set of commands for working with data sets, UNIX System Services, JES, and issuing TSO and console commands.

6.7.2 Zowe CLI in a Docker container

For the technical showcase, we run the Zowe CLI in a Docker container along with Node.js. Zowe CLI is one of the three methods that performs our actions and calls on z/OS through Jenkins and Ansible.

By using Zowe CLI, we can take Zowe commands, translate them into API calls against our z/OS system, and plug them into bash, python, or Java scripts to complete some of the complex functions on z/OS. Examples of these complex functions are creating or manipulating a file, submitting a job, and API calls against plug-ins for middleware.

For more information about Zowe CLI, see this website.

6.8 Db2

IBM Db2 is a family of hybrid data management products that offers a complete suite of AI-empowered capabilities to help you manage structured and unstructured data that is on-premises and in private and public cloud environments. Db2 is built on an intelligent common SQL engine that is designed for scalability and flexibility. It drives high-impact data insights, seamless business continuity, and real business transformation.

Db2 for z/OS is an industrial-strength database system that is known for its reliability, security, performance, and recoverability. Key components for recoverability of Db2 and its user databases are dual BSDS, active log, and archive log datasets, along with a powerful suite of program utilities to support performance, backup, and recovery.

Db2 supports many languages and supports remote application access. Db2's key to availability is the use of data sharing technology to allow 24x7 access in the course of maintaining system and database updates.

6.8.1 Db2 as our database

Db2 serves as our database for the technical showcase. It is where the data for the stock trading application is stored. When transactions come into CICS, that data is written to or removed from our tables in Db2. This data includes names of portfolios, stocks in portfolio, and the quantity of stocks in portfolios.

For more information about Db2, see this web page.

6.9 Customer Information Control System

CICS is a general-purpose transaction processing subsystem for the z/OS operating system. It provides services for running an application online, by request, at the same time as many other users are submitting requests to run the same applications by using the same files and programs.

CICS manages sharing resources, the integrity of data, and prioritizing the execution with fast response. CICS authorizes users, allocates resources (real storage and cycles), and passes on database requests by the application to the suitable database manager, such as Db2. We might say that CICS acts like and performs many of the same functions as the z/OS operating system.

6.9.1 CICS as transactional manager

CICS serves as the transaction manager for our stock trading application. When a user adds stocks, deletes stocks, creates a portfolio, and so on, that process drives the business logic that is provided by our COBOL application that is running in CICS.

To interact with this application, our front-end stock trading application uses APIs that are provided by z/OS Connect that make these services available. When z/OS Connect receives these requests, it converts our JSON payload to a COBOL copybook and passes the transaction to CICS to be processed. When that process is successful, the transaction is hardened in our Db2 database.

For more information about CICS, see IBM Knowledge Center.

6.10 z/OS Connect

IBM z/OS Connect facilitates calling z/OS target applications by using REST calls. It provides a standard way to identify assets that are on z/OS and reach these assets by using REST.

z/OS Connect includes the following features:

- Expose your applications and data through RESTful APIs or enhance existing applications with the ability to call external APIs.
- Point and click API mapping; no code changes are required.
- Expose your assets without modifying the underlying programs. An intuitive visual editor makes creating APIs as easy as point and click.

6.10.1 Interacting with COBOL application

z/OS Connect provides us a way to interact with our COBOL application running in CICS. We defined services in z/OS Connect that convert JSON payloads to COBOL copybooks, which allows us to interact with our COBOL application through REST APIs. This communication is achieved by defining an IPIC connection in CICS, which allows z/OS Connect to relay the payloads it receives from our microservices.

CICS processes the incoming transactions and forwards the payload to our COBOL application in copybook form. Our COBOL application processes the copybook and makes updates to, or retrieves the relevant data from, our Db2 database.

A payload is then passed back to z/OS Connect and converted back into JSON before being returned to our application. z/OS Connect also allows us to omit or transform fields before they are returned to our front-end application, letting us remove fields that might not be relevant to the application. This capability is powerful because it allows us to give our developers a way to interact with our traditional application in a way that is familiar to them, regardless of their experience with z/OS.

For more information about z/OS Connect Enterprise Edition, see this web page.

6.11 IBM MQ

IBM MQ supports the exchange of information between applications, systems, services, and files by sending and receiving message data through messaging queues. This feature simplifies the creation and maintenance of business applications. IBM MQ provides the following benefits:

- Versatile messaging integration across platforms that provides a single, robust messaging backbone for dynamic heterogeneous environments.
- Message delivery with security-rich features that produce results that can be audited.
- Qualities of service that provide once and only once, and delivery of messages to ensure that messages will withstand application and system outages.
- High-performance message transport to deliver data with improved speed and reliability.
- Highly available and scalable architectures to support an application's needs.
- Administrative features that simplify messaging management and reduce time that is spent by using complex tools.
- Open standards development tools that support extensibility and business growth.

6.11.1 IBM MQ as messaging queue

In the context of this showcase, IBM MQ serves as our messaging queue for loyalty status updates. When the user reaches the threshold for the next level loyalty status, those notification messages are sent out to the queue, go to IBM Cloud Functions, and then are sent to the endpoint for which we are looking. That endpoint can take the form of a Slack message, SMS message, email, and so on. We use CP&M to provision an IBM MQ instance and to create the queue we use in our application.

For more information about IBM MQ, see IBM Knowledge Center.

6.12 Summary

Created by the IBM Garage for Systems team in IBM Poughkeepsie, NY, US, this technical showcase was a culmination of several different technologies to demonstrate one method to automate deployments across the z/Architecture.

We used Ansible for most of our automation to deploy our entire application stack. This process included a few different steps in multiple parts of the Z architecture.

First, we needed to look at automation that focused on z/OS to deploy and configure our middleware. After that process completed, we then focused on compiling our COBOL code to install to our recently created CICS region.

Finally, we used Red Hat OpenShift to build and run our microservices to drive our business logic on z/OS.

Along this journey, we constantly passed dynamically created information to get the solution interconnected. The use of Ansible was ideal because we configured all of our systems with a single language, despite having large architectural differences. This common framework made it easy to tie everything together to create our fully automated solution.

In our case, we tied all of our different automation together with Jenkins. However, because of the modularity of Ansible, this task can easily be accomplished by using any other type of orchestrator.

This showcase demonstrated how IBM Z can play a key role in your hybrid cloud.

To learn more about how the technical showcase works, see the zModernization Technical Showcase Presentation YouTube video.

Related publications

The publications that are listed in this section are considered particularly suitable for a more detailed discussion of the topics that are covered in this paper.

IBM Redbooks

The following IBM Redbooks publications provide more information about the topic in this document. Note that some publications referenced in this list might be available in softcopy only.

- *Modernizing Applications with IBM CICS*, REDP-5628
- *Accelerating Modernization with Agile Integration*, SG24-8452
- *IBM Z Integration Guide for Hybrid Cloud*, REDP-5319
- *Getting Started with Z/OS Container Extensions and Docker*, SG24-8457
- *Mainframe Modernization and Skills: The Myth and the Reality*, REDP-5115
- *IBM z/OS Container Extensions (zCX) Use Cases*, SG24-8471
- *Cloud Workloads on the Mainframe*, REDP-5108
- *Batch Modernization on z/OS*, SG24-7779
- *Red Hat OpenShift on IBM Z Installation Guide*, REDP-5605
- *IBM Storage for Red Hat OpenShift Blueprint*, REDP-5565

You can search for, view, download or order these documents and other Redbooks, Redpapers, web docs, draft and additional materials, at the following website:

ibm.com/redbooks

Other publications

The following publications are also relevant as further information sources:

- IBM Garage Method for Cloud Field Guide:

  https://www.ibm.com/cloud/architecture/files/ibm-garage-method-for-cloud-field-guide.pdf

- IDC Whitepaper: The Business Value of the Transformative Mainframe:

  https://www.ibm.com/downloads/cas/O6Q76WXV

- IDC Whitepaper: The Business Value of the Connected Mainframe for Digital Transformation:

  https://docs.broadcom.com/doc/idc-the-business-value-of-connected-mainframe-for-digital-transformation

- MIT Tech Review:

  https://wp.technologyreview.com/wp-content/uploads/2020/09/Mainframe-2020-A-catalyst-for-transformation_0920.pdf

Help from IBM

IBM Support and downloads

ibm.com/support

IBM Global Services

ibm.com/services

REDP-5627-00

ISBN 0738459534

Printed in U.S.A.