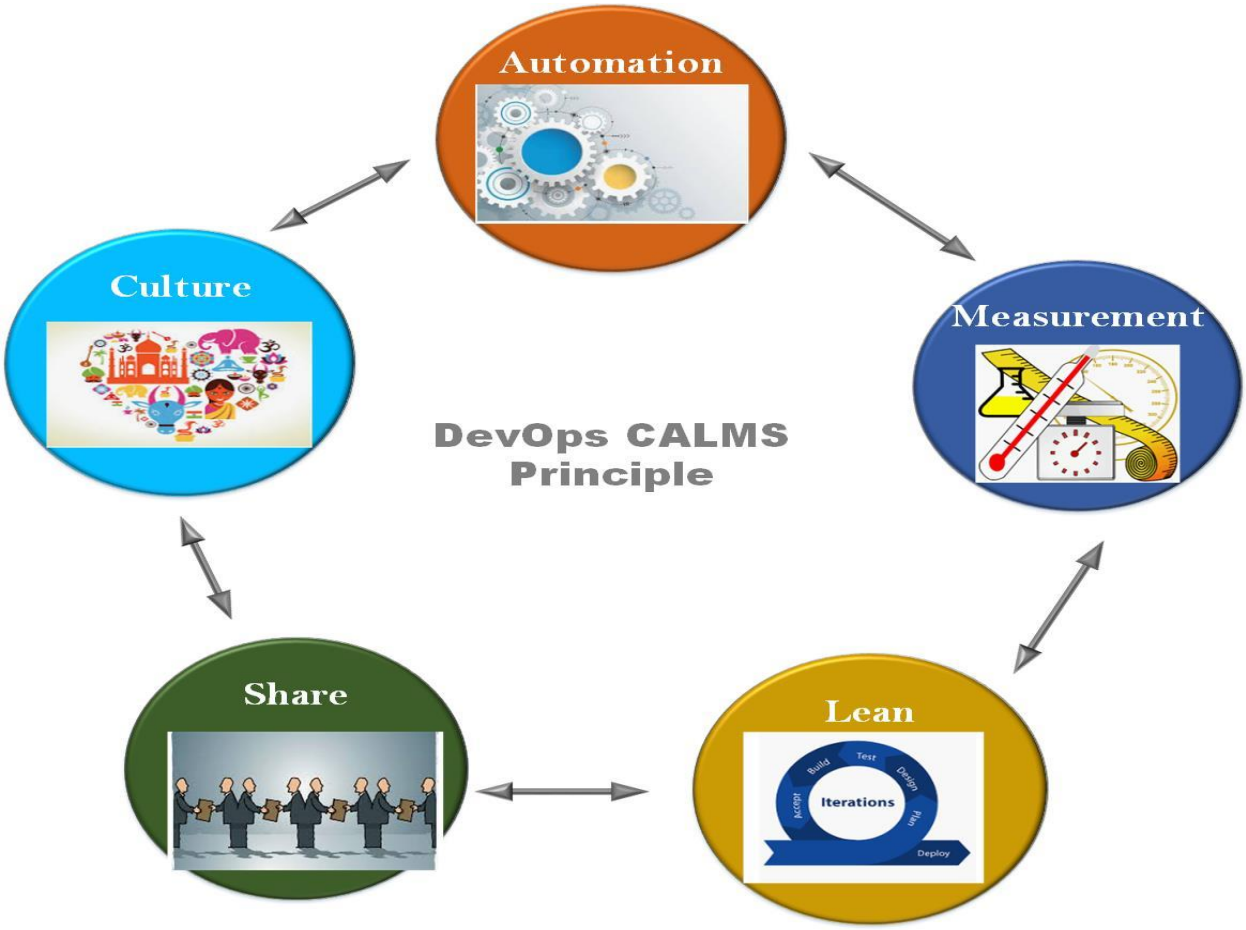
Microservice + DevOps is like 1+1 = 11 not 2?

I recommend to go through https://www.linkedin.com/pulse/realization-devops-equivalent-speaking-truth-jyoti-ranjan/ so that we do not get carried away by the content of this article which might allude in believing that DevOps is JUST an automation activity.

Let us recollect!

Realization of DevOps requires leaving traditional approach and mindset, adopting new approach in totality instead of bits and pieces. In short, DevOps requires a cultural change as is reflected in its founding stone i.e. CALMS principle as depicted below.



In this article, we will delve with into following aspects:

1. Vocabulary used by Microservice developer and Release engineer. They intend same but they focus at very different degree on different aspects of software engineering but with common goal of delivering beautiful things for customers.
2. How does Mircroservice constituent aligns naturally with automated part of DevOps?
3. Deeper view on CI/CD/CD workflow.

Let us start!

As we know, Microservice is a fine grained atomically deployable service accessed by platform agnostic network API. It's implicit or explicit focus is on solving one portion of problem. It is designed to follow Liskov Substitution Principle (LSP) religiously with Immutability as basic tenet. Each service's life-cycle is completely independent and is expected scale up and down at its own will. On the other hand, DevOps is one of the emerging paradigm for new way of looking at software delivery model affecting every stake holders whether it be developer, operations or customer. The need can be business to get early pulse of customer or respond to customer issue as quickly as possible.

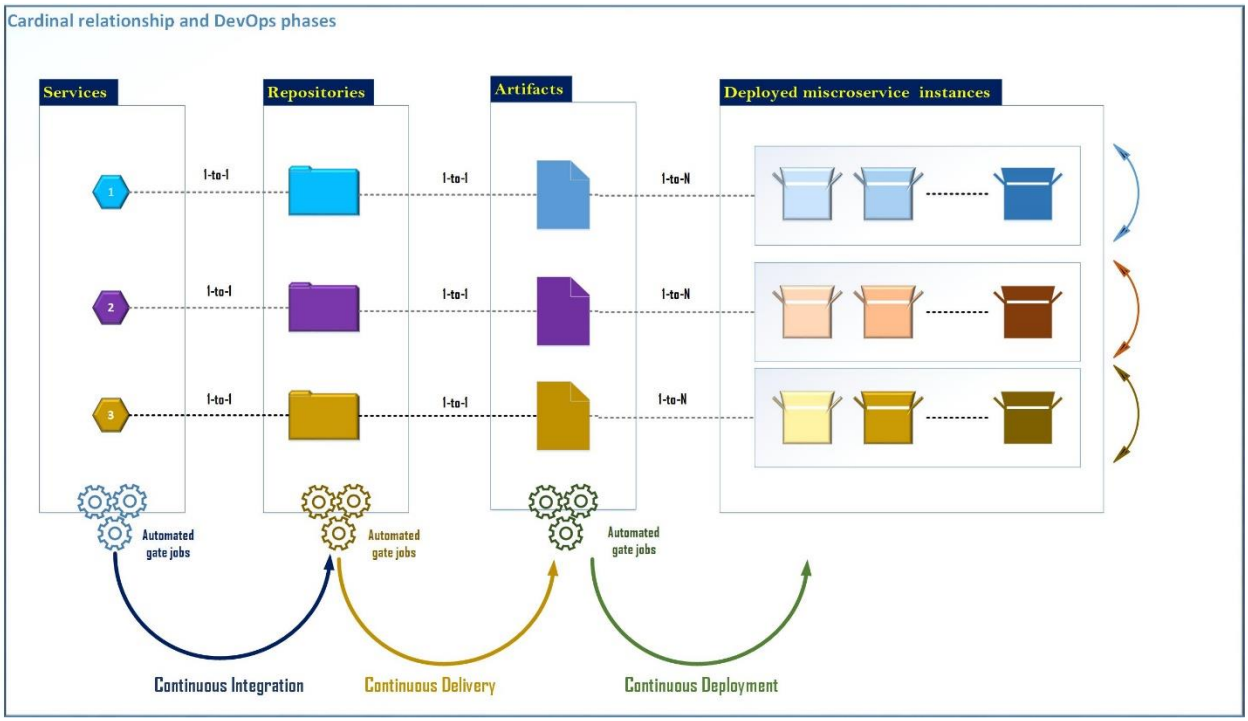
Developer Vocabulary	Release Engineer Vocabulary
Microservice	Artifact repository
Code repository	Continuous Integration
Artifact	Automated artifact generation
Container (service instance)	Continuous Delivery
Automated Unit test	Canary release
Automated functional test	Service lifecycle management
Code analysis tools	Continuous Deployment
	24x7 Service observability
	Automated rollback or rollforward

If you had any affair with code implementing any Microservice, you must have observed the following.

1. Usage of dedicated but only one repository to host source code of service
2. Policy to generate Only one artifact from the repository
3. Deployment of different instances of same service but of different shades by just varying configuration injected dynamically at deployment time. Mechanism can be environment variable, service configuration file, kubernetes configmap or something else.

Is your memory telling that you came across some Microservice which did not follow above guideline? ***You are not alone!*** *Many still do not shy away of packaging existing code in to container and calling as Microservice .*

The below picture depicts what cardinal relationship is among different entities when it comes across realization of DevOps for Microservice.



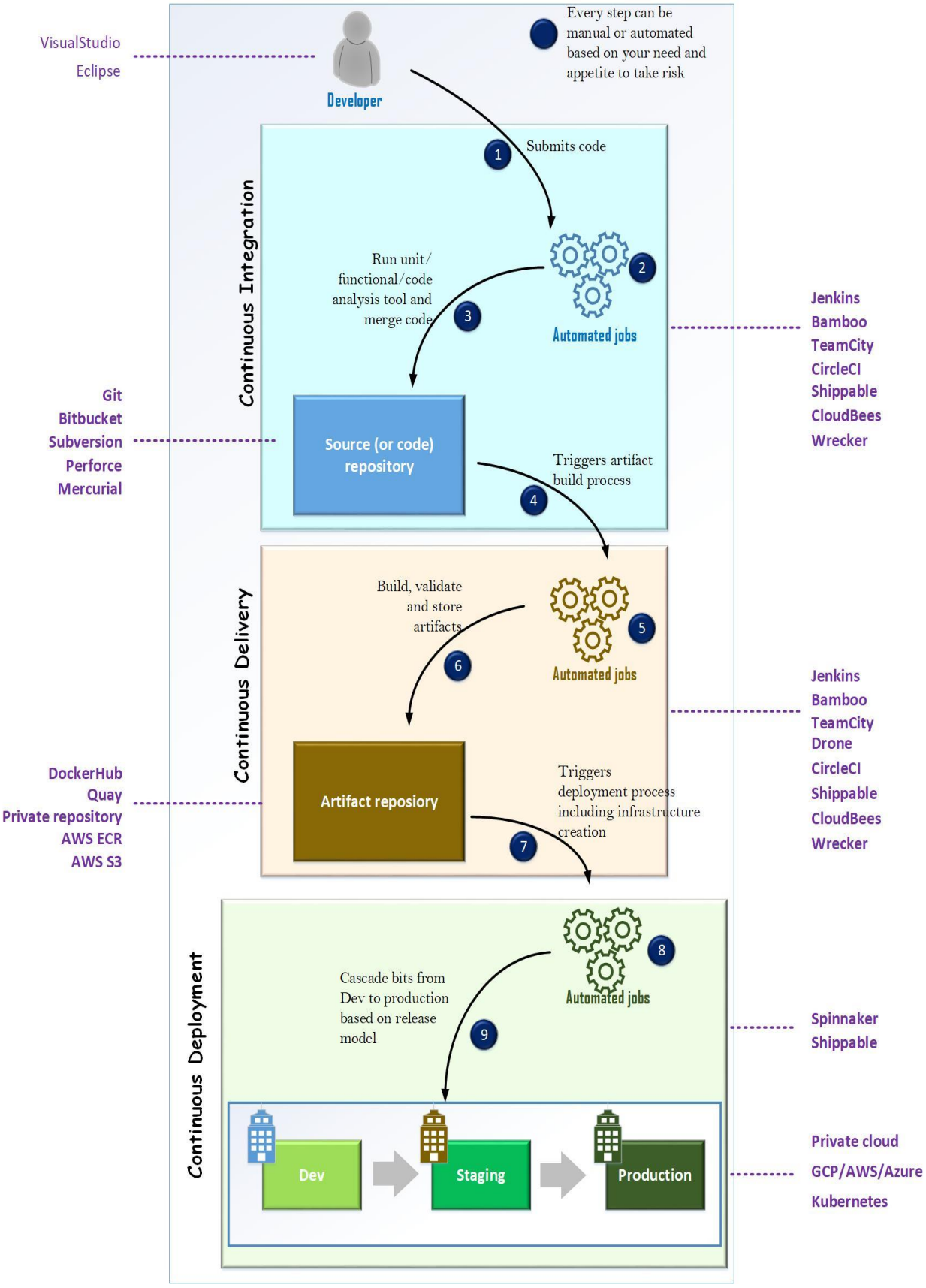
So, if you implement service adhering cardinal relationship then you are ripe for taking benefit of CI/CD/CD workflow depicted below. You just need to implement a specific system functionality within its own bounded context and subject it to below workflow. And you will be ready to get benefit of 1+1=11 not 2 as it will take away all worries like mentioned below of releasing your bits:

1. How do I automated update or upgrade?

2. What will happen if update or upgrade fails? Let tool decide!

3. How frequently should I release?

4. How do I release bits without breaking customer production environment?


VisualStudio
Eclipse
Developer
Every step can be manual or automated based on your need and appetite to take risk
Continuous Integration
1
Submits code
Run unit/ functional/code analysis tool and merge code
3
2
Automated jobs
Jenkins
Bamboo
TeamCity
CircleCI
Shippable
CloudBees
Wrecker
Git
Bitbucket
Subversion
Perforce
Mercurial
Source (or code) repository
Triggers artifact build process
4
Continuous Delivery
Build, validate and store artifacts
6
5
Automated jobs
Jenkins
Bamboo
TeamCity
Drone
CircleCI
Shippable
CloudBees
Wrecker
DockerHub
Quay
Private repository
AWS ECR
AWS S3
Artifact reposiory
Triggers deployment process including infrastructure creation
7
Continuous Deployment
8
Automated jobs
Cascade bits from Dev to production based on release model
9
Spinnaker
Shippable
Dev
Staging
Production
Private cloud
GCP/AWS/Azure
Kubernetes

As of now everyone is doing automation or at least thinking to do. So, the question to ask is where your company stands in the journey. **The company like HPE are very ahead of the curve. They have very automated process to realize every phase of CI/CD/CD, which empowers them to release bits more than once every day.** Some have adopted CI/CD part. Some have adopted CI part. But, it is very evident that everyone is moving with their own agility in a direction which is going to be future of software development, operability and release process.

Are you ready to do shopping? Are you ready to establish start of art CI/CD/CD pipeline for your organization?

I will not be a surprised to see new set of Unicorn companies emerging in next few years who will be able to solve this entire workflow with less friction among Developer, Operation and Customer.

Microservices + Automated CI/CD/CD is like 1 + 1 = 11 not 2.

Are you ready for 1+1 = 11?