CONTINO

Geekfest: 3 Musketeers 101

Consultant Profile

Gregory Patmore

Technical Principal Consultant

For more than 15 years, Greg has been leading teams in a number of industries including Financial Services, Ad Technology, Real Money Gaming, and Streaming Content Providers.

Over the last 5 years, Greg has been leading teams for Contino, and now Cognizant, on a variety of Data, Cloud, and Devops projects that are truly transforming the organizations he works with. His projects are centered around rapid assessment, cloud transformation, workload and technology optimization, security and compliance integration, and process engineering.

Expertise

- Rapid assessments of complex organizations
- Governance, Security & Compliance Integration and Management
- Cloud Migrations and Workload Optimization
- Process Engineering

Qualifications & Certifications

- Rapid Assessments
- Critical Systems Analysis and Migration
- SDLC Process and Workload Optimization
- Security and Compliance integration
- Data Management and Optimization.

CONTINO | Cognizant

Agenda

01 | Unified Application Pipeline and Verification Engine Overview

02 | 3 Musketeers Overview

03 | Hands on Exercises

04 | Q&A/Discussion

Unified Application Pipeline & Verification Engine

The Unified DevOps Framework

This framework defines the DevOps best practices for the organization. It aims to deliver faster and more reliable releases by automating the verification of security, compliance, and quality standards for code before it goes to production. The framework covers the following areas:

Standard CI/CD Pipeline

- Is a starter GitLab pipeline following DevOps best practices
- Use the 3 Musketeers pattern, a method of software development that makes the work easier by using standard tools and processes
- Use Artifactory to track scan and compliance test results, and stores them with the application artifacts as metadata
- Invokes the Verification Pipeline

Verification Pipeline

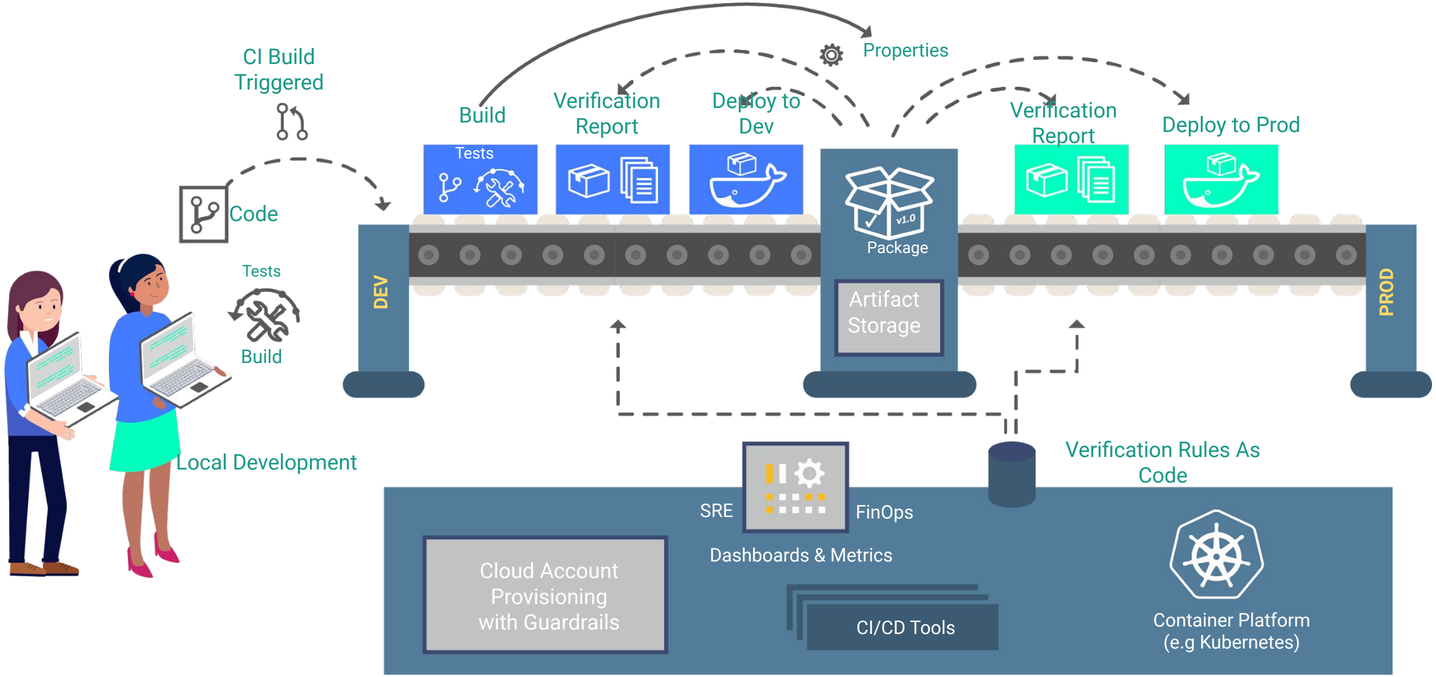
- Retrieve software metadata from Artifactory and find all the reports that measure the quality and security of the software
- Run in a centralized repository that is updated automatically for all application products to leverage
- Check each report against a list of rules to verify if the software meets standards
- Create a summary report and stores it as a file for proof and auditing compliance

Improved path to production

The framework applies the trust and verify approach to achieve faster and more reliable releases that meet the quality, security, and compliance standards for the code and the system.

Trust development teams to manage their own pipelines.
Continuous integration & delivery (deploy to dev)

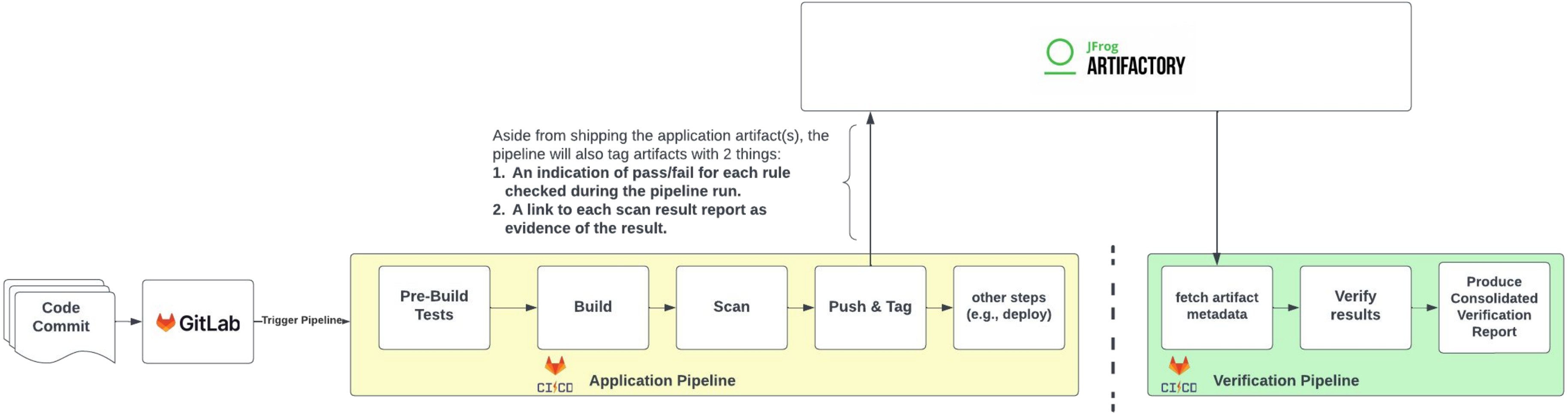
Verify all software before it's deployed to prod.
Continuous deployment (deploy to prod)



cognizant

DevOps Unified Framework – Pipelines Working Together

The Application and the Verification Pipeline enables teams to quickly start new projects or customize the verification process for any existing environment.



Aside from shipping the application artifact(s), the pipeline will also tag artifacts with 2 things:

1. **An indication of pass/fail for each rule checked during the pipeline run.**
2. **A link to each scan result report as evidence of the result.**

(verification pipeline can be triggered at the end of the app pipeline or separately)

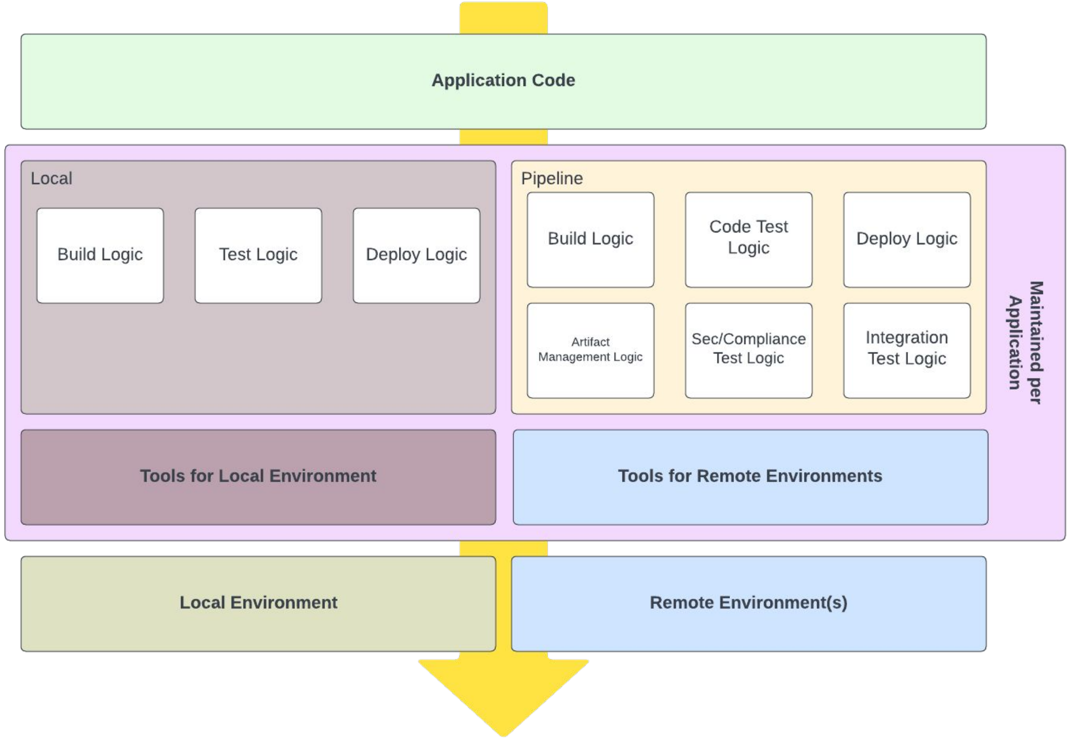
Use Cases for the Unified DevOps Framework

- ✓ Create a pipeline for a **new project** that follows the best practices and standards for DevOps.
- ✓ Refactor an **existing project** to align with the DevOps principles and improve its quality and performance.
- ✓ Have a **simple and consistent way** to install and configure the **tools and dependencies** for the project.
- ✓ Avoid **security** and **compliance** issues late in the process that can delay or prevent the release of the software.
- ✓ Collect and document **evidence** for **audit requirements**.
- ✓ Have a **reliable** and **reproducible** environment for building and testing the software across different stages of the pipeline.
- ✓ Ensure that all the developers on the project use the same **tools** and **language versions** to avoid **compatibility** issues and bugs.

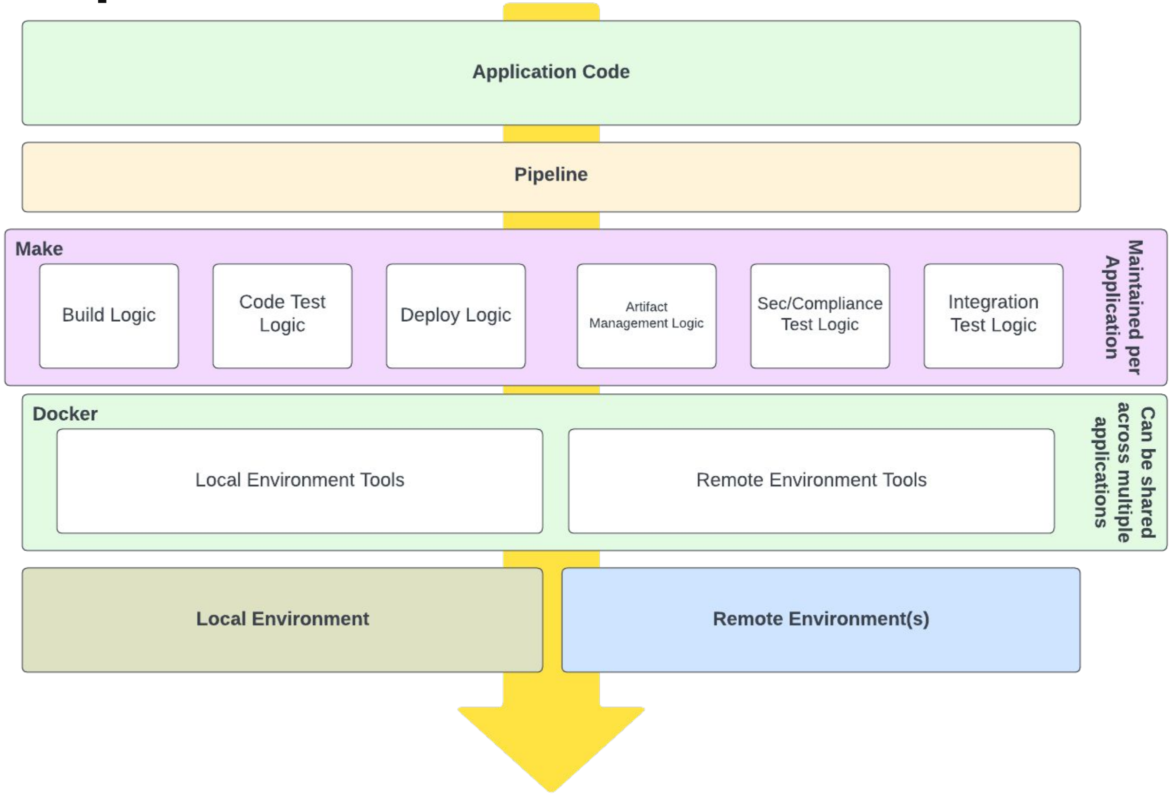
Benefits

Speed and Quality	Accelerate onboarding of new projects through a streamlined framework Shift left and reduce bugs and defects through a comprehensive verification Maintain software security and compliance with automated verifications
Flexibility	Adapt to different development scenarios and technologies based on needs Leverage powerful and common tooling widely available Minimal setup required to immediately implement
Enhanced Visibility	Automated reporting and dashboarding for improved insights. Security and Compliance teams can make informed decisions based on accurate data.
Consistency	A systematic approach to testing and validation of software Teams can leverage centralized improvements to verification tools Standard framework increases collaboration between development teams

Current State



Unified Pipeline

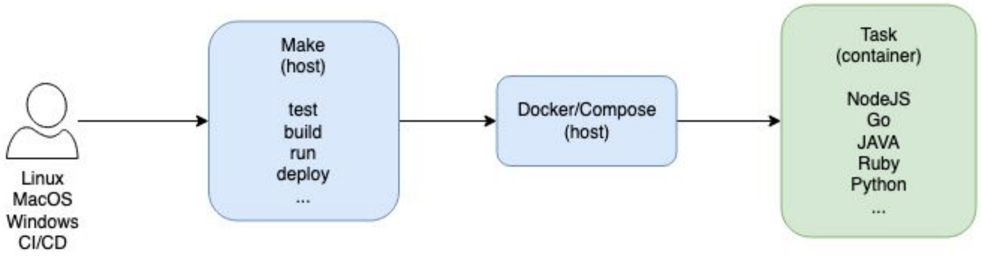


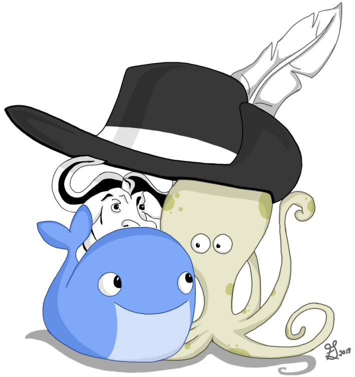
3 Musketeers

3 Musketeers - What?

3musketeers.io

The 3 Musketeers is a pattern for developing software in a repeatable and consistent manner. It leverages Make as an orchestration tool to test, build, run, and deploy applications using Docker and Docker Compose. The Make and Docker/Compose commands for each application are maintained as part of the application’s source code and are invoked in the same way whether run locally or on a CI/CD server.





3 Musketeers Who

Who

Docker

- Builds, tests, and deployments can happen independent of the host system.
- Jenkins VS local execution.

Compose

- Orchestrating multiple docker containers.
- Simplifies makefile docker commands.

Make

- Multi-platform build automation tool.
- Uses Makefiles and associated targets.

3 Musketeers Why

Why

Consistency

Run the same commands no matter where you are.

- Linux
- CI/CD tools

Control

Take control of languages, versions, and tools you need, and version source control your pipelines with your preferred SCM.

Confidence

Test your code and pipelines locally before your CI/CD tool runs it. Feel confident that if it works locally, it will work in your CI/CD server.

Make

Make

- Make
  - A powerful cross platform build automation tool.
  - Make uses ‘Makefiles’ and associated targets to run build steps.
  - Originally created Stuart Feldman at Bell Labs in 1976!
- Common make targets
  - `Make Clean:`
  - `Make Build:`
  - `Make Run:`
  - `Make Help:`

Docker

Docker

What is docker and how are we using it?

- A standard unit in which an application resides. Containerization uses the host operating systems Kernel while implementing its own file system.
    - Verification images
        - Images used in the Hello-World sample app pipeline
        - Verify-sonarqube
    - Utility scanner images
        - Utility-scanner-sonarqube
- Docker containers are NOT the same as VMs, but they can be described this way for new learners to understand the concept.

Docker

- Single-use
    - Each docker image has a specific use. Verifier or utility.

- Additional Highlights
    - Code packaged with dependencies so applications can run across environments
    - Dockerfiles -> Docker images.
    - Docker images -> Containers at runtime.
    - Standalone environment.

Docker Compose

Docker Compose

- Compose
  - A tool for defining and running multi-container Docker applications.
  - Creates a network of running/stopped containers.
  - Can help with cleaning up after yourself.
- Utilise compose to help clean up cluttered makefiles.
  - When docker run commands become too long an complex, consider implementing compose.
  - Makefile without compose.

    -
      ```
      docker run -v ${WORKSPACE}/app -it ${IMAGE_NAME} bash
      ```
  - Makefile with compose.

    -
      ```
      docker-compose run local
      ```

Let’s try it!

https://github.com/gregp-cognizant/cncb-3musketeers

Q & A

Links

- Demo Code - https://github.com/gpatmore/cncb-3musketeers
- 3M Site - https://3musketeers.io
- Make Docs - https://www.gnu.org/software/make/manual/make.html
- Docker Docs - https://docs.docker.com
- Docker-Compose Docs - https://docs.docker.com/compose/

Contino Unified Devops Framework Repositories

- https://gitlab.int.bell.ca/contino/devsecops-framework

CONTINO

Thank You

London	New York	Melbourne	Sydney	Atlanta
london@contino.io	newyork@contino.io	melbourne@contino.io	sydney@contino.io	atlanta@contino.io

www contino.io

continohq

in contino