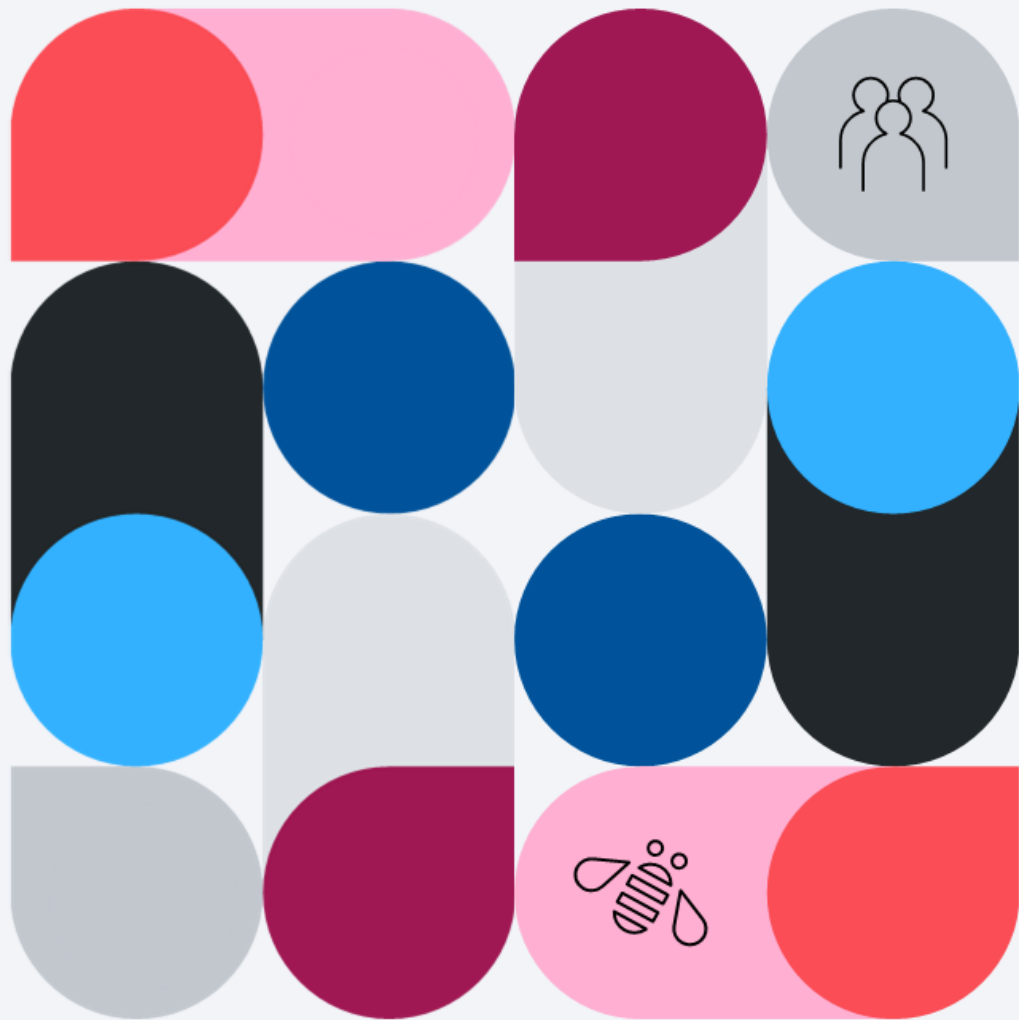# IBM Community Festival 2021

—

17 November 2021

How and Why to Modernize in Place with IBM Z

Elizabeth K. Joseph
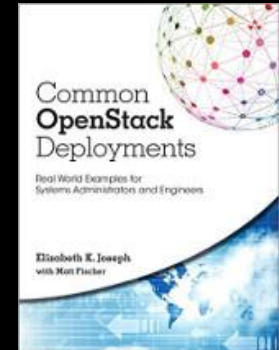@pleia2

IBM

Elizabeth K. Joseph

Linux Systems Administrator by trade

Working as a Developer Advocate on IBM Z

An open Source Software Contributor

Member of the Open Mainframe Project

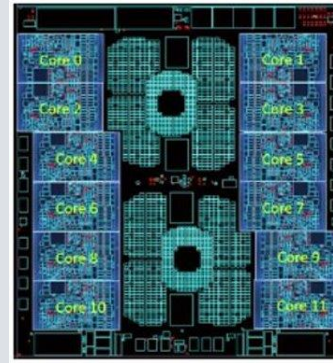Author of books on Ubuntu and OpenStack

What is IBM Z?

z/Architecture (not x86)

Mainframe

Big Iron

A big computer

A large server (but now fits in a 19" rack footprint!) built with data processing, cryptography, and security in mind.

- 190 5.2ghz processor units (PUs), with 12 cores per chip

- 40TB of RAM

- 60 PCIe control units across 12 PCIe I/O drawers

- 22 dedicated I/O offload processors (SAPs) pre-allocated per system

Various Operating Systems

Traditional mainframe operating systems:

- z/OS

- z/VM

- z/TPF

+ Linux:

SUSE Enterprise Linux

Red Hat Enterprise Linux

Ubuntu

Plus:
- Debian
- openSUSE
- Fedora
- ClefOS

"Modernization"

IBM Z is already modern!

What I'm talking about is:

- Application Modernization

- Process Modernization

- Hybrid Cloud

Transforming the way you use your Z to better integrate with cloud, DevOps practices, and all kinds of other modern tooling and processes.

Modernization: Everyone wants everything to be fast

Companies need to introduce new technologies, more quickly.

More interconnectedness.

Developers also expect to get up to speed with new systems quickly, which is tricky when your toolset is not similar to the rest of the tech industry.
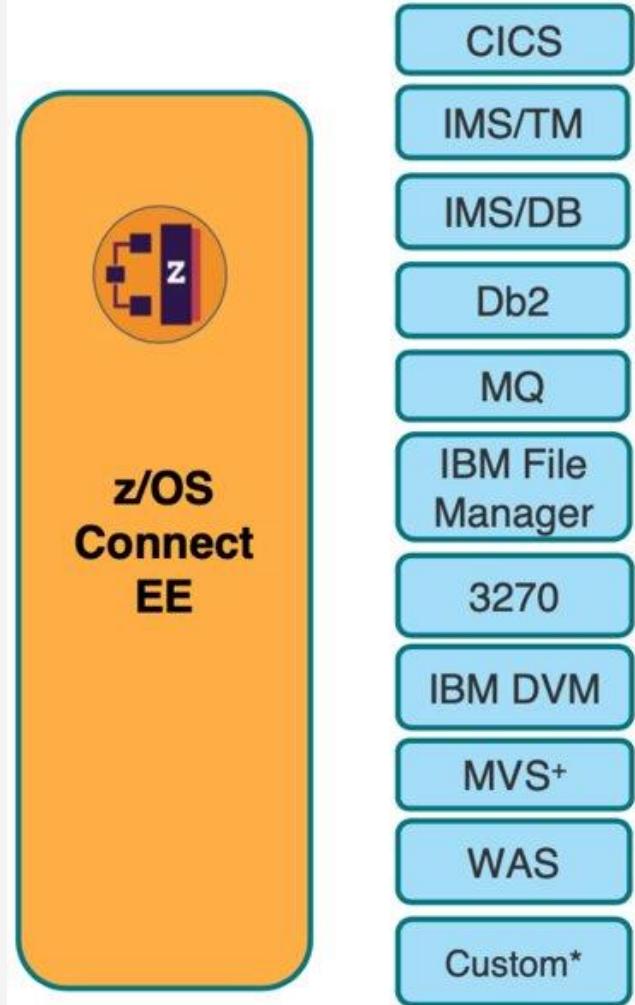
We also need to empower developers to self-provision environments and tooling so they're never left waiting.

Application Modernization: "RESTification"

Expose assets hosted on z/OS to other parts of your hybrid cloud with REST APIs with z/OS Connect EE.

https://www.ibm.com/products/zos-connect-enterprise-edition

CICS
IMS/TM
IMS/DB
Db2
MQ
IBM File Manager
3270
IBM DVM
MVS+
WAS
Custom*

z/OS Connect EE

* Other Vendors or your own implementation

Process
Modernization: IDEs

Syntax highlighting and real-time checking

Problem views

Unreachable code detection

Variable declaration hovers and 'go to declaration'

Code and variable completion

Refactoring like 'rename symbol'

Code snippets for easy code reuse

Runtime debugging with breakpoints and variable tracing

Process Modernization: Putting polish on traditional languages

Developers expect IDE integration, but what if you're developing COBOL?

No problem!

IBM Z Open Editor: https://ibm.github.io/zopeneditor-about/

Open Mainframe Project COBOL Course: https://www.openmainframeproject.org/projects/cobolprogrammingcourse

COBOL Programming
with VSCode on Coursera: https://www.coursera.org/learn/cobol-programming-vscode

Build Your First COBOL Application: https://developer.ibm.com/components/ibmz/patterns/my-first-cobol-application/

Process Modernization: DevOps tooling

Goal: Make z/OS applications just another part of your CI/CD pipeline, fueled by the same tools you are probably already familiar with:

- Jenkins
- Git
- Ansible

Create a Fully Automated CI/CD Pipeline for z/OS Testing
https://developer.ibm.com/components/ibmz/tutorials/jenkins-ibm-zos-connector-plugin-fully-automated-cicd-pipeline-for-zos-testing/

Build a pipeline with Jenkins, Dependency Based Build, and UrbanCode Deploy
https://developer.ibm.com/components/ibmz/tutorials/build-a-pipeline-with-jenkins-dependency-based-build-and-urbancode-deploy/

Check out the "IBM z/OS Automation via Red Hat® Ansible®" Z Trial: https://www.ibm.com/it-infrastructure/z/software-trials

Process Modernization: IBM Wazi Developer

IBM Wazi Developer is made up of:

- **Wazi Code**: Edit, build, and debug in your choice of VS Code™, Eclipse®, or Red Hat CodeReady Workspaces IDEs.
- **Wazi Sandbox**: Develop and test z/OS application components in a personal, virtual, containerized OpenShift z/OS environment.
- **Wazi Analyze**: Understand the impact of potential application changes by discovering the relationships between application components.

https://www.ibm.com/products/wazi-developer

Bring it all together with the "Bring Your Own (BYO) IDE for Cloud Native Development" Z Trial:

With this trial of IBM Wazi Developer and IBM Developer for z/OS Enterprise Edition you can choose your preferred IDE (Eclipse® or Microsoft® VS Code™) integrated with familiar DevOps tools such as Git and Jenkins to develop a z/OS application. https://www.ibm.com/it-infrastructure/z/software-trials

Hybrid Cloud: Red Hat OCP

Red Hat OpenShift Container Platform (OCP) on IBM Z + IBM Cloud Paks

- Get a free, temporary OCP developer environment via https://developer.ibm.com/gettingstarted/ibm-linuxone/

Hybrid Cloud: Hyper Protect

IBM Cloud Hyper
Protect Services

- Hyper Protect
  Virtual Servers

- Hyper Protect
  Crypto Services

- Hyper Protect
  DBaaS

IBM Z and Cloud Modernization Center

- Case studies

- Demos

- Tutorials and code patterns

- Badges

- And more!

https://www.ibm.com/community/z-and-cloud/

Continuing your modernization journey with…

IBM Z Xplore @ https://www.ibm.com/it-infrastructure/z/education/zxplore

Learn traditional, core IBM Z concepts while using VS Code, and new tooling like Zowe, Ansible, and Docker containerized environments.

IBM Z Community @ https://community.ibm.com/z

Blog posts, forums, library, and more covering

- Hardware (IBM Z & LinuxONE)

- Solutions

- Software

- Languages

- Open Source

Thank you

Elizabeth K. Joseph
Developer Advocate
—
lyz@ibm.com
ibm.com