Enabling DevOps on Premise or Cloud with Jenkins

Sam Rostam

srostam@shaw.ca

Cloud & Enterprise Integration Consultant/Trainer

Certified SOA & Cloud Architect

Certified Big Data Professional

MSc @SFU & PhD Studies – Partial @UBC

Topics

The Context - Digital Transformation

An Agile IT Framework

What DevOps bring to Teams?

- Disrupting Software Development
- Improved Quality, shorten cycles
- highly responsive for the business needs

What is CI /CD ?

Simple Scenario with Jenkins

Advanced Jenkins : Plug-ins , APIs & Pipelines

Toolchain concept
Q/A

Digital Transformation – Modernization

As stated by a As established enterprises in all industries begin to evolve themselves into the successful *Digital Organizations* of the future they need to begin with the realization that the road to becoming a *Digital Business* goes through their IT functions. However, many of these incumbents are saddled with IT that has organizational structures, management models, operational processes, workforces and systems that were built to solve "turn of the century" problems of the past.

Many analysts and industry experts have recognized the need for a new model to manage IT in their Businesses and have proposed approaches to understand and manage a hybrid IT environment that includes slower legacy applications and infrastructure in combination with today's rapidly evolving Digital-first, mobile-first and analytics-enabled applications.

http://www.ntti3.com/wp-content/uploads/Agile-IT-v1.3.pdf

Digital Transformation requires building an ecosystem

- Digital transformation is a *strategic approach* to IT that treats IT infrastructure and data as a potential product for customers.
- Digital transformation requires shifting perspectives and by looking at new ways to use data and data sources and looking at new ways to engage with customers.
- At the core it will include ..

  Virtualization, public clouds, containers and orchestration, Data virtualization, memory caches and storage, multiple messaging protocols, various data formats from different data sources, management and deployment tools, test automation plus of course various applications & business process automations.

RedHat – A Platform for Modernizing Java Applications – Whitepaper Oct 2016

DevOps

- DevOps is essentially a philosophy, a perspective on how to address the problems faced by development and operations teams.
- DevOps began in a software development environment and as such many of its methodologies and tools focus on improving software deployment.
- Continuous Integration (CI) and Continuous Delivery (CD) transforms the deployment process and its scope is beyond Agile software development.
- Part of the reason DevOps is commonly described as a philosophy is that it requires transforming the organization with its ideas.
- This requires taking a step back and reassessing some of the basic assumptions of how the IT works in that organization.
- Within the context of Agility you need to always look for continuous improvement by using automation & LeanIT principles.

CI-CD-CD

- **Continuous Integration** basically just means that the developer's working copies are synchronized with a shared mainline several times a day.
- **Continuous Deployment** is described as the logical next step after continuous delivery: Automatically deploy the product into production whenever it passes QA!
- **Continuous Delivery** is described as the logical evolution of continuous integration: Always be able to put a product into production, or alike.

Continuous deployment is simply the discipline of continuously being able to move the result of a development process to a production-like environment where functional testing can be executed in full scale.

Continuous Integration (CI) in a nutshell

- Originated in Extreme Programming , used by many Agile shops
- Deploying continuously is essential to streamlining the feedback loop, a core Agile tenant
- CI usually means an entire project is rebuilt upon change to code base
- Laborious integration efforts to be avoided with short automated ones
- The process of CI is about building software components often, which, in many instances, means any time code within a repository changes.
- the term *Continuous Integration* often is associated with one or more particular tools.

http://agilemanifesto.org/principles.html

CI Benefits

Bugs more obvious –

Catch:: if system tests tightly coupled with CI implementation

Reduces Risk -

- CI makes identifying problems easier to predict & more obvious
- The earlier, the cheaper to resolve

Catch:: The above assumes frequent code integration by developers

Catch:: At least update & recompile before committing new code!

https://martinfowler.com/articles/continuousIntegration.html

Getting Started with CI

- An automated build process with a platform like Ant or Maven
- A code repository like CVS or Subversion
- A CI server such as Hudson, although a cron job could suffice

- In the Java world, Ant stands as the ubiquitous build platform.
- With Ant, you can perform automated builds or even manual tasks like compilation, testing, and even more interesting things like software inspection and deployment.

https://martinfowler.com/articles/continuousIntegration.html

A Basic Build Process

- Compiling source code, including tests
- Executing tests, such as those written with JUnit or TestNG
- Running code inspections, such as PMD
- Archiving the final product into a JAR, WAR, or series of files
- Deploying the final assets (assuming the final product warrants it)

For CI to add value to a software development process, the build itself must do more than just compile code. Because a build is run every time code changes, it presents an excellent opportunity to run tests and code inspections.

IBM developer work portal

DEMO 1

Continuous Deployment (CD)

Automate deployment – one of the tenants of Agility

- Continuous Integration is not the same thing as Continuous Deployment but it makes sense to consider the two concurrently
- Other names include “one click” deployment
- CI systems support custom scripting to integrate this into automated workflows
- Many CI Systems also provide APIs to support CD

https://martinfowler.com/articles/continuousIntegration.html

Continuous Delivery

- described as the logical evolution of *continuous integration*:
- The goal is to be able to put a product into production (or production like environment.
- CI systems support custom scripting to integrate this into automated workflows
- Many CI Systems also provide APIs to support CD

Continuous integration is focused on automatically building and testing code, as compared to continuous delivery, which automates the entire software release process up to production potentially"

CI Platforms

- Jenkins
  - Open source (MIT License)
  - Forked from Hudson following dispute with Oracle
- AnthillPro
  - Been around for a long time, very mature
  - Full enterprise solution
- Bamboo – Atlassian (i.e. Jira & Confluence dev tools)
  - Support for Java, .NET, PHP, Java Script
  - Continuous deployment made easy
  - Bamboo onDemand – in AWS EC2
- TeamCity – from JetBrains
  - runs in a Java environment, usually on an Apache Tomcat server
  - TeamCity offers equal support for .NET and open stack projects, integrating into both Visual Studio and IDEs like Eclipse.

Jenkins

- Jenkins, formerly Hudson, is an open source Continuous Integration and Continuous Delivery tool used widely and considered by many to be the de facto standard for CI.
- Jenkins is an open-source automation server that lets you flexibly orchestrate your build, test, and deployment pipelines.
- Jenkins also runs in Java, so you can install it on Windows or Linux.
- Highly extensible and with a gallery of plug-ins readily at hand, Jenkins offers granular customization for any size CI operation.
- Jenkins offer a RESTful API and comes in three flavors: XML, JSON and Python.

https://jenkins-ci.org

DEMO 2

CI Server – Hudson / Jenkins

- The CI server's main role in the entire developmental dance is that of the director:
- When the server detects a change within a code repository, it merely defers the responsibility of running a build to the build process itself.
- If the build indicates it failed, then the CI server will notify interested parties and go back to monitoring a repository for changes.
- Its role may seem passive in nature; however, it is the vehicle by which rapid feedback of issues is made possible..

https://github.com/jenkinsci

Installing Jenkins

- One of the primary benefits of using Hudson is its simple set up. At a minimum, Jenkins requires two steps:

  1- Download the latest version (which is bundled as a WAR file).

  2- Run java -jar jenkins.war.

- Because the download is a WAR file, you are free to deploy it to a container such as Tomcat or JBoss, if you want.
- fire up a browser and go to the default location of your installation.

  → http://localhost:8080/.

https://

DEMO 3

Next Generation IT – Key Considerations

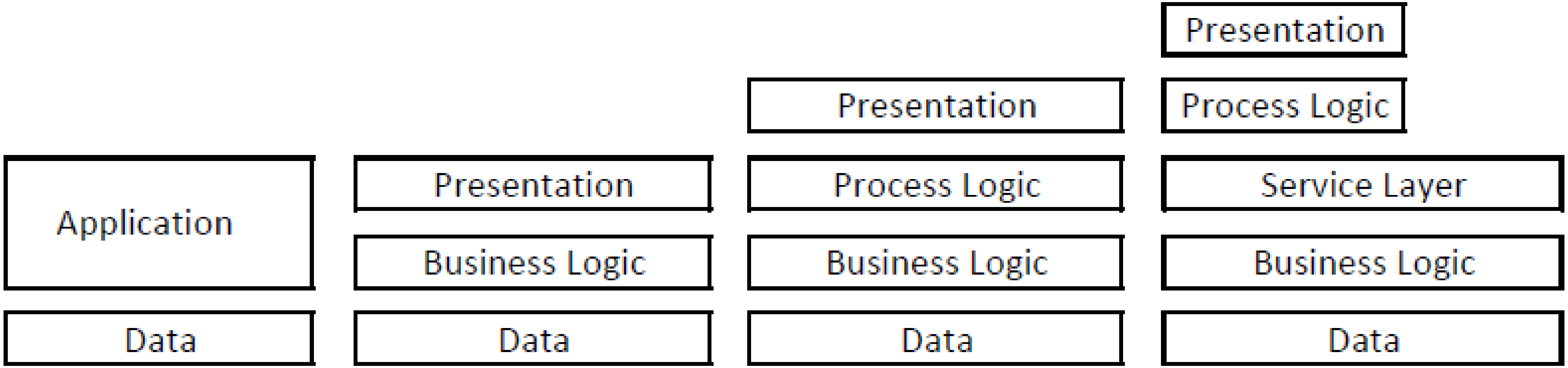
The current model for IT, which is built around delivering IT *Faster*, *Better* and *Cheaper*, is not adequate to deal with the increased volume, velocity and variety of change driven by the rapidly evolving digital landscape.

NTT Innovation Institute

Speed of operation and ‘fluidity’ are important prerequisites of next generation IT

Gartner

Architectural Layering



Application

Data

Presentation

Business Logic

Data

Presentation

Process Logic

Business Logic

Data

Presentation

Process Logic

Service Layer

Business Logic

Data

Changing Systems Assumptions

Traditional →

- Homogeneous hardware/software
- Stable, centrally managed configuration
- Synchronous & Local: processing, state, time, and communication
- Infrequent, monolithic failures

Transaction
Design
Creational

Distributed

- Homogenous hardware/software
- Evolving configurations
- Remote, autonomous processing
- Distributed, replicated, non-uniform: state & time
- Asynchronous, insecure & variable: communications
- Frequent partial system failures

Compensation
Architecture
Compositional

Software Architect Bootcamp, ISBN: 0-13-027407-0

Jenkins Scenarios

- Software Development – or SDLC
- Application Build – CI
- Application Deployment – CD
- Application QA
- Platform set up
- Continuous Delivery

Pipelines in Jenkins

- Pipeline is a suite of plugins which supports implementing and integrating continuous delivery pipelines into Jenkins.
- Pipeline provides an extensible set of tools for modeling simple-to-complex delivery pipelines "as code" via the Pipeline DSL.
- Jenkins fundamentally is an automation engine which supports a number of automation patterns.
- Pipeline adds a powerful set of automation tools onto Jenkins, supporting use cases that span from simple continuous integration to comprehensive continuous delivery pipelines.

https://jenkins.io/doc/book/pipeline/

DEMO 4

ToolChains concepts

- A **toolchain** is a set of distinct software development tools that are linked (or chained) together by specific stages. Optionally, a toolchain may contain other tools such as a Debugger or a Compiler for a specific programming language.
- The DevOps toolchain is composed of the tools needed to support a DevOps continuous integration, continuous deployment, and continuous release and operations initiative.
- A toolchain strategy allows IT organizations to implement tools aligned to the activities required to deliver a DevOps practice.
- For DevOps, many of the toolchain activities are combined and delivered in parallel. It is important to recognize that activities are not autonomous and that the pipeline is not a set of linear silos.
- One technique that works well is to map tools to activity within an organization.

Source: http://elinux.org/Toolchains https://www.xmatters.com

Q/A & Discussions

Additional Resources - Questions

Very thorough Bamboo documentation
http://www.atlassian.com/software/bamboo/overview

Additional information about CI
http://www.ibm.com/developerworks/java/tutorials/j-cq11207/

Gartner: Enabling DevOps with ToolChains
https://www.gartner.com/doc