SKILUP SM DAYS

by: DevOps Institute
ADVANCING THE HUMANS OF DEVOPS

Simplifying Microservices by Visualizing Your Application

Why the Logical Version of the Application is still relevant.

Tracy Ragan
CEO
DeployHub
@TracyRagan
Tracy@DeployHub.com

Agenda

Today we use many ‘visibility’ tools for tracking an application version.

Microservices obfuscate the view of the Application.

The impact is low confidence in deployments, ad hoc updates, discovery via incident and an inability to be pro-active.

We need to re-imagine CI and re-create the visualizations for clarifying the Application.

Tracy Ragan
CEO
@TracyRagan

- CEO and Co-Founder – DeployHub, Inc.
- Founding Board member of the CD Foundation
- Founding Board Member Eclipse Foundation
- DevOps Institute Ambassador,
- 20+ DevOps Experience.

SKILUP DAYS

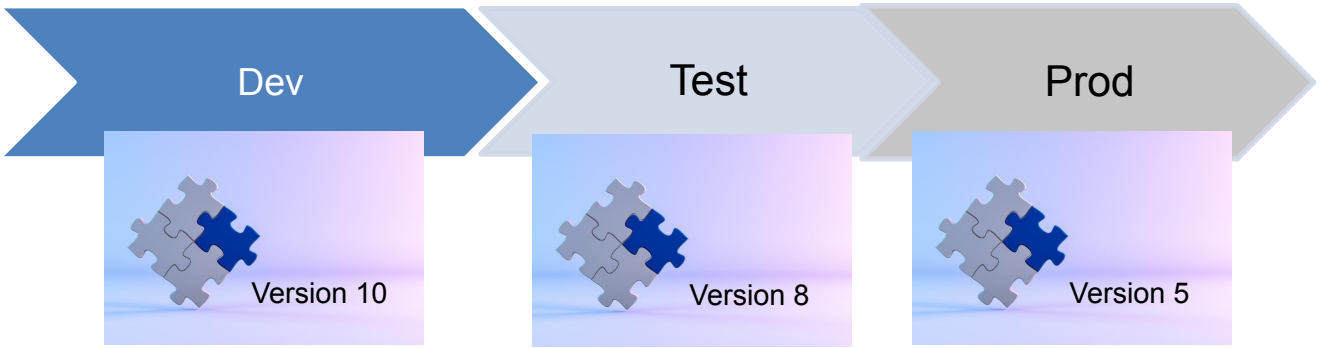
Microservices are loosely coupled and independently deployed functions that flow through the Continuous Delivery pipeline alone. They are the foundation of business agility.

Microservices create an endless cycle of changes moving out to K8s clusters continuously.

What We Do Now

Monolithic Pipeline

The all-important Check-in and Build – The Foundation of CD



The CI Step:

- Pull
- Compile/Link the Application, create a new version, generate BOM, Diff and Impact Reports
- Deploy to Dev

The Test Step:

- Deploy to Test
- Run Test
- Approve for Prod

The Prod Step:

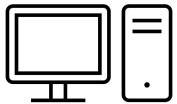
- Approve
- Deploy to Prod

Microservices Vs. Monolithic

We are taking our static application and breaking it into smaller puzzle pieces.

The New CI

Creates a Container – no “Integration” required.



Pull code based on labels, scan for transitive dependencies.

Create a Docker Container that includes a set services that change together.

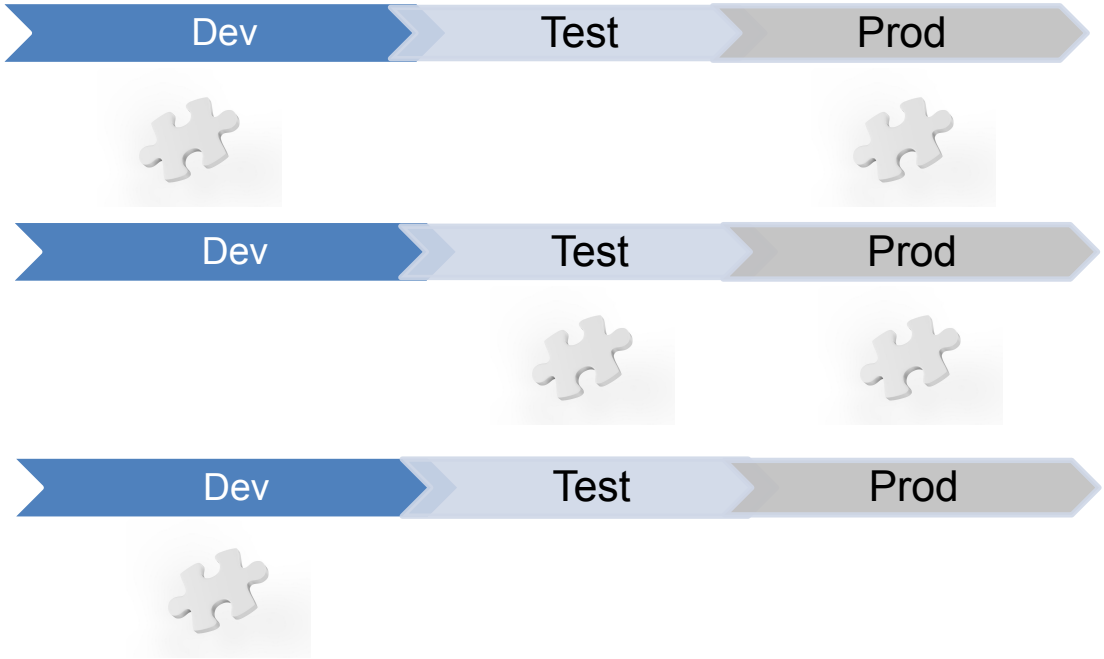
Check-in to a Container Registry

Repeat for many new microservices.

The New CD Pipeline

Microservice are Independent and Shared Across Applications

- Dev does not create an “application” much less tracks versions.
- You may not know when a new version of a service was released – you now have a new version of your application.

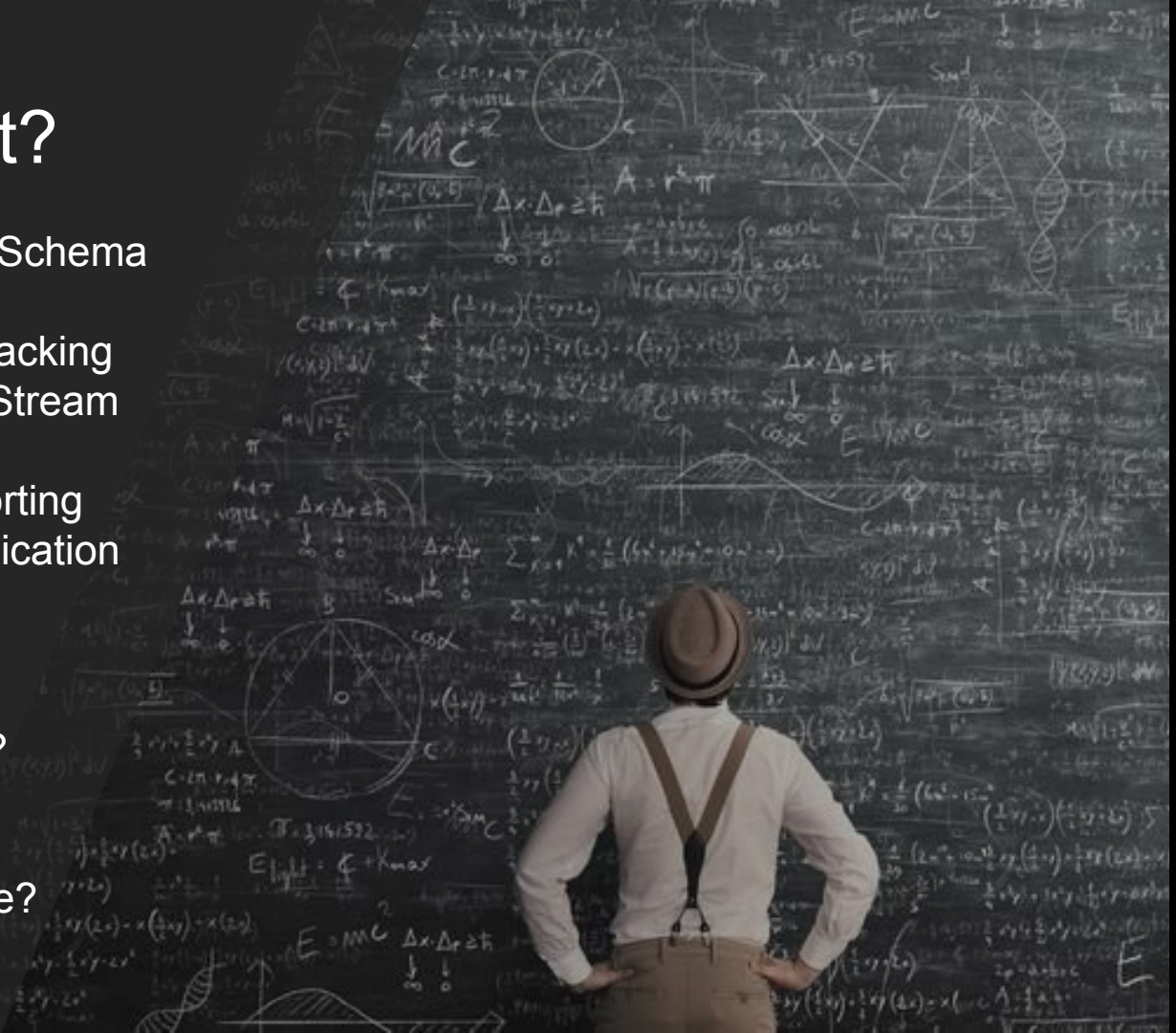


Only 15% of companies report ‘massive’ success with microservices

O’Reilly Survey Business Wire December 201

What is Lost?

- Application Version Schema
  - Impacts Testing
  - Impacts Bug Tracking
  - Impacts Value Stream
- Bill of Material Reporting
  - How is the Application Configured?
- Difference Reports
  - What was new?
- Impact Analysis
  - Should I release?



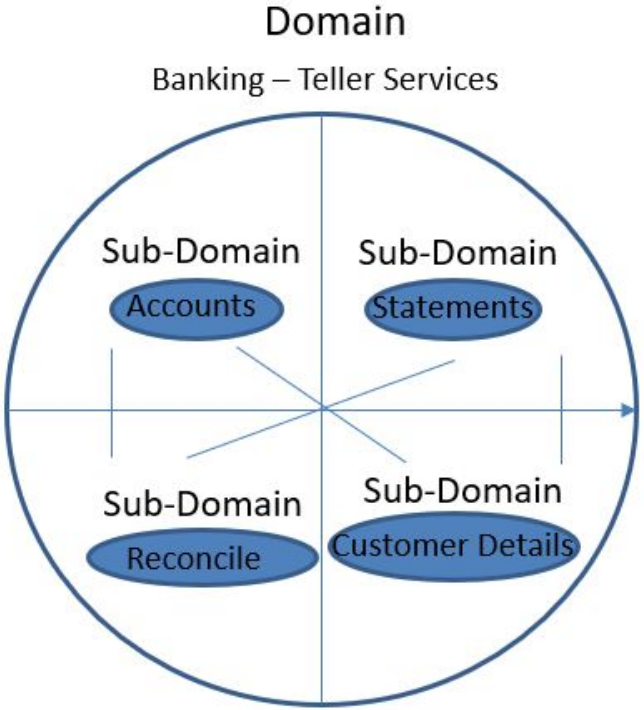
Visualizing the Logical View of the Application

Domain Driven Design

Organizing Your Microservices

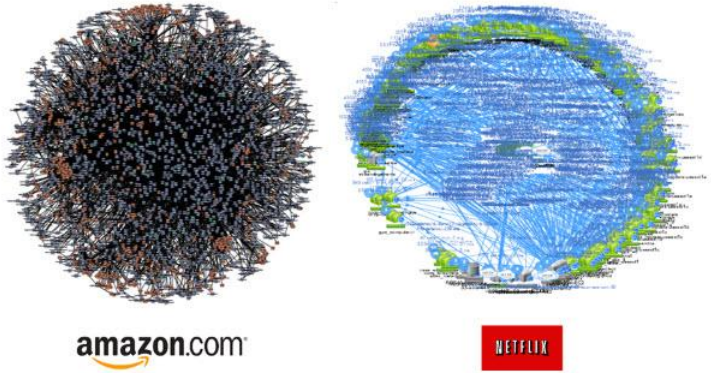
- Domain Driven Design is where you are managing an architecture based on the microservice 'problem space.'

- Domains can be defined based on your organizational patterns. Start by decomposing a few applications and you will begin recognizing their commonality. What is common are potential domains.
  - Login routines
  - Database calls
  - Logging



Reality of Configuration

Navigating the Deathstar



Haunted Graveyards, Frankenstein Clusters, when do we deprecate?

Configuration Management

Critical Data for both developers and SREs:

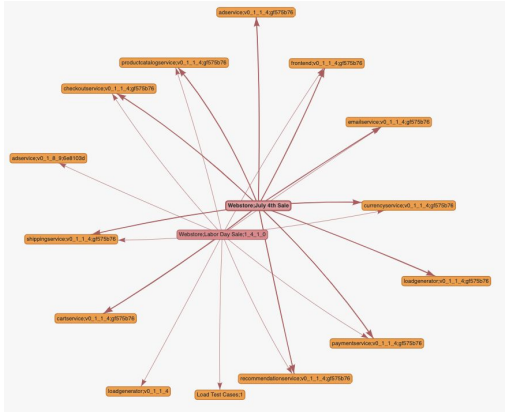
- Tracking what microservices your application consumes (Version and BOM).
- Knowing when a particular microservice is about to be updated or has been (Difference Reports).
- What cluster is the new service active in (Deployment Tracking).
- If I update a microservice who will I impact (Impact Analysis).

DeployHub

Microservice Configuration Management for Site Reliability, DevOps Engineers, and Release Teams

A Self-Service SaaS Solution for:

- **Sharing** Microservices based on a Domain Driven Design.
- **Restoring and Mapping** the “Logical” Application Versions.
- **Deploying and Tracking** Microservice Versions with Deployment Data across clusters.



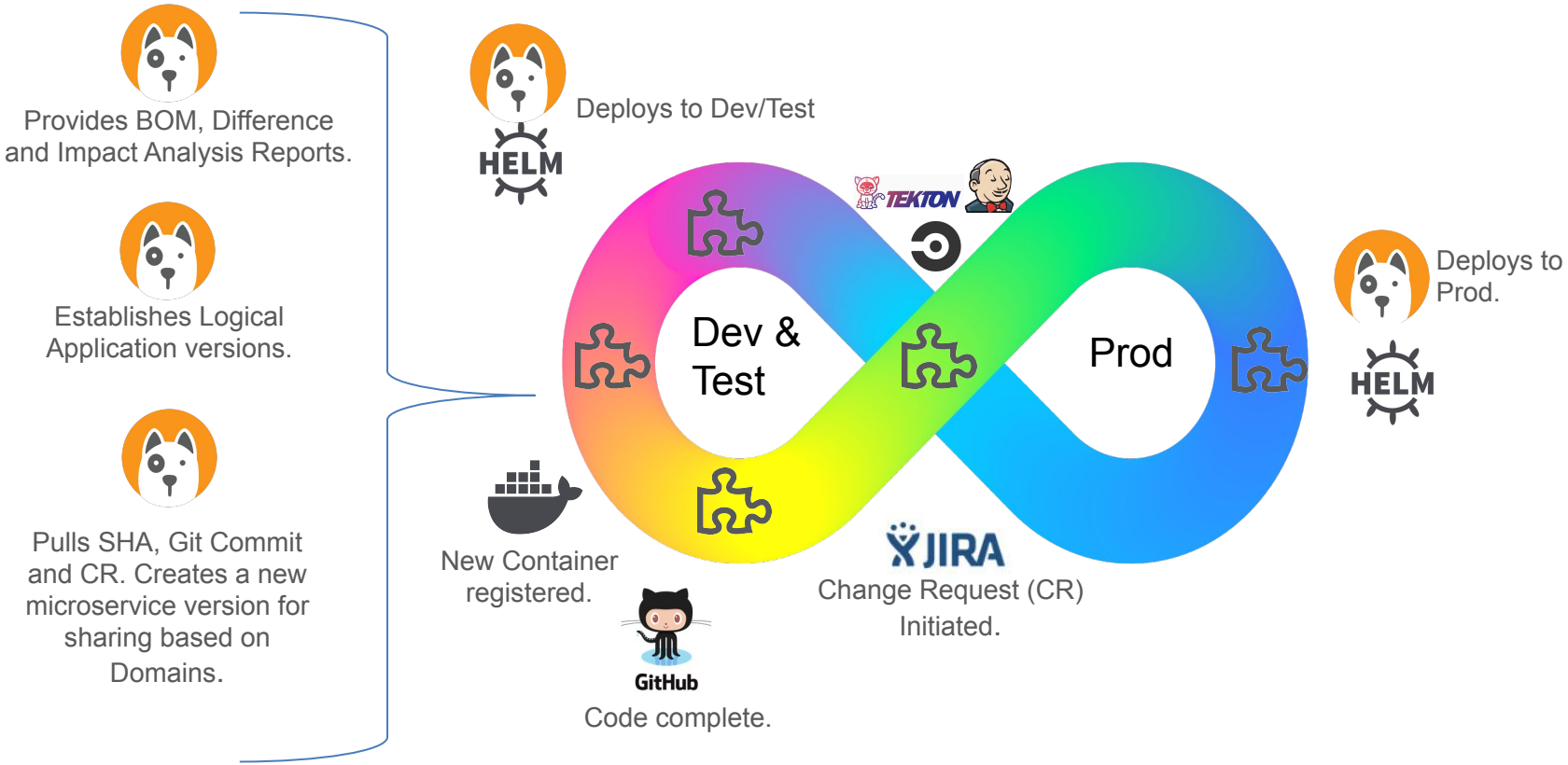
We open sourced the mapping - DeployHub Team is free and based on our Open source Project – Ortelius.io

Ortelius.io
Open Source Project

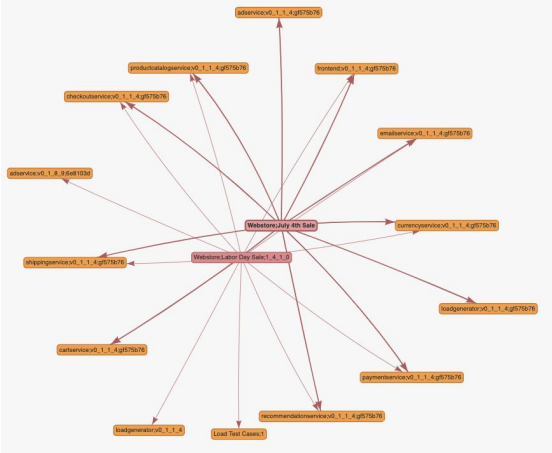
DeployHub Team
Hosted Open Source

DeployHub

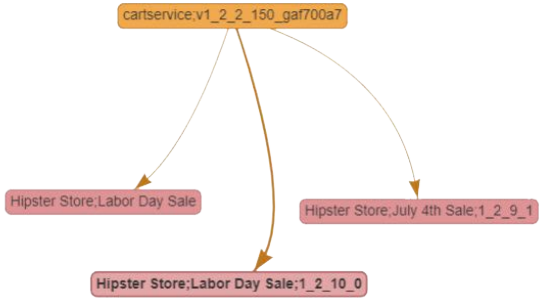
Enhancing the CD Ecosystem with Visualization



Providing Visibility into the Logical Application



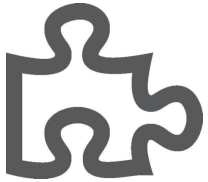
BOM Report



Impact Report

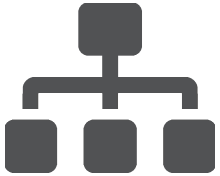
Difference Report

Results



Takes the guess work out of tracking who is consuming your microservices. Saves 3-6 hours of manual tracking work per team each week.

Microservice sharing via Domains reduces redundant coding by up to 50% and creates a flexible and dynamic ecosystem.



Confidently deploys microservices across clusters with the knowledge of their impact, before they land reducing incidents and confusion.

Visibility gives Site Reliability Engineers the confidence of knowing what is going on and the ability to make data-driven decisions quickly

THANK YOU!

Meet Me in the Network Chat Lounge for Questions

LinkedIn: https://www.linkedin.com/in/tracy-ragan-oms/
Twitter: @TracyRagan
Calendar: https://drift.me/tracyragan/meeting/coffeechat
Email: TracyRagan@DeployHub.com
Dig In at: DeployHub.com or Ortelius.io

SKILUP DAYS