

Tekton

IBM's open-source framework for creating CI/CD systems

James Gosnell & Suyash Amatya

Our Goals:

- Users need a way to prevent the failure of unrelated Tasks from affecting the execution of a set of related Tasks that should execute to completion once the first one has started.
- Pipelines (a data structure that holds tasks) should be able to store other pipelines in itself and should consecutively run through Pipeline by Pipeline.

What we've done:

- Most of the work we have completed relate to the long process of setting up a development environment
- We created a repository that clarifies steps in the process where we got stuck such as:
 - The basic architecture of Tekton
 - Setting up docker desktop to work with Tekton
 - Troubleshooting kubernetes clusters
 - how to push to a cluster and verify that it was successful
 - What to do after a successful installation to get to grips with building and running tasks

Use of each tool in the pipelines:

Go (Golang):

- Developers write code in Go to create their applications.
- Go code can be compiled into executable files and containerized for easy deployment.

YAML Files:

- CI/CD pipelines, tasks, and resources are defined in YAML files, making it easier to manage and version control these components.
- Kubernetes, Tekton, and many other CI/CD tools use YAML files for configuration and orchestration.

Kubernetes:

- Kubernetes is often the deployment target for containerized applications.
- CI/CD pipelines can interact with Kubernetes to deploy and manage applications.
- YAML files are used to define Kubernetes resources, such as pods, services, and deployments.

KO:

- Developers can use KO to streamline the containerization and deployment of Go applications.
- KO simplifies the process of creating container images for Kubernetes.
- CI/CD pipelines can integrate KO to automate the build and deployment of Go applications to Kubernetes clusters.

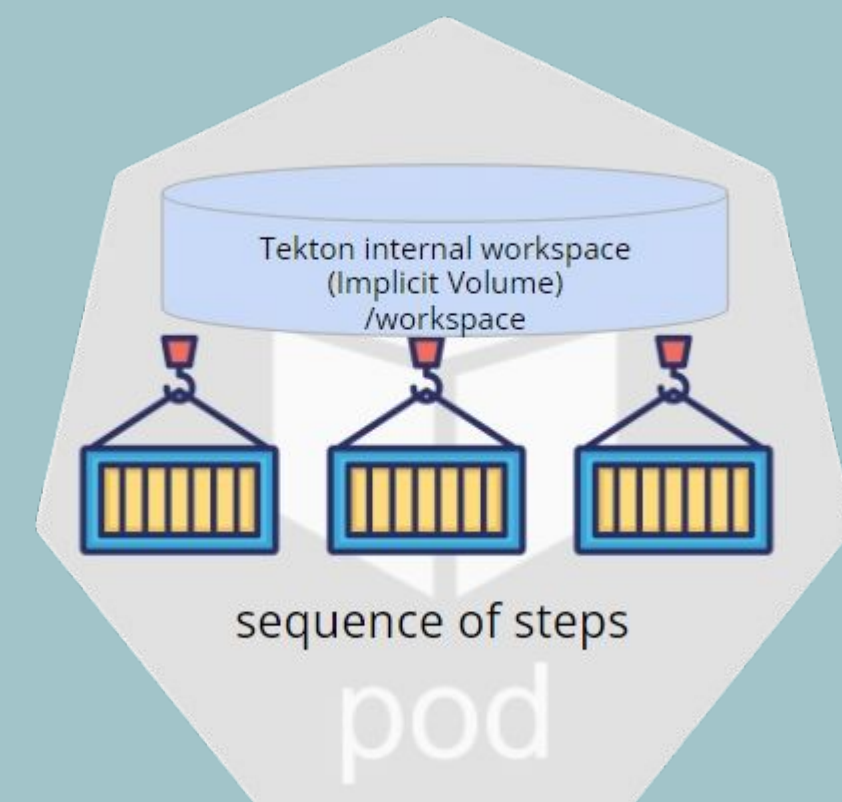
Docker:

- Developers use Docker to containerize their applications, including Go applications.
- Docker images are created from applications, and these images can be deployed to Kubernetes clusters.
- CI/CD pipelines can include steps for building, testing, and pushing Docker images to container registries. In summary, Go (Golang) is the language used to develop applications, YAML files define configurations, Kubernetes manages containerized applications, KO streamlines containerization for Go apps, and Docker helps create and manage container images. These tools and technologies work together in the CI/CD process to automate the building, testing, and deployment of software for Tekton.

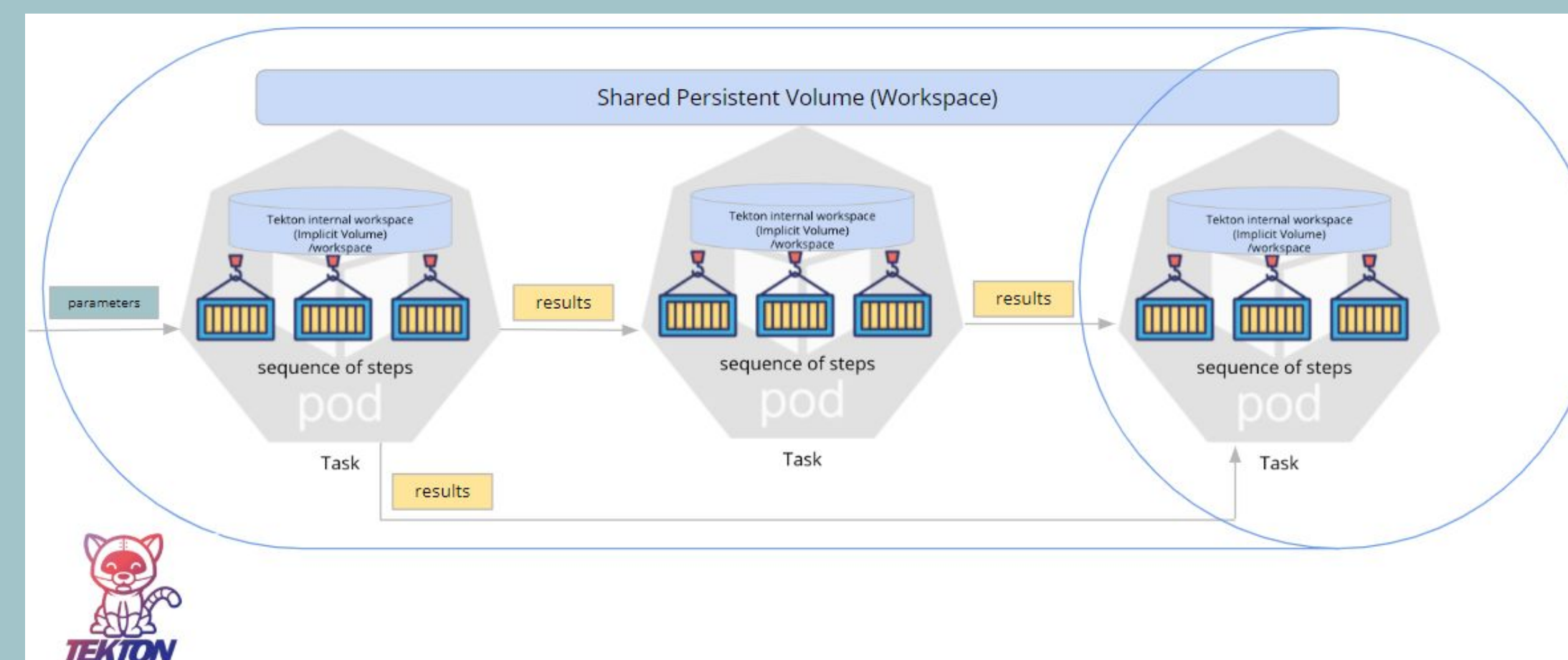
Basic Architecture

CI/CD, short for Continuous Integration and Continuous Delivery, is a streamlined process that automates the building, testing, and release of software. In essence, it's a powerful automation tool that simplifies the journey from code development to its deployment.

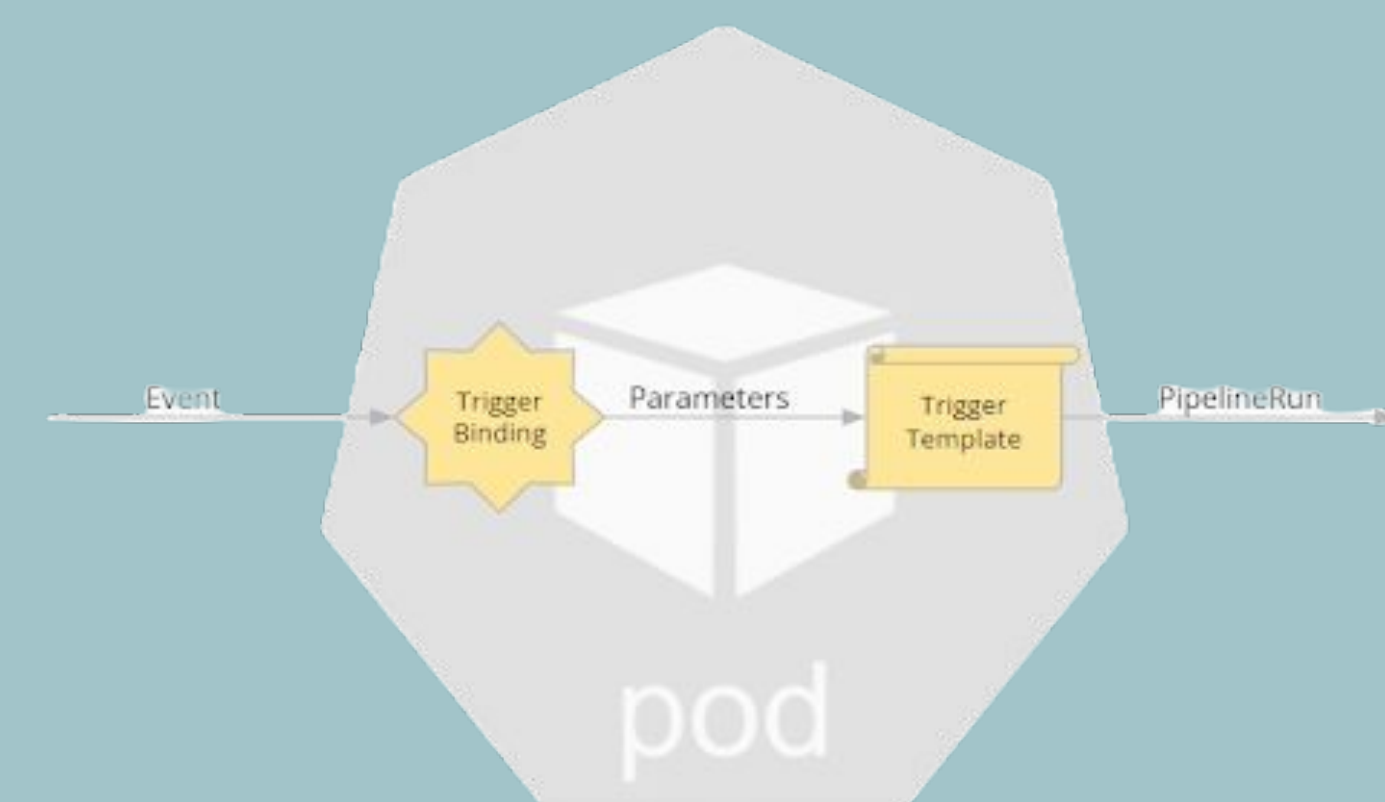
Steps: In Tekton, steps (a node) are like individual actions, similar to steps in a recipe. Each step is a discrete operation, such as compiling code, running tests, or packaging software. Steps are designed to be simple and focused, making it easier to manage and troubleshoot specific tasks within a larger process



Tasks: Tasks in Tekton are groups of related steps. Think of them as subtasks within a project. A task bundles together a series of steps that need to be performed together to achieve a specific goal. For example, a "Build" task might include steps for compiling code and generating executable files, making it easier to reuse and share common procedures across different pipelines.



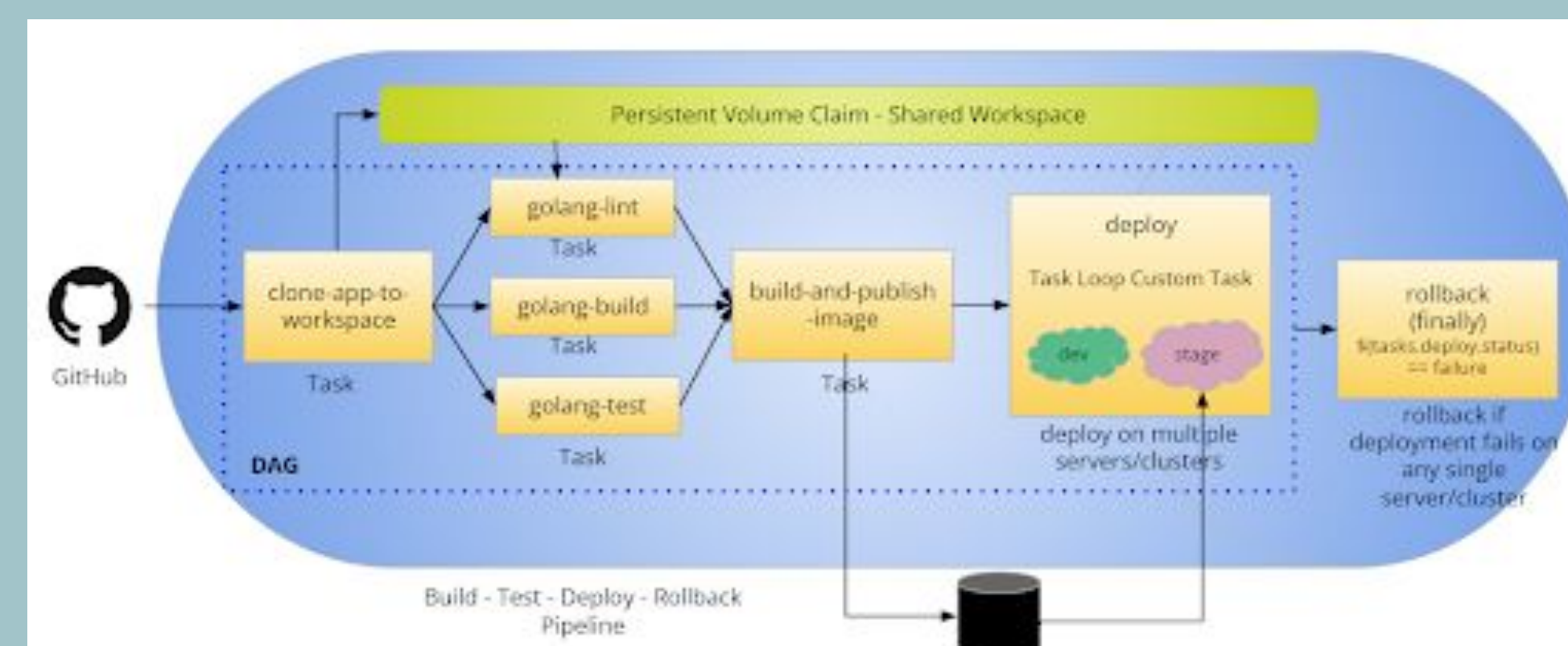
Pipelines: Pipelines provide the overarching structure for orchestrating tasks in Tekton. You can consider pipelines as the complete recipe for your software development or delivery process. They define the order in which tasks should be executed, creating a clear roadmap for your entire project. A pipeline can include multiple tasks, ensuring that each step is performed in a logical sequence.



Triggers: Triggers in Tekton act as the initiation mechanism for pipelines. They are like the "start" button for your plan. Triggers respond to specific events or conditions, such as code changes in a code repository or manual approvals. When a trigger is activated, it sets the pipeline into motion, enabling automated and controlled execution of tasks.

Pipeline Catalog: A pipeline catalog, or pipeline marketplace, is a repository or collection of predefined, reusable pipelines. These pipelines are like well-crafted recipes for common development or deployment processes. Teams can use a pipeline catalog to access and share standardized, battle-tested pipeline templates. It simplifies the process of creating new pipelines by allowing developers to pick and choose from a catalog of pre-built, proven workflows. This enhances efficiency, reduces duplication of effort, and promotes best practices within an organization's software development processes.

Here is a showcase of how CI/CD can be used in the build - test - deploy - rollback process:

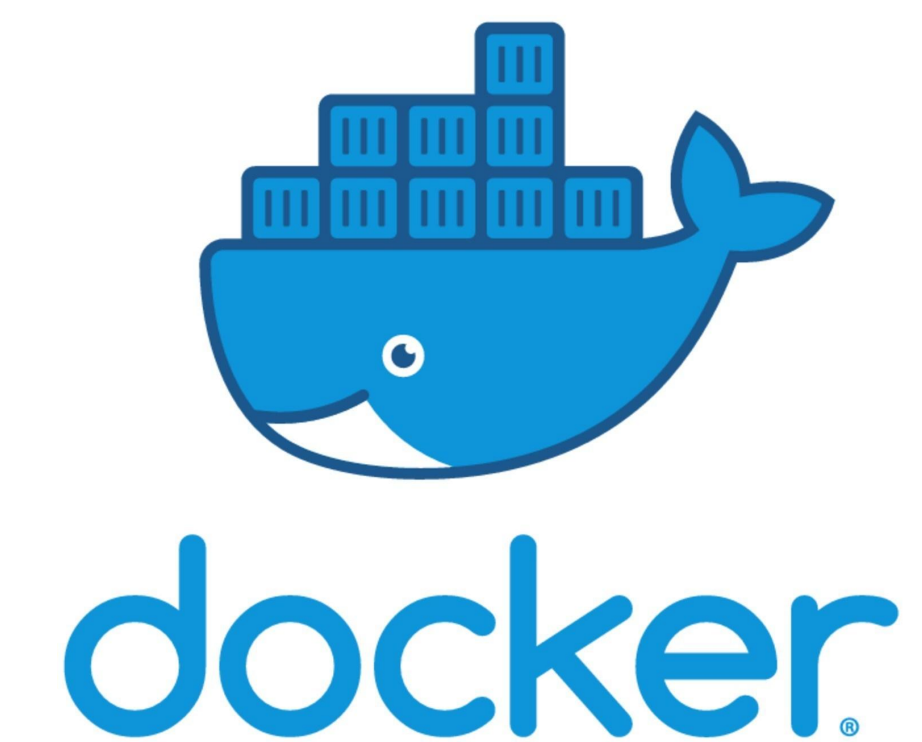


In the case of an inner pipeline failure, the 'on-error' section within the inner pipeline is invoked. After the 'on-error' actions are executed, the outer pipeline continues without interruption. Additionally, any actions specified in the 'finally' section within the inner pipeline are executed prior to the corresponding 'finally' section in the outer pipeline, regardless of what transpires within the inner pipeline.

Tools Used

Docker Desktop:

Docker is an application with a straightforward GUI which allows you to build, share, and run containerized applications and microservices. The real engine for our containers is Kubernetes but Docker provides a straightforward GUI to control and manage our containers



Windows Subsystem for Linux (WSL):

WSL is a virtual machine running linux using a modified linux kernel to run on top of your windows installation. WSL is a stripped down, command line only version that gives all the tools and versatility of linux without the overhead of a full virtual machine or the inconvenience of a dual boot.



Golang:

golang is an open-source programming language supported by Google. It is a statically typed language built for concurrency. It's high performance



Kubernetes:

Kubectl is a more powerful containerization tool then docker that can host more microservices than Docker on its own. Tekton runs a task in the form of a Kubernetes pod, where each step becomes a running container in the pod.



Ko:

The Tekton project uses ko to simplify the building of its container images from go source, push these images to the configured image repository and deploy these images into Kubernetes clusters.