

# IBM Z DevOps

## Git-based CI/CD pipeline demo for building Z applications

A developer's journey

---

Lauren Li

[Lauren.K.Li@ibm.com](mailto:Lauren.K.Li@ibm.com)

DevOps Acceleration Team

Visit:

[https://ibm.github.io/mainframe-downloads/DevOps\\_Acceleration\\_Program/devops-acceleration-program.html](https://ibm.github.io/mainframe-downloads/DevOps_Acceleration_Program/devops-acceleration-program.html)

# Agenda

**Introduction to IBM Z DevOps**

**End-to-end CI/CD pipeline demo with an open and modern pipeline based on IDZ/Gitlab/DBB/UCD**

**Roadmap to CI/CD Practices on the Mainframe**

**Discussion, Q&A**



# THE VISION



For agile processes and modern methodologies to be part of our Mainframe customers' Software Development Life Cycle (SDLC)

# THE MISSION



To drive DevOps transformation initiatives and adoption of CI/CD pipelines facilitating migrations from legacy library managers to Git



Selecting the  
right  
technology is  
the key to  
success

### Top Technology Investments

- Continuous integration, deployment, and delivery
- Automation and containers
- DevOps

### Top 5 Most Important Tools

- Version control system
- Text editors/IDE
- Chat/Collaboration tools
- Bug/Issue tracker
- Continuous integration and delivery

92%

agree that open source tools are critical to software innovation

Open source tool provides a cost effective, sustainable development environment.

58%

think the biggest challenges organization faces when it comes to adopting new practices or tools is replacing ingrained practices

GitLab 2018 Global Developer Report. (n.d.).  
Retrieved from  
<https://about.gitlab.com/developer-survey/2018/>



# Enterprise-wide CI/CD standardization

## Standardized CI/CD Pipeline

- Open source tools such as Git, Jenkins and Artifactory are becoming the de facto standard for powering Continuous Integration and Continuous Delivery in the enterprise.
- Reduce silos and harmonize development toolchains across the enterprise.
- Building blocks are loosely coupled and highly customizable; No vendor lock-in. IBM provides reference implementation.

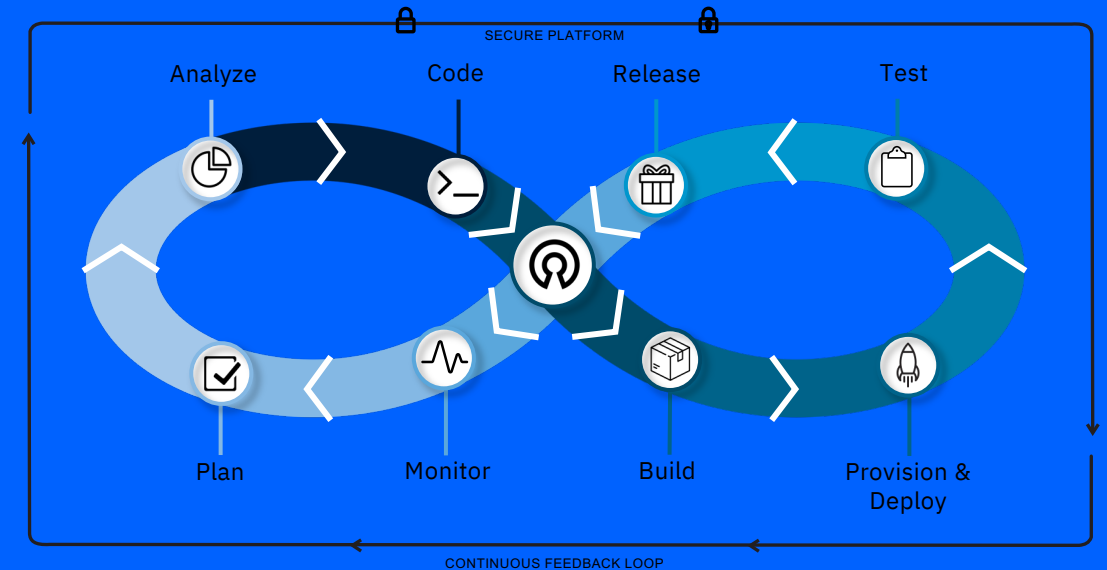
## z/OS has been transitioning to remove differences that serve no purpose

- Open Source works on and for IBM Z. Break down silos and address the skill challenges for the next developer generation, by maintaining and modernizing your existing assets.
- Applying provisioning practices like spinning up an execution environment by pushing a single button (IBM z/OS Cloud broker), requires a modern pipeline which can dynamically deploy a baseline of the application to that new environment.

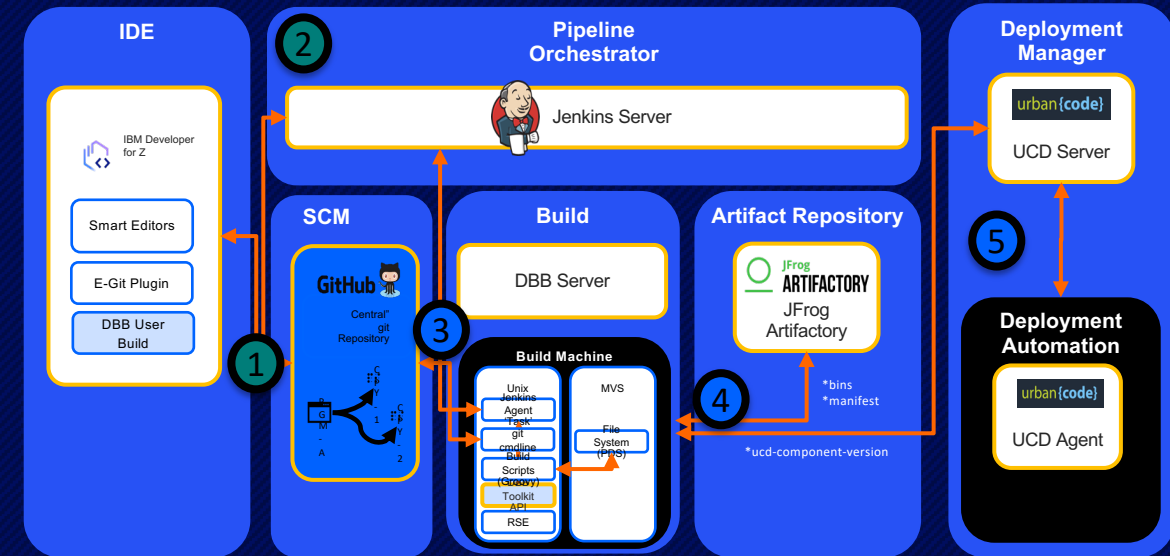
## Adoption of agile / DevOps practices

- Leveraging the same development practices across all platforms, across the entire enterprise.
- Take mainframe teams into the 21<sup>st</sup> century by adopting DevOps practices to improve their agility and customer focus while delivering smaller high quality increments faster.

## Enterprise DevSecOps Standardization of Practices



## High-Level Pipeline Architecture



\* named products for Pipeline orchestrator, Git provider, Artifact repository and Deployment manager in this picture are used as a sample. Different solutions possible.

# CI/CD Pipeline unification with IBM's Continuous Integration Solution

## Open by Design

- **IBM Developer for Z** offers an **Integrated Development Environment** that provides core capabilities needed to link z/OS development seamlessly with an established, open and modern DevOps toolchain.
- **IBM Wazi for RedHat CodeReady Workspaces** delivers the zero-install cloud native development experience
- **IBM Dependency Based Build** enables to implement **mainframe build automation** with Git and your preferred CI orchestrator for traditional mainframe artifacts
- **Ansible for IBM z/OS integrations and IBM Urbancode** Deploy and Velocity enable a common approach for hybrid infrastructure management and application delivery.

## Benefits

- All solutions integrate with your GIT provider to easily **identify and make changes** quickly and efficiently; leverage **real parallel development capabilities** which are necessary for DevOps and CI/CD
- **Build and debug** application changes directly from the development environment
- Pipeline unification with IBM's solution supports your **mainframe modernizing projects breaks down silos**

## References being raised

*By implementing DBB and GIT in ZD&T environment, IBM helps a large banking group to implement technical and organizational means to increase the autonomy of its AMS centers and improve the quality of the overall process.*

*This client shared their story at THINK 2019*



# A typical User Story

## Deb, new z/OS developer

- At school I learned to develop Java with Eclipse, but I could also use my **favorite editor** I downloaded from the web.
- I am used to writing a **unit test** for all my programs - that makes it easier and safer to modify them in the future.
- **Test coverage** for Java is great. It tells me if my tests also hit the changed code. I wish I could use that in COBOL.

## Pain points with current tooling

- Slow
- Antiquated tools
- Many tools
- Program understanding
- Deployment steps / complexity



Let me explain to you how I would like to work :

I received a task to fix a defect, so I start by **cloning** the repository of my application into my work environment.

I create a **new branch** for this task, so I can work in isolation, and not worry about being disturbed by the development activities of my team.

In my IDE I use IBM **DBB user build** to compile / link and unit test the program.

Then I **commit and push** my branch to the central Git repository along with my commit.

Let's have a teammate review my change. I create a **pull request** to merge my changes to the common branch of our team.

- The pull request needs to be approved.
- It triggers a build which runs a set of tests and code scans .

After the pull request is approved and merged in, I can delete my personal branch. On the integration branch, a **pipeline build** is performed, so I can move the changes forward or associate them with a release.

# The big picture

- A generic CI/CD pipeline setup includes several building blocks each serving a dedicated purpose

Graphical  
User  
Interface

**IDE**

Provides check-out and check-in capabilities to the version control system.

Text User  
Interface

## Pipeline Orchestration

Also known as the CI Orchestrator; providing connectors to version control, build environment, packaging and deployment. This is the place where all the automation happens.

**SCM**

Storing different versions of your application configuration like **source code**, application specific configuration data, test cases etc.

**Build**

Platform and language agnostic build environment. In mainframe environments understanding dependencies, compile, link, unit test.

**Artifact  
Repository**

Storage for build outputs – an application package containing load modules, DBRMs, DDL, configuration files of subsystems, WAR, EAR files. It decouples scm layouts and runtime environments.

**Deployment  
Manager**

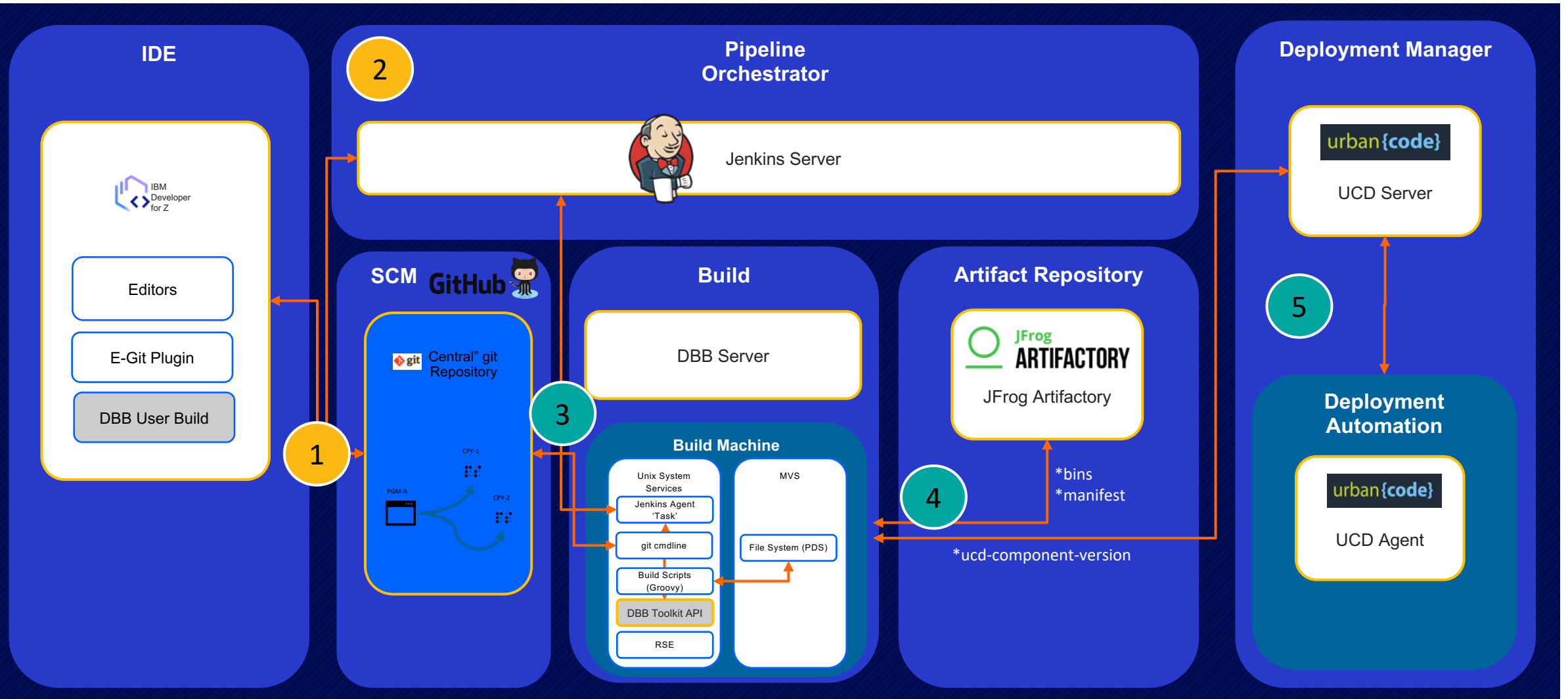
Knows about the execution environments and their inventory.

Used to rollout application packages including installation steps like newcopy and bind for traditional mainframe apps.

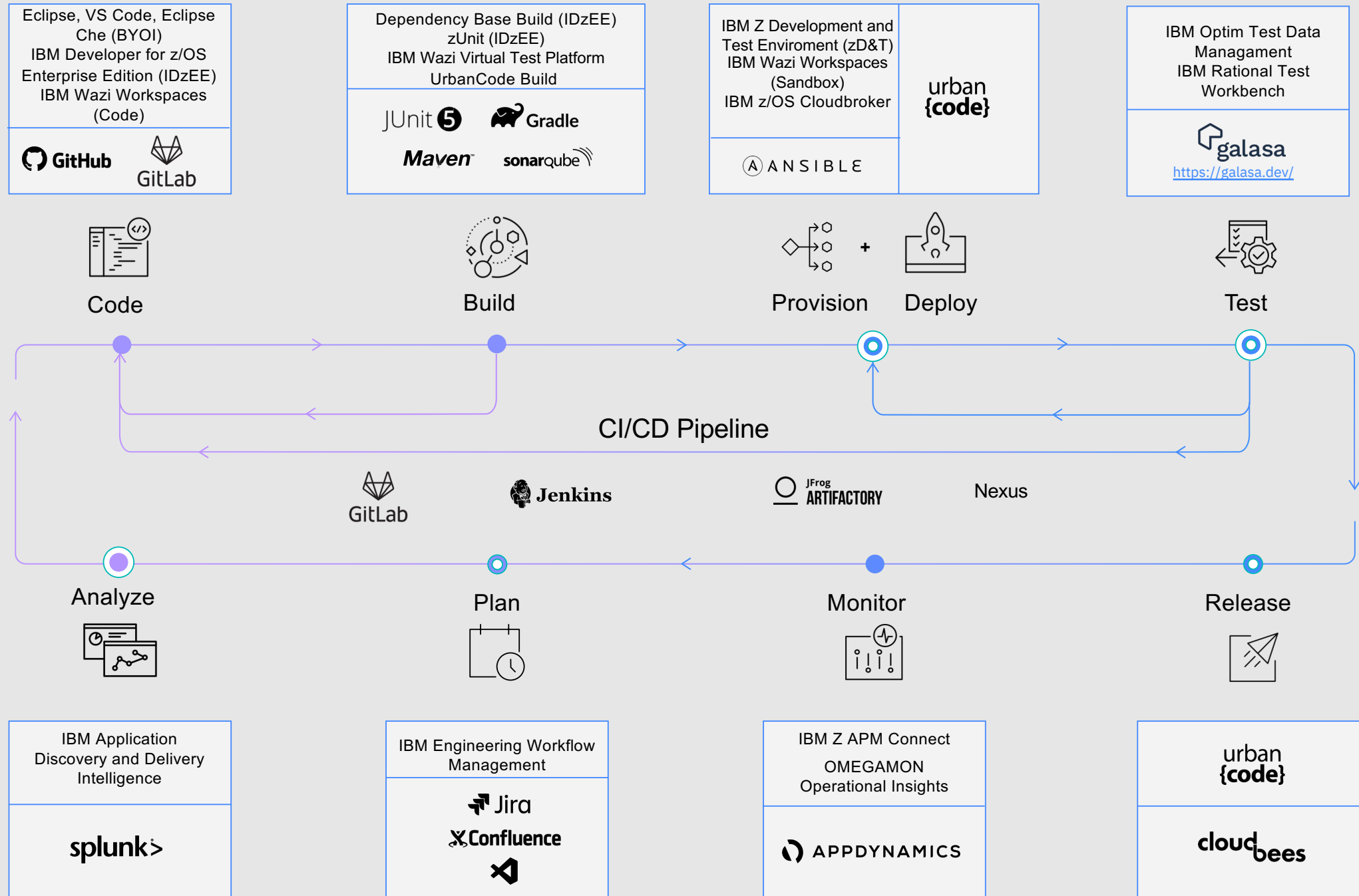


# Sample Pipeline Overview

1 Edit & check-in → 2 Pipeline → 3 Build → 4 Packaging → 5 Deploy & Test



# End to end DevOps The IBM way Open and Flexible



## LEGEND

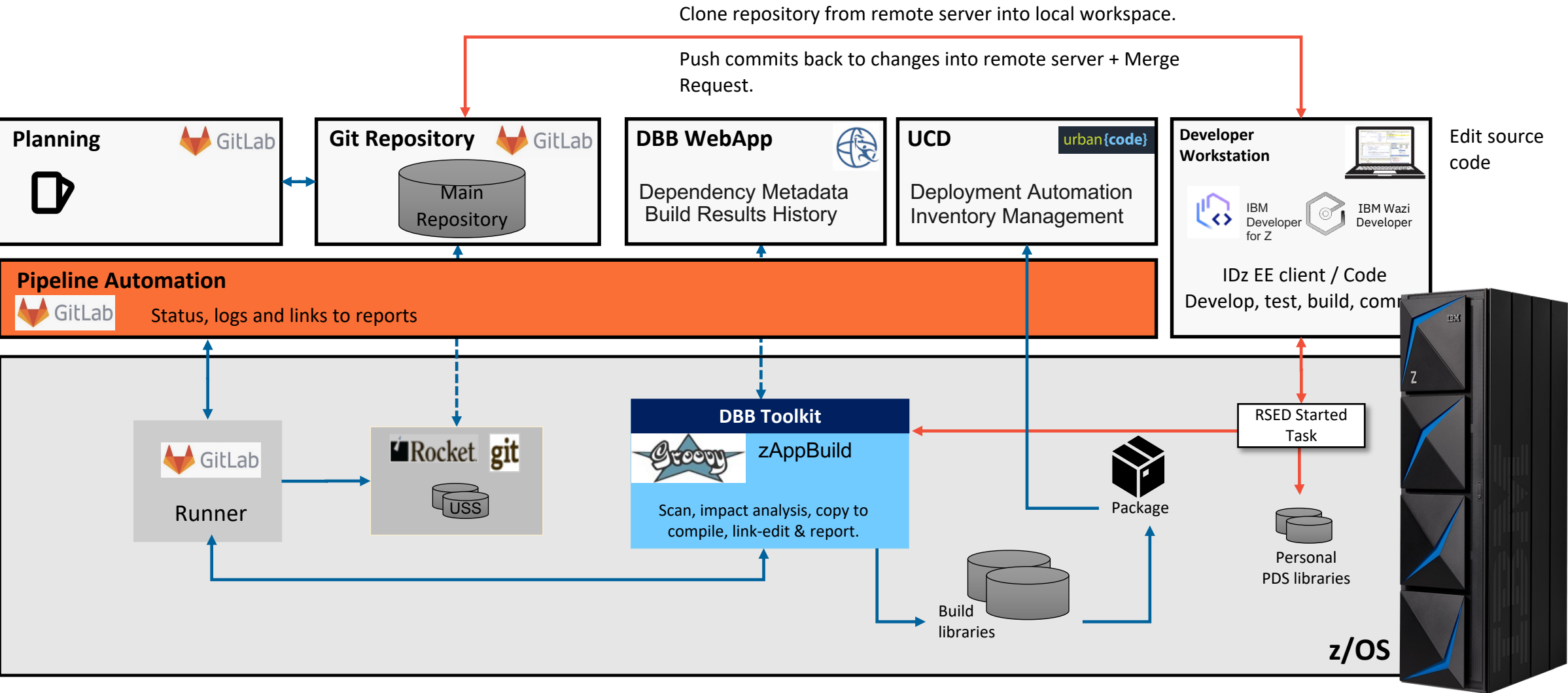
Analyze

Feedback, approvals

**Note:** Starting from the upper left, products and significant capabilities appear once, the first time they are used in the pipeline. Products and capabilities are used at multiple points.

**DEMO TIME!**

# Pipeline Architecture





# **Roadmap to CI/CD Practices on the Mainframe**





# The journey to a CI/CD pipeline

## Design and Validate the high-level future design

- Develop your high-level to-be future state
- Understand the technical composition of the pipeline
- Build a pilot to prove feasibility
- Establish the Champions to drive the project
- Achieve Senior Management Support

## Deep-dive analysis to build business case and migration strategy

- Analysis of the existing build management system
- Analysis of the existing repository layouts
- Develop a migration strategy which fits the customers' needs

## Architect and implement the future state and new way of working

- Refinement of future state – define workflows and integrations
- Low-level design
- Architect and implement the e2e future state
- Communication of the journey - sandbox system

## Rollout the new way of working

- Onboard and train the development and operation teams
- Execute the migration plan and migrate applications to the new pipeline

## Realize the benefits

- Work in a standardized pipeline
- Return any ISV license(s)
- Enable new capabilities in the pipeline

# Learn more

Our remote instructor-led and self-paced training provides you with the perfect start to begin your DevOps journey



## **IDz instructor led virtual remote training**

<https://ibm.github.io/mainframe-downloads/Training/idzrdz-remote-training.html>

## **CI/CD Learning Collection**

<https://www.ibm.com/training/collection/zdevopstransformationcicdpipeline/swithdbbgit>

## **IDz Basics - self paced**

[https://ibm.github.io/mainframe-downloads/DevOps\\_Acceleration\\_Program/idz-self-paced-learning.html](https://ibm.github.io/mainframe-downloads/DevOps_Acceleration_Program/idz-self-paced-learning.html)

## **DBB Fundamentals - self paced**

<https://ibm.github.io/mainframe-downloads/Training/dbb-self-paced-learning.html>

Visit our website for more content:

<https://ibm.github.io/mainframe-downloads/Training/Training.html>

**QUESTIONS?**