

CICS Tools Update and Open-Source Solutions

SS (SangSoo) HAN

Asia Pacific - CICS product manager

sshan@sg.ibm.com

Agenda

CICS Tools Update

CICS PA / Omegmon

CICS CM

CICS IA

Hybrid Cloud – Transactional Integration with CICS TX and CTG

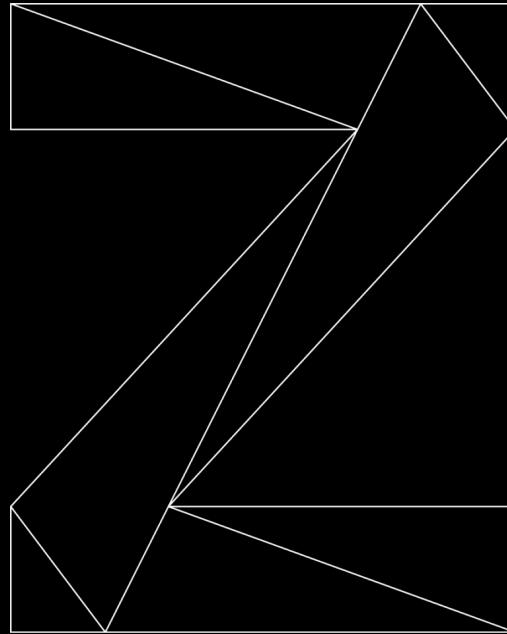
Open-Source Solutions

Spring Boot

MyBatis

Kafka Integration

Kafka for z/OS



CICS Tools Portfolio: Overview

Six products designed to support customers with their understanding, management and monitoring of CICS TS

At all stages of an application's lifecycle



**CICS
Performance
Analyzer
(CICS PA)**

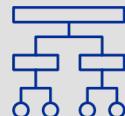


**CICS
Configuration
Manager
(CICS CM)**

Rocket Software
IBM-owned CICS Explorer plug-ins



**CICS
VSAM
Recovery
(CICS VR)**



**CICS
Interdependenc
y Analyzer
(CICS IA)**



**CICS
Deployment
Assistant
(CICS DA)**



**CICS
VSAM
Transparency
(CICS VT)**

Precisely

HCL
Including CICS Explorer plug-ins

Day-to-day support and in case of emergencies

CICS Performance Analyzer for z/OS

Monitor and manage CICS system and application performance

Analyze performance of CICS transactions and resource (Link, TSQ, VSAM) usage

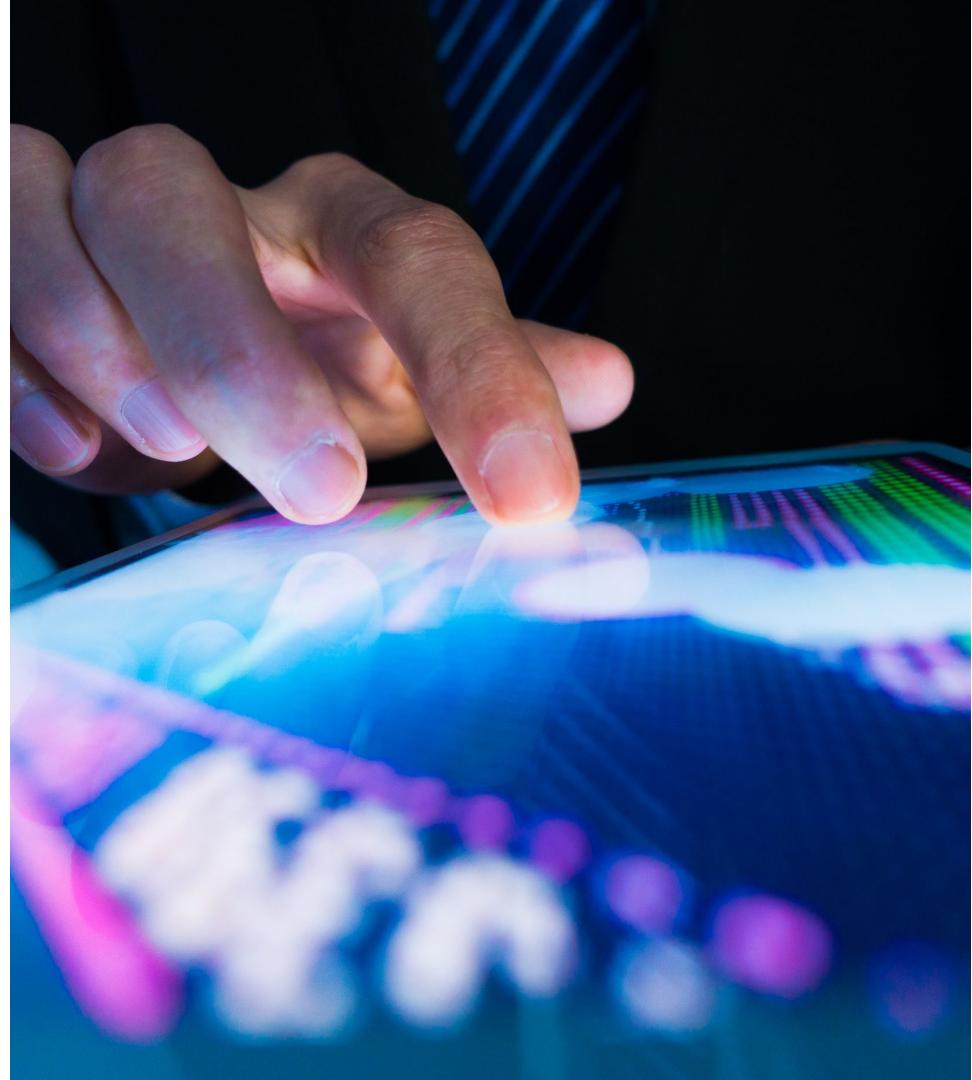
Statistics and performance alerts to quickly highlight performance degradation

Usage trends for capacity planning

Comprehensive reporting and analysis, visible through CICS Explorer or external visualization with sending JSON data

Program analysis report from the Omegamon SMF

**Stay aware of your system
and make informed decisions**



CICS PA – Resource Usage Summary for VSAM Files

CICS Transaction

VSAM File Summary

Tuning Targets

- Log file
- Customer / Account Relation File
- Sequence No Master File
- Transaction History File

Tran	#Tasks	Get	Put	Browse	Add	Delete	Total	File	RLS	CFDT	Control	Requests
AAAA	767	Elapse	Avg					.0015	.0000	.0000	.0000	
		Max						.0271	.0000	.0000	.0000	
	Count	Avg	44	0	0	0	46	2	0	0	0	4
		Max	48	2	0	8	55	15	0	0	0	19
		***** FC Calls *****										
File	#Tasks	Get	Put	Browse	Add	Delete	Total	File	RLS	CFDT	Control	AccMeth Requests
AAAAAAA	2	Elapse	Avg	.0000	.0000	.0000	.0338	.0338	.0038	.0000	.0000	.0000
		Max	.0000	.0000	.0000	.0370	.0370	.0045	.0000	.0000	.0000	
	Count	Avg	0	0	0	6	0	6	0	0	0	6
		Max	0	0	0	6	0	6	0	0	0	6
BBBBBBB	767	Elapse	Avg	.0000	.0000	.0000	.0000	.0000				
		Max	.0004	.0000	.0000	.0000	.0000					
	Count	Avg	7	0	0	0	0	0				
		Max	9	0	0	0	0	0				
CCCCCCC	767	Elapse	Avg	.0016	.0029	.0000	.0022	.0000				
		Max	.2544	.0392	.0000	.1310	.0000	.2572	.0079	.0000	.0000	.0000
	Count	Avg	0	0	0	0	0	1	1	0	0	3
		Max	2	2	0	1	0	0	0	0	0	0
DDDDDDD	307	Elapse	Avg	.0001	.0006	.0000	.0053	.0000				
		Max	.0020	.0199	.0000	.0364	.0000					
	Count	Avg	0	0	0	1	0	0	0	0	0	0
		Max	1	1	0	1	0	0	0	0	0	0
EEEEEEE	767	Elapse	Avg	.0000	.0000	.0000	.0000	.0000				
		Max	.0004	.0000	.0000	.0000	.0000					
	Count	Avg	37	0	0	0	0	0				
		Max	38	0	0	0	0	0				
Total	2610	Elapse	Avg	.0005	.0009	.0000	.0013	.0000				
		Max	.2544	.0392	.0000	.1310	.0000	.2572	.0191	.0000	.0000	.0000
	Count	Avg	13	0	0	0	0	13	0	0	0	0
		Max	38	2	0	6	0	38	7	0	0	10

Transaction Resource Usage Reports
— File Usage Summary
— Temporary Storage Usage Summary
— DPL Usage Summary
— URIMAP Usage Summary
— WEBSERVICE Usage Summary
— Transaction Resource Usage List
— Tuning Targets
— Detailed List Reports Required:

Detailed List Reports Required:
/ File Usage
/ Temporary Storage
/ DPL
/ URIMAP
/ WEBSERVICE

Omegamon for CICS V5.6 : Task Program Details

The screenshot shows the Omegamon for CICS V5.6 interface. At the top, there's a menu bar with File, Edit, View, Tools, Navigate, Help, and a timestamp of 02/04/2022 13:27:24. To the right of the menu, it says Display : HISTORY, CICSprix : SB3, and Region : CICD5501. Below the menu, a command line shows Command ==> KCPTASHP. The main window title is "Task History Detail". Below the title, there are tabs: Details, Statistics, Storage, Timings, Programs, and Related. The "Timings" tab is selected. A sub-section titled "Transaction Program Details" is displayed. It includes columns for Program Name, Invoked Count, CPU Time, Elapsed Time, Dispatch Time, CPU Time on QR TCB, Number of EXEC calls, Number of Abends, and Number of Mode Switches. The data table lists six programs: DPLLSTRT, DPLLINKA, DPLLINKB, DPLLINKC, DPLLINKD, and DPLXCTLA, with their respective details.

Program Name	Invoked Count	CPU Time	Elapsed Time	Dispatch Time	CPU Time on QR TCB	Number of EXEC calls	Number of Abends	Number of Mode Switches
DPLLSTRT	1	.000033s	.000066s	.000063s	.000033s	3	0	0
DPLLINKA	3	.000221s	.000381s	.000372s	.000221s	12	0	0
DPLLINKB	1	.000054s	2.19312s	.000054s	.000054s	6	0	0
DPLLINKC	2	.000274s	4.19449s	.000299s	.000274s	18	0	0
DPLLINKD	1	.000086s	2.09694s	.000089s	.000086s	7	0	0
DPLXCTLA	1	.000175s	2.09709s	.000221s	.000175s	7	0	0

- OMEGAMON CICS provides details on each CICS defined program that has been used by a task.
- This is available via the OMEGAMON CICS Active Task and Task History displays.

CICS PA – Program Task List Report

- Historically, CICS Performance Analyzer customers have frequently asked if it could provide a list of task programs with metrics for those programs.
- Unfortunately, this level of program information was not available to CICS PA until now.
- New OMEGAMON Program List and Summary reports have been introduced to provide CICS task program details. These reports will be generated using the new OMEGAMON for CICS SMF 112 subtype 202 records.
- CICS PA Reports Sets have been enhanced to include new Subsystem Reports OMEGAMON Program report definition.

CICS Performance Analyzer OMEGAMON - Program List										
OMP0001 Printed at 12:19:58 4/26/2022 data from 23:11:33 2/04/2022										
Start Date	Start Time	Tran	Task No	APPLID	UOW Seq	UOWID	Netname	# Progs		
2/04/2022	19:48:15.255	STRS	36682	CACD5501	1	046D0D811572	ROCKNET1.CACD5501	1		
Program	Count	CPU	Elapsed	Dispatch	QR CPU	Othr CPU	# ModesW	# Exec	# Abends	
MICKSTRS	1	.000009	.000015	.000015	.000009	.000000	0	6	0	
Total	1	.000009	.000015	.000015	.000009	.000000	0	6	0	
Start Date	Start Time	Tran	Task No	APPLID	UOW Seq	UOWID	Netname	# Progs		
2/04/2022	19:48:15.255	STRS	36675	CACD5501	1	046D0D80FE10	ROCKNET1.CACD5501	3		
Program	Count	CPU	Elapsed	Dispatch	QR CPU	Othr CPU	# ModesW	# Exec	# Abends	
MICKSTRS	1	.000158	.000545	.000159	.000158	.000000	0	6	1	
DFHTFP	1	.000037	.000038	.000038	.000037	.000000	0	0	0	
DFHPEP	1	.000004	.000004	.000004	.000004	.000000	0	2	0	
Total	3	.000199	.000587	.000200	.000199	.000000	0	8	1	
Start Date	Start Time	Tran	Task No	APPLID	UOW Seq	UOWID	Netname	# Progs		
2/04/2022	19:48:15.255	STRS	36683	CACD5501	1	046D0D812686	ROCKNET1.CACD5501	1		
Program	Count	CPU	Elapsed	Dispatch	QR CPU	Othr CPU	# ModesW	# Exec	# Abends	
MICKSTRS	1	.000015	.000015	.000015	.000015	.000000	0	6	0	
Total	1	.000015	.000015	.000015	.000015	.000000	0	6	0	
Start Date	Start Time	Tran	Task No	APPLID	UOW Seq	UOWID	Netname	# Progs		
2/04/2022	19:48:15.255	STRS	36684	CACD5501	1	046D0D812A0A	ROCKNET1.CACD5501	1		
Program	Count	CPU	Elapsed	Dispatch	QR CPU	Othr CPU	# ModesW	# Exec	# Abends	
MICKSTRS	1	.000010	.000010	.000010	.000010	.000000	0	6	0	
Total	1	.000010	.000010	.000010	.000010	.000000	0	6	0	
Start Date	Start Time	Tran	Task No	APPLID	UOW Seq	UOWID	Netname	# Progs		
2/04/2022	19:48:15.255	STRS	36685	CACD5501	1	046D0D812CA2	ROCKNET1.CACD5501	1		

CICS PA – Task Program Summary Report

- The Summary report provides summarized task and program statistics based on a user specified key. By default, this report is ordered by key value.

CICS Performance Analyzer OMEGAMON - Program Summary											Page 47											
Tran APPLID Start Date Start Time Program			Data from 6:37:57 1/23/2022 to 13:44:47 2/05/2022								Page 47											
<hr/>																						
<hr/>																						
Tasks	Count	CPU	Elapsed	Dispatch	QR CPU	Othr CPU	# ModesW	# Exec	# Abends													
35080	35080	Avg .000009	.000346	.000019	.000009	.000000	0	6	0													
	Tot .326529	12.13904	.673619	.326529	.000000	0	210480	0	0													
	Max .000070	.211172	.118804	.000070	.000000	0	6	0	0													
	Min .000008	.000008	.000008	.000008	.000000	0	6	0	0													
	Dev .000003	.005142	.000755	.000003	.000000	0	0	0	0													
	90% .000013	.006938	.000987	.000013	.000000	0	6	0	0													
Tran APPLID Start Date Start Time Program																						
STRS CACD5501	2/04/2022 22:00:00	MICKSTRS																				
Tasks	Count	CPU	Elapsed	dispatch	QR CPU	Othr CPU	# ModesW	# Exec	# Abends													
26	26	Avg .000003	.000003	.000003	.000003	.000000	0	2	0													
	Tot .000076	.000076	.000076	.000076	.000000	0	52	0	0													
	Max .000004	.000004	.000004	.000004	.000000	0	2	0	0													
	Min .000002	.000002	.000002	.000002	.000000	0	2	0	0													
	Dev .000000	.000000	.000000	.000000	.000000	0	0	0	0													
	90% .000003	.000003	.000003	.000003	.000000	0	2	0	0													
Tran APPLID Start Date Start Time Program																						
STRS CACD5501	2/04/2022 23:00:00	DFHPEP																				
Tasks	Count	CPU	Elapsed	dispatch	QR CPU	Othr CPU	# ModesW	# Exec	# Abends													
26	26	Avg .000039	.000039	.000039	.000039	.000000	0	0	0													
	Tot .001006	.001023	.001023	.001006	.000000	0	0	0	0													
	Max .000084	.000084	.000084	.000084	.000000	0	0	0	0													
	Min .000035	.000035	.000035	.000035	.000000	0	0	0	0													
	Dev .000009	.000010	.000010	.000009	.000000	0	0	0	0													
	90% .000051	.000052	.000052	.000051	.000000	0	0	0	0													
Tran APPLID Start Date Start Time Program																						
STRS CACD5501	2/04/2022 23:00:00	MICKSTRS																				
Tasks	Count	CPU	Elapsed	dispatch	QR CPU	Othr CPU	# ModesW	# Exec	# Abends													
6874	6874	Avg .000009	.000975	.000014	.000009	.000000	0	6	0													
	Tot .063751	6.701784	.096778	.063751	.000000	0	41244	0	0													
	Max .000072	.370256	.021671	.000072	.000000	0	6	0	0													
	Min .000008	.000008	.000008	.000008	.000000	0	6	0	0													
	Dev .000003	.012956	.000266	.000003	.000000	0	0	0	0													
	90% .000013	.017584	.000355	.000013	.000000	0	6	0	0													

CICS PA – z/OS Connect Support

Requirement:

- RFE 104690 - Enhance CICS PA to process zOS Connect SMF 123 records

New functionality:

- System Definitions subsystem type ZCON
- Report Set -> Subsystem Reports -> z/OS Connect
- Report Set -> Extracts -> Record Selection -> z/OS Connect
- Report Set -> Extracts -> z/OS Connect
- Report Set -> Forwarding -> z/OS Connect
- Splunk sample app -> z/OS Connect Dashboard

CICS Configuration Manager for z/OS

Increase operational efficiency and service agility

Manage resource definitions on multiple repositories across diverse environments (including the z/OS File System)

Transform definitions automatically to match the target environment

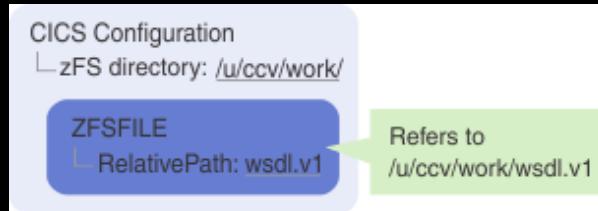
Assess COLD START impact using updated CICS definitions when compared with the CICS region's current runtime resources

Single point of control

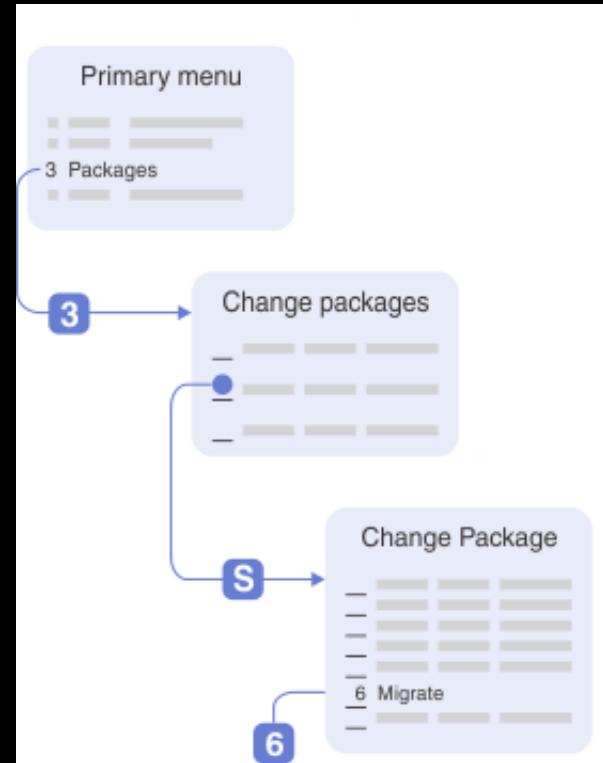


CICS Configuration Manager – zFS File resource definition

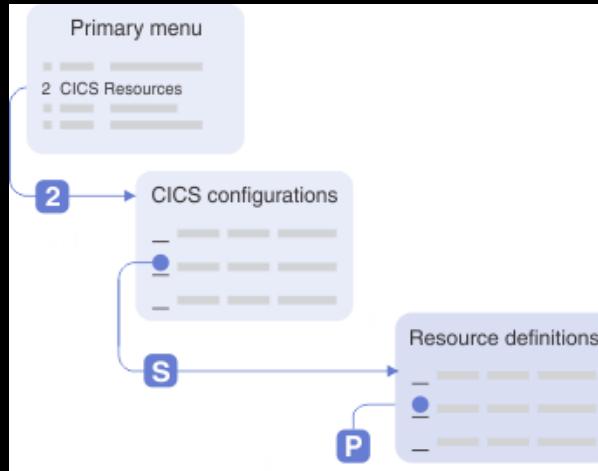
Determining path of a ZFSFILE resource



Migrating a ZFSFILE from one CICS configuration to another



Packaging a ZFSFILE resource definition



CICS Interdependency Analyzer for z/OS

Understand relationships and dependencies

Isolation and removal of affinities

Identification of threadsafe and non-threadsafe programs

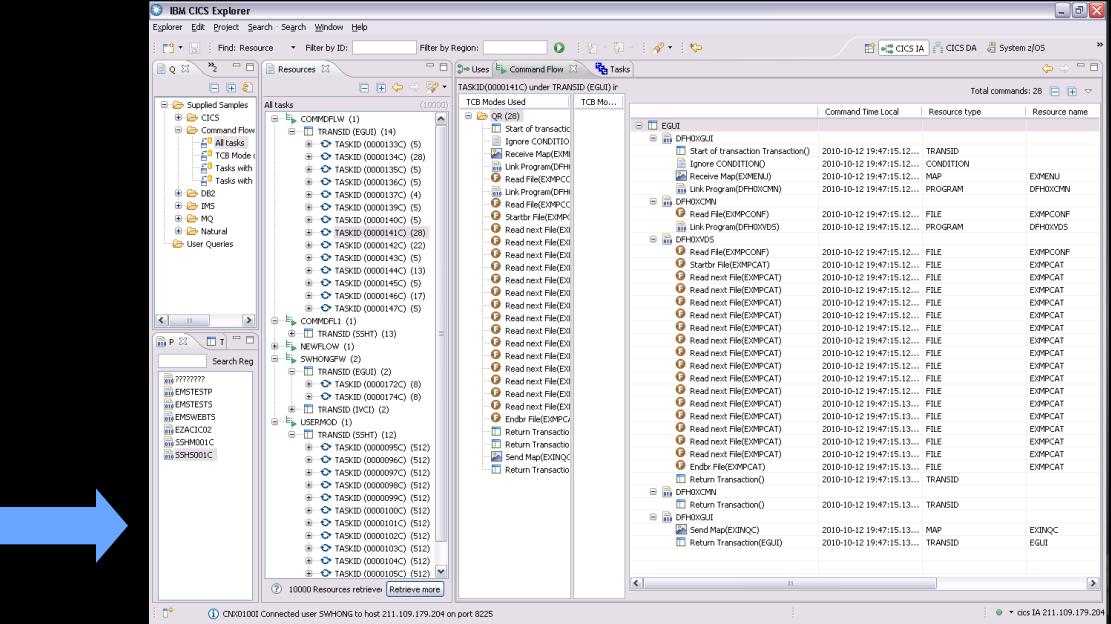
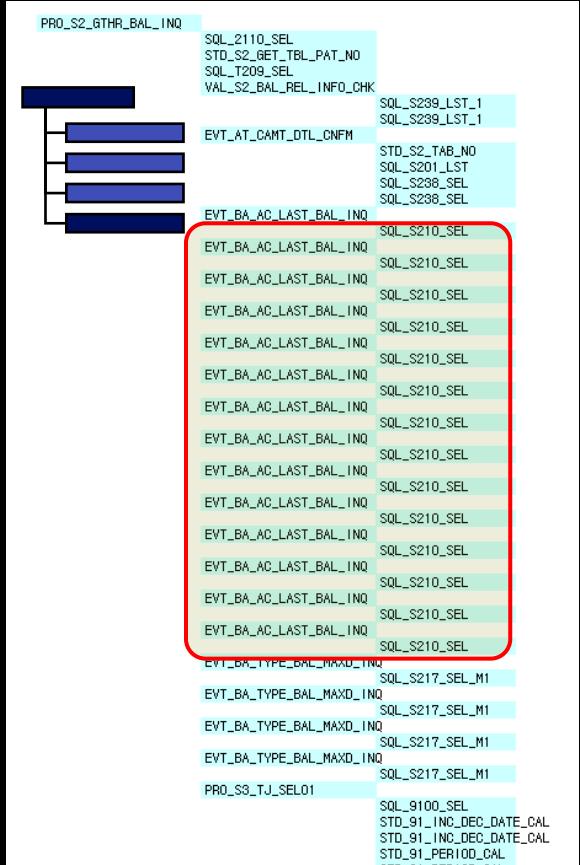
Transaction/Program profiling using Command Flow

Runtime understanding of your CICS environment

Assists with modernization and API enablement of CICS applications



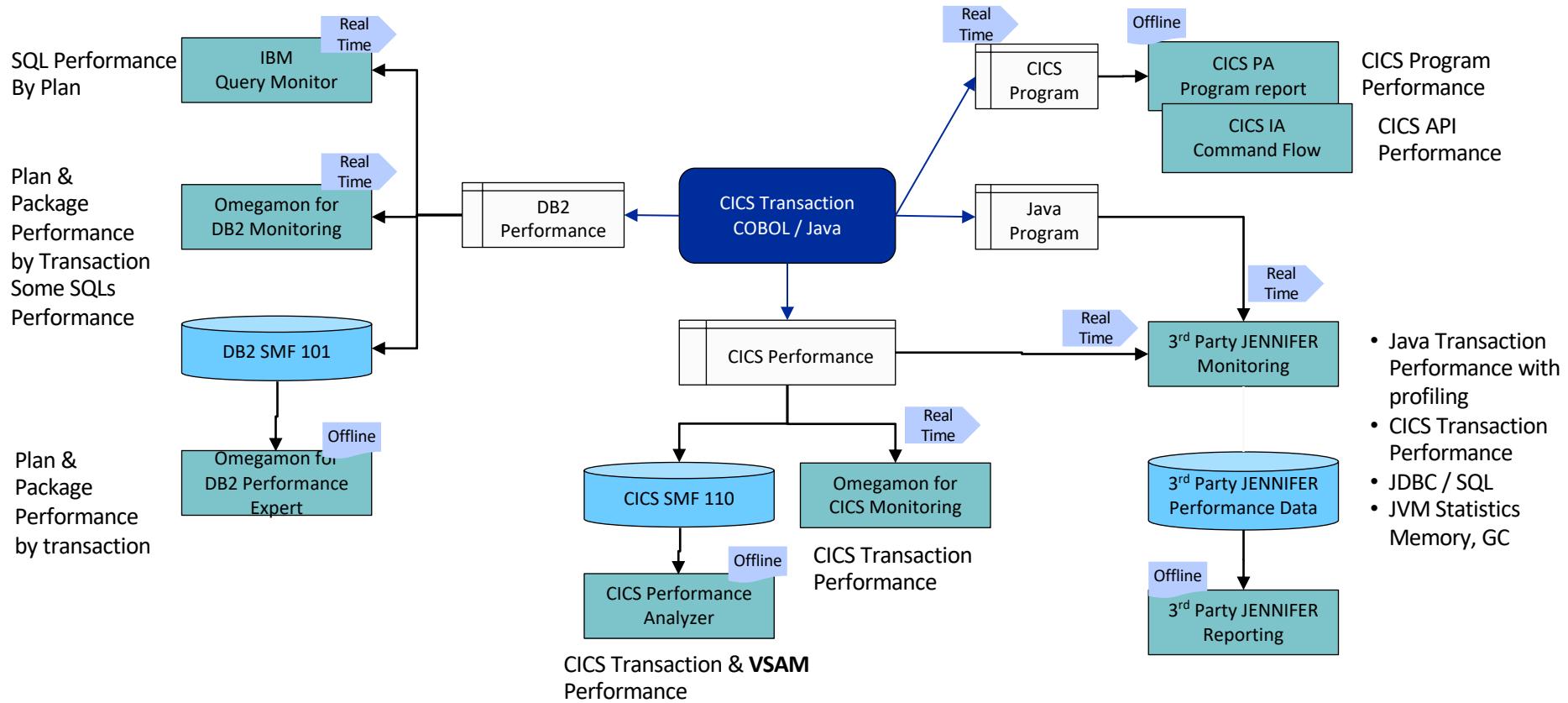
CICS IA – Command Flow Feature for Profiling



**Capture task information and all call flows
including Dynamic Cobol Call, DB2 SQL, IMS
DBCTL and VSAM Call**

Performance Tuning Process for COBOL/Java applications – CICS Services

COBOL/Java Performance Data Source and Tuning Tool

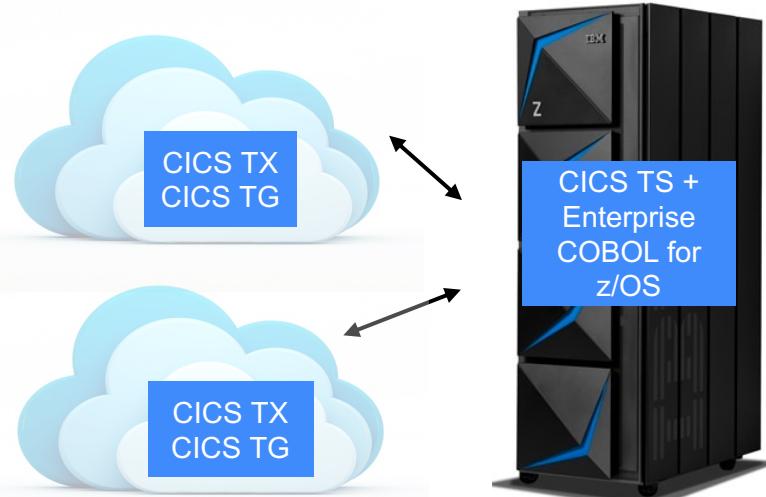


Hybrid Cloud Integration

CICS TX and CTG

Hybrid Cloud	CICS TX	CICS TG	Resources
<ul style="list-style-type: none">- Key drivers- What & how of modernization- Use cases	<ul style="list-style-type: none">- Use cases- Architecture- capabilities	<ul style="list-style-type: none">- Use cases- Capabilities- CICS TG 9.3 features	<ul style="list-style-type: none">- Links & Resources

Hybrid cloud architecture



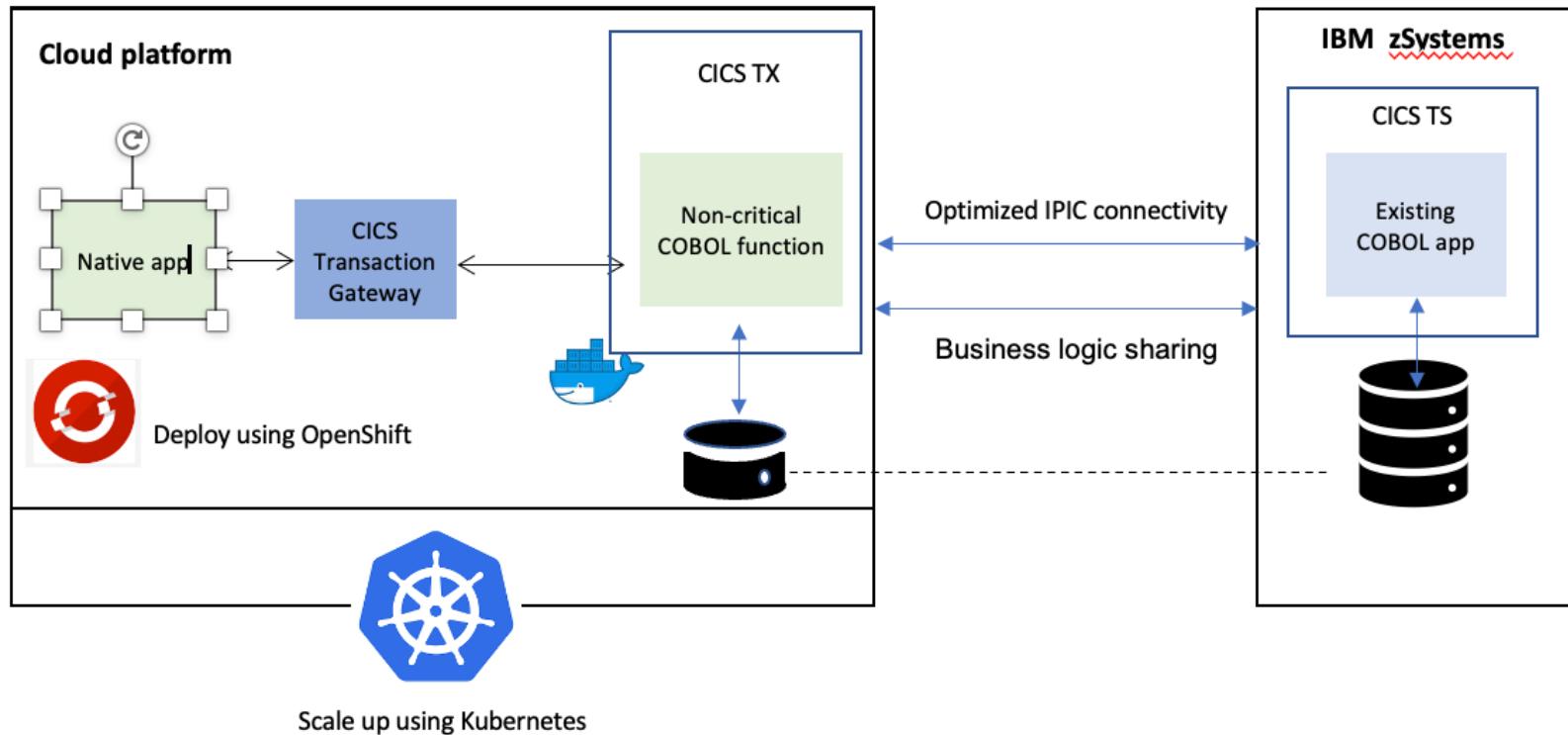
Hybrid cloud strategy is emerging as a popular choice for digital transformation

- Deploy non-performance critical COBOL/CICS applications on cloud
- Scale and interoperate with core CICS Transaction Server for z/OS (CICS TS)
- Share data and logic seamlessly across z/OS and cloud

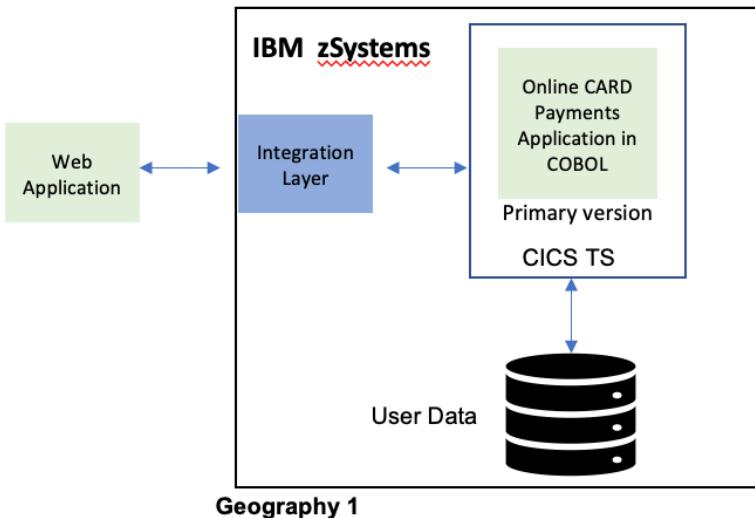
Usage scenarios

1. Read-only applications, ex
 - Use CICX TX for quotes
 - Use CICS TS on z/OS for actual purchases
 - Core data resides on IBM Z
2. Data sovereignty
 - Core COBOL/CICS applications and data resides on IBM Z
 - Smaller geographies can use CICS TX
3. Enhance department servers with cloud infrastructure
 - Improve flexibility, reduce maintenance overhead
 - Core COBOL/CICS applications and data resides on IBM Z

Hybrid cloud use case – Extend transactions

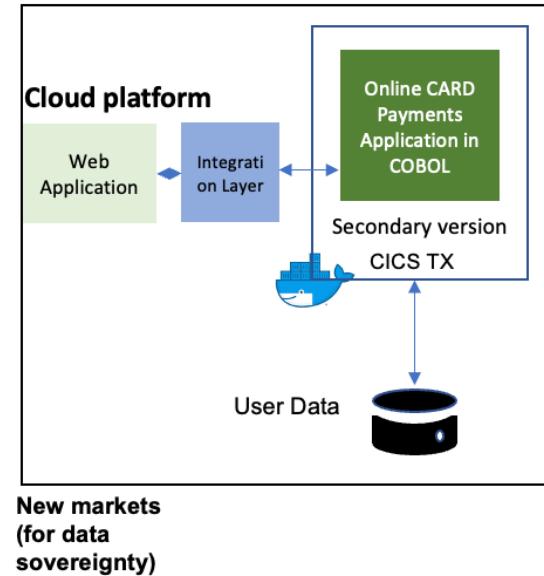


Hybrid cloud use case – Data Sovereignty



Geography 1

Additionally modify platform specific elements of CARD Payments application to deploy in cloud
Original application remains intact



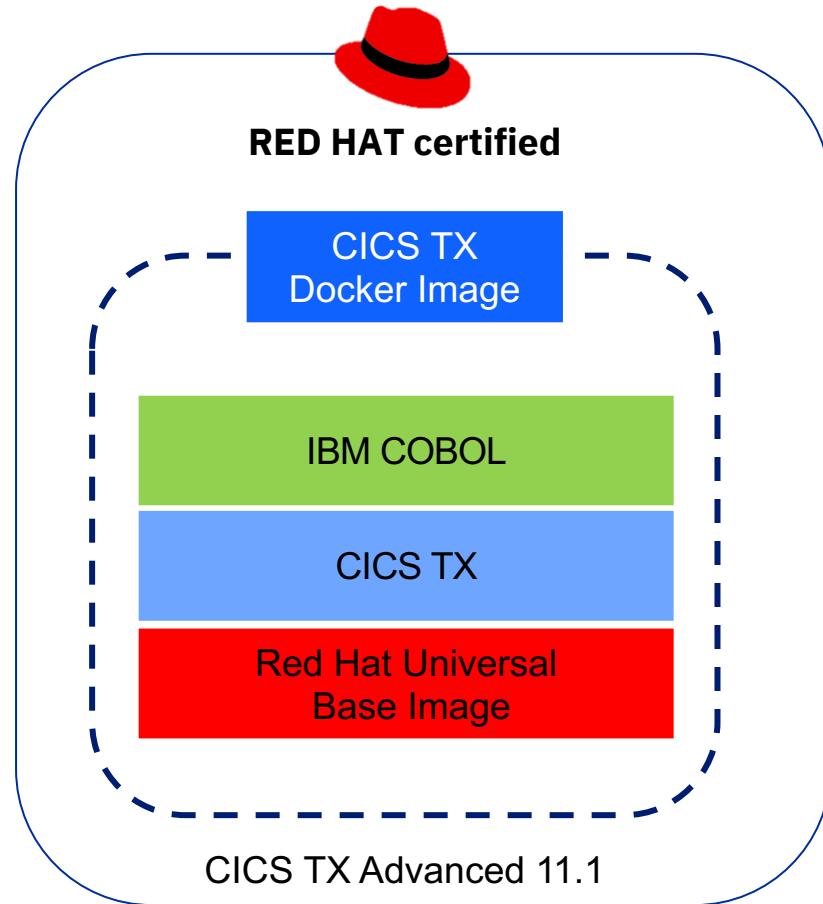
New markets
(for data sovereignty)

CICS TX container

Supports Core set of CICS APIs in container

Fully Red Hat® certified container image

- Guaranteed to be portable – build once and run anywhere
- Includes IBM COBOL for x86
- Guaranteed to be secure and current
- Improved enterprise support from Red Hat and IBM
- Continuous monitoring for assured security - notifies you of new and/or known vulnerabilities - Red Hat Build Service automatically updates and publishes images.



IBM COBOL for Linux x86

100% IBM COBOL compilation technology

Native, optimizing compiler for developing and deploying COBOL applications on Linux on x86

Runs on x86-64 servers running:

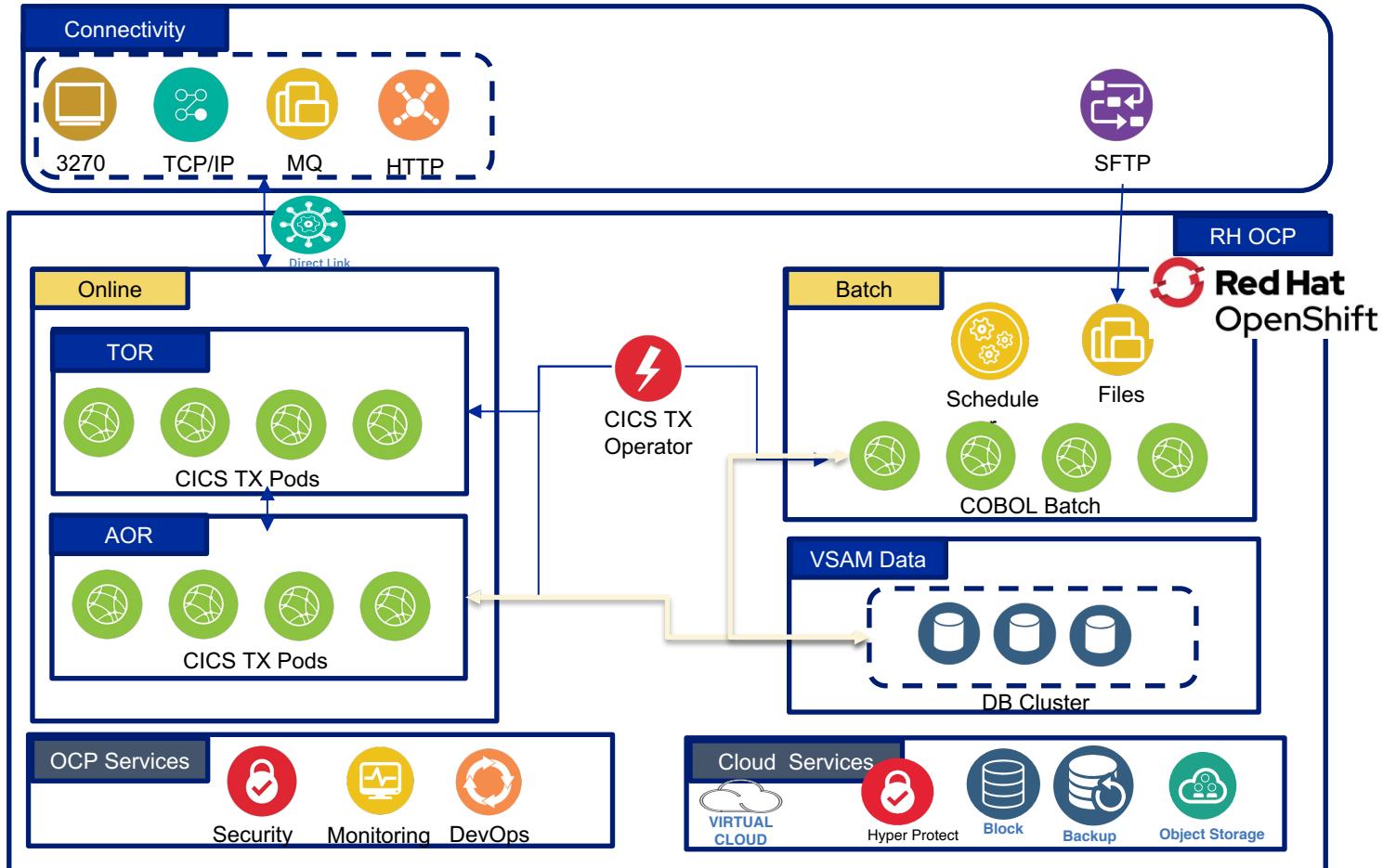
- Red Hat® Enterprise Linux (RHEL)7.8 or higher
- Ubuntu server (Ubuntu)16.04, 18.04, or higher

Integrated Debugger support

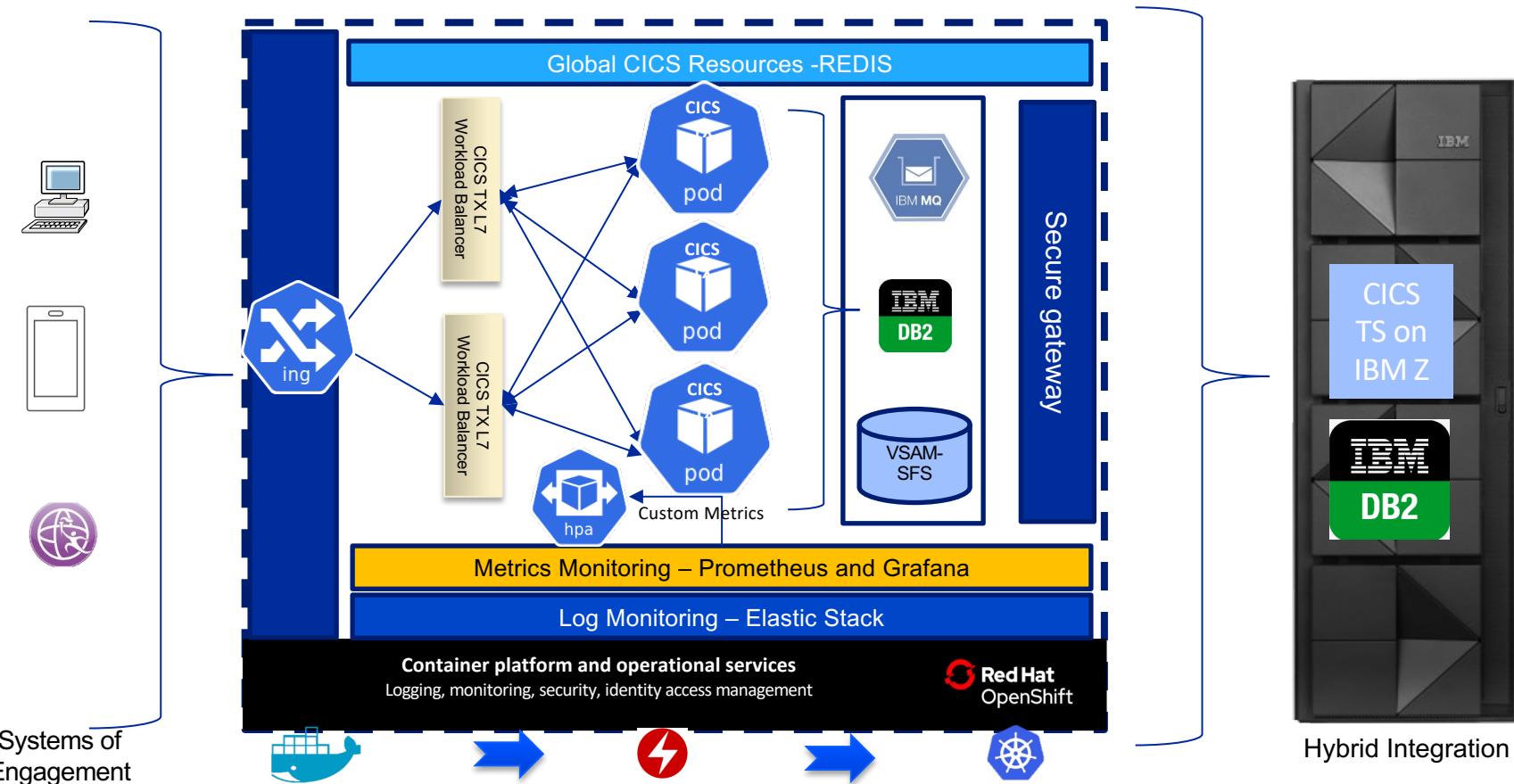
- Provides core debugging functionality for COBOL & CICS applications



CICS TX in cloud reference architecture



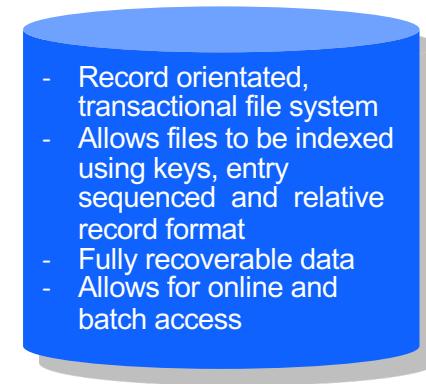
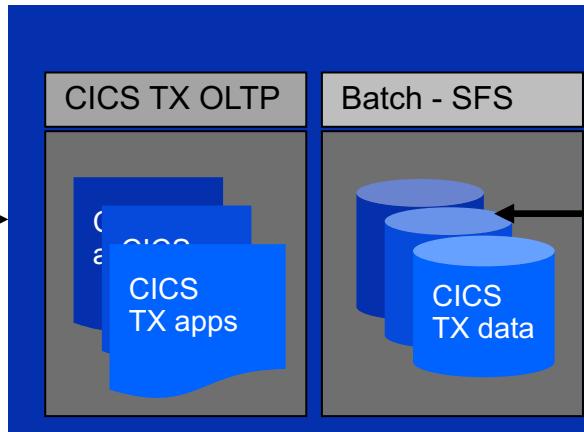
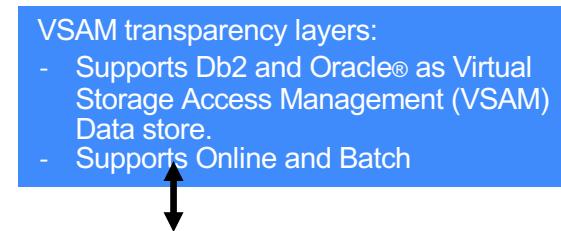
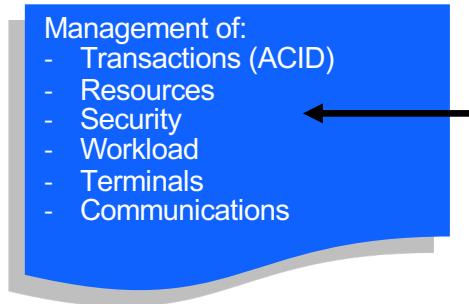
CICS TX High Level Architecture



Key components of CICS TX

Includes two core components:

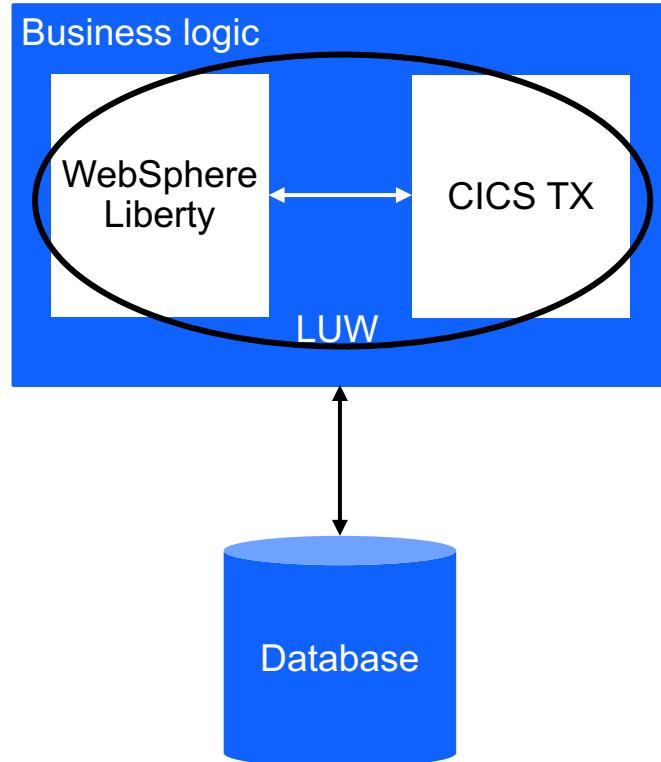
- CICS Online Transaction Processing environment (OLTP).
- Batch Component - Structured File Server (SFS).



Interoperate COBOL with Java

Bi-directional transactional integration with WebSphere Liberty

Write new business logic in Java™ and extend them to interact with existing COBOL assets.

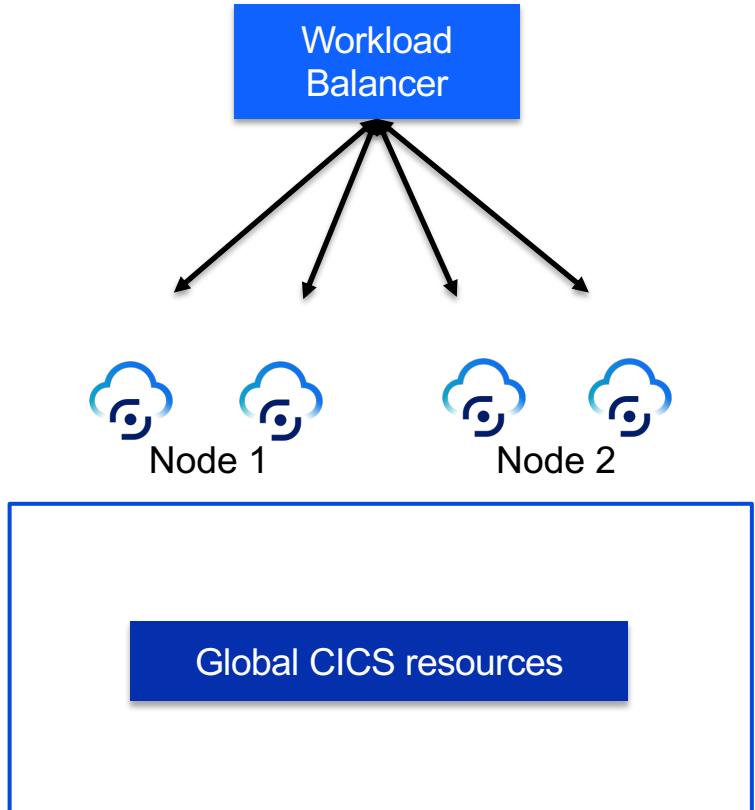


CICS resources in cloud

Moving applications to cloud can be tedious, time consuming and costs money.

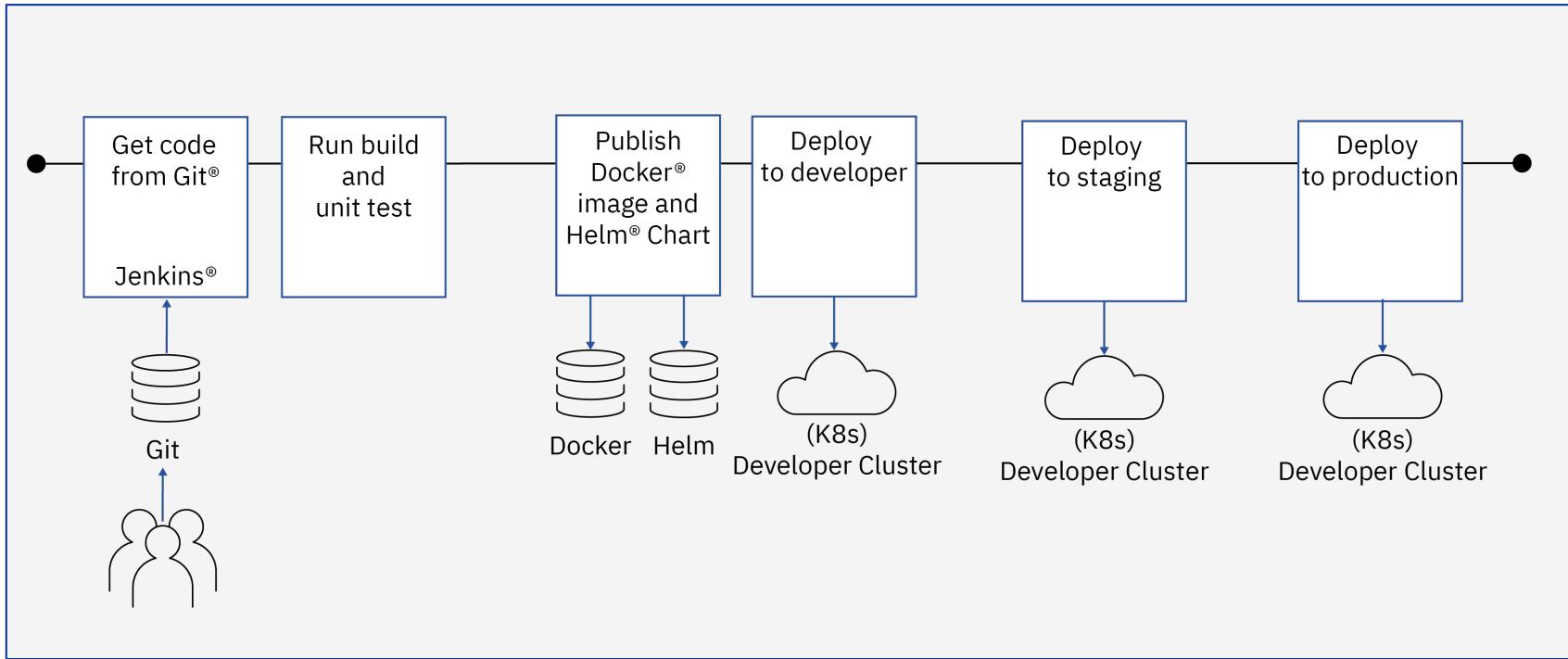
CICS TX uses REDIS® an in-memory data store manage resources like ENQ, DEQ, TSQ, etc globally.

This ensures that minimum application refactoring is needed when moving to cloud.



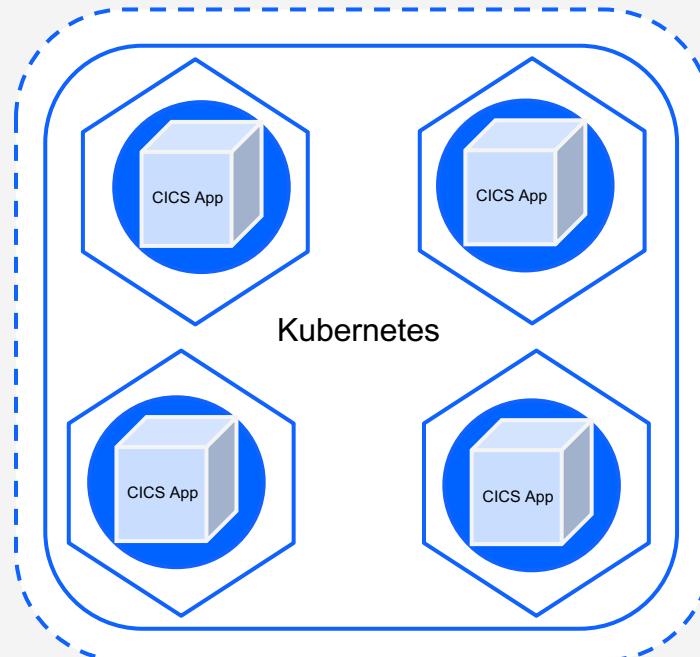
Global CICS resources

DevOps with CICS TX



CICS TX and Kubernetes

- Seamless scaling with Kubernetes®
- Open-source container management system
- Highly available
- Horizontal autoscaling
- Automated deployments.
- Automated rollouts and rollbacks
- Self healing



Administrator

Home

Overview

Projects

Search

Explore

Events

Operators

OperatorHub

Installed Operators

Workloads

Pods

Deployments

Deployment Configs

Stateful Sets

Secrets

Config Maps

You are logged in as a temporary administrative user. Update the [cluster OAuth configuration](#) to allow others to log in.

Based on recent sampling, the PersistentVolume claimed by elasticsearch-elasticsearch-cdm-rk2mwrzn-3 in Namespace openshift-logging is expected to fill up within four days. Currently 6.275% is available.

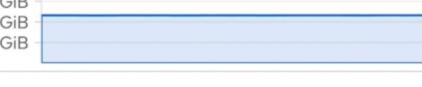
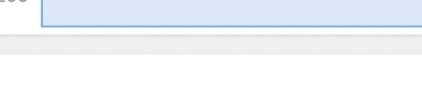
⚠ May 16, 5:53 am

Based on recent sampling, the PersistentVolume claimed by image-registry-storage in Namespace openshift-image-registry is expected to fill up within four days. Currently 6.275% is available.

[View details](#)

Cluster Utilization

1 Hour ▾

Resource	Usage	13:15	13:30	13:45	14:00
CPU	12.88 of 96				
Memory	46.43 GiB of 188.4 GiB				
Filesystem	357.6 GiB of 556.8 GiB				
Network Transfer	9.34 MBps in 10.73 MBps out				
Pod count	297				

Details

[View settings](#)

Cluster API Address

Home

Overview

Projects

Search

Explore

Events

Operators

OperatorHub

Installed Operators

Workloads

Pods

Deployments

Deployment Configs

Stateful Sets

Secrets

Config Maps

Cron Jobs

Jobs

You are logged in as a temporary administrative user. Update the cluster OAuth configuration to allow others to log in.

Project: operator ▾

Stateful Sets

Create Stateful Set

Name

Search by name...



Name ↑

Status ↑

Labels ↑

Pod Selector ↑

SS cicctx-app

2 of 2 pods

app.kubernetes.io/c... =bac...

app.kubernetes.io... =cicctx...

app.kubernetes.i... =cicctx-...

app.kubernetes.io...=cicctx...

app.kubernetes.io... =cicctx...

kappnav.app.auto-cre... =tr...

Q app.kubernetes.io/instance=...

cicctx-app



SS cicctx-hybrid

2 of 2 pods

app.kubernetes.io/c... =bac...

app.kubernetes.i... =cicctx...

app.kubernetes.i... =cicctx-...

app.kubernetes.i... =cicctx-h...

app.kubernetes.i... =cicctx-...

kappnav.app.auto-cre... =tr...

Q app.kubernetes.io/instance=...

cicctx-hybrid



SS redis

app.kubernetes.io/c... =bac...

app.kubernetes.i... =cicctx...

app.kubernetes.i... =cicctx-...

app.kubernetes.i... =cicctx-s...

app.kubernetes.i... =cicctx-t...

app.kubernetes.i... =cicctx-t...

kappnav.app.auto-cre... =tr...

Q app=redis



SS svc

app.kubernetes.io/c... =svc-t...

app.kubernetes.i... =svc-t...

app.kubernetes.i... =svc-t-s...

app.kubernetes.i... =svc-t-t...

app.kubernetes.i... =svc-t-t...

kappnav.app.auto-cre... =tr...

Q app.kubernetes.io/instance=...

svc-test



CICS TX deployed on OCP

CICS TX container with Employee Management application deployed on OCP

Home

Overview

Projects

Search

Explore

Events

Operators

OperatorHub

Installed Operators

Workloads

Pods

Deployments

Deployment Configs

Stateful Sets

Secrets

Config Maps

Cron Jobs

Jobs

Project: operator ▾

Stateful Sets > Stateful Set Details

 cicctx-hybrid

Actions ▾

Details YAML Pods Environment Events

Name	Status	Ready	Restarts	Node	Memory	CPU
 cicctx-hybrid-0	 Running	1/1	1	 openshiftci-cstxoncloud-10-worker2	241.6 MiB	0.016 cores
 cicctx-hybrid-1	 Running	1/1	0	 openshiftci-cstxoncloud-10-worker0	232.8 MiB	0.025 cores

CICS TX deployed on OCP

CICS TX container with Employee Management application deployed on OCP

You are logged in as a temporary administrative user. Update the [cluster OAuth configuration](#) to allow others to log in.

Project: operator ▾

Services

[Create Service](#)

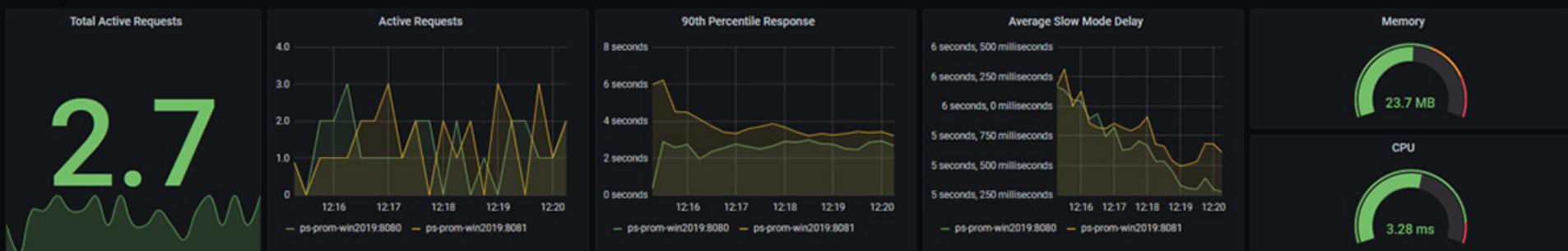
Name	Labels	Pod Selector	Location
cicstx-app	app.kubernetes.io... =bac... app.kubernetes.io... =cicst... app.kubernetes.io... =cicstx... app.kubernetes.io... =cicstx... app.kubernetes.io... =cicst... monitor.cicstx.io/en...=tr...	app.kubernetes.io/instan...ce=cicstx-app	172.30.0.72:1435
cicstx-hybrid	app.kubernetes.io... =bac... app.kubernetes.io... =cicst... app.kubernetes.io... =cicstx... app.kubernetes.io... =cicstx... app.kubernetes.io... =cicstx... monitor.cicstx.io/en...=tr...	app.kubernetes.io/instan...ce=cicstx-hybrid	172.30.175.38:1435
cics-tx-term	app.kubernetes.io... =bac... app.kubernetes.io... =cicst... app.kubernetes.io... =cicstx... app.kubernetes.io... =cicstx... app.kubernetes.io... =cicstx... monitor.cicstx.io/en...=tr...	app.kubernetes.io/instan...ce=cicstx-hybrid	172.30.92.81:3270
egress-2	name=mysql	app.kubernetes.io/instan...ce=cicstx-hybrid	172.30.142.54:3306
redis	No labels	app=redis	None
svc-test	app.kubernetes.io... =bac... app.kubernetes.io... =cicst... app.kubernetes.io... =cicstx... app.kubernetes.io... =cicstx... app.kubernetes.io... =cicstx... monitor.cicstx.io/en...=tr...	app.kubernetes.io/instan...ce=svc-test	172.30.123.189:1435

Two instance of CICS TX PODs are running

Batch processor



Web App



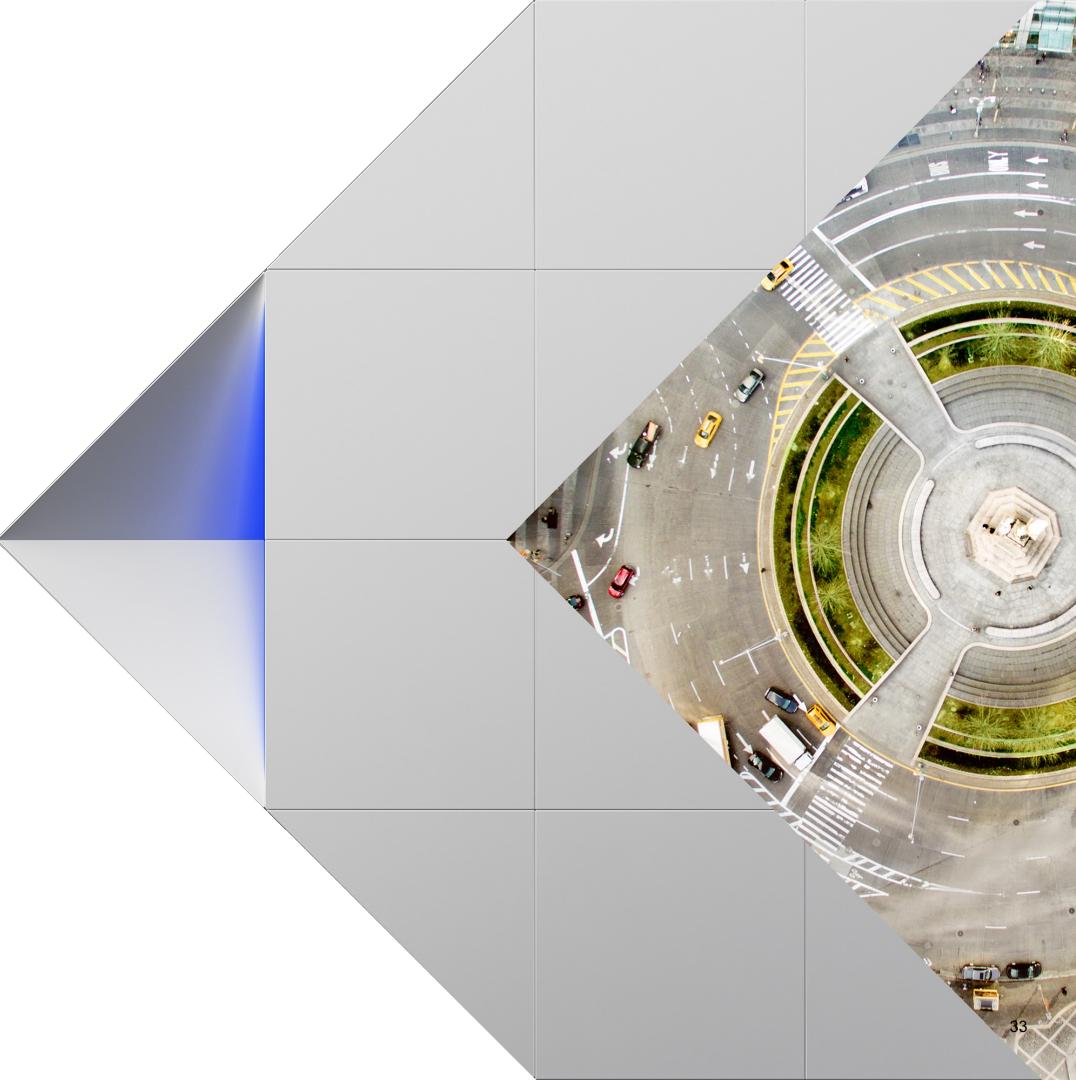
Info

Linux Server			Batch app				Web app			
	Instance	App Version	Assembly Name	.NET Version	Instance	App Version	Assembly Name	.NET Version		
linux	ps-prom-ub1804:9100	4.15.0-101-generic	ps-prom-ub1804:8080	1.6.2	PrometheusDemo.Ba...	3.1.5	ps-prom-win2019:8081	2.0.1	PrometheusDemo.Web	3.1.5
	ps-prom-ub1804:8081	1.6.2	PrometheusDemo.Ba...	3.1.5	ps-prom-win2019:8080	2.0.1	PrometheusDemo.Web	3.1.5		
windows	Windows Server		ps-prom-win2019:9182	10.0.17763						

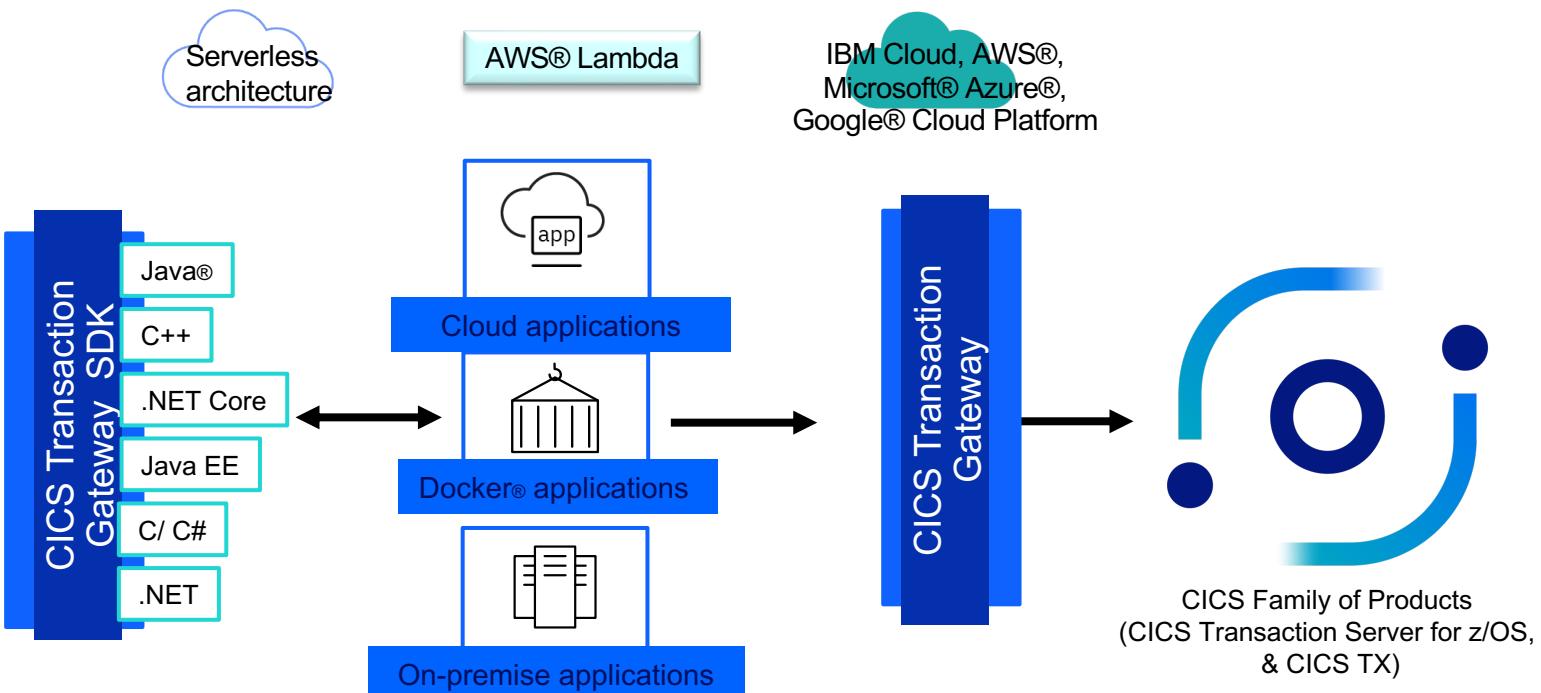
CICS TG

Connect your applications to CICS using native SDKs

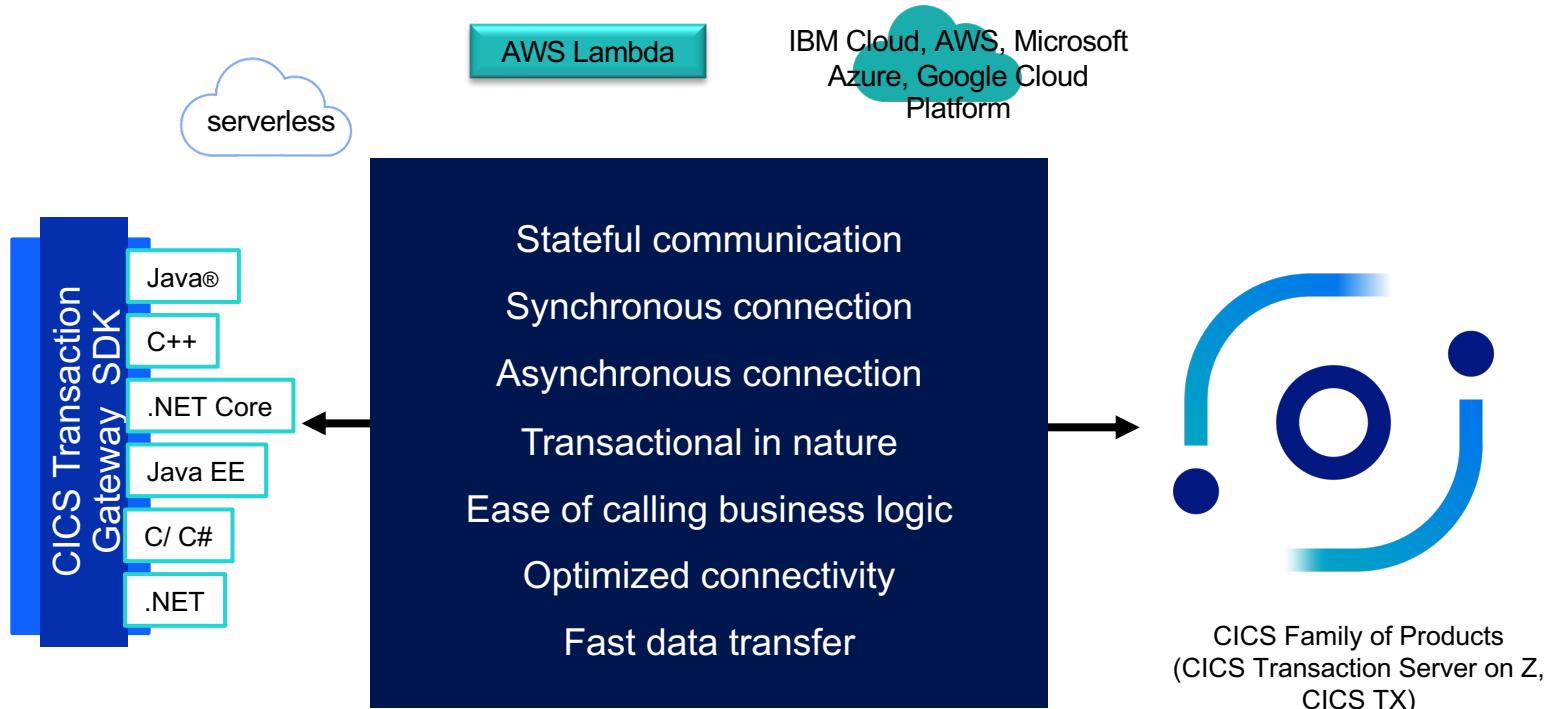
Unleash the benefits of SDKs and containers.



CICS Transaction Gateway



Why SDK connectivity

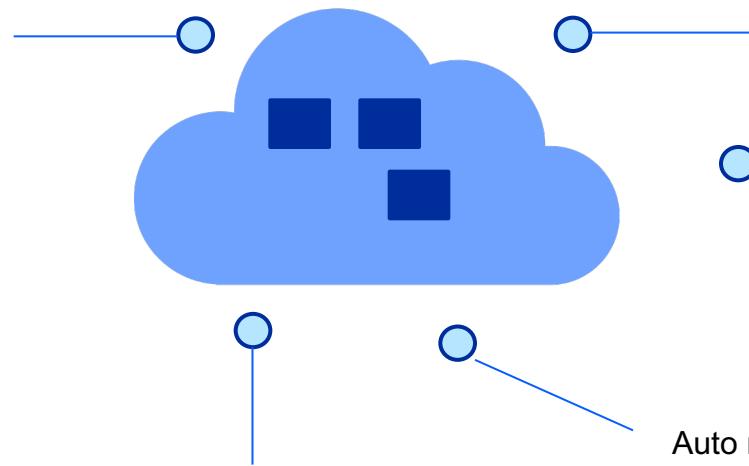


Containerized CICS TG V9.3

GA - 29th
Apr, 2022

Unleash the benefits of containers!

Multiple CICS Transaction
Gateway (CICS TG)
gateways on same host



Easier administration
of multiple gateway
instances

Easier upgrades, roll
backs of fixes

Auto recovery of
unhealthy CICS TG
containers

Deploy easily of
Red Hat® Open Shift®

CICS TG 9.3



Implement standard security solution using JWT
New APIs to support JWT for Java®, JCA, .NET, .NET Core applications



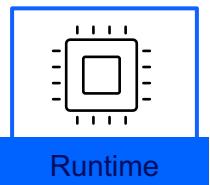
ID Propagation support with WebSphere Liberty for better integration and security



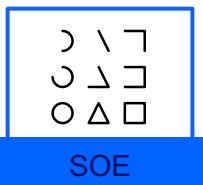
End to end transaction tracking enablement for monitoring products like IBM Z Application Performance Management Connect



Connection Pool support for .NET Core for optimal resource utilization
NLS support for .NET applications
Support for zCX for application development



Ability to reset gateway health to 0
Selectively reject back level clients for simplification during upgrades



Leverage z16 features, upgrade stack .NET core and .NET framework & other SOE

Transaction tracking with ZAPM



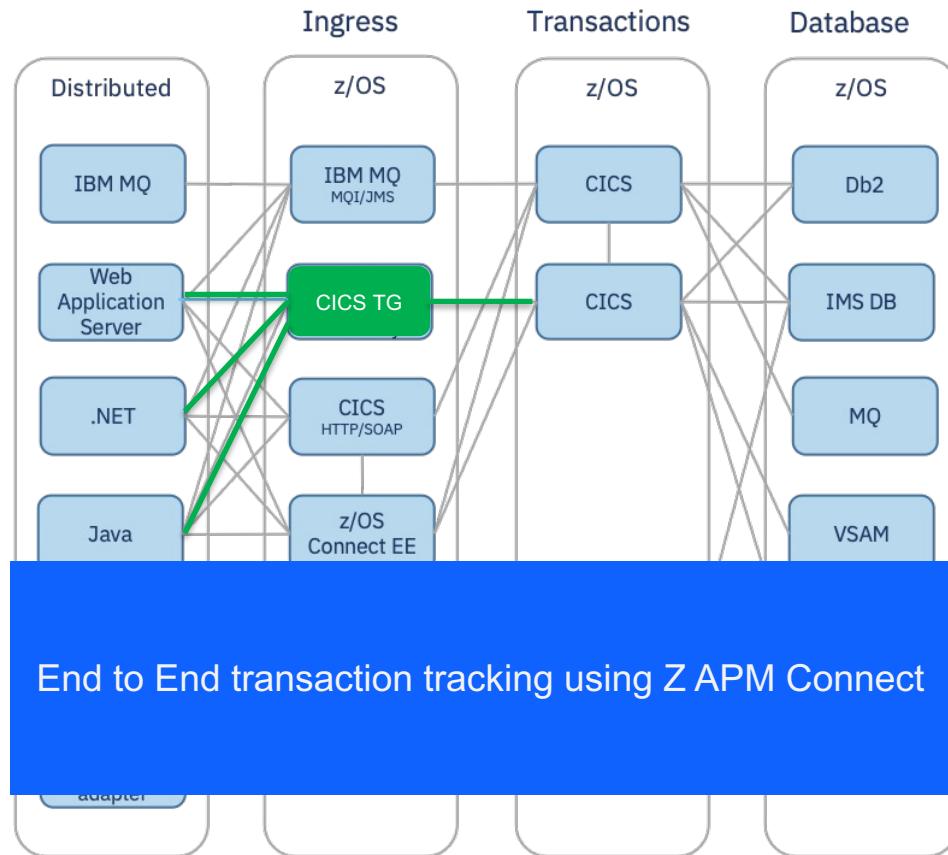
- I can see transactions flow end to end starting from WebSphere Application Server-> CICS TG-> CICS Transaction Server -> Db2
- Ease of tracking the delay/issues



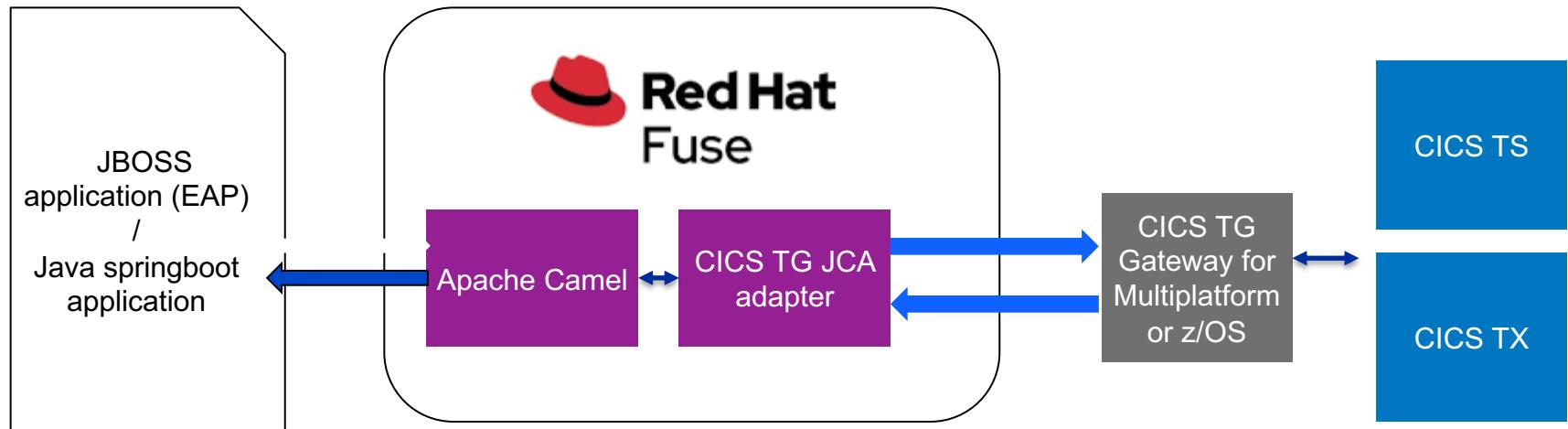
- Rapidly identify and resolve performance issues.



- I can use one solution for most of my products.
- Enhances my team's ability to isolate subsystem which is root cause of the problem.
- Can also be used for products deployed in cloud



Red Hat Fuse Integration



No specific dev activities involved from CICS TG side

CICS TG & Red Hat team activities :

- Test the integration and provide support statement
- Document the integration steps

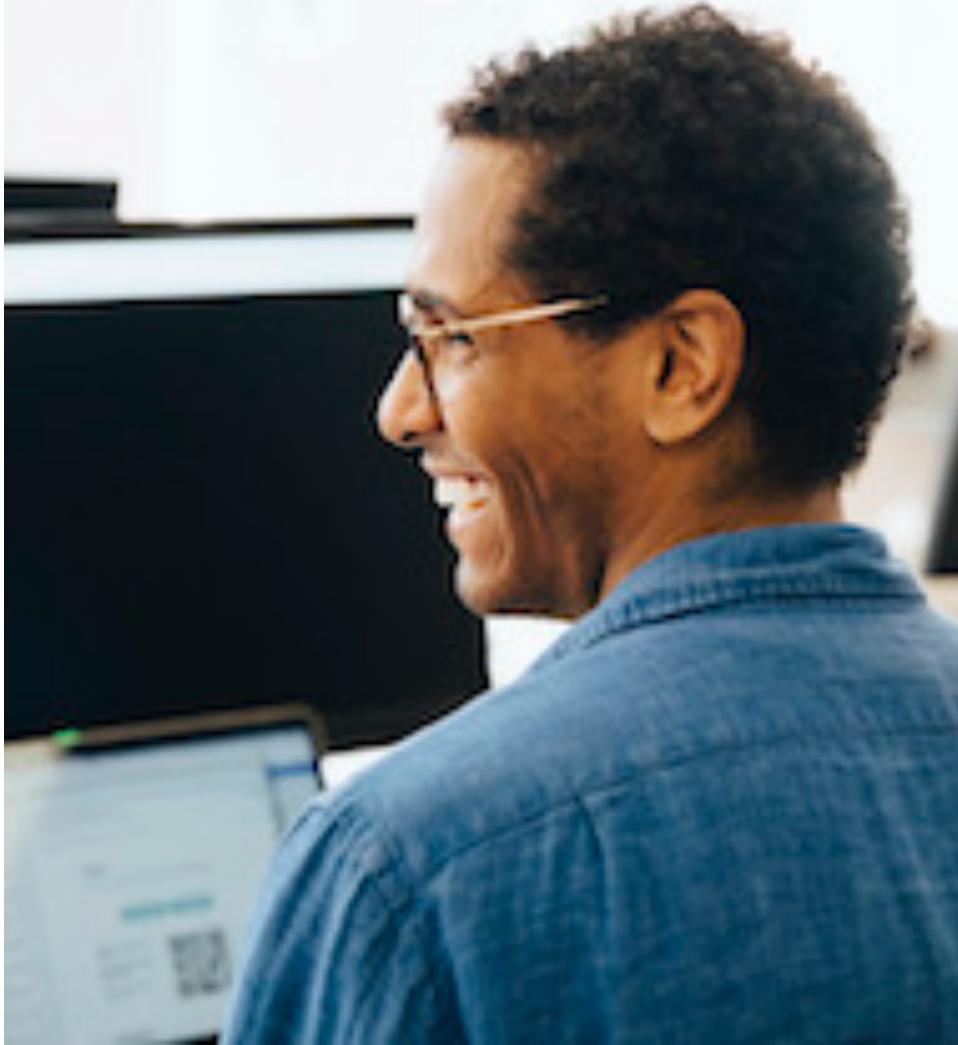
No foreseeable impact due to RH certification change as Fuse will embed CICS TG SDK only.

Want to know more?

Visit our product page where you can learn more and access a free CICS TX trial. [Link](#)

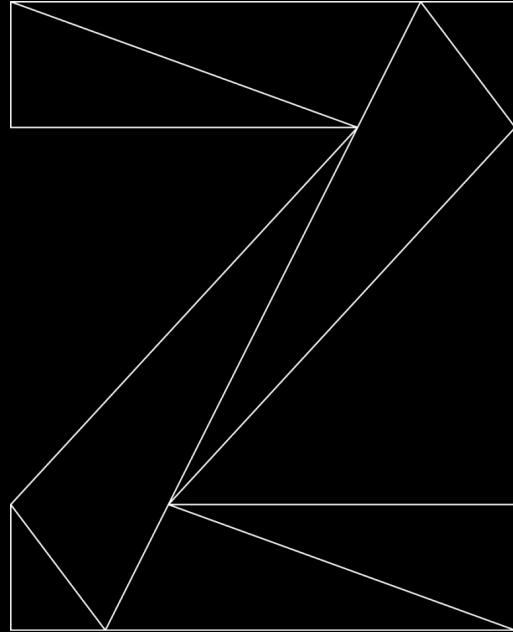
- Announcement letter- [Link](#)
- IBM COBOL for Linux on x86 - [Link](#)
- Latest information, how-to and more – [Link](#)
- Announcement letter– [Link](#)
- IBM Documentation – [Link](#)
- [CICS TG link](#)

[CICS TX and CICS TG community](#)



Open-source Solutions

- Spring Boot
- MyBatis
- Kafka



Springboot in CICS Liberty

Springboot features in Liberty

springBoot-1.5 : Provides support for Spring Boot applications using Spring Boot version 1.5.x.

springBoot-2.0 : Provides support for Spring Boot applications using Spring Boot version 2.0.x.

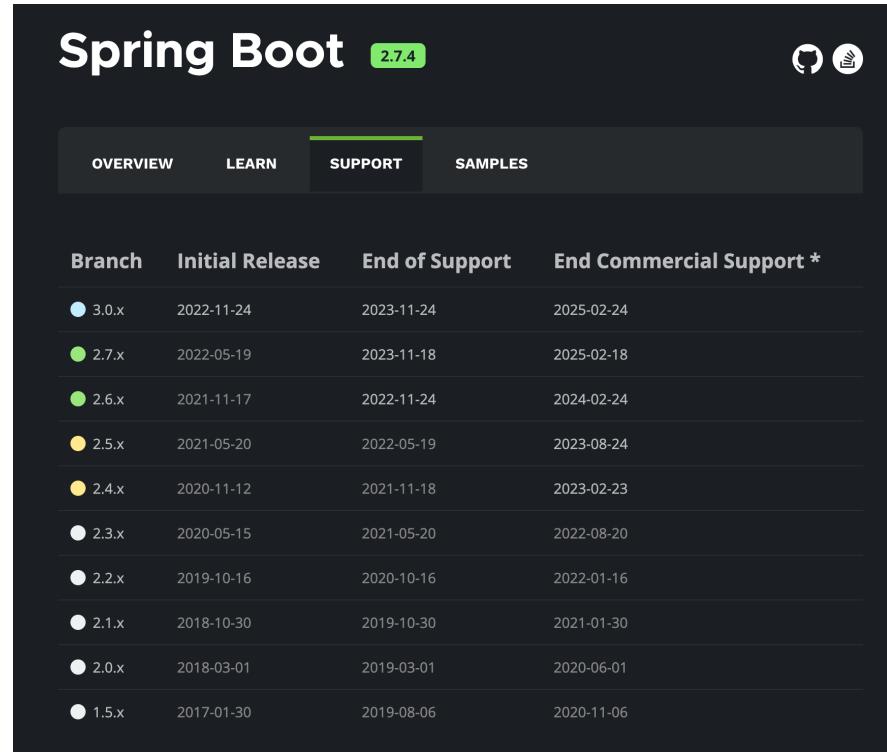
Without the feature, you can test the latest Spring boot in a web project.

Deploying Spring Boot applications in JARs

If you build and deploy your Spring Boot application as a JAR, the application can use the JCICS and JCICSX APIs. JAR deployments cannot integrate with Java EE and Liberty.

Deploying Spring Boot applications in WARs

If you build and deploy your Spring Boot application as a WAR, the application can use the JCICS and JCICSX APIs, and the integrated with Java EE and Liberty. Follow the best practices that are described in topic [Building and deploying Spring Boot applications](#). Each subtopic describes an important aspect of integration and how to ensure Spring Boot integrates with Java EE, Liberty, and CICS capabilities.



The screenshot shows the Spring Boot website with the title "Spring Boot 2.7.4". Below the title is a navigation bar with tabs: OVERVIEW, LEARN, SUPPORT (which is highlighted in green), and SAMPLES. To the right of the tabs are two icons: a GitHub logo and a bell icon. The main content area contains a table titled "Support Timeline" with four columns: Branch, Initial Release, End of Support, and End Commercial Support *. The table lists ten branches from 1.5.x to 3.0.x, each with its release date, end of support date, and end of commercial support date.

Branch	Initial Release	End of Support	End Commercial Support *
3.0.x	2022-11-24	2023-11-24	2025-02-24
2.7.x	2022-05-19	2023-11-18	2025-02-18
2.6.x	2021-11-17	2022-11-24	2024-02-24
2.5.x	2021-05-20	2022-05-19	2023-08-24
2.4.x	2020-11-12	2021-11-18	2023-02-23
2.3.x	2020-05-15	2021-05-20	2022-08-20
2.2.x	2019-10-16	2020-10-16	2022-01-16
2.1.x	2018-10-30	2019-10-30	2021-01-30
2.0.x	2018-03-01	2019-03-01	2020-06-01
1.5.x	2017-01-30	2019-08-06	2020-11-06

Springboot Support – Transactional integration with CICS

Using the Java™ Transaction API (JTA) you can coordinate CICS, Liberty, and third-party resource managers, such as a type 4 database driver connection, together as one global transaction. For more information about JTA support in CICS, see [Java Transaction API \(JTA\)](#).

JTA is available for use in a Spring Boot WAR application in various ways:

- Spring Boot's @Transactional annotation: This annotation, which is specified at the class or method level denotes the code segment to be contained within a single global transaction.
- Spring templates: The Spring framework provides two templates for use with programmatic transaction management: the TransactionTemplate and the PlatformTransactionManager interface.
- UserTransaction: It is also possible to use the JTA UserTransaction interface within a Spring Boot application by obtaining the UserTransaction initial context of the hosting Application server (Liberty) through a JNDI lookup. For example, `ctx.lookup("java:comp/UserTransaction");`. The developer can employ a Bean-managed approach to transactions by explicitly coding UserTransaction 'start' and 'end' calls around the resources to be managed.

Data Source Definition and Behaviors

1. Liberty dataSource with Autoconfigure or manual

```
<feature>jdbc-4.2</feature>
<dataSource id="defaultCICSDataSource" jndiName="jdbc/jdbcDataSource" transactional="false">
    <jdbcDriver libraryRef="defaultCICSDb2Library"/>
    <properties.db2.jcc driverType="2"/>
    <connectionManager agedTimeout="0"/>
</dataSource>
<library id="defaultCICSDb2Library">
    <fileset dir="/usr/lpp/db2v11/jdbc/classes" includes="db2jcc4.jar db2jcc_license_cisuz.jar"/> <fileset dir="/usr/lpp/db2v11/jdbc/lib"
includes="libdb2jcct2zos4_64.so"/>
</library>
```

2. cicsts_dataSource definition (Manual, not autoconfigure)

```
<feature>cicsts:jdbc-1.0</feature>
<cicsts_dataSource id="defaultCICSDataSource" jndiName="jdbc/CICSDataSource">
    <properties.db2.jcc currentSchema="DSN8110"/>
</cicsts_dataSource>

<cicsts_jdbcDriver libraryRef="defaultCICSDb2Library"/>
<library id="defaultCICSDb2Library">
    <fileset dir="/usr/lpp/db2v11/jdbc/classes" includes="db2jcc4.jar db2jcc_license_cisuz.jar"/>
    <fileset dir="/usr/lpp/db2v11/jdbc/lib" includes="libdb2jcct2zos4_64.so"/>
</library>
```

dataSource	type	autocommit	autocommit default	Default commit behaviour
cicsts_dataSource	T2	false	false	commit CICS UOW
Liberty datasource	T2	false	false	rollback CICS UOW
Liberty dataSource	T4	true or false	true	commit database update

JTA Usage in CICS Liberty

Unlike a CICS unit of work, a UserTransaction must be explicitly started using the begin() method. Invoking begin() causes CICS to commit any updates that may have been made prior to starting the UserTransaction. The UserTransaction is terminated by invoking either of the commit() or rollback() methods, or by the web container when the web application terminates. While the UserTransaction is active, the program can not invoke the JCICS Task commit() or rollback() methods.

The JCICS methods Task.commit() and Task.rollback() will not be valid within a JTA transaction context. If either is attempted, an InvalidRequestException will be thrown.

MyBatis - Spring Boot Application Properties Configuration

<https://docs.spring.io/spring-boot/docs/current/reference/html/appendix-application-properties.html#spring.datasource.jndi-name>

- Set to use the DataSource set in Liberty as the default DataSource of Spring Boot
 - JNDI location of the datasource. Class, url, username and password are ignored when set.

```
spring.datasource.jndi-name=jdbc/defaultCICSDataSource
```

<https://mybatis.org/spring-boot-starter/mybatis-spring-boot-autoconfigure/>

- Spring Boot's MyBatis integration setting
- Please refer to the reference site because there are various settings related to the MyBatis function.
- Default settings for very basic functional tests were used
 - Locations of Mapper xml config file.

```
mybatis.mapper-locations=classpath:test/aBank/*.xml
```

MyBatis - Mapper Configuration

<https://mybatis.org/spring/mappers.html>

- Scanning for mappers
 - You can configure Scanning for mappers in various ways. Here, Annotation is used.
 - Mapper Scan configuration requires very high initialization priority, so configure it in SpringBootApplication class or Configuration class.

```
@Configuration  
 @MapperScan("test.aBank")  
 public class TestaBankConfig {  
 ...}
```

MyBatis Mapper XML Creation

<https://mybatis.org/mybatis-3/sqlmap-xml.html>

- There are many ways to define SQL in MyBatis, but here we use Mapper XML.
- **Note that it must be defined in the resource path, not the source path, according to the packaging method.**
 - Mapper XML Files

```
<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE mapper PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN"
    "http://mybatis.org/dtd/mybatis-3-mapper.dtd">

<mapper namespace="test.aBank.TestaBankMapper">

    <select id="selectById" resultType="test.aBank.TestaBank">
        SELECT ID, PN FROM aBank.aBankTRXT WHERE ID = #{id}
    </select>

    <select id="selectAll" resultType="test.aBank.TestaBank">
        SELECT ID, PN FROM aBank.aBankTRXT
    </select>

    <insert id="insert" parameterType="test.aBank.TestaBank">
        INSERT INTO aBank.aBankTRXT(PN) VALUES (#{pn})
    </insert>
</mapper>
```

MyBatis Mapper Interface Creation

<https://mybatis.org/mybatis-3/configuration.html#mappers>

- Although it is possible to execute SQL even if the mapper is not registered in the interface method, in general, the Mapper Interface is explicitly created in order to reduce mistakes and use the automatic completion function of the tool.

```
@Mapper  
public interface TestaBankMapper {  
  
    public TestaBank selectById(long id);  
    public List<TestaBank> selectAll();  
    public void insert(TestaBank testaBank);  
}
```

Use MyBatis in user application

- Spring Boot extension of MyBatis registers Mapper as Spring Bean, and application can access through automatic injection function like using general Spring Bean.

```
@Controller  
 @RequestMapping("/mybatis")  
 public class TestaBankControllerMyBatis {  
  
     @Autowired  
     private TestaBankMapper testaBankMapper;  
  
     @RequestMapping(value = "/test", method = RequestMethod.GET)  
     @Transactional  
     public String test(HttpServletRequest httpServletRequest,  
                        TestaBank testaBank) {  
         testaBank.setPn("TEST");  
         testaBankMapper.insert(testaBank);  
         for (TestaBank th : testaBankMapper.selectAll()) {  
             ~~~
```

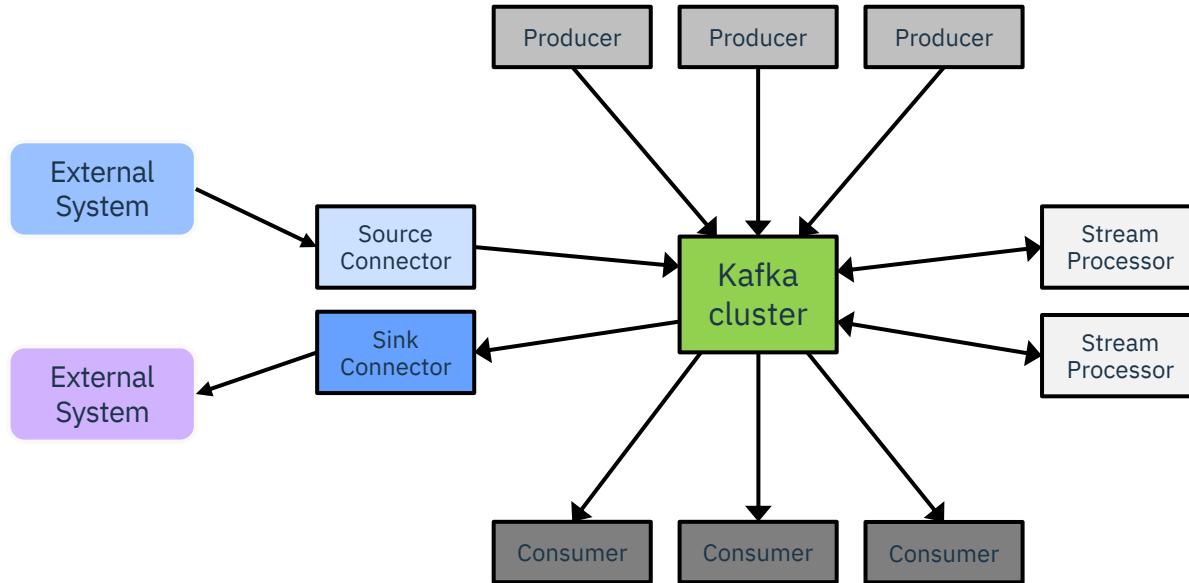
Producing and consuming messages in Kafka from CICS applications

- What is Kafka?
- CICS interactions and use cases with Kafka
- KAFKA for z/OS
- Demo
- Questions?

Apache Kafka is a distributed event streaming platform typically used for high-performance data pipelines, streaming analytics, data integration, and interconnecting mission-critical applications. This session introduces options for CICS applications to produce and consume messages in Kafka, including example code of the Kafka API in CICS.

What is Apache Kafka?

Apache Kafka is an open-source distributed event streaming platform used by thousands of companies (more than 80% of Fortune 100 companies) for high-performance data pipelines, streaming analytics, data integration, and mission-critical applications. Source <https://kafka.apache.org/>

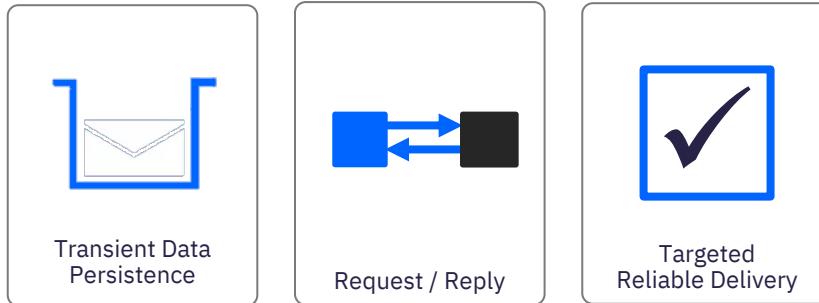


Kafka use cases

- Pub/sub messaging
- Website activity tracking
- Metrics
- Log aggregation
- Stream processing
- Event sourcing
- Commit log

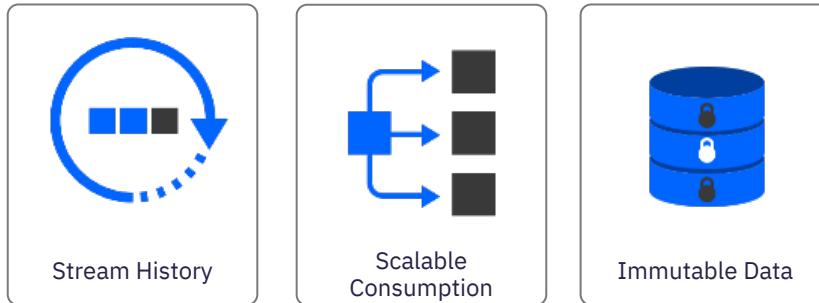
Message Queueing vs Event Streaming

Message Queuing



Destructive read
Server to Server Communication
Assured Delivery
Shared Data store for HA

Event Streaming



Marked as read by offset and expired cleanup
Client to Server Communication
Exactly-Once Semantics
Replication for HA

CICS application options to interact with Kafka

There are many ways to interact with Kafka instances - some of the primary options include:

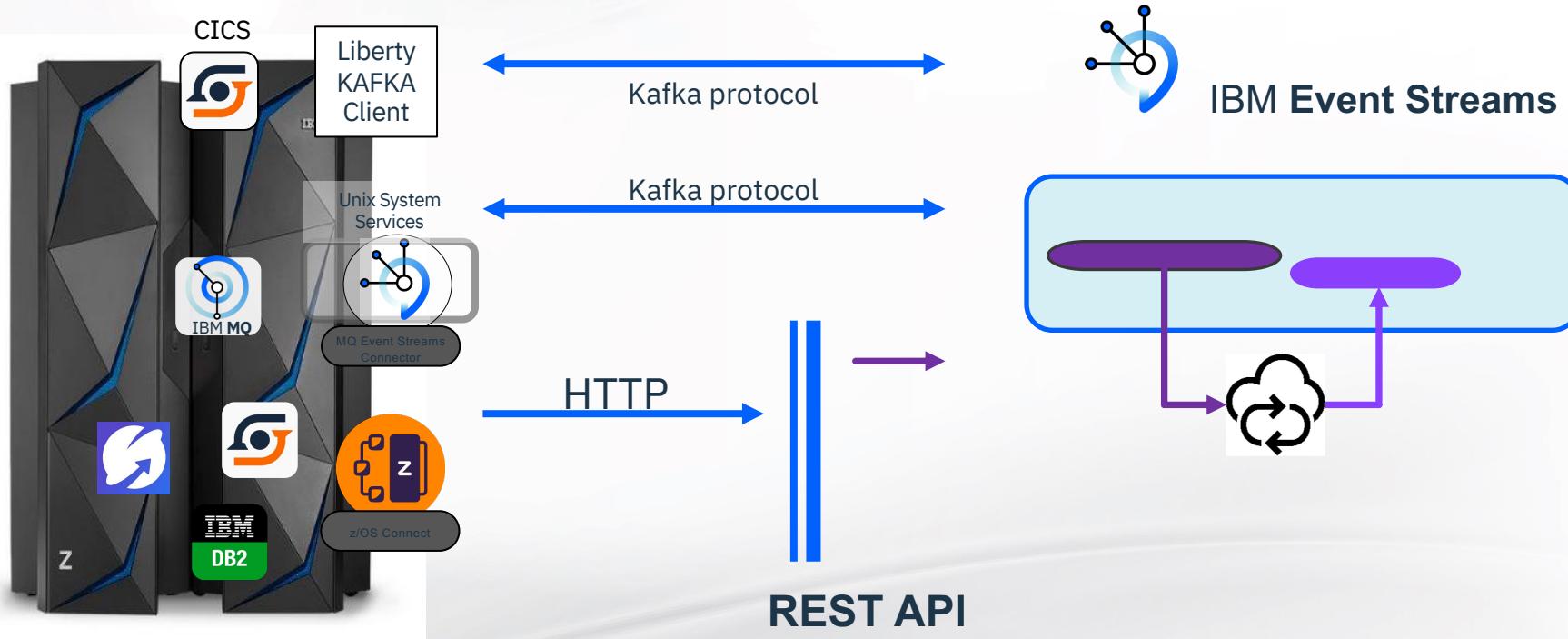
1. Develop a Java application using the [Kafka APIs](#) – consumer, producer, stream
 - Or use one of the [clients](#) that provide bindings for Node.js, C, and other languages.
2. Use [Kafka Connect](#) to copy / replicate data with other systems. This also provides [REST APIs](#).
3. Use the Kafka command line tools. These are particularly useful when experimenting and writing automation scripts.

Main options for CICS applications:

1. Write a CICS application in Java that uses the Kafka API, packaged using OSGi, Jakarta, or Spring framework
2. Write a CICS application to call the Kafka REST APIs, for example via [z/OS Connect EE](#), or your own Java REST component.
3. Write a CICS application to get/put messages to IBM MQ, and then replicate the IBM MQ queue with a Kafka topic using either:
 - [Kafka Connect source connector for IBM MQ](#): This is an open source project and can run on z/OS co-located with IBM MQ, or another platform.
 - Confluent [IBM MQ Sink Connector for Confluent Platform](#).

Connecting to IBM Event Streams

IBM Z Connectivity Options



KAFKA Client options

1. KAFKA Java client

```
@SpringBootApplication
public class Application {

    public static void main(String[] args) {
        SpringApplication.run(Application.class, args);
    }

    @Bean
    public NewTopic topic() {
        return TopicBuilder.name("topic1")
            .partitions(10)
            .replicas(1)
            .build();
    }

    @KafkaListener(id = "myId", topics = "topic1")
    public void listen(String in) {
        System.out.println(in);
    }
}
```

2. Spring KAFKA template

```
@SpringBootApplication
public class Application {

    public static void main(String[] args) {
        SpringApplication.run(Application.class, args);
    }

    @Bean
    public NewTopic topic() {
        return TopicBuilder.name("topic1")
            .partitions(10)
            .replicas(1)
            .build();
    }

    @Bean
    public ApplicationRunner runner(KafkaTemplate<String, String> template) {
        return args -> {
            template.send("topic1", "test");
        };
    }
}
```

```
public class Sender {
    public static void main(String[] args) {
        AnnotationConfigApplicationContext context = new AnnotationConfigApplicationContext(Config.class);
        context.getBean(Sender.class).send("test", 42);
    }
}

private final KafkaTemplate<Integer, String> template;
public Sender(KafkaTemplate<Integer, String> template) {
    this.template = template;
}
public void send(String toSend, int key) {
    this.template.send("topic1", key, toSend);
}
}

public class Listener {
    @KafkaListener(id = "listen1", topics = "topic1")
    public void listen1(String in) {
        System.out.println(in);
    }
}

@Configuration
@EnableKafka
public class Config {

    @Bean
    ConcurrentKafkaListenerContainerFactory<Integer, String> kafkaListenerContainerFactory(ConsumerFactory<Integer, String> consumerFactory) {
        ConcurrentKafkaListenerContainerFactory<Integer, String> factory =
            new ConcurrentKafkaListenerContainerFactory<>();
        factory.setConsumerFactory(consumerFactory);
        return factory;
    }

    @Bean
    public ConsumerFactory<Integer, String> consumerFactory() {
        return new DefaultKafkaConsumerFactory<>(consumerProps());
    }

    private Map<String, Object> consumerProps() {
        Map<String, Object> props = new HashMap<>();
        props.put(ConsumerConfig.BOOTSTRAP_SERVERS_CONFIG, "localhost:9092");
        props.put(ConsumerConfig.GROUP_ID_CONFIG, "group");
        props.put(ConsumerConfig.KEY_DESERIALIZER_CLASS_CONFIG, IntegerDeserializer.class);
        props.put(ConsumerConfig.VALUE_DESERIALIZER_CLASS_CONFIG, StringDeserializer.class);
        props.put(ConsumerConfig.AUTO_OFFSET_RESET_CONFIG, "earliest");
        //...
        return props;
    }

    @Bean
    public Sender sender(KafkaTemplate<Integer, String> template) {
        return new Sender(template);
    }

    @Bean
    public Listener listener() {
        return new Listener();
    }

    @Bean
    public ProducerFactory<Integer, String> producerFactory() {
        return new DefaultKafkaProducerFactory<>(producerProps());
    }

    private Map<String, Object> producerProps() {
        Map<String, Object> props = new HashMap<>();
        props.put(ProducerConfig.BOOTSTRAP_SERVERS_CONFIG, "localhost:9092");
        props.put(ProducerConfig.LINGER_MS_CONFIG, 10);
        props.put(ProducerConfig.KEY_SERIALIZER_CLASS_CONFIG, IntegerSerializer.class);
        props.put(ProducerConfig.VALUE_SERIALIZER_CLASS_CONFIG, StringSerializer.class);
        //...
        return props;
    }

    @Bean
    public KafkaTemplate<Integer, String> kafkaTemplate(ProducerFactory<Integer, String> producerFactory) {
        return new KafkaTemplate<Integer, String>(producerFactory);
    }
}
```

CICS application considerations when interacting with Kafka

1. *Transactional scope* between Kafka and resources being updated in CICS, Db2, IMS, MQ
 - Background reading: [Does Apache Kafka do ACID transactions?](#) and [Transactions in Apache Kafka](#)
2. *Network latency and capacity* to the Kafka instance that may adversely effect CICS transaction response times.
 - Ensure using connection pooling eg. using a Java object pooling framework
3. *Network reliability* to the Kafka instance that may adversely effect CICS transaction availability.
4. *Facilities to format / transform the data* appropriate for consumption by consumers of the topic, for example from EBCDIC binary format into JSON.
5. *Skills* to develop Java applications and setup CICS.
6. *Existing use of products* such as IBM MQ for z/OS and z/OS Connect EE.

CICS use case with Kafka

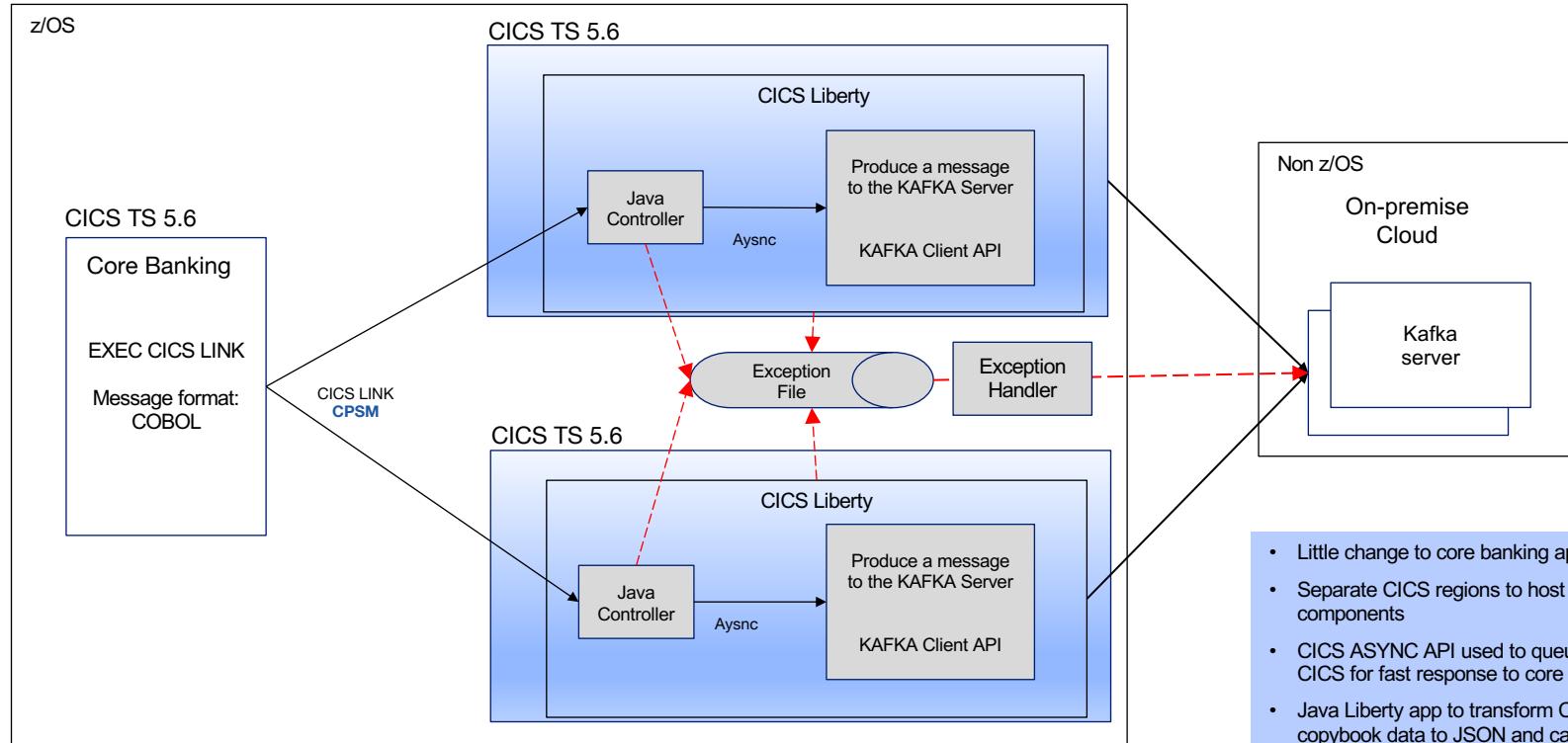
Business Case

- Implement a financial portal system in the on-premise Cloud for real time data analysis for corporate banking users.
- KAFKA is defined for the data source for the new system and there is no MQ in mainframe.
- Traceability is critical to resolve all data sync issues.
- Cost effective not to increase the M/F cost.

Design Principles

- Utilizing zIIP engine for data producer clients.
- HA configuration for the outbound client.
- Async from the main backend business not to have any performance impact to the core system.
- Eventually all data must be sent out to the KAFKA.

CICS – Kafka Case : CICS COBOL / Liberty Java / Kafka

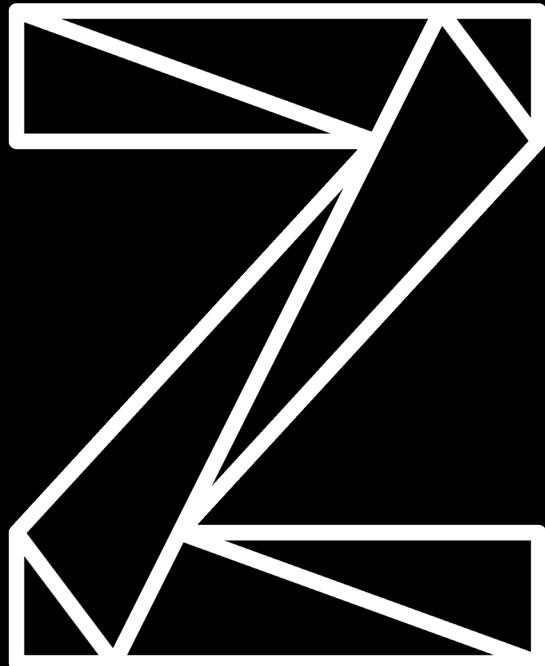


- Little change to core banking app
- Separate CICS regions to host new components
- CICS ASYNC API used to queue work in CICS for fast response to core banking
- Java Liberty app to transform COBOL copybook data to JSON and call Kafka client API in Java
- Exception handling with zFS files

New option for KAFKA runtime

KAFKA for z/OS

- To run Apache Kafka on z/OS, your z/OS operating system must meet the following requirements: z/OS version 2.4 or later. With z/OS 2.4, the following fixes must be installed:
 - OA60306/UJ90013
 - OA60310/UJ05191
 - OA60316/UJ05214
 - PH32235/UI74844
- One of the following Java™ SDK Products on z/OS must be installed and operational on your z/OS system:
 - IBM® 64-bit SDK for z/OS, Java Technology Edition, V8 SR6 FP20 (5655-DGH).
 - IBM Semeru Runtime Certified Edition for z/OS, Version 11.0 (5655-DGJ).
- It is recommended to use the latest service release. To find the latest service release or fix pack, see [Java SDK Products on z/OS](#).
- Bash 4.3 or later. You can download Bash from the [Rocket software website](#).



Limitations

- Apache Kafka provided producer(`kafka-console-producer.sh`) and consumer(`kafka-console-consumer.sh`) console commands do not work on z/OS USS.
- Develop Java programs that can issue these commands instead.

For more details on Installation and configuring Kafka on z/OS, refer :

<https://www.ibm.com/docs/en/z-logdata-analytics1/5.1.0?topic=configuring-setting-up-apache-kafka-z>

Kafka on z/OS

DEMO

