DevOps on z Proof of Technology

David Hawreluk
dhawrel@us.ibm.com

Vanessa Ramirez Callaghan
vanessa.r.callaghan@ibm.com

Carlos a Hirata
chirata@us.ibm.com

Wilbert Kho
wilbert@us.ibm.com

Audience

- This Proof of Technology is targeted to, but not limited to, architects, technical specialists or developers
- Non-technical attendees with an understanding of application lifecycle management are welcome

Pre-requisites

- Familiarity with z/OS concepts
- Awareness (not necessarily proficiency) of basic z/OS System Programming tasks
- Basic familiarity with JCL concepts (JCL skills NOT required)
- No Java, COBOL or PL/I skills are required
- Developers are welcome

Proof of Technology Objectives

The objective of this session is to demonstrate through hands-on workshops, the power of collaborative development and delivery enabled by the Enterprise DevOps solutions.

Application teams can learn how they can modernize their application portfolio using modern technology and tools. We will focus on code and build, automate testing and deploy.

Production teams can learn how they can extend and automate their existing build and deployment solutions for enterprise applications.

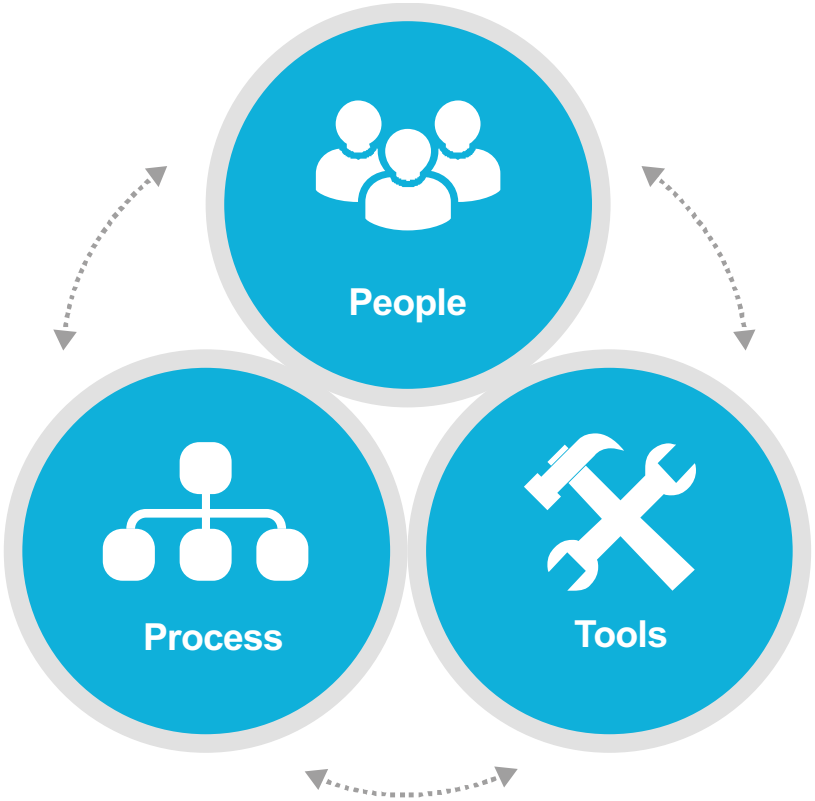
Areas covered:

- Enterprise analytical tool for application understanding and modernization (ADDI)
- Integrated application development and problem analysis (ADFz)
- Managing your source code using modern technology (including Git)
- Building automation (DBB) for COBOL without a specific source code manager but using a pipeline automation tool (including Jenkins)
- Using a unit testing framework (zUnit) for COBOL as part of the development and build environment.
- Using IBM Z Virtual Test Platform (VTP) for application Integration Testing of a COBOL/CICS/DB2 application
- Running Integration tests using Galasa framework
- Application deployments automation to many environments (UCD)

Introductions

- Introduce yourself and your team:
  - What's your role within your organization? (Architect, Developer, DBA, Manager, etc.)
  - Current development tools set in use? (IDz, TSO, SCLM, Changeman, Endevor)
- What do you hope to get out of this session?

DevOps is not one of these things, it's *all* of them!



DevOps is a set of practices that combines software development (Dev) and IT operations (Ops). It aims to shorten the system development life cycle and provide continuous delivery with high software quality.

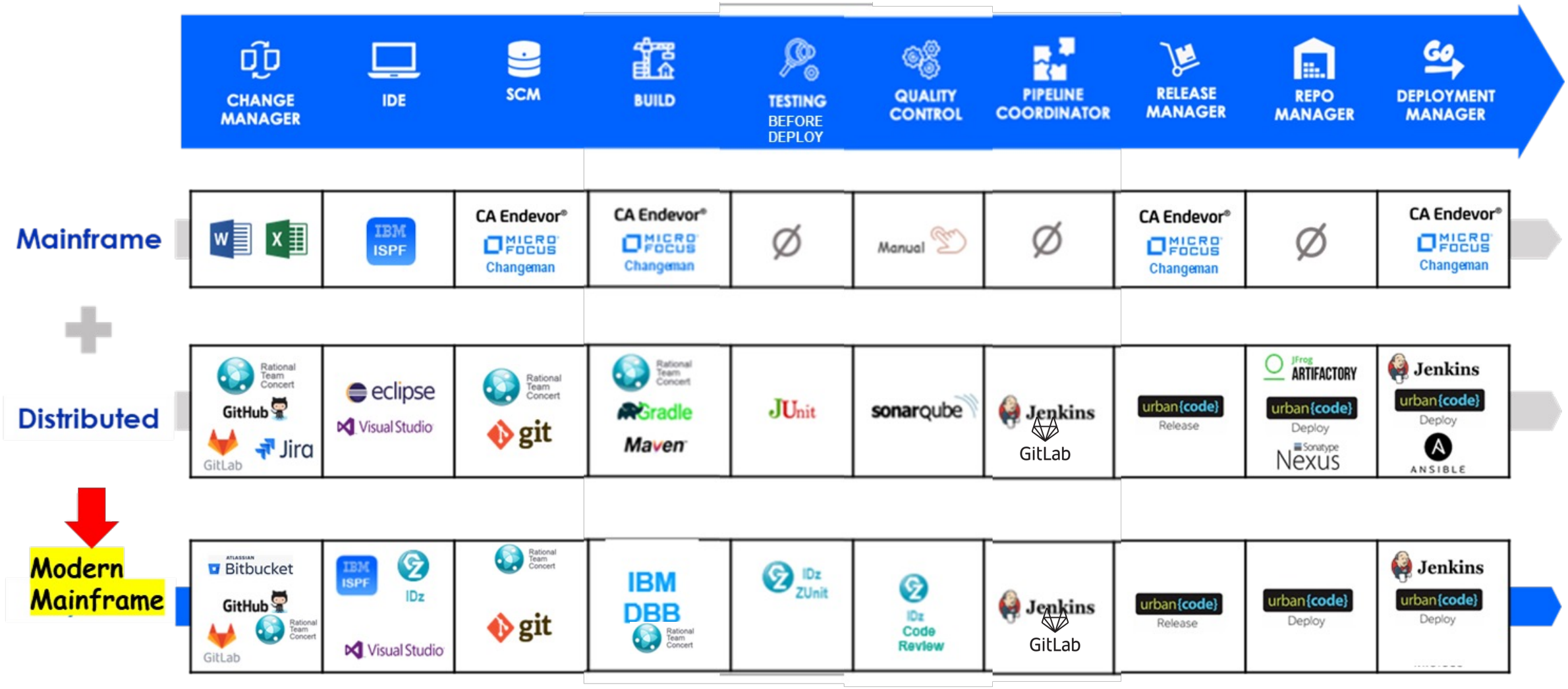
Why DevOps for z/OS?

- Client Value through increased quality
  - Automated test (at any level) at code delivery and
  - Code reviews will include results of build, package and tests

- Modernizing to industry standard tools
  - Enables z/OS to leverage relevant open source tooling. E.g *Jenkins* and *Git*
  - Unlocks an ecosystem not currently available to z/OS dev *
  - Decreased time to market

- Retention and Skills
  - New hires start out knowing some of the DevOps tools
  - Using homegrown, hard to maintain, proprietary tools is a problem
  - Lack of automation increases the time until a new team member is contributing fully
  - Current tools support not sustainable

Imagine a unified pipeline on z/OS



Topology used on the labs

zD&T on Cloud

z/OS

DBB

Jenkins Agent

UCD-Agent

DBB App server

Jenkins

UCD Server urban{code}

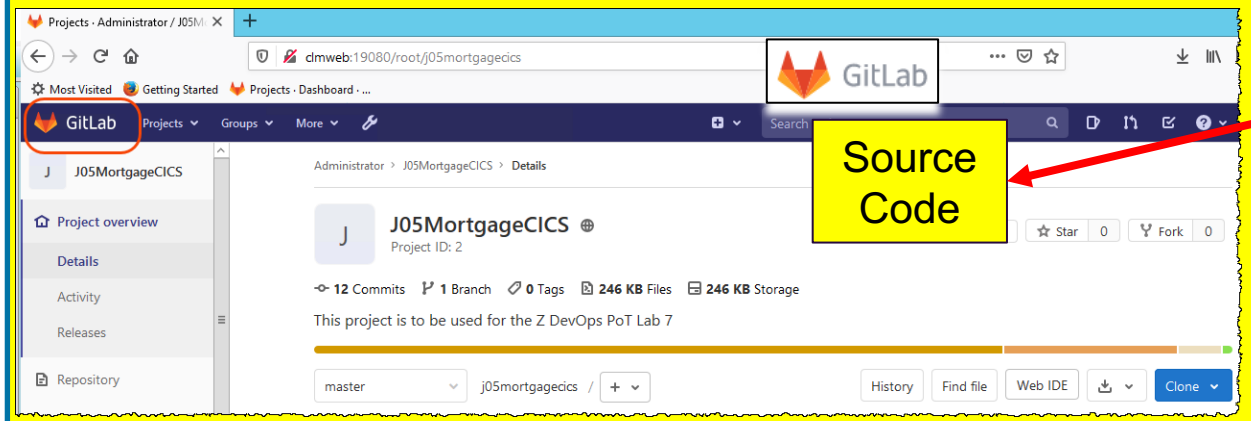
Executables

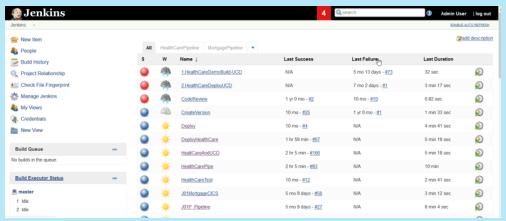
Windows

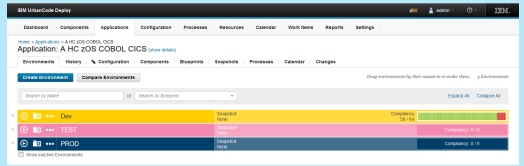
Developer using ADFz

Git

Source Code

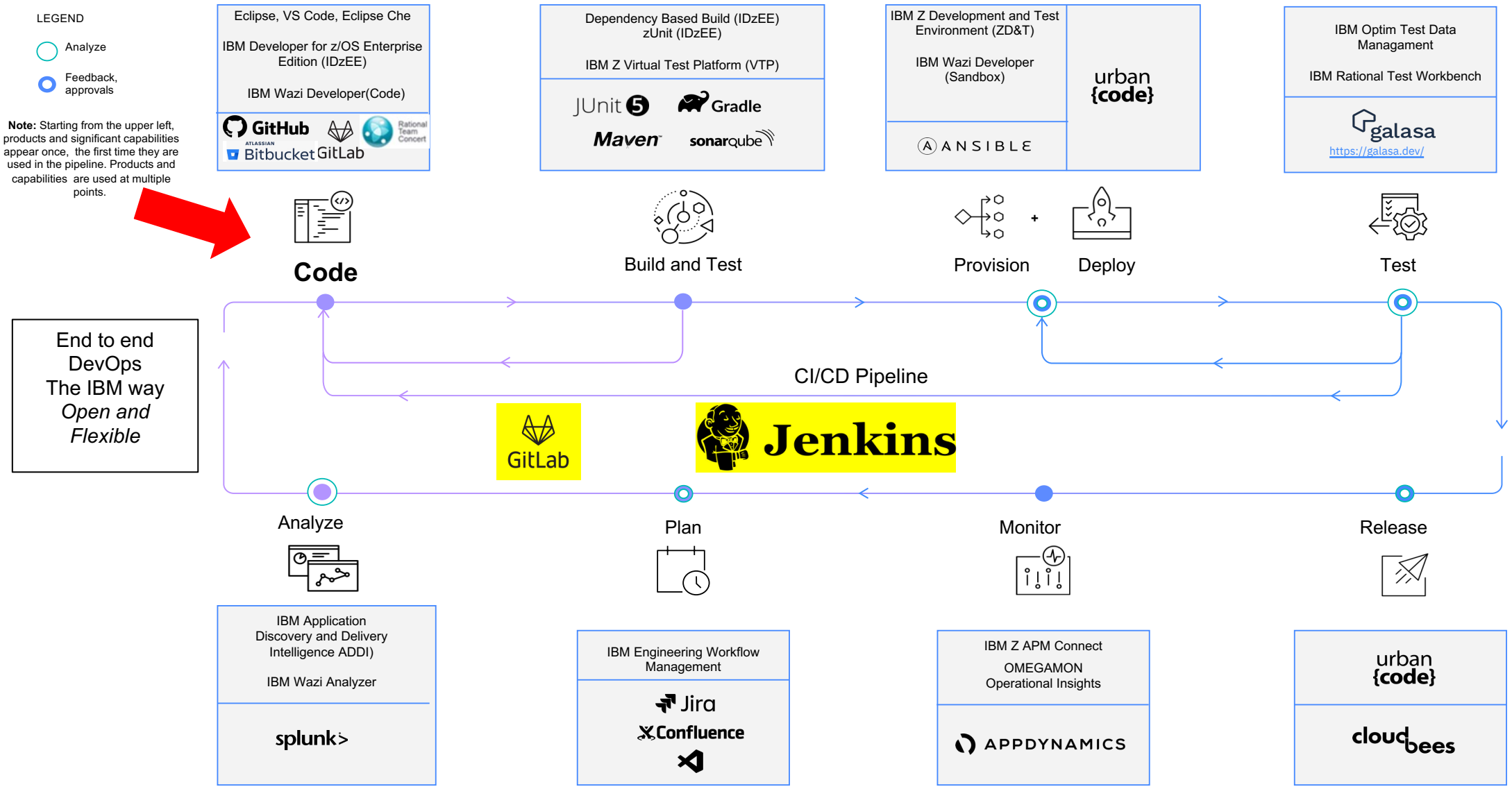






Linux

Z DevOps Pipeline



What is Jenkins ?

“Jenkins is a self-contained, open-source automation server which can be used to automate all sorts of tasks related to building, testing, and delivering or deploying software.”

Note: Jenkins is not the only orchestrator. You can use Azure DevOps, TeamCity, Bambo, etc.

Jenkins

Ref: https://www.jenkins.io/doc/

What is Git?

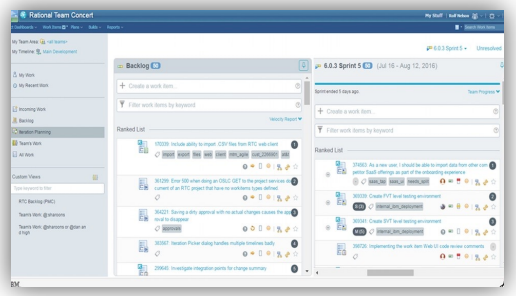
- A distributed, open-source version control system
  - Runs on all platforms (z/OS, Linux, Windows, Mac)
    - z/OS uses Rocket® open-source implementation on USS
  - Command line interface
  - Community supported

- The de-facto standard for software development
  - Integrated into many tools
    - Jenkins, Slack, Urban Code Deploy, Ansible, Artifactory, Jira, JUnit, Maven, Gradle, Make, DBB, Azure and many more.

z/OS and subsystems

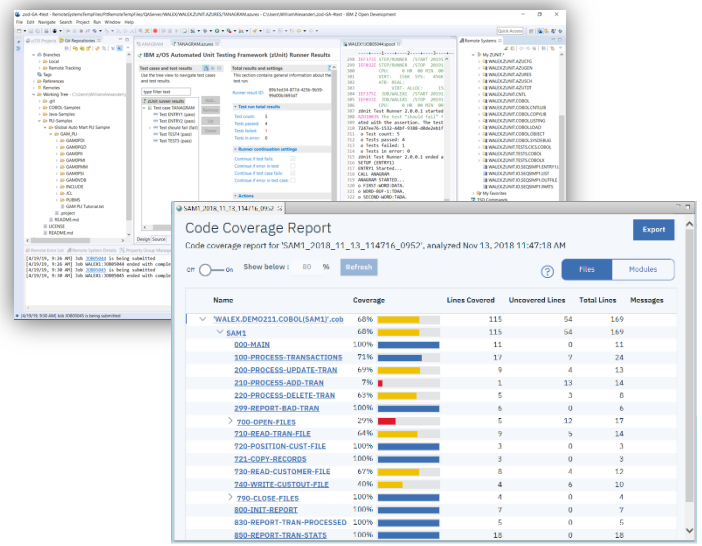
Build and SCM

Integration with Git, RTC and Endevor for Lifecycle and Source Management

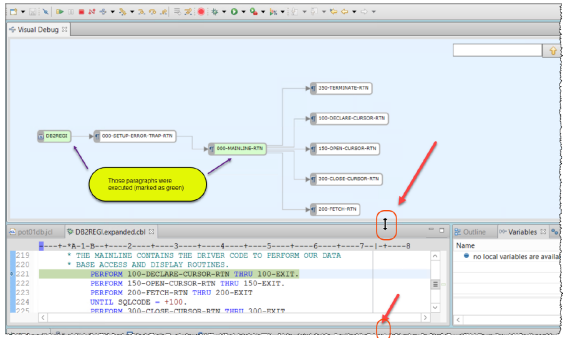


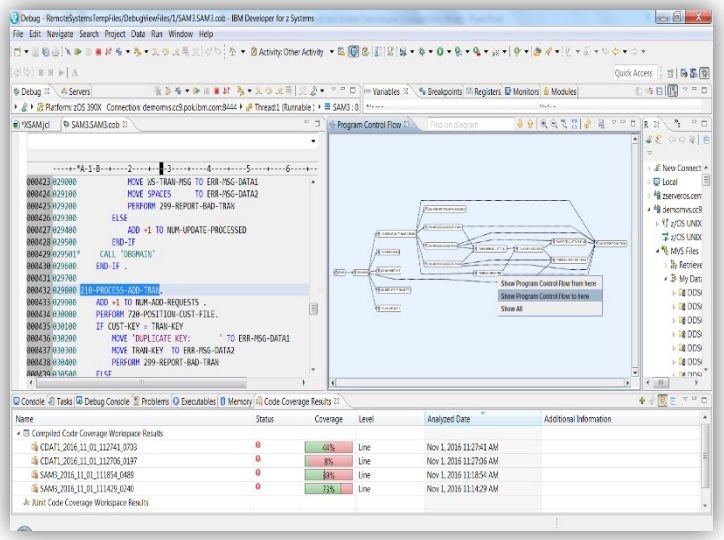
IBM Developer for z/OS Enterprise Edition- the developer's cockpit!

Unit test and code coverage

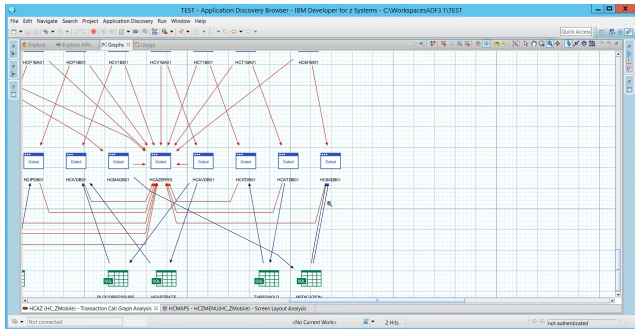


Debug and code coverage

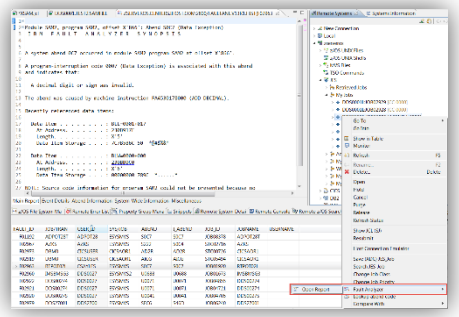




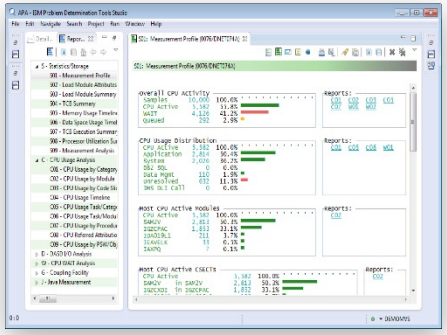
Application Discovery



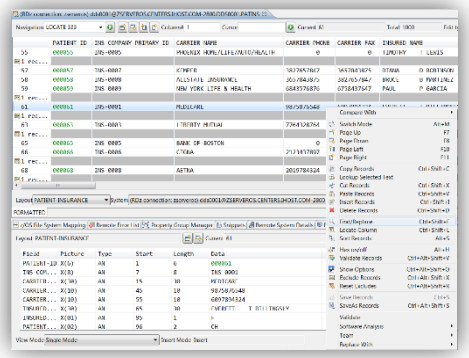
Abend analysis



Performance analysis



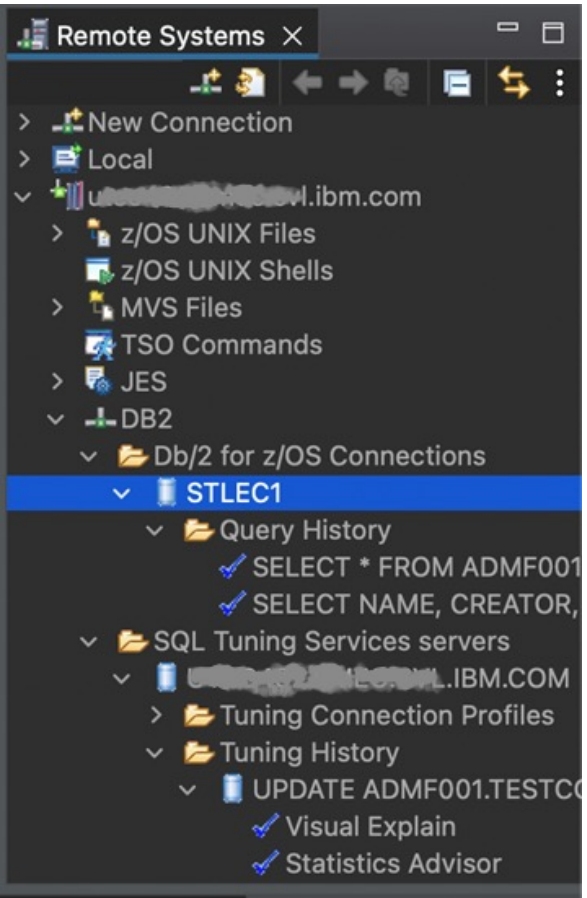
File Management



IDz - Db2 for z/OS Application Development

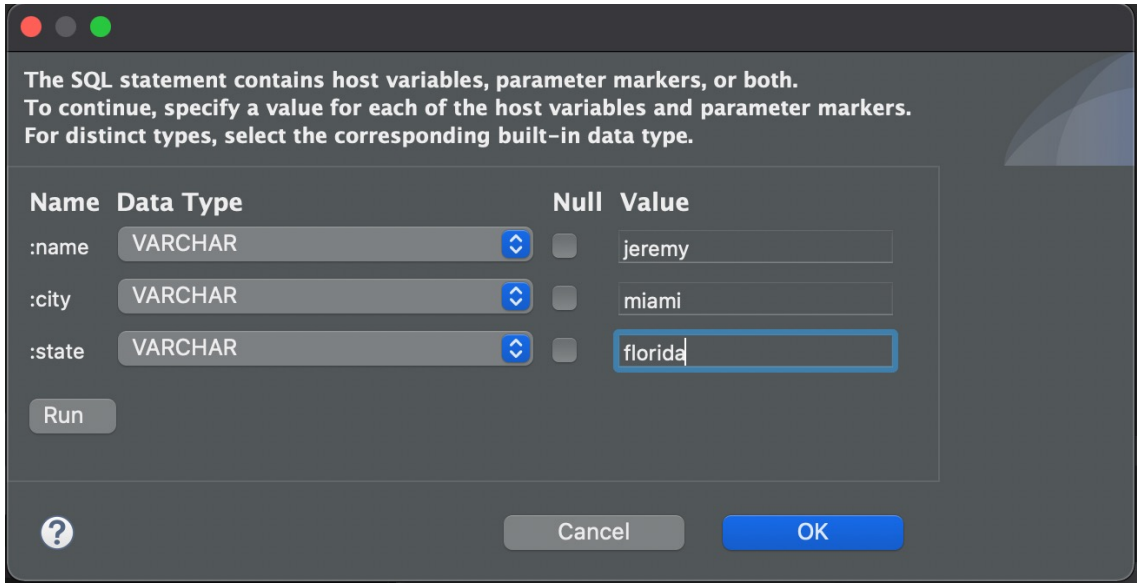
Db2 for z/OS connection integration with RSE view makes it easier to share credentials between RSE and Db2

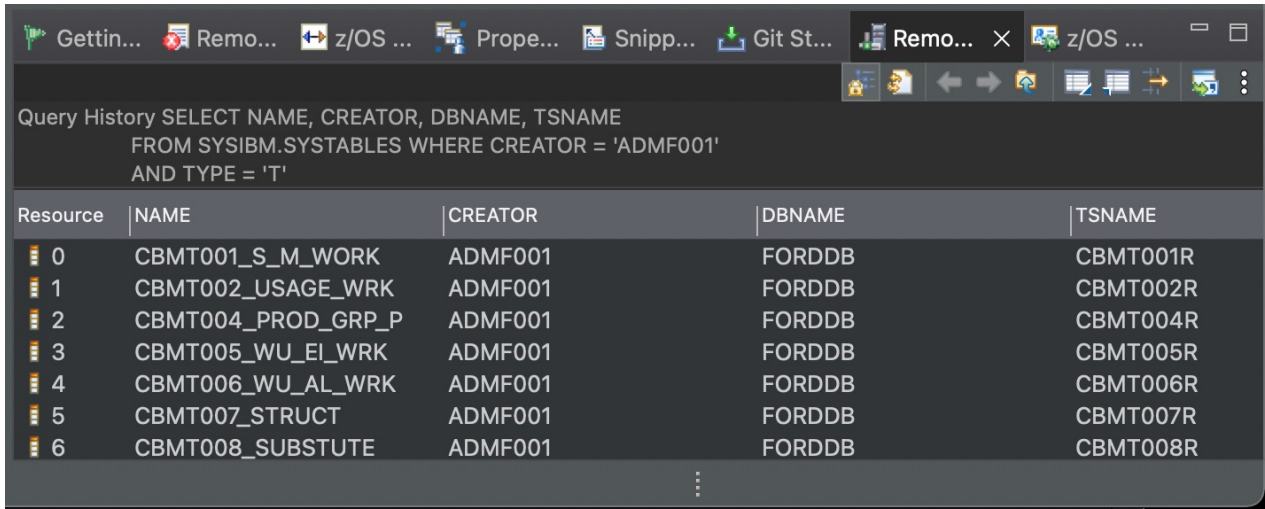
Connections to Db2 Accessories Suite provides SQL Tuning Services such as Visual Explain



While editing COBOL or PL/I source files users can highlight an SQL statement, run it and browse the results.

If the SQL statement contains host variables a prompt allows for input of values to use during execution.

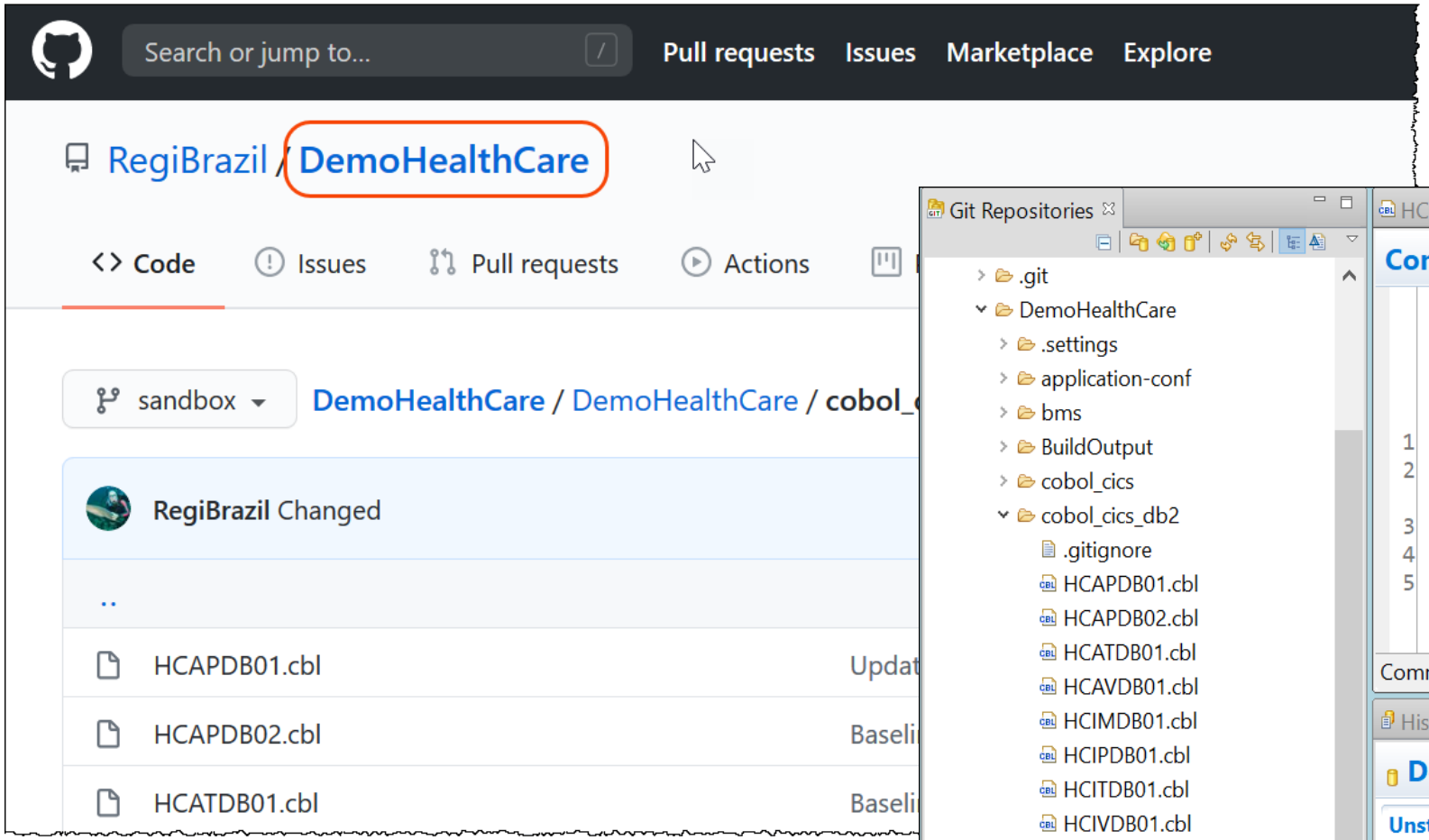


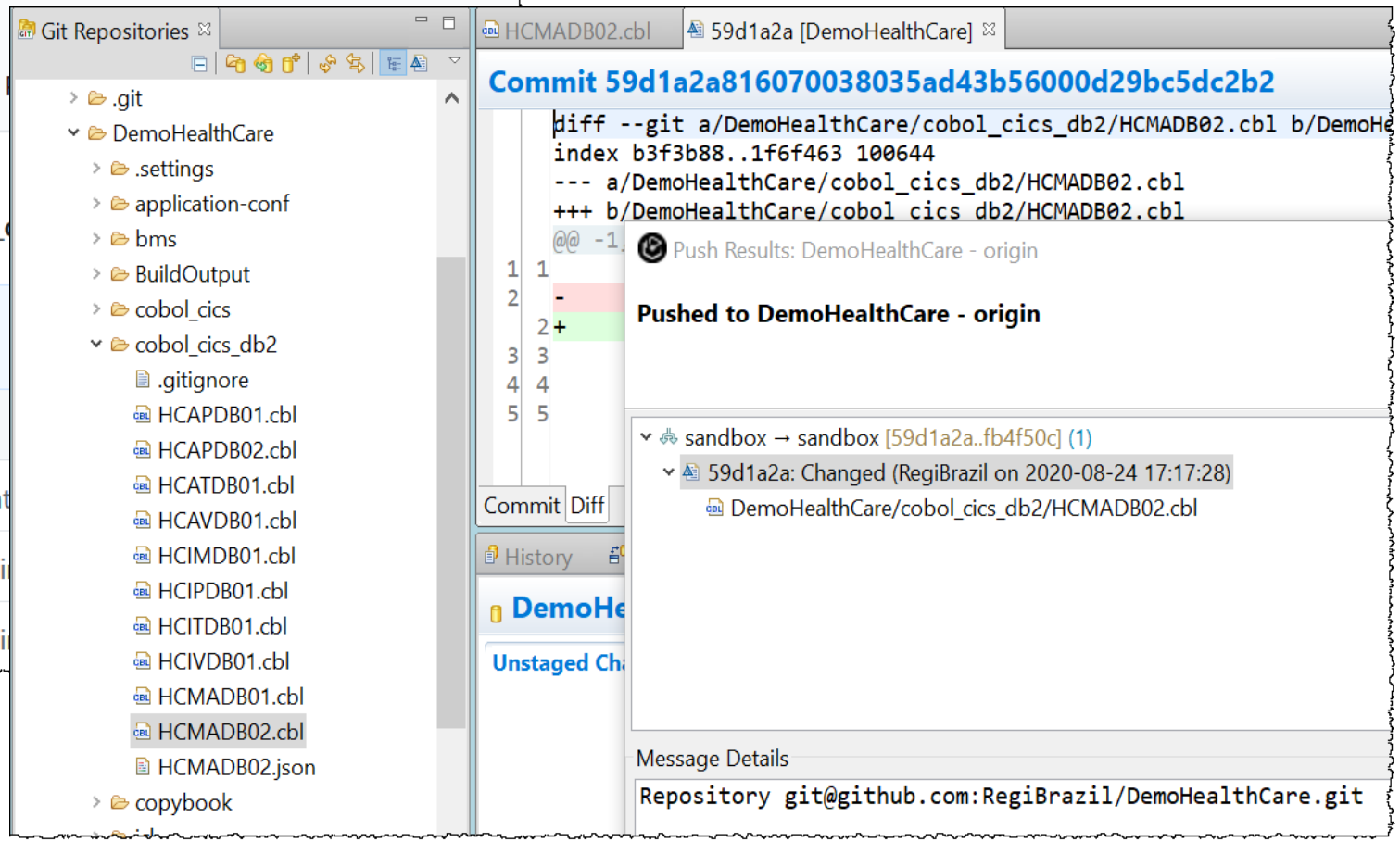


Source Code Management

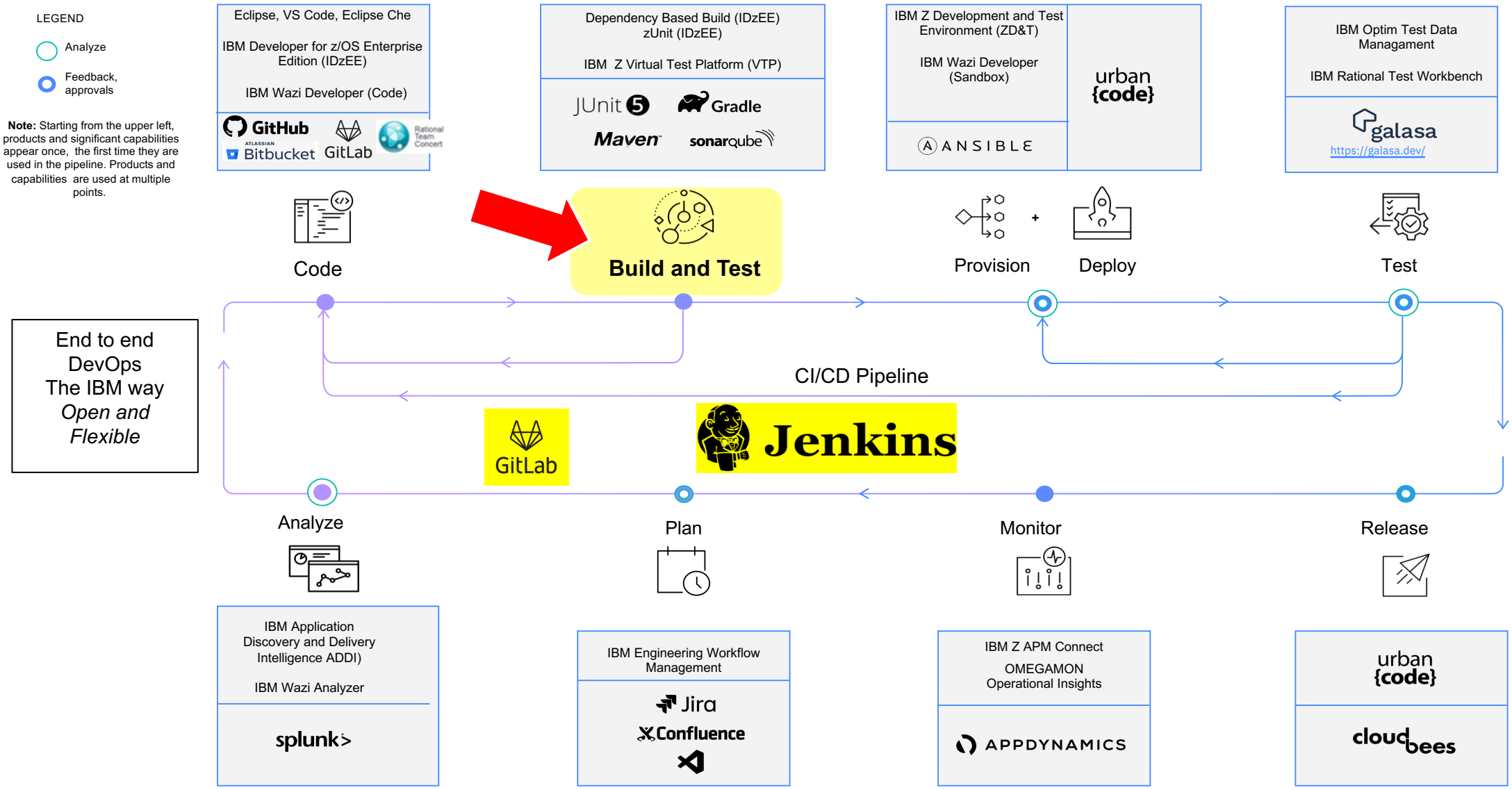
EGit included in IDz

IBM **IDz** offers out-of-the-box **integration** for the IBM Rational Team Concert, CA **Endevor** and IBM SCLM.



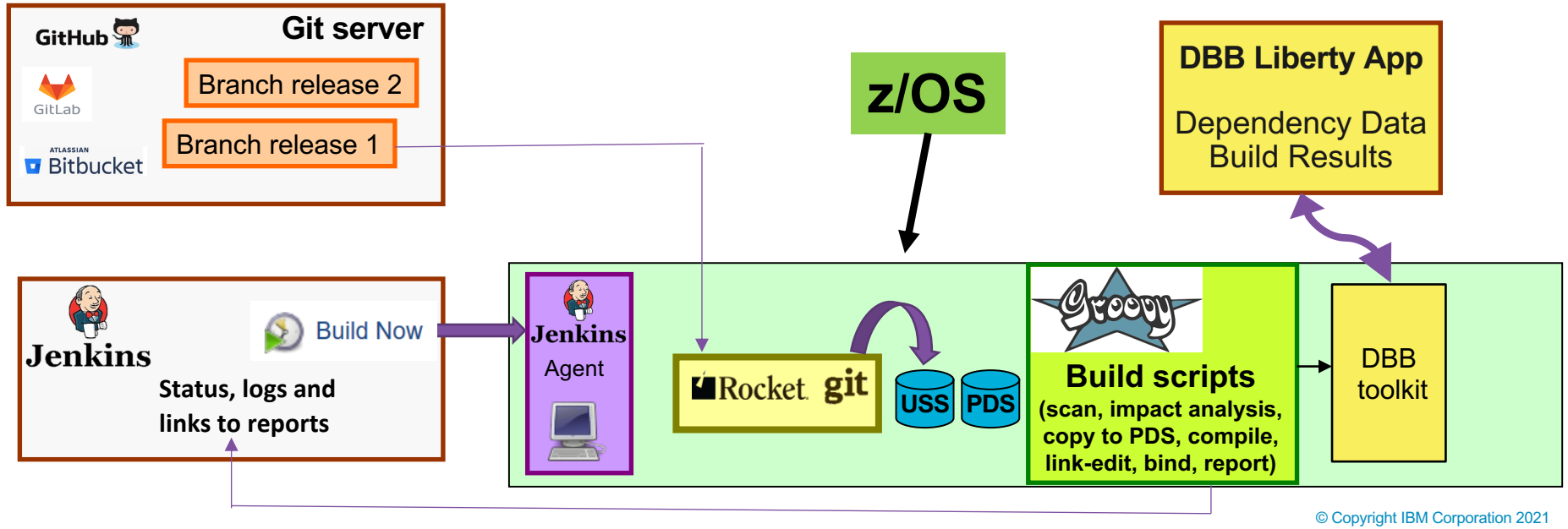


Z DevOps Pipeline



What is IBM Dependency Based Build (DBB)?

- Provides a modern scripting languagebased automation capability that can be used on z/OS.
  - The DBB API can be called by Java based scripting languages such as **Groovy**, JRuby, Jython, Ant, Maven, Python, etc.
- Consists of a build toolkit (Java API, Groovy Installation) installed on USS (z/OS) and a Liberty application server that hosts build metadata (dependency data, build results) installed on Linux.



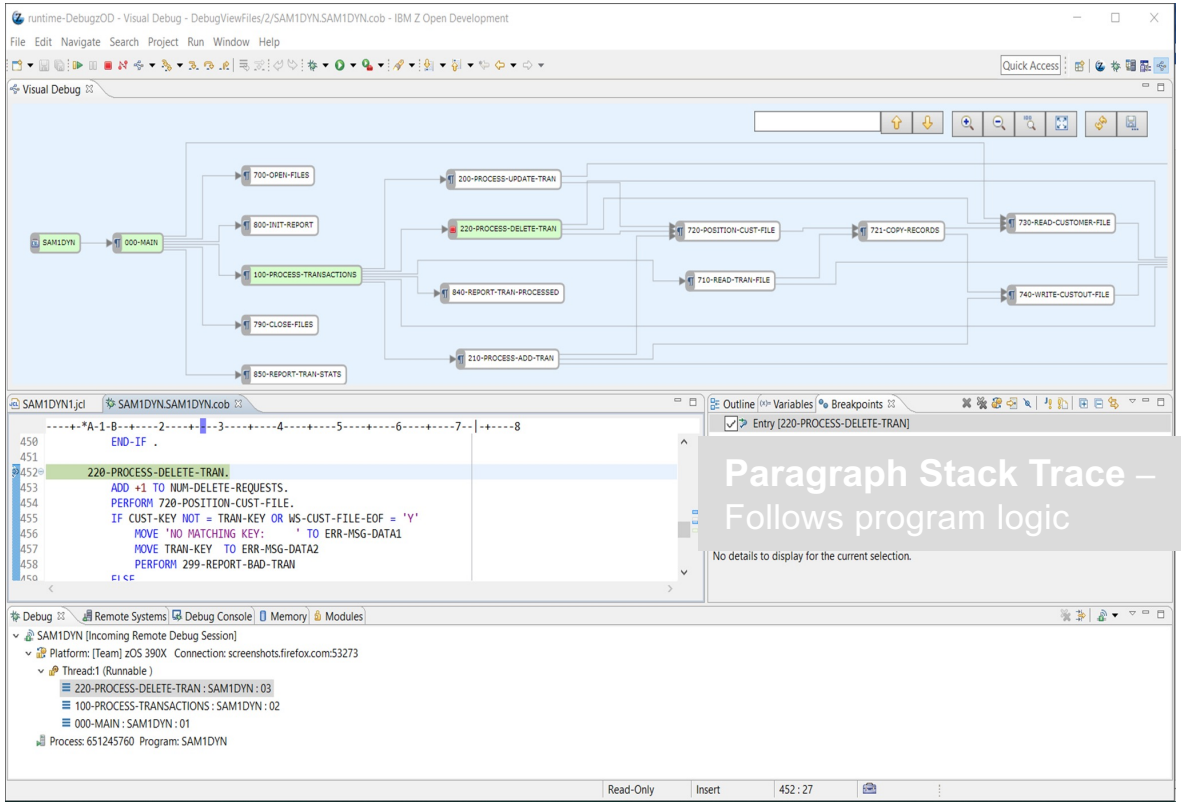
Diagnose (z/OS Debugger)

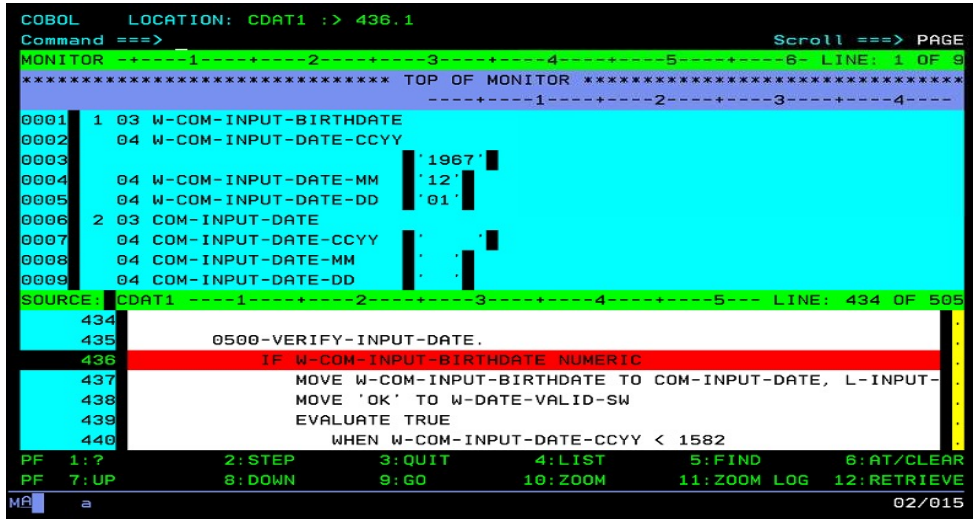
Program testing and analysis

Helps you examine, monitor, and control the execution of application programs on z/OS

With:

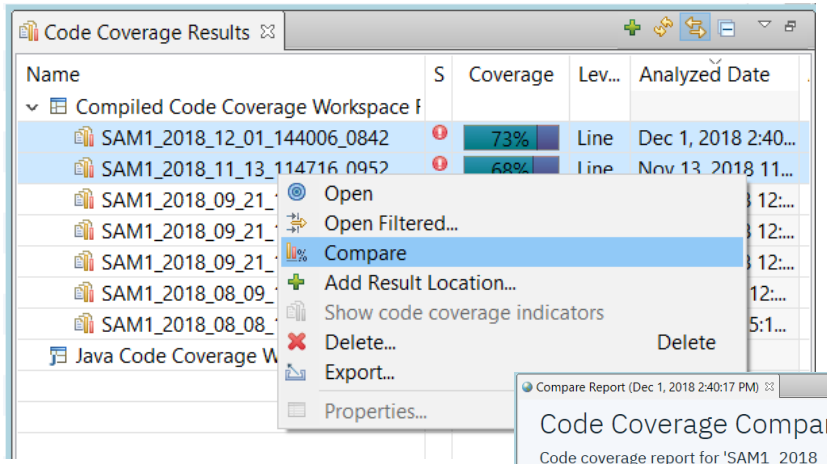
- Visual debug
- Code coverage facilities
- Eclipse GUI
- 3270 UI
  (Debug for z/OS and IDZ-EE)





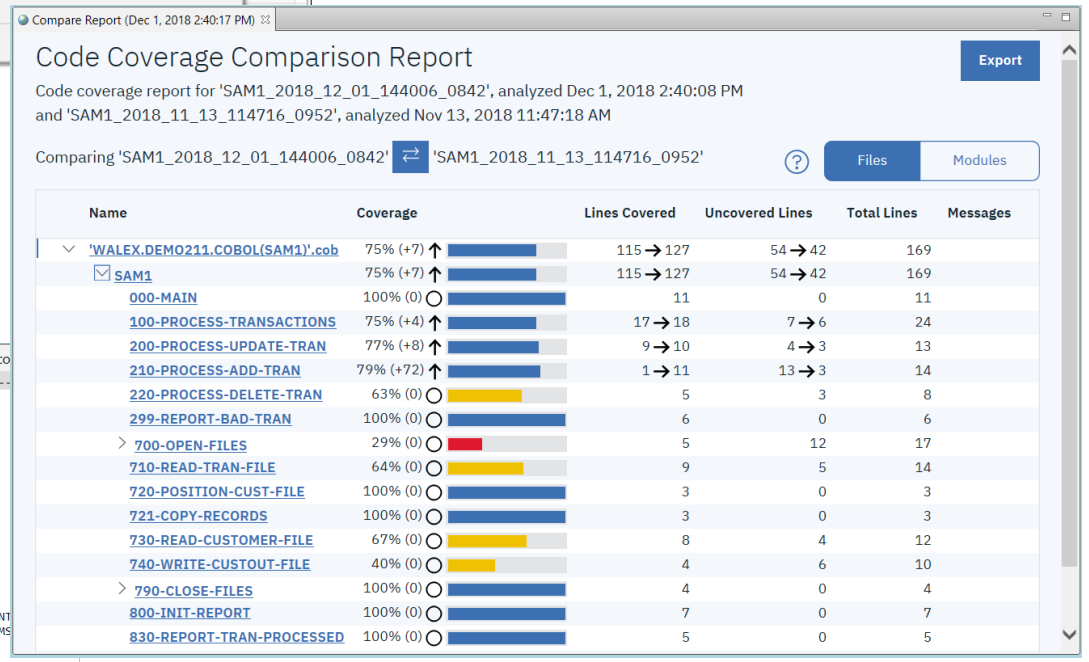
Test (code coverage)

- Tracks lines of code that have been executed during test
- Improves application testing quality
- Focuses testing resource usage
- Numerous reports and options to assess testing and trends
  - Source view
  - Summary
  - Comparisons
  - Merging
- Supports: Batch, CICS and IMS
- Integration with ADDI and SonarQube
- Run in batch as part of your DevOps pipeline, with RESTful APIs for Debug profile creation

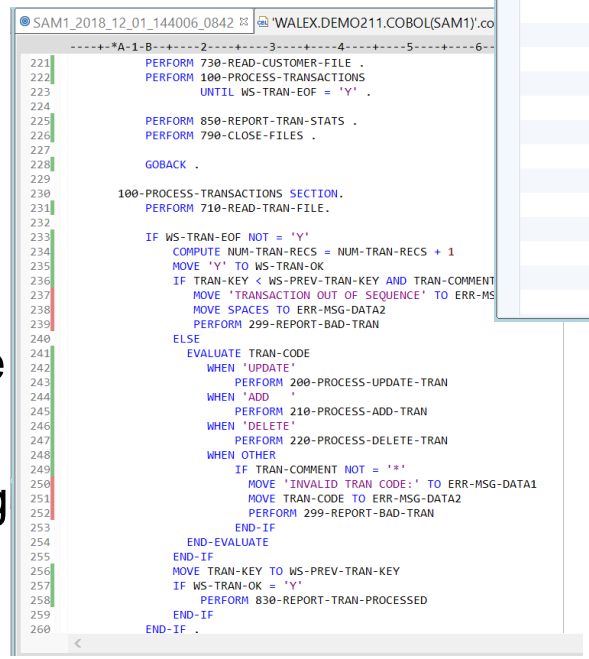


Compare results

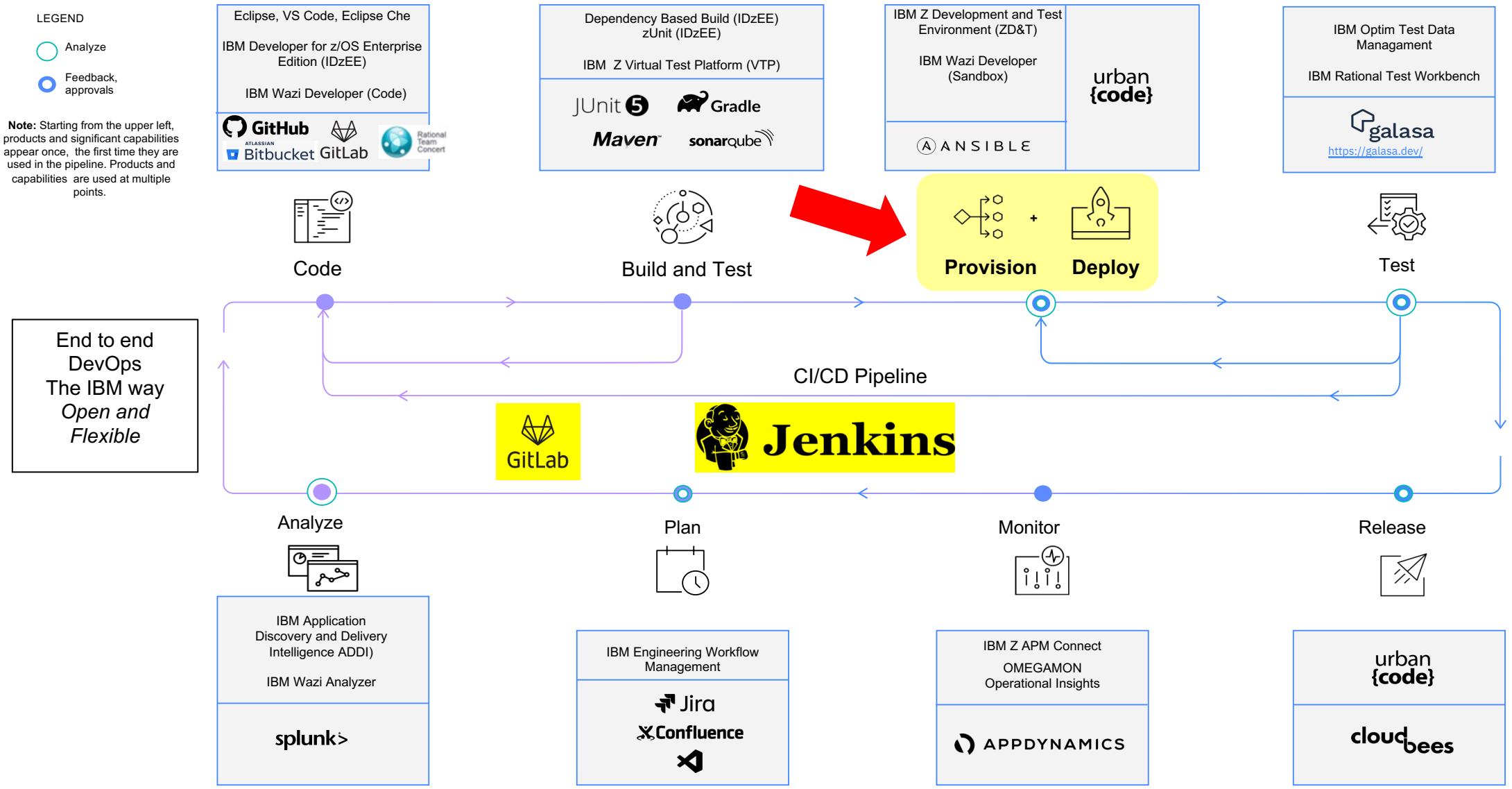
New! Web UI look and feel with easier filtering, grouping and comparisons, plus PDF output (14.1.5)



Coverage



Z DevOps Pipeline

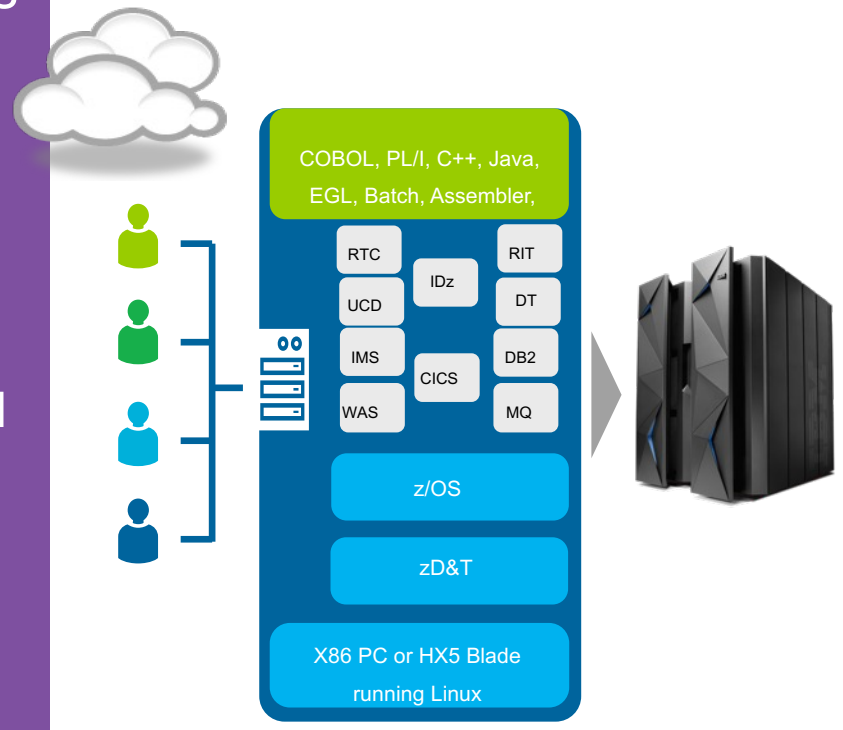


Provisioning

—Shift left with **Z Development and Test Environment**

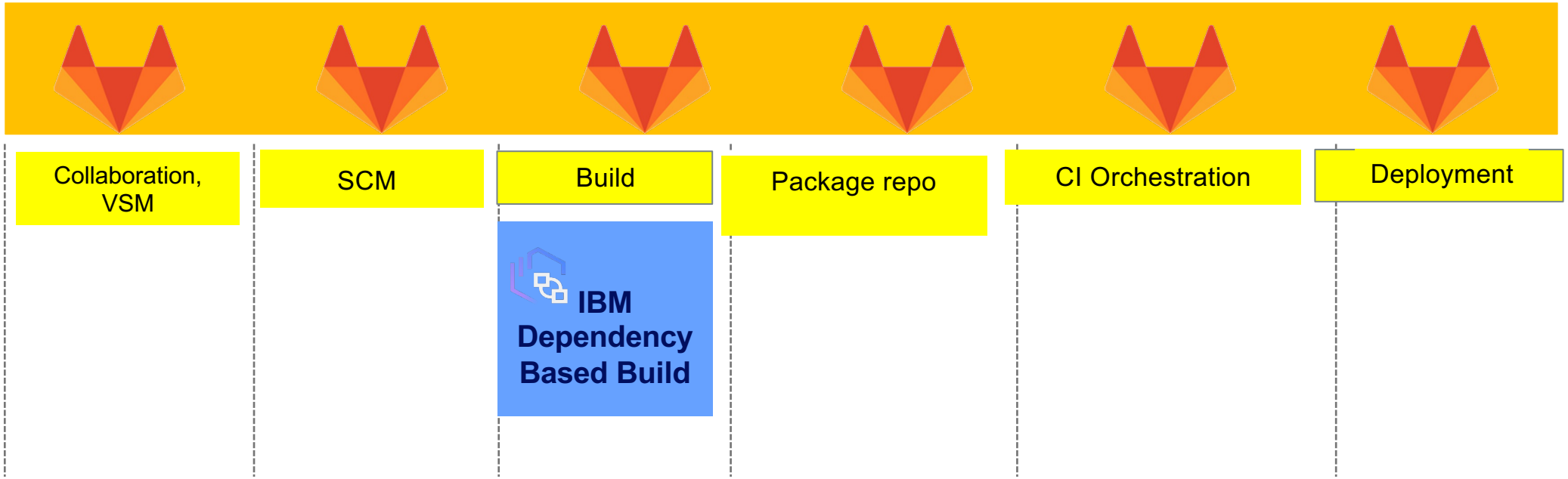
Develop and test IBM Z applications anywhere, anytime with z/OS optimized for x86 hardware

- **zD&T tools to accelerate provisioning and image management**
- Cloud friendly, software-based licensing for enterprise customers (managed service on IBM Cloud)
- Latest z/OS 2.4 software and middleware
- ***Developer autonomy just like distributed, web and mobile developers!***



DevOps mainframe pipeline with **GitLab Ultimate for z/OS**

One solution provides: Project Management, SCM, CI/CD, issue tracking, container registries, logging, dependency scanning and license management. Many IBM, 3rd party and open source tools integrate.



Extend GitLab with Specialists

Jira

GitHub

Bitbucket

Artifactory

Jenkins

urban{code} Deploy

GALAXY

IBM Rational Test

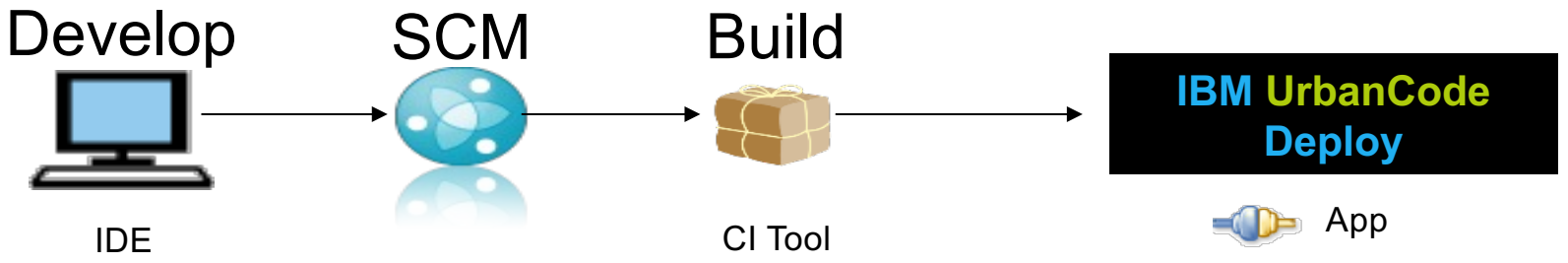
urban{code}

Velocity (VSM, data, pipeline orchestration)

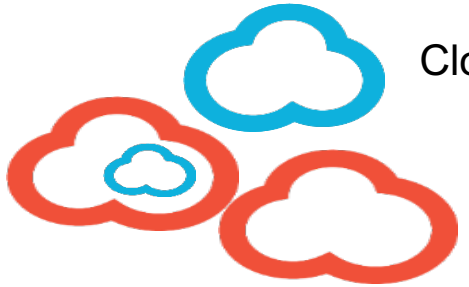
UrbanCode for Deployment automation (UCD)

urban{code}

FROM MAINFRAME TO I-SERIES TO DISTRIBUTED TO CLOUD-NATIVE



Mobile Device



Cloud: Public / Private / Hybrid

amazon web services

Microsoft Azure

Traditional

Mainframe

IBM UrbanCode Deploy automates the deployment of applications, databases and configurations into development, test and production environments, helping to drive down cost, speed time to market with reduced risk.

Deployment orchestration for Distributed and z/OS Applications – including CICS, IMS, Db2, MQ, across multiple environments

Integrates with Jenkins

Manages artifact versions via an in-built repository called codestation

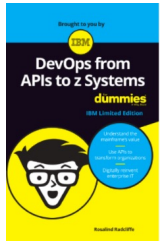
IBM

Jenkins

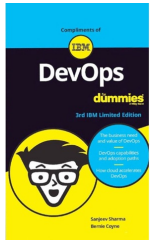
Additional Resources

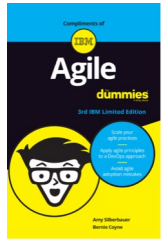
— Mainframe CI/CD with an open toolchain:
- https://www.linkedin.com/pulse/mainframe-cicd-open-toolchain-its-real-spectacular-minaz-merali/

— Mainframe Dev Center:
- Downloads - Mainframe DEV (ibm.github.io)



— IBM DevOps for Enterprise Systems:
- https://www.ibm.com/it-infrastructure/z/capabilities/enterprise-devops





— For Dummies books:
- https://www.ibm.com/ibm/devops/us/en/resources/dummiesbooks/

— IBM Z Trial:
- https://ibm.biz/z-trial


QUESTIONS?