

# Digital transformation towards microservices on multi-cloud: how to remove the anxiety

Manuel Schuller

Let me briefly introduce myself...



CloudBees



DevOps  
INSTITUTE  
Human of DevOps



**Manuel Schuller,**  
*DevOps Institute  
Ambassador*

## Digital transformation towards microservices on multi-cloud: How to remove the anxiety

Companies are transforming. Transforming towards what exactly? Sometimes the answer is “towards complexity, towards anxiety ... because we do not know exactly. In this session, learn how to mitigate anxiety with adequate principles and tools.

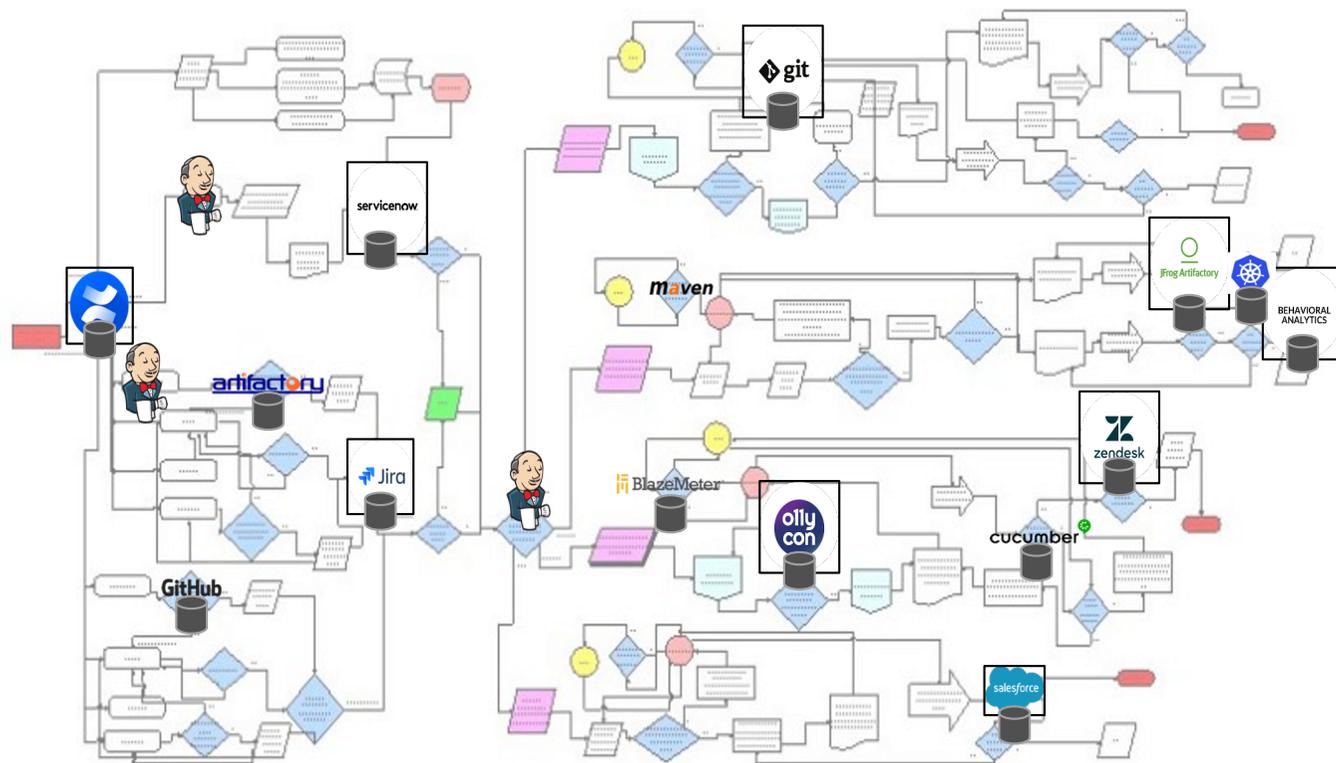


[linkedin.com/in/manuelschuller](https://www.linkedin.com/in/manuelschuller)

[@myDevOpsExpert](https://twitter.com/myDevOpsExpert)



# The problem is... building and releasing software is complicated.





# THERE IS DISCONNECT!

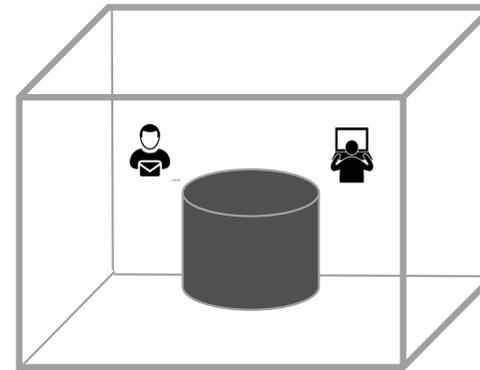
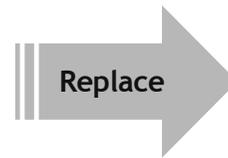
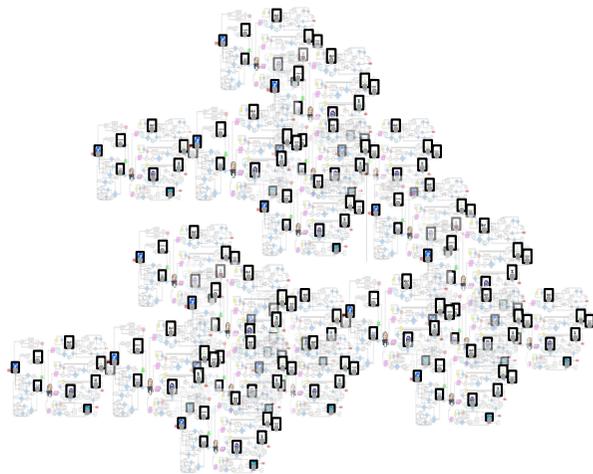




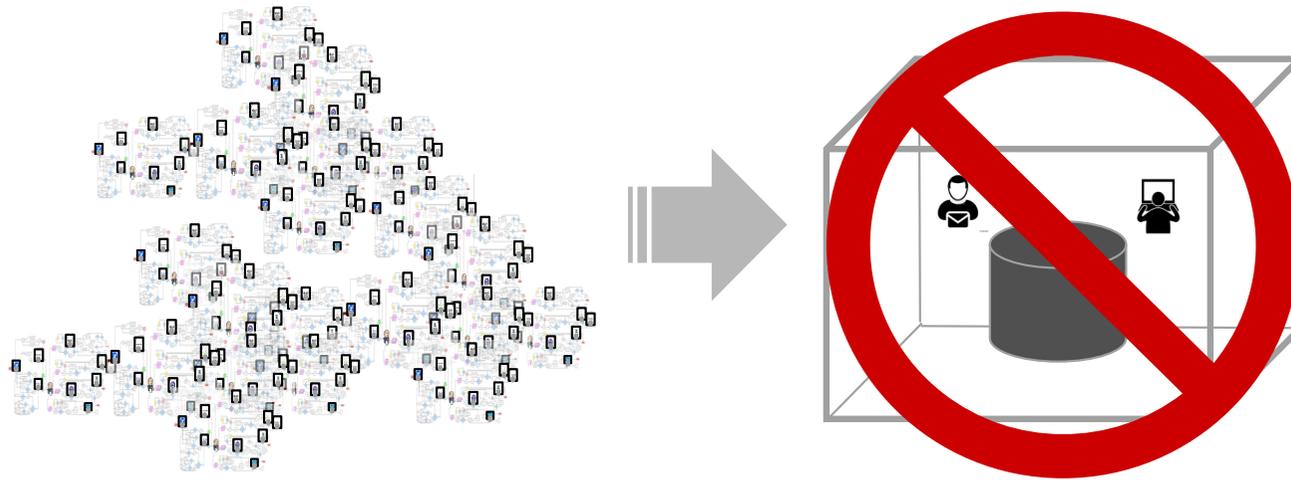
Scale and Complexity Exacerbate the Disconnect, and Break Traditional Tools and Methods



**Some might tell you the solution is...**  
...replace with one giant all-in-one system.

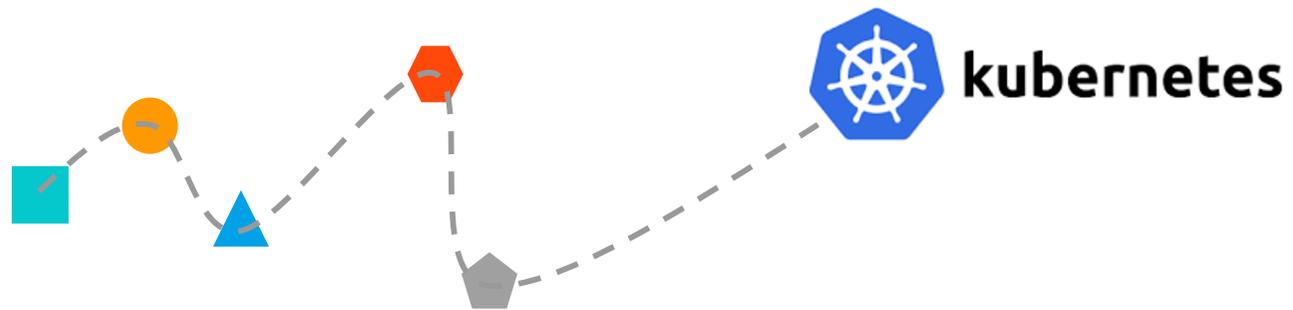


But it's a devops and software anti-pattern



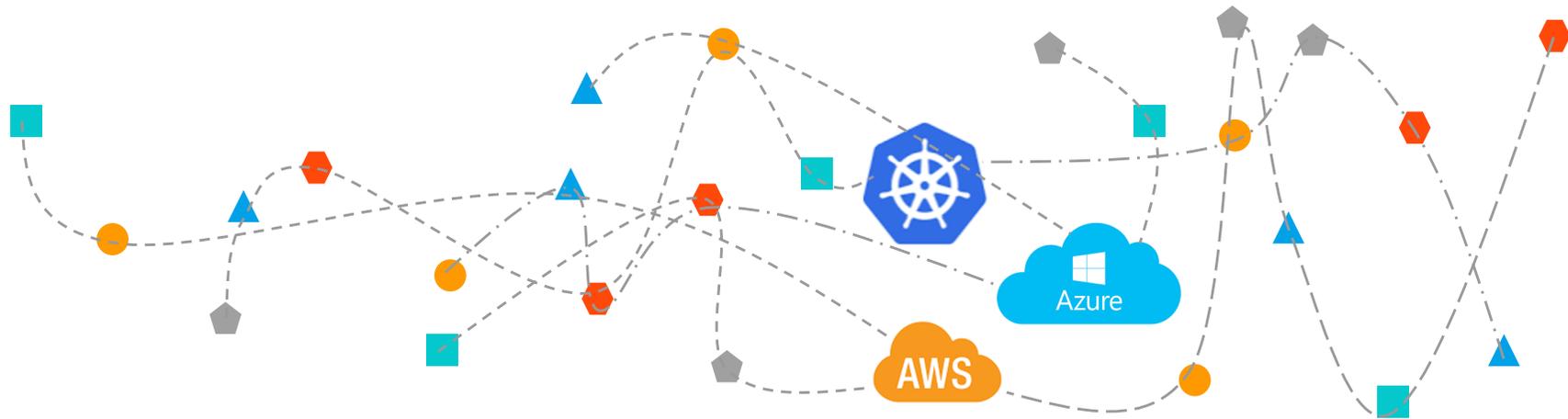
---

## In a Perfect World...



■ ● ▲ = Various DevOps Tooling (eg, Jira, Jenkins, HPQC, Selenium)

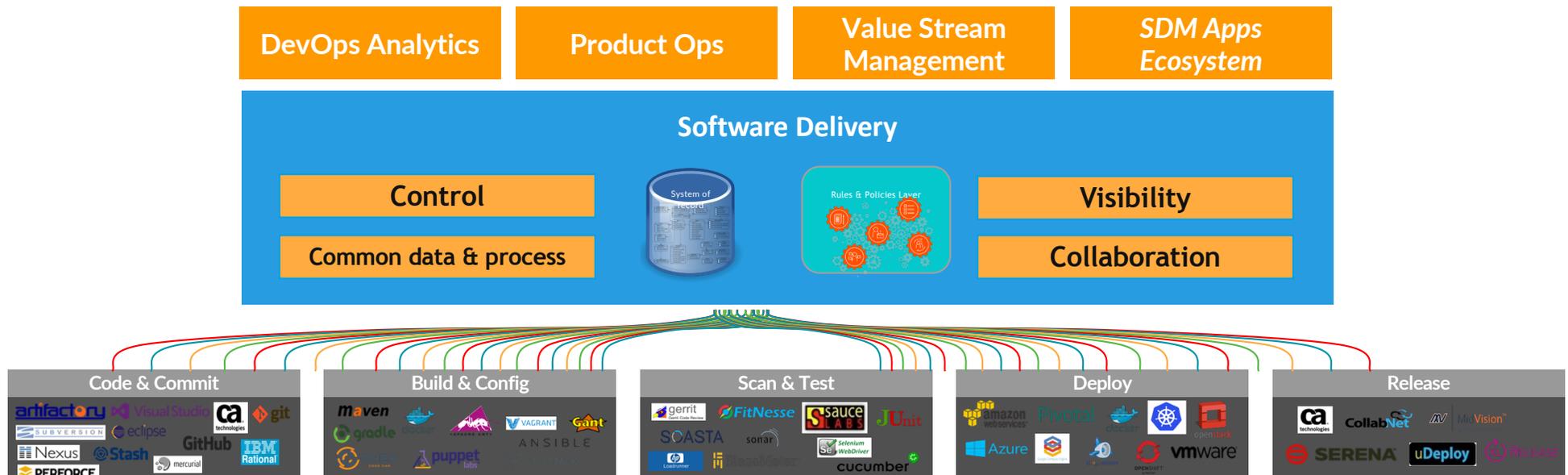
# But In Reality...



■ ● ▲ = Various DevOps Tooling (eg, Jira, Jenkins, Nexus, Selenium)

# What's better and more realistic ...

But add above



Embrace not  
replace

## 2019 Definition



Building a combination of **deployment automation**, **pipeline and environment management**, and **release orchestration** capabilities to simultaneously improve the quality, velocity and governance of application releases. These tools enable enterprises to scale release activities across multiple, diverse and multigenerational teams (e.g. DevOps), technologies (from legacy to traditional to microservices), development methodologies (agile, etc.), delivery patterns (e.g. continuous), pipelines, processes and their supporting toolchains...

---

## Three Key Ingredients

### Deployment Automation & Environment Management

Predictably deploy applications and microservices to any environment, at any scale.

### Pipeline Management & Release Orchestration

Manage release pipelines and dependencies across all teams, DevOps tools, and environments

### VSM

Track progress and identify patterns with 360° visibility into metrics across software supply chain.

Decouple

Decouple

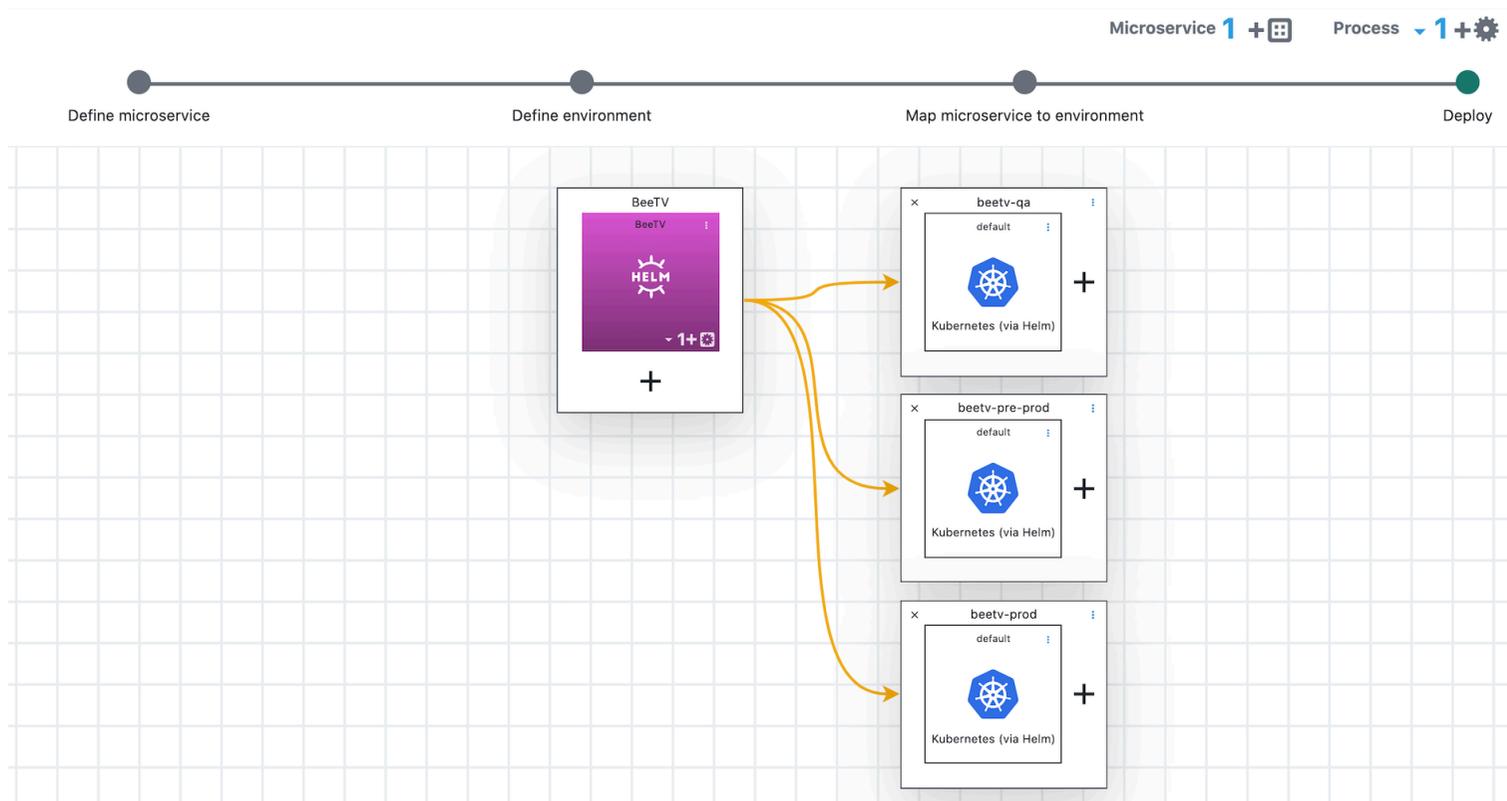
Decouple

Decouple

Decouple

# Deployment Automation & Environment Management

Predictably deploy applications and microservices to any environment, at any scale.



# Pipeline Management & Release Orchestration

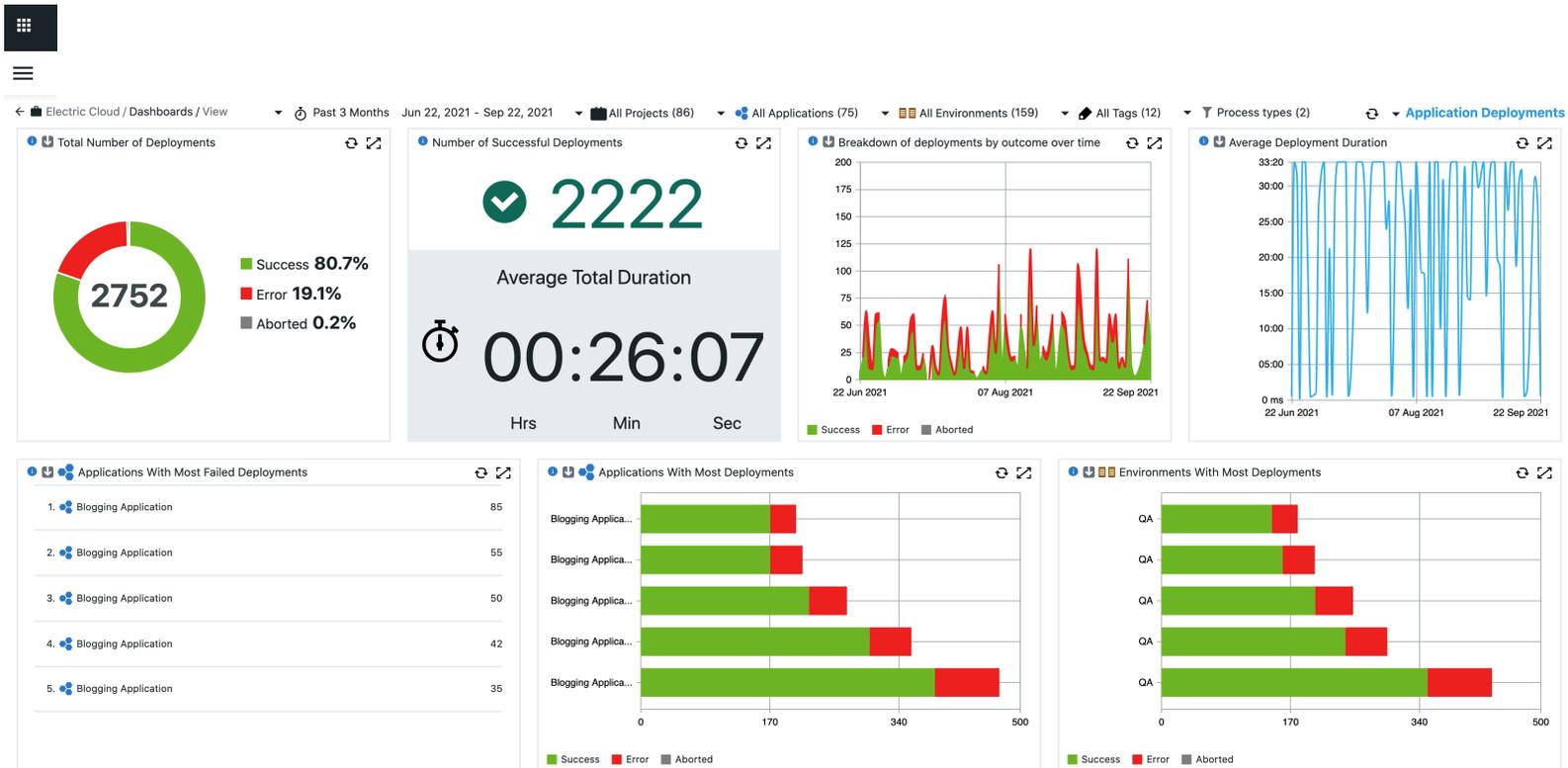
Manage release pipelines and dependencies across all teams, DevOps tools, and environments

The image displays a pipeline management interface with five stages, each represented by a vertical panel with a colored header:

- DEV (Orange header):** 7 Tasks
  - 1. Pull Jira Tickets (EC-JIRA GetIssues)
  - 2. Pull (ECSCM-Git CheckoutCode)
  - 3. Jenkins Build (EC-Jenkins RunAndWait)
  - 4. Book Environments (bookAllEnvts)
  - 5. Dev Tasks
    - a. Deploy to DEV (Deployer)
    - b. Run SonarQube (EC-SonarQube Run Sonar Scanner)
  - 6. Unit Tests (Unit test)
- QA (Purple header):** 3 Tasks
  - 1. Provision QA and Deploy (Deployer)
  - 2. Run Functional Tests (Functional Tests)
  - 3. Decommission QA (EF-Utilities Decommission Environments)
- Pre-Production (Blue header):** 5 Tasks
  - 1. Book PreProd and Deploy (Deployer)
  - 2. Run SIT Tests (Integration Tests)
  - 3. Send Slack Notification (EC-Slack Send Realtime Message)
  - 4. Verify QA / Notify (Manual)
  - 5. Create ServiceNow Ticket (EC-ServiceNow CreateChangeRequest)
- PROD (Green header):** 4 Tasks
  - 1. Check for Outage (EC-Moogsoft Get System Summary)
  - 2. Deploy to Prod (Deployer)
  - 3. Run Smoke Tests (Smoke Test)
  - 4. Update Prod Status (Command)
- Audit Reports (Grey header):** 4 Tasks
  - 1. Approval Report (EC-AuditReports generateApprovalAuditReport)
  - 2. Evidence Report (EC-AuditReports generateEvidenceLinksAuditReport)
  - 3. Duration Report (EC-AuditReports generateTaskDurationAuditReport)
  - 4. Generate Environment Audit (EC-AuditReports generateProductionAuditReport)

# Insight

Track progress and identify patterns with 360° visibility into metrics across DevOps processes



# Insight

Value stream

Motorcycle Store Demo / Releases / Motorcycle Release / Command Center

Release Plan **30%**

Open Defects **36** +0%

Days to Delivery **21**

**100%** →  
Planning Complete  
**Planning**

**63%** →  
Development Complete  
**Dev**

**89%** ↗  
Build Success Rate  
**Build**

**90%** →  
Test Success Rate  
**Test**

**80%** ↗  
Success Rate  
**Deploy**

**11**  
Total Incidents  
**Operate**

**33**  
Stories

**21 of 33**  
Stories Resolved

**225 of 252**  
Successful Builds

**100%**  
Automated Tests

**121 of 139**  
Successful Application Deployments

**5**  
Resolved Incidents

**33**  
Stories Estimated

**99%**  
Unit Test Success Rate

**00:12:00**  
Time to Build

**00:00:11**  
Time to Test

**108 of 144**  
Successful Microservice Deployments

**1 Day**  
**00:00:00**  
Average Time to Resolve

# Insight

## Production Audit Report

### General Information

Label	Value
AWS_REGION	us-east-2
BUILD_ID_NUM	9766
CURRENT_APP_VERSION	1.2.1
CURRENT_RELEASE	2
CURRENT_VERSION	1

### Microservice Information

MICROSERVICE	ARTIFACT_DIR	CURRENT_BUILD	CURRENT_IMAGE_TAG	DEPLOYMENT_TYPE	SCHEDULER_ORCH	SCM_USER	IMAGE_REPOSITORY	PULL_DIR	REGISTRY_TYPE	REGISTRY_VALUE	SCM_COMMENT	SCM_LAST_COMMIT	SCM_TYPE	SCM_ADDR	
CARTS	/builds3/carts	160	1.2.159	CONTAINER	KUBERNETES	elektrikTomcat	carts	/builds3	AMAZON ECR REGISTRY	731977058523.dkr.ecr.us-east-2.amazonaws.com	Changed cart color	2017-06-13 04:03:07+00:00	GITHUB	https://github.com/elektrikTomcat/carts.git	
CARTS-DB			latest	CONTAINER	KUBERNETES		mongo		DOCKER HUB						
CATALOGUE	/builds3/catalogue	157	1.2.156	CONTAINER	KUBERNETES	elektrikTomcat	catalogue	/builds3	AMAZON ECR REGISTRY	731977058523.dkr.ecr.us-east-2.amazonaws.com	Added merchandise item	2017-06-14 05:02:47+00:00	GITHUB	https://github.com/elektrikTomcat/catalogue.git	
CATALOGUE-DB	/builds3/catalogue	149	1.2.1	CONTAINER	KUBERNETES	elektrikTomcat	catalogue-db	/builds3	AMAZON ECR REGISTRY	731977058523.dkr.ecr.us-east-2.amazonaws.com	Initial commit	2017-06-14 05:32:48+00:00	GITHUB	https://github.com/elektrikTomcat/catalogue.git	
FRONT-END	/builds3/front-end	201	1.2.200	CONTAINER	KUBERNETES	elektrikTomcat	front-end	/builds3	AMAZON_ECR_REGISTRY	731977058523.dkr.ecr.us-east-2.amazonaws.com	Changed banner color, carousel pics	2017-08-08 10:53:28+00:00	GITHUB	https://github.com/elektrikTomcat/front-end.git	att-late
ORDERS	/builds3/orders	157	1.2.156	CONTAINER	KUBERNETES	elektrikTomcat	orders	/builds3	AMAZON ECR REGISTRY	731977058523.dkr.ecr.us-east-2.amazonaws.com	Added calculator functionality	2017-06-24 10:32:06+00:00	GITHUB	https://github.com/elektrikTomcat/orders.git	
ORDERS-DB			latest	CONTAINER	KUBERNETES		mongo		DOCKER HUB						
PAYMENT	/builds3/payment	153	1.2.152	CONTAINER	KUBERNETES	elektrikTomcat	payment	/builds3	AMAZON ECR REGISTRY	731977058523.dkr.ecr.us-east-2.amazonaws.com	Added foreign currency	2017-06-16 04:12:41+00:00	GITHUB	https://github.com/elektrikTomcat/payment.git	
QUEUE-MASTER	/builds3/queue-master	152	1.2.151	CONTAINER	KUBERNETES	elektrikTomcat	queue-master	/builds3	AMAZON ECR REGISTRY	731977058523.dkr.ecr.us-east-2.amazonaws.com	Modified existiting queue	2017-06-19 11:00:58+00:00	GITHUB	https://github.com/elektrikTomcat/queue-master.git	
RABBITMQ			3.6.8	CONTAINER	KUBERNETES		rabbitmq		DOCKER HUB						
SHIPPING	/builds3/shipping	154	1.2.153	CONTAINER	KUBERNETES	elektrikTomcat	shipping	/builds3	AMAZON ECR REGISTRY	731977058523.dkr.ecr.us-east-2.amazonaws.com	Modified shipping calculator	2017-06-22 01:30:02+00:00	GITHUB	https://github.com/elektrikTomcat/shipping.git	
USER	/builds3/user	157	1.2.151	CONTAINER	KUBERNETES	elektrikTomcat	user	/builds3	AMAZON ECR REGISTRY	731977058523.dkr.ecr.us-east-2.amazonaws.com	Added Oauth	2017-06-20 02:02:08+00:00	GITHUB	https://github.com/elektrikTomcat/user.git	
USER-DB	/builds3/user	158	1.2.10	CONTAINER	KUBERNETES	elektrikTomcat	user-db	/builds3	AMAZON ECR REGISTRY	731977058523.dkr.ecr.us-east-2.amazonaws.com	Updated db schema	2017-06-12 05:33:59+00:00	GITHUB	https://github.com/elektrikTomcat/user.git	
ZIPKIN			latest	CONTAINER	KUBERNETES		zipkin		DOCKER HUB	openzipkin					
ZIPKIN-CRON			1.4.0	CONTAINER	KUBERNETES		zipkin-dependencies		DOCKER HUB	openzipkin					
ZIPKIN-DB			1.20.0	CONTAINER	KUBERNETES		zipkin-mysql		DOCKER HUB	openzipkin					

https://github.com/elektrikTomcat/catalogue.git

# Insight

## Environment Inventory Report

Search Filter:

Store	Current Inventory/Version					Inventory Deployment History Details - Date Deployed (Default) ▼				
104th Avenue-6024	oe 1.1	oe-dtf 2.2	voltage 2.2	path 2.0	ccm 1.0	oe version: 1.1 2017-10-04T18:50:38.660Z	oe-dtf version: 2.2 2017-10-04T18:50:39.066Z	voltage version: 2.2 2017-10-04T18:50:39.434Z	path version: 2.0 2017-10-04T18:50:39.760Z	ccm version: 1.0 2017-10-04T18:50:40.171Z
Albuquerque-7194	voltage 2.0	oe-dtf 2.2	oe 1.1	path 2.2	ccm 1.1	voltage version: 2.0 2017-10-04T18:50:51.900Z	oe-dtf version: 2.2 2017-10-04T18:50:52.204Z	oe version: 1.1 2017-10-04T18:50:52.414Z	path version: 2.2 2017-10-04T18:50:52.646Z	ccm version: 1.1 2017-10-04T18:50:52.841Z
Augusta-7209	oe-dtf 2.1	path 2.2	voltage 2.2	oe 2.2	ccm 2.0	oe-dtf version: 2.1 2017-10-04T18:51:09.930Z	path version: 2.2 2017-10-04T18:51:10.229Z	voltage version: 2.2 2017-10-04T18:51:10.480Z	oe version: 2.2 2017-10-04T18:51:10.664Z	ccm version: 2.0 2017-10-04T18:51:10.811Z
Austin South-7190	oe-dtf 2.2	path 1.0	oe 1.1	voltage 2.2	ccm 2.2	oe-dtf version: 2.2 2017-10-04T18:51:30.364Z	path version: 1.0 2017-10-04T18:51:30.671Z	oe version: 1.1 2017-10-04T18:51:30.821Z	voltage version: 2.2 2017-10-04T18:51:30.997Z	ccm version: 2.2 2017-10-04T18:51:31.172Z
Austin-7154	oe-dtf 2.2	oe 2.1	path 2.0			oe-dtf version: 2.2 2017-10-04T18:51:45.812Z	oe version: 2.1 2017-10-04T18:51:46.092Z	path version: 2.0 2017-10-04T18:51:46.258Z		
Bakersfield-7281	path 2.0					path version: 2.0 2017-10-04T18:52:01.858Z				
Baton Rouge-7187	voltage 1.1	path 2.1	ccm 1.1			voltage version: 1.1 2017-10-04T18:52:26.520Z	path version: 2.1 2017-10-04T18:52:26.930Z	ccm version: 1.1 2017-10-04T18:52:27.105Z		
Beverton-7164	ccm 1.0	oe 1.1	path 2.0			ccm version: 1.0 2017-10-04T18:52:47.115Z	oe version: 1.1 2017-10-04T18:52:47.511Z	path version: 2.0 2017-10-04T18:52:47.696Z		
Birmingham-7224	voltage 2.0	path 2.0	ccm 1.1	oe 1.1		voltage version: 2.0 2017-10-04T18:53:10.171Z	path version: 2.0 2017-10-04T18:53:10.476Z	ccm version: 1.1 2017-10-04T18:53:10.796Z	oe version: 1.1 2017-10-04T18:53:10.992Z	
Boynton Beach-7113	voltage 2.0	oe-dtf 2.1	path 2.2	ccm 2.0		voltage version: 2.0 2017-10-04T18:53:30.698Z	oe-dtf version: 2.1 2017-10-04T18:53:30.953Z	path version: 2.2 2017-10-04T18:53:31.252Z	ccm version: 2.0 2017-10-04T18:53:31.427Z	
Bradenton CBC-7955	oe 1.1	voltage 2.0	path 2.0			oe version: 1.1 2017-10-04T18:53:48.918Z	voltage version: 2.0 2017-10-04T18:53:49.309Z	path version: 2.0 2017-10-04T18:53:49.508Z		
Brandywine-7257	oe 2.1	oe-dtf 2.2	path 2.0			oe version: 2.1 2017-10-04T18:54:09.297Z	oe-dtf version: 2.2 2017-10-04T18:54:09.576Z	path version: 2.0 2017-10-04T18:54:09.754Z		
Brooklyn Park-6008	oe-dtf 2.2	path 2.2	oe 2.1	voltage 2.1	ccm 2.1	oe-dtf version: 2.2 2017-10-04T18:45:00.349Z	path version: 2.2 2017-10-04T18:45:00.541Z	oe version: 2.1 2017-10-04T18:45:00.696Z	voltage version: 2.1 2017-10-04T18:45:00.997Z	ccm version: 2.1 2017-10-04T18:45:01.215Z

# Insight

Track progress and identify patterns with 360° visibility into metrics across DevOps processes

Production Audit Report Cluster Status Microservices Dependencies Environment Overview Puppet

Dev

APP/SERVICE NAME	VERSION	DEPLOYMENT TYPE	SONAR	DEPLOYMENT DATE
🔗 Carts	1.2.22	K8 Deployment	93%	7/1/2017 17:50:01
🔗 Carts-DB	1.2.22	K8 Deployment	-	7/1/2017 17:50:01
🔗 Catalogue	1.2.22	K8 Deployment	91%	7/1/2017 17:50:01
🔗 Catalogue-DB	1.2.22	K8 Deployment	-	7/1/2017 17:50:01
🔗 Front-End	1.2.23	K8 Rolling Update	94%	8/8/2017 10:53:45
🔗 Orders	1.2.22	K8 Deployment	99%	7/1/2017 17:50:01
🔗 Orders DB	1.2.22	K8 Deployment	97%	7/1/2017 17:50:01
🔗 Payment	1.2.22	K8 Deployment	90%	7/1/2017 17:50:01
🔗 Queue Master	1.2.22	K8 Deployment	89%	7/1/2017 17:50:01
🔗 RabbitMQ	1.2.22	K8 Deployment	-	7/1/2017 17:50:01
🔗 Shipping	1.2.22	K8 Deployment	93%	7/1/2017 17:50:01
🔗 User	1.2.23	K8 Deployment	93%	7/1/2017 17:50:01
🔗 User DB	1.2.22	K8 Deployment	-	7/1/2017 17:50:01
🔗 Zipkin	1.2.22	K8 Deployment	-	7/1/2017 17:50:01
🔗 Zipkin Cron	1.2.22	K8 Deployment	-	7/1/2017 17:50:01
🔗 Zipkin DB	1.2.22	K8 Deployment	-	7/1/2017 17:50:01

QA

APP/SERVICE NAME	VERSION	DEPLOYMENT TYPE	SELENIUM	DEPLOYMENT DATE
🔗 Carts	1.2.22	K8 Deployment	✓	7/1/2017 17:50:01
🔗 Carts-DB	1.2.22	K8 Deployment	✓	7/1/2017 17:50:01
🔗 Catalogue	1.2.22	K8 Deployment	✓	7/1/2017 17:50:01
🔗 Catalogue-DB	1.2.22	K8 Deployment	✓	7/1/2017 17:50:01
🔗 Front-End	1.2.23	K8 Rolling Update	✓	8/8/2017 10:54:46
🔗 Orders	1.2.22	K8 Deployment	✓	7/1/2017 17:50:01
🔗 Orders DB	1.2.22	K8 Deployment	✓	7/1/2017 17:50:01
🔗 Payment	1.2.22	K8 Deployment	✓	7/1/2017 17:50:01
🔗 Queue Master	1.2.22	K8 Deployment	✓	7/1/2017 17:50:01
🔗 RabbitMQ	1.2.22	K8 Deployment	✓	7/1/2017 17:50:01
🔗 Shipping	1.2.22	K8 Deployment	✓	7/1/2017 17:50:01
🔗 User	1.2.23	K8 Deployment	✓	7/1/2017 17:50:01
🔗 User DB	1.2.22	K8 Deployment	✓	7/1/2017 17:50:01
🔗 Zipkin	1.2.22	K8 Deployment	✓	7/1/2017 17:50:01
🔗 Zipkin Cron	1.2.22	K8 Deployment	✓	7/1/2017 17:50:01
🔗 Zipkin DB	1.2.22	K8 Deployment	✓	7/1/2017 17:50:01

Prod

APP/SERVICE NAME	VERSION	DEPLOYMENT TYPE	HEALTH	DEPLOYMENT DATE
🔗 Carts	1.2.22	K8 Deployment	✗	7/1/2017 17:50:01
🔗 Carts-DB	1.2.22	K8 Deployment	✓	7/1/2017 17:50:01
🔗 Catalogue	1.2.22	K8 Deployment	✗	7/1/2017 17:50:01
🔗 Catalogue-DB	1.2.22	K8 Deployment	✓	7/1/2017 17:50:01
🔗 Front-End	1.2.23	K8 Rolling Update	✓	8/8/2017 10:56:31
🔗 Orders	1.2.22	K8 Deployment	✗	7/1/2017 17:50:01
🔗 Orders DB	1.2.22	K8 Deployment	✓	7/1/2017 17:50:01
🔗 Payment	1.2.22	K8 Deployment	✓	7/1/2017 17:50:01
🔗 Queue Master	1.2.22	K8 Deployment	✓	7/1/2017 17:50:01
🔗 RabbitMQ	1.2.22	K8 Deployment	✓	7/1/2017 17:50:01
🔗 Shipping	1.2.22	K8 Deployment	✓	7/1/2017 17:50:01
🔗 User	1.2.23	K8 Deployment	✗	7/1/2017 17:50:01
🔗 User DB	1.2.22	K8 Deployment	✓	7/1/2017 17:50:01
🔗 Zipkin	1.2.22	K8 Deployment	✓	7/1/2017 17:50:01
🔗 Zipkin Cron	1.2.22	K8 Deployment	✓	7/1/2017 17:50:01
🔗 Zipkin DB	1.2.22	K8 Deployment	✓	7/1/2017 17:50:01

The background is a solid blue color with a faint, light blue grid pattern that recedes into the distance, creating a sense of depth. The grid lines are slightly blurred and fade out towards the top and right edges.

**Secure and Notarize**

# BoM is back (especially since May 12<sup>th</sup>)!



Join Extra Crunch

Login

Search Q

## Biden's executive order on cybersecurity should include behavior transparency

Ben Higgins @ExtraHop / 4:58 PM GMT+2 • June 21, 2021

THE WHITE HOUSE



Administration Priorities COV

(j) the term “Software Bill of Materials” or “SBOM” means a formal record containing the details and supply chain relationships of various components used in building software. Software developers and vendors often create products by assembling existing open source and commercial software components. The SBOM enumerates these components in a product. It is analogous to a list of ingredients on food packaging. An SBOM is useful to those who develop or manufacture software, those who select or purchase software, and those who operate software. Developers often use available open source and third-party software components to create a product; an SBOM allows the builder to make sure those components are up to date and to respond quickly to new vulnerabilities. Buyers can use an SBOM to perform vulnerability or license analysis, both of which can be used to evaluate risk in a product. Those who operate software can use SBOMs to quickly and easily determine whether they are at potential risk of a newly discovered vulnerability. A widely used, machine-readable SBOM format allows for greater benefits through automation and tool integration. The SBOMs gain greater value when collectively stored in a repository that can be easily queried by other applications and systems. Understanding the supply chain of software, obtaining an SBOM, and using it to analyze known vulnerabilities are crucial in managing risk.



Comment

---

## Typical use cases for the BoM

### Automation

- Deploying and releasing software

### Compliance

- Know what has been released
- Review, audit, cross-organization

### Security

- Assessment, audit, DevSecOps automation

### Understand the complexity

... and, like Unix' YACC (Yet another compiler-compiler) in 1975, BoM could be described as YAAA (Yet another artifact-artifact)!

---

## How DevOps principles can help

Continuous Integration - executes a set of steps that integrate the solution on a regular basis, ideally upon each change: typically upon check-in, a set of tests are executed that validates the process. From a software BoM perspective CI can also update the BoM with the current build, and then push a working version to an escrow repository owned and operated by the 3rd parties.

Continuous Delivery - delivers software in a working state on a frequent basis. In this sense the escrow can just be another deployment target to deploy the outputs to, albeit probably not actively spinning up infrastructure on the 3rd party fabric.

Containers - a straightforward way to package and deliver! But also the ability to quickly recreate a specific environment, ensuring the ability to recreate any product release build, hence that any app can be rebuilt.

# Application Release Pipeline, stage: PROD

The screenshot displays the configuration for the '8. PROD - Production Release' stage. It is organized into two main sections:

- 4 Entry Gate Rules:**
  - 1. BoM Entry Check (Approval) - Define
  - 2. CMDB Check (Approval) - Define
  - 3. Immutability Check (Approval) - Define
  - 4. SKU Check (Approval) - Define
- 3 Tasks:**
  - 1. Deploy in PROD (deployWithJenkins) - Define
  - 2. Run Smoke Tests (Procedure) - Define
  - 3. Close CR in GSD (Procedure) - Define

At the bottom of each section, there are 'Add +' and 'Copy' icons.

---

## BoM: Single Source Of Software Delivery Truth

Where BoM (**B**ill **o**f **M**aterial) was meant to describe how a software was *built*, it has now become how the software is *released (Delivered)*.

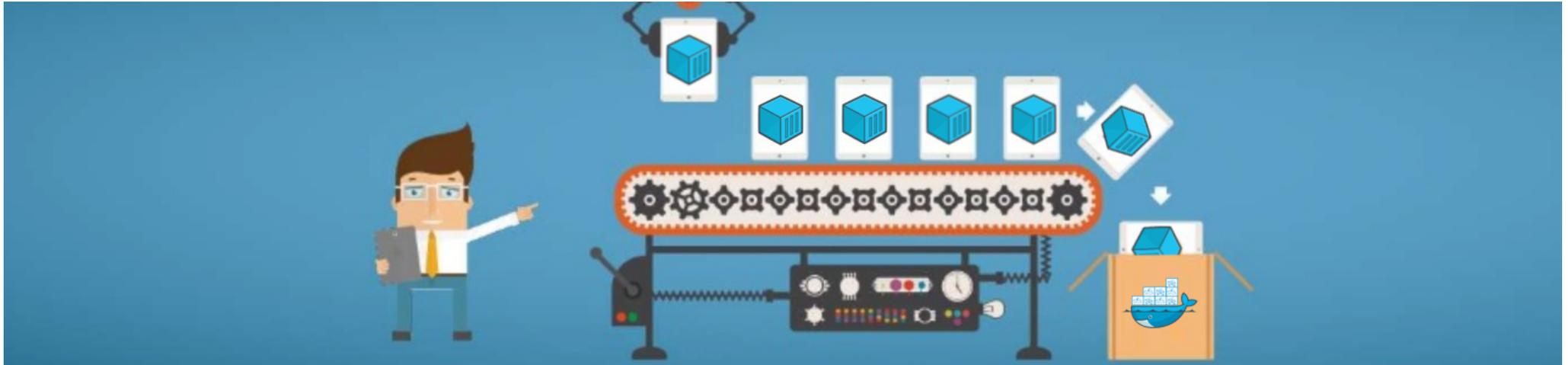
Which, at the end of the day, is just an extension of the notion

Extension which is comparable to audit and compliance evolutions over time

Focused on Software Delivery Security (or *how to secure my supply chain*)

Typically escrowed (just like you do at your notary)

Where BoM was made of *sources*, it is now also made of *results*, because results *will* change over time. BoM, as a consequence, depends on the time it is generated, and has become an artifact



# Digital transformation towards microservices on multi-cloud: how to remove the anxiety

Manuel Schuller

**Thank you!**