Des silos au Platform Engineering en passant par le DevOps

Adopter GitOps et aller au-delà de l'hype

Horacio Gonzalez
Clever Cloud

Sébastien Blanc
Aiven

Who are we?

Sébastien Blanc

DevRel
Aiven
`@sebi2706`

Horacio Gonzalez

DevRel
Clever Cloud
`@LostInBrittany`

What are we going to see?

1. IT in the 90s
2. Tooling evolves
3. XP, agility and DevOps
4. Enter the Cloud
5. Declarative infrastructure
6. Operators to the rescue
7. GitOps?
8. Platform Engineering?
9. Build your own platform
10. Some examples
11. What about not using K8s?

IT in the 90s

Once upon a time…

In a time almost forgotten

When even internet was young...

When Windows 95 was the cutting edge

And a 100 Mb disk was huge…

Big companies still used mainframes

Bigger, fancier, but still the same old IBM

Bare-metal based IT reigned

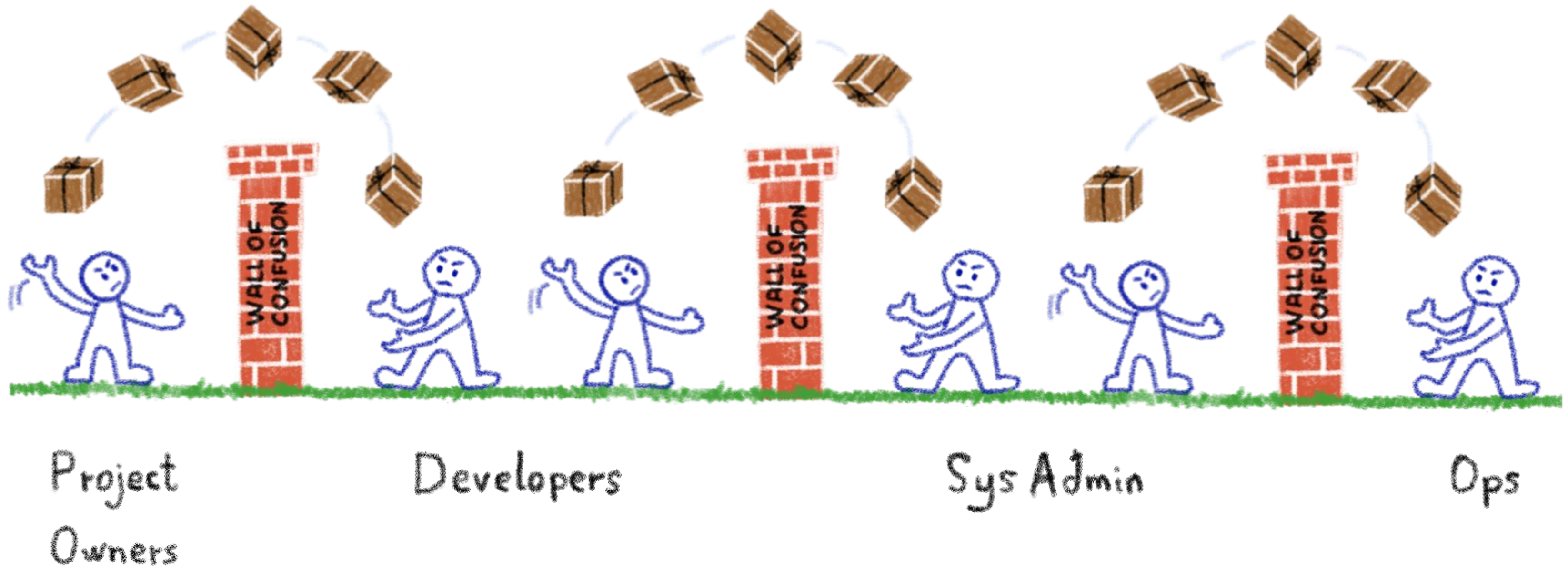
Control, reliability, security…

But cost, rigidity, logistics...

Applying the industrial model

Trying to shoehorn IT into a model where it doesn't fit

Walls & Silos

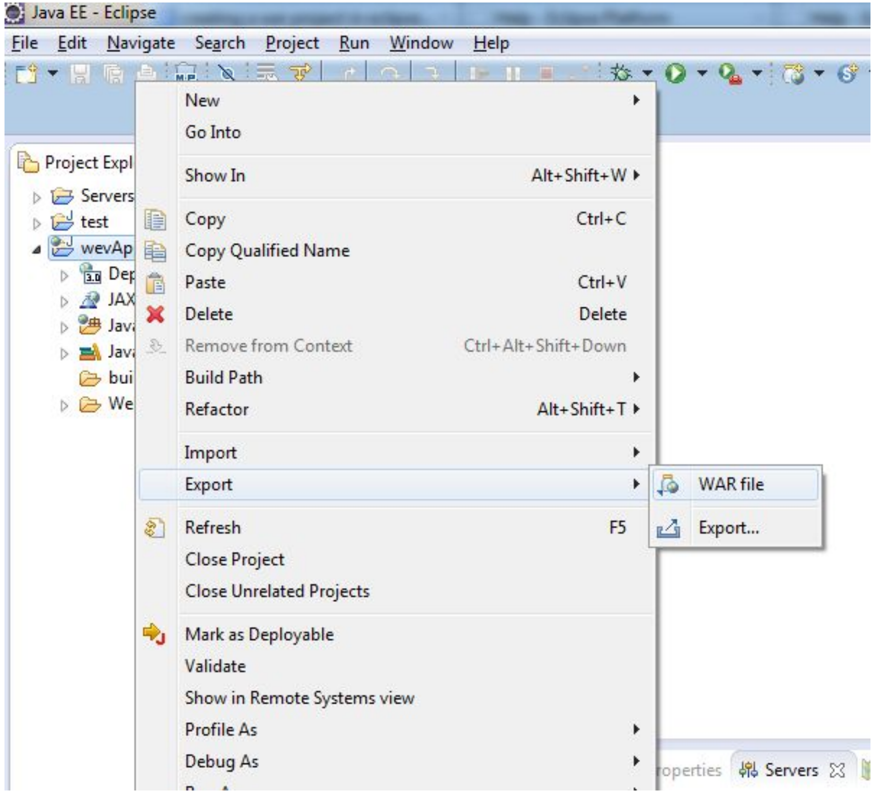


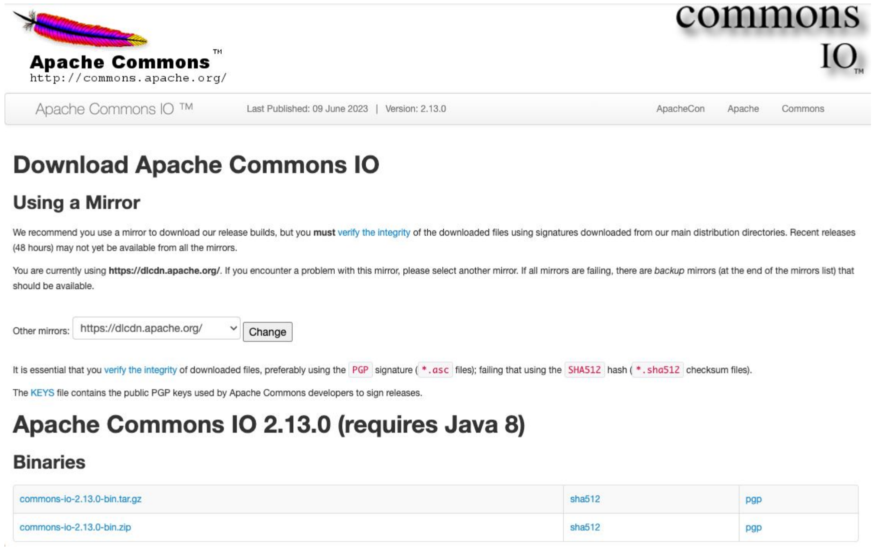
And procedures, and hierarchy, and corporate politics

Tooling evolves

CVS, Ant and (Leeeroy) Jenkins

Old school procedures





Apache Commons™
http://commons.apache.org/

commons IO™

Apache Commons IO ™ Last Published: 09 June 2023 | Version: 2.13.0 ApacheCon Apache Commons

Download Apache Commons IO

Using a Mirror

We recommend you use a mirror to download our release builds, but you **must** verify the integrity of the downloaded files using signatures downloaded from our main distribution directories. Recent releases (48 hours) may not yet be available from all the mirrors.

You are currently using **https://dlcdn.apache.org/**. If you encounter a problem with this mirror, please select another mirror. If all mirrors are failing, there are *backup* mirrors (at the end of the mirrors list) that should be available.

Other mirrors: https://dlcdn.apache.org/ Change

It is essential that you verify the integrity of downloaded files, preferably using the `PGP` signature (`*.asc` files); failing that using the `SHA512` hash (`*.sha512` checksum files).

The KEYS file contains the public PGP keys used by Apache Commons developers to sign releases.

Apache Commons IO 2.13.0 (requires Java 8)

Binaries

commons-io-2.13.0-bin.tar.gz	sha512	pgp
commons-io-2.13.0-bin.zip	sha512	pgp

Tooling empowering changes

Theory existed since 1999
But without the right tooling…

Source control tools

ClearCase

CVS

SUBVERSION

Better than copying and renaming folders…

Dependency management & build

<APACHE ANT>

Apache Maven™

ivy

The agile dependency manager

Better than grabbing each dependency in their website and running `javac` by hand…

Unit testing and continuous integration

Hudson CI

continuum™

JUnit

Jenkins

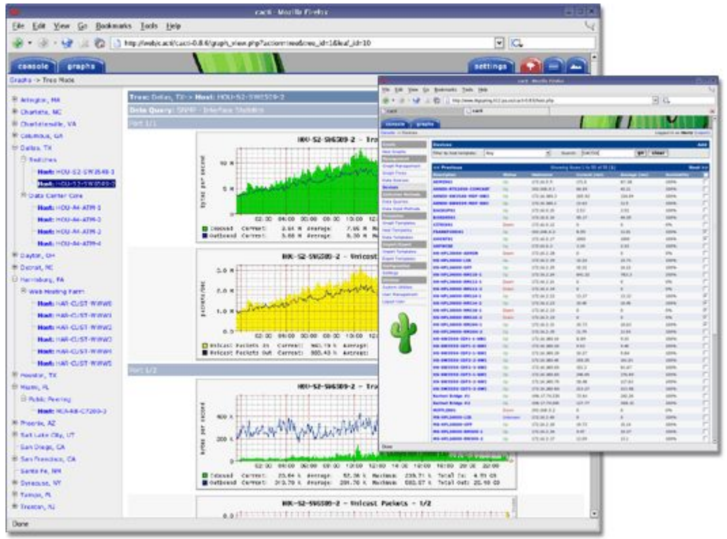
If *Testing is Doubting*, let's doubt automatically

Monitoring tools

cacti

Nagios

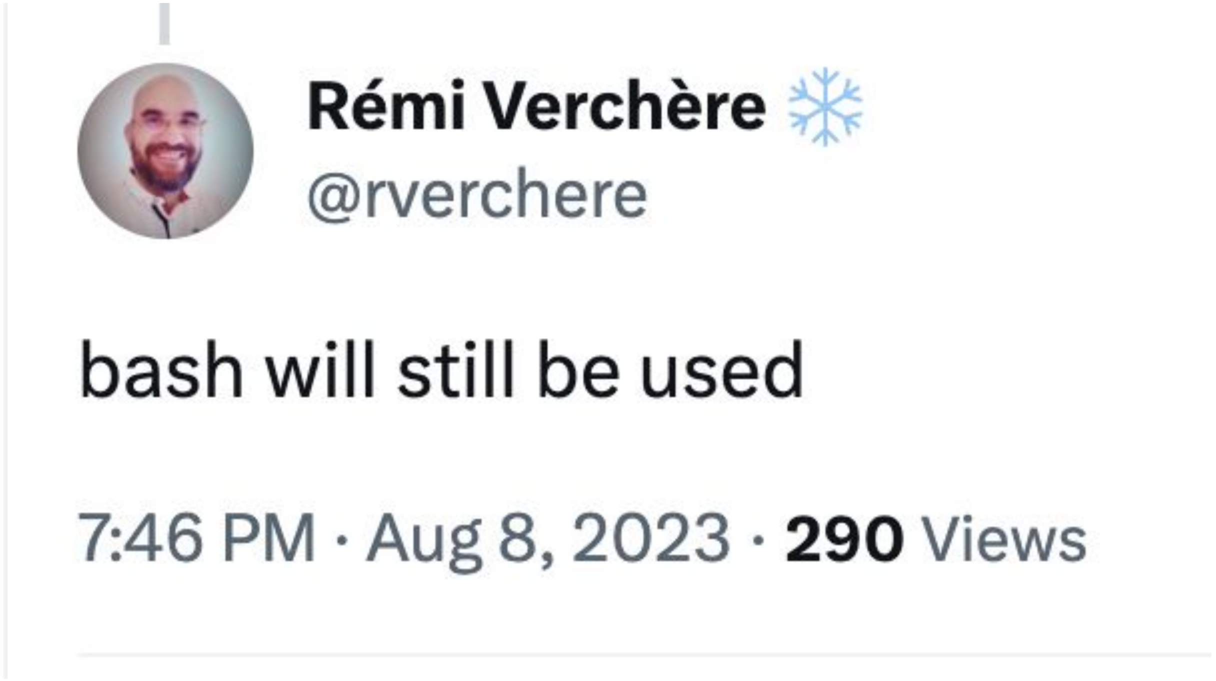
ZABBIX



No more spending nights looking at a status screen

Too many changes in a few years

Old ways were difficult to change



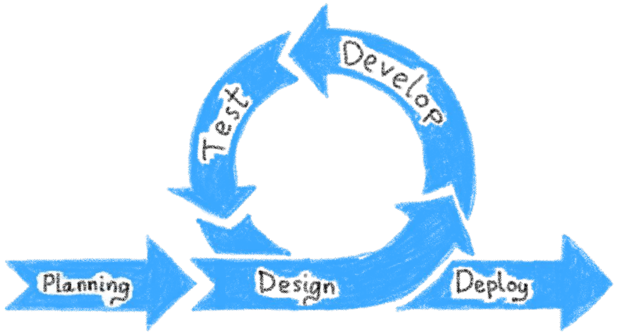
Rémi Verchère ❄️
@rverchere

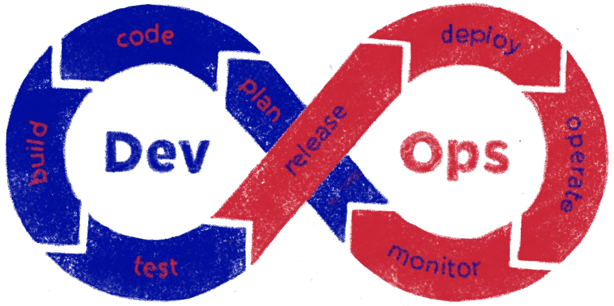
bash will still be used

7:46 PM · Aug 8, 2023 · **290** Views

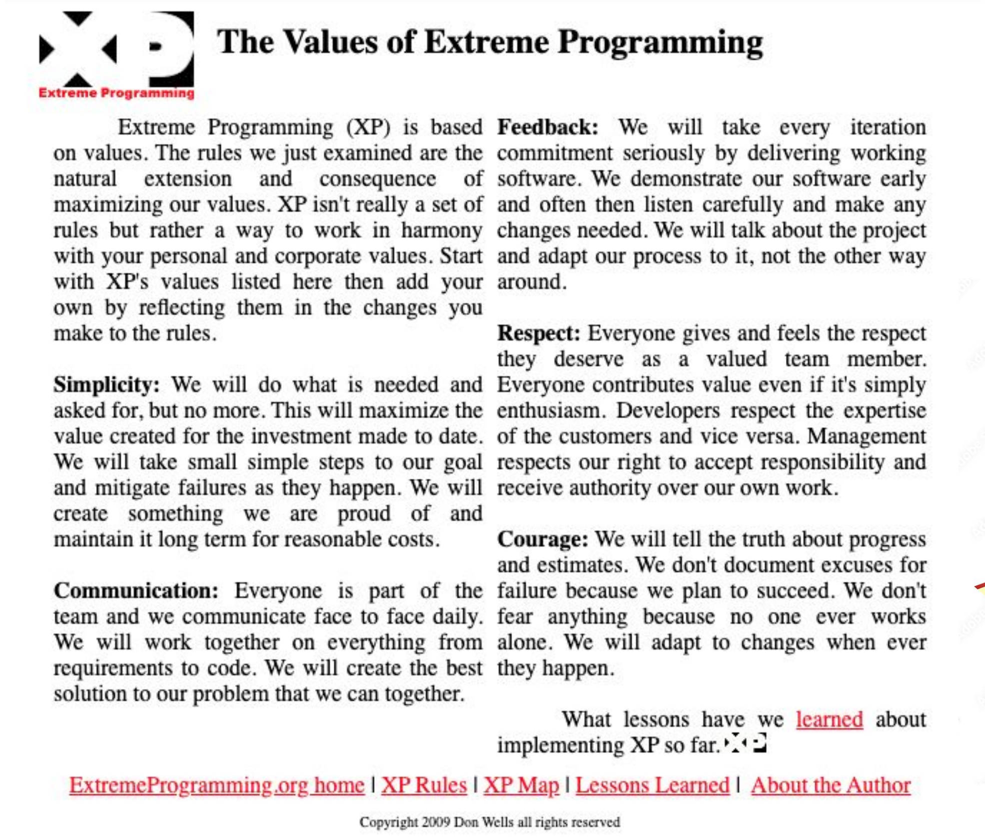
XP, agility and DevOps

Buzzwords that changed the IT





Extreme Programming



Extreme Programming

The Values of Extreme Programming

Extreme Programming (XP) is based on values. The rules we just examined are the natural extension and consequence of maximizing our values. XP isn't really a set of rules but rather a way to work in harmony with your personal and corporate values. Start with XP's values listed here then add your own by reflecting them in the changes you make to the rules.

Simplicity: We will do what is needed and asked for, but no more. This will maximize the value created for the investment made to date. We will take small simple steps to our goal and mitigate failures as they happen. We will create something we are proud of and maintain it long term for reasonable costs.

Communication: Everyone is part of the team and we communicate face to face daily. We will work together on everything from requirements to code. We will create the best solution to our problem that we can together.

Feedback: We will take every iteration commitment seriously by delivering working software. We demonstrate our software early and often then listen carefully and make any changes needed. We will talk about the project and adapt our process to it, not the other way around.

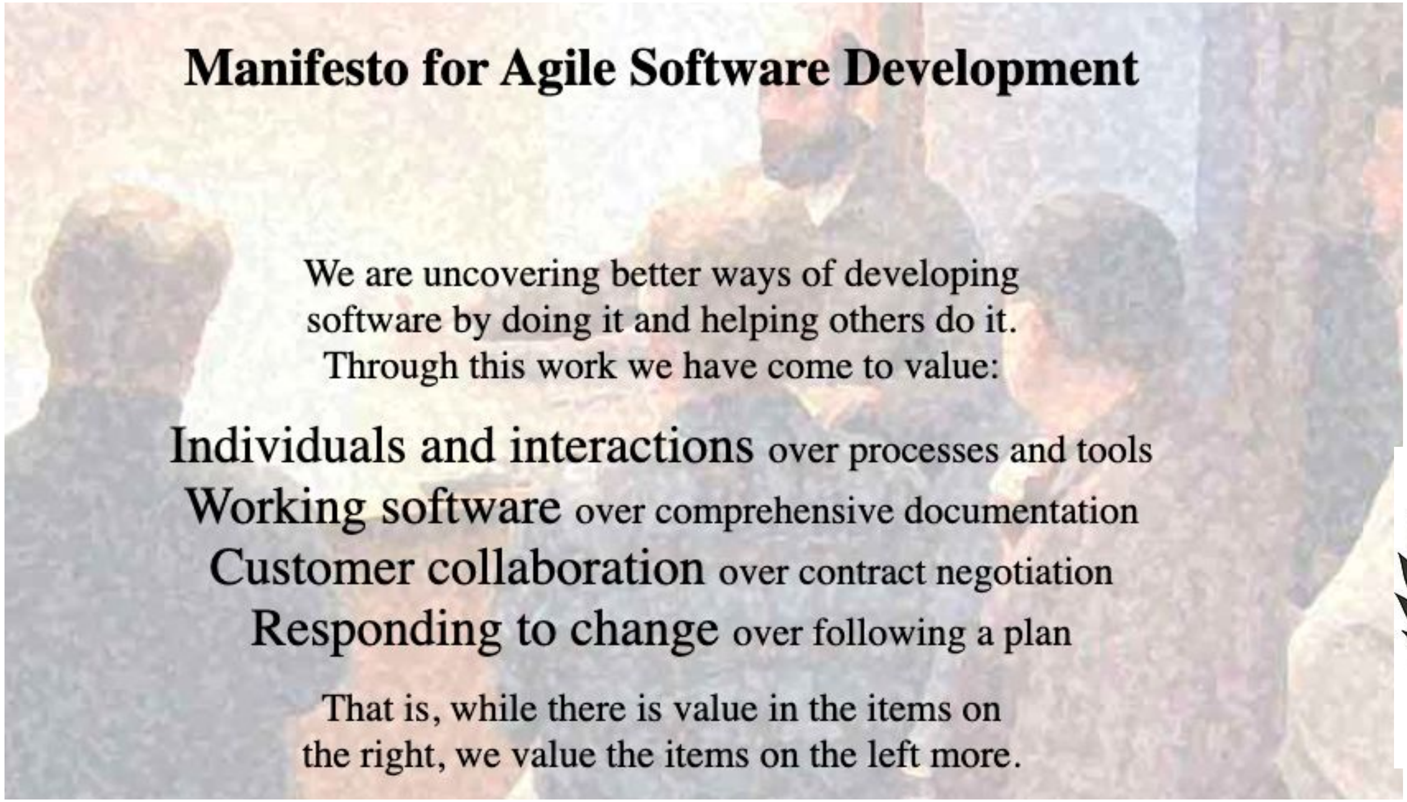
Respect: Everyone gives and feels the respect they deserve as a valued team member. Everyone contributes value even if it's simply enthusiasm. Developers respect the expertise of the customers and vice versa. Management respects our right to accept responsibility and receive authority over our own work.

Courage: We will tell the truth about progress and estimates. We don't document excuses for failure because we plan to succeed. We don't fear anything because no one ever works alone. We will adapt to changes when ever they happen.

What lessons have we learned about implementing XP so far.

ExtremeProgramming.org home | XP Rules | XP Map | Lessons Learned | About the Author

Manifesto for Agile Software Development



Manifesto for Agile Software Development

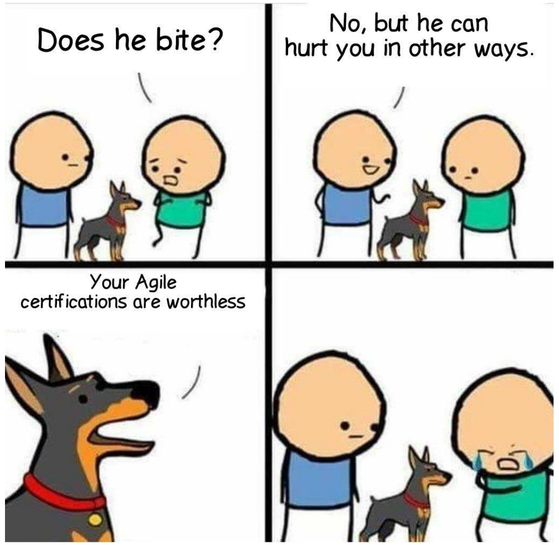
We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:

Individuals and interactions over processes and tools
Working software over comprehensive documentation
Customer collaboration over contract negotiation
Responding to change over following a plan

That is, while there is value in the items on the right, we value the items on the left more.

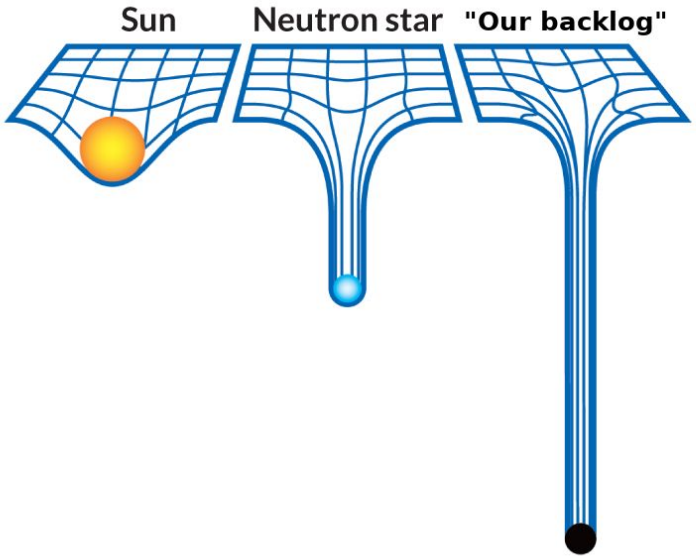
Breaching walks, breaking down silos

The business of Agility

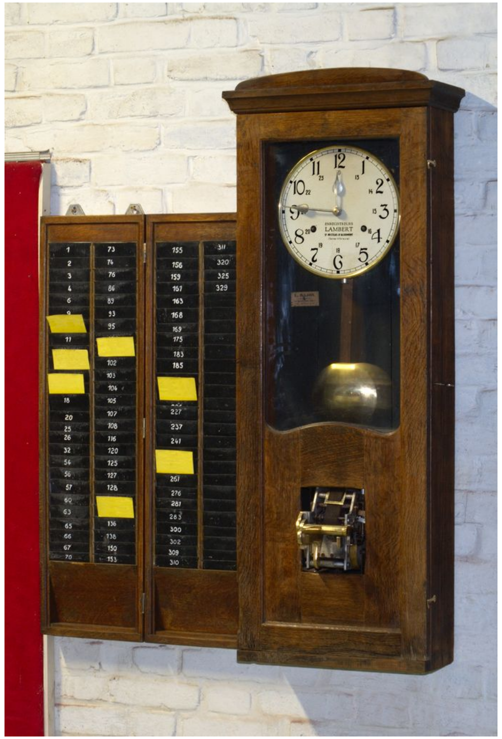
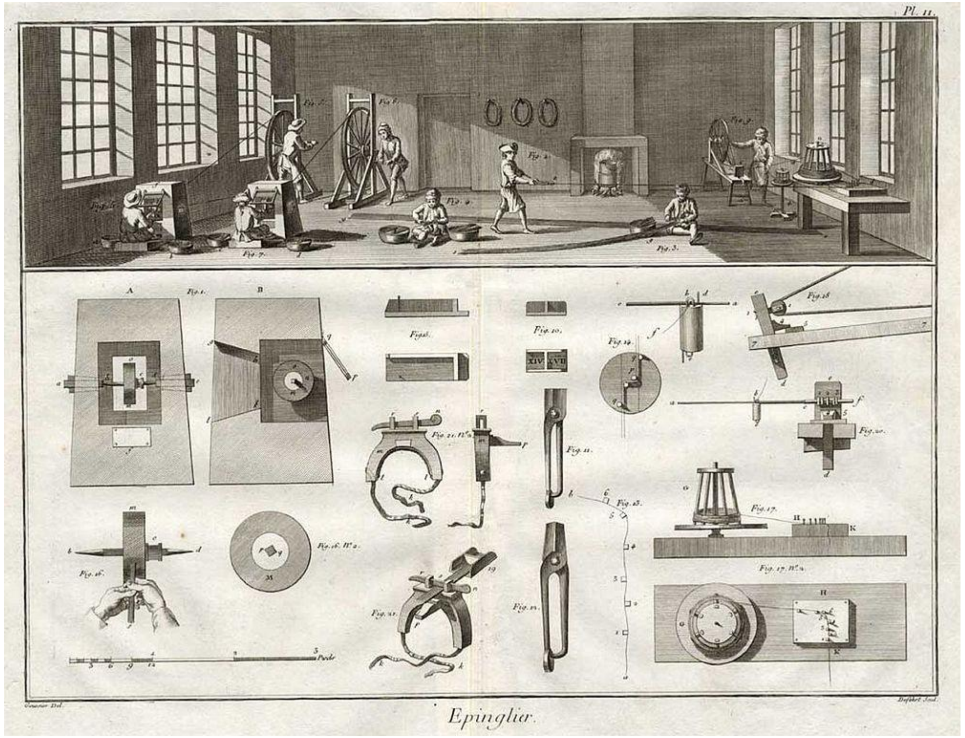

Does he bite?
No, but he can hurt you in other ways.
Your Agile certifications are worthless

The Dark Side rises

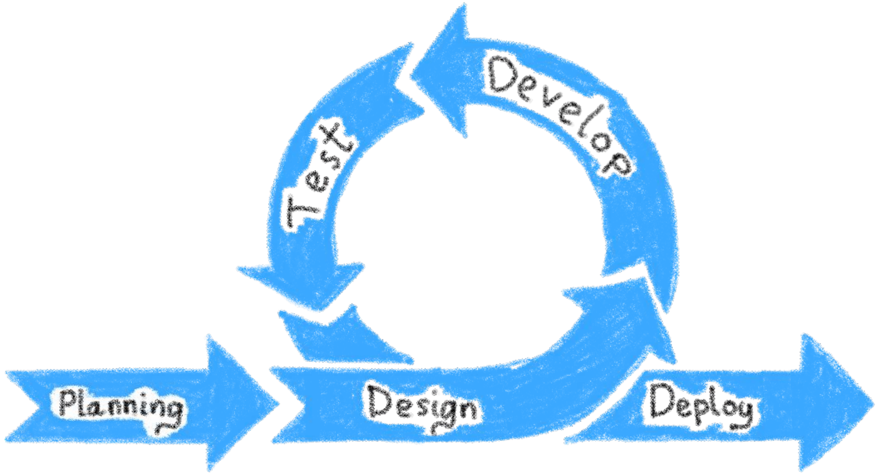
Agile Tooling

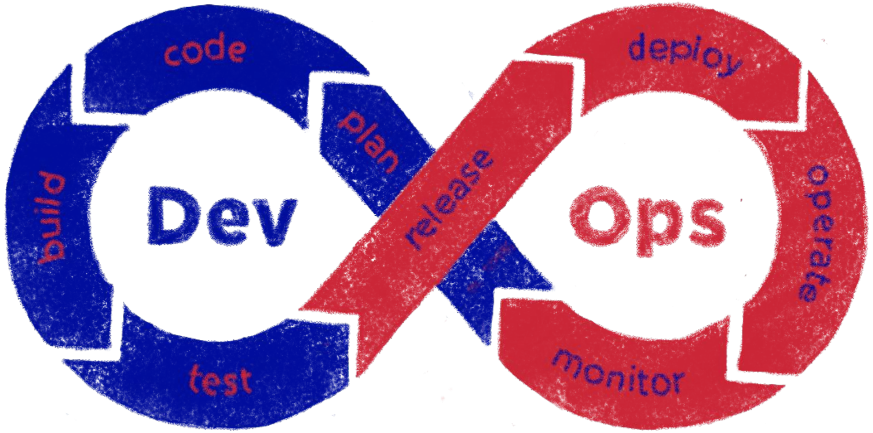


Back to industrial practices ?

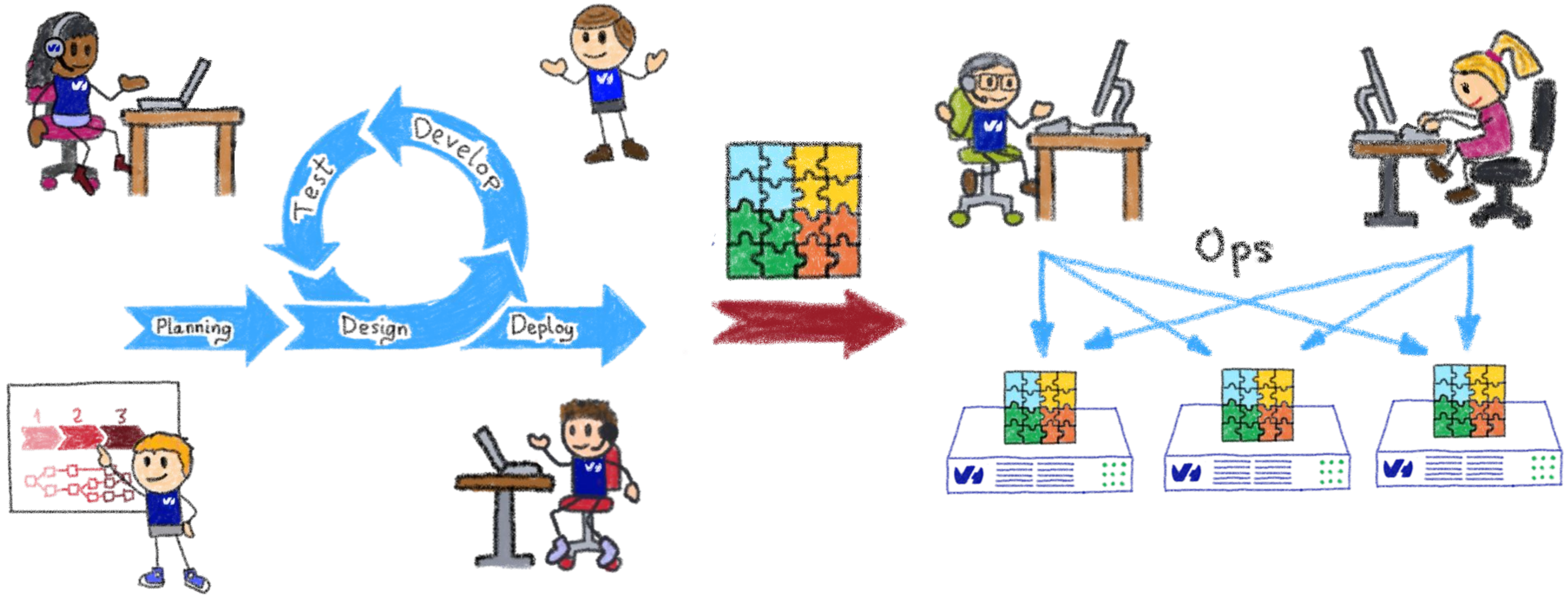
Is DevOps the same than Agility?





Can you have one without the other?

You could have Agility without DevOps



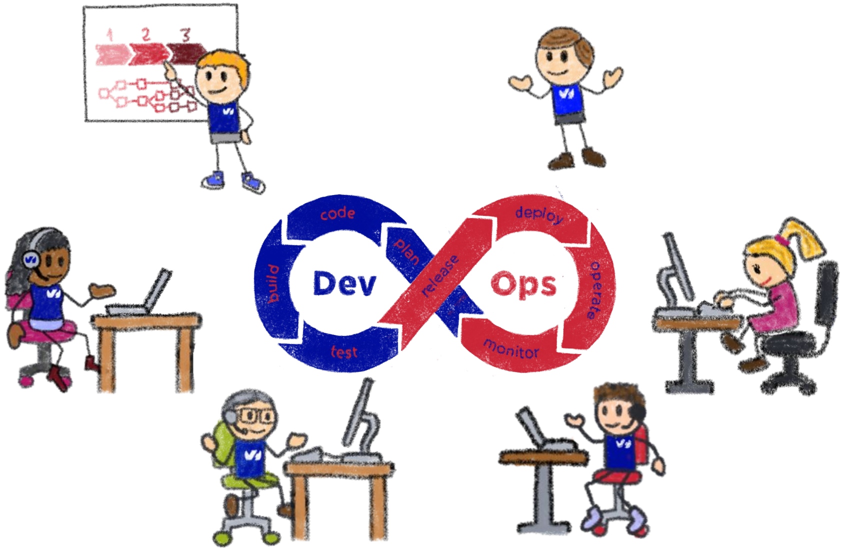
Even if I fail to see why you would want...

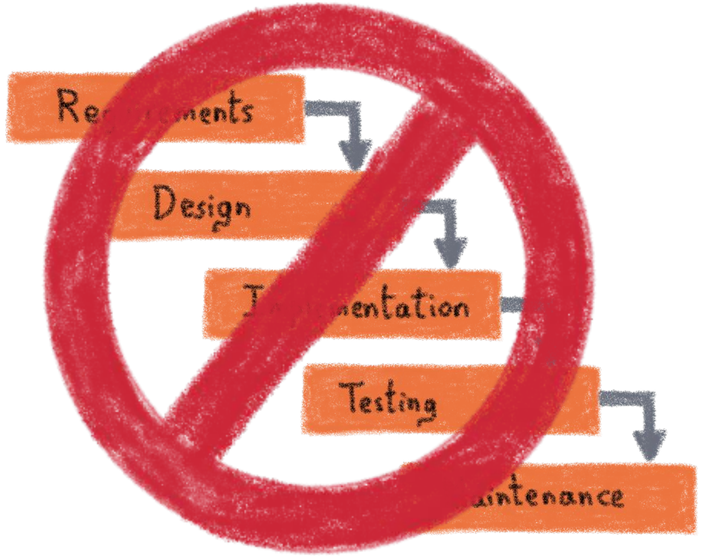
DevOps is a reaction to the wall of confusion

WALL OF CONFUSION
Developers
Operations

Making the different stakeholders
to work together in sync

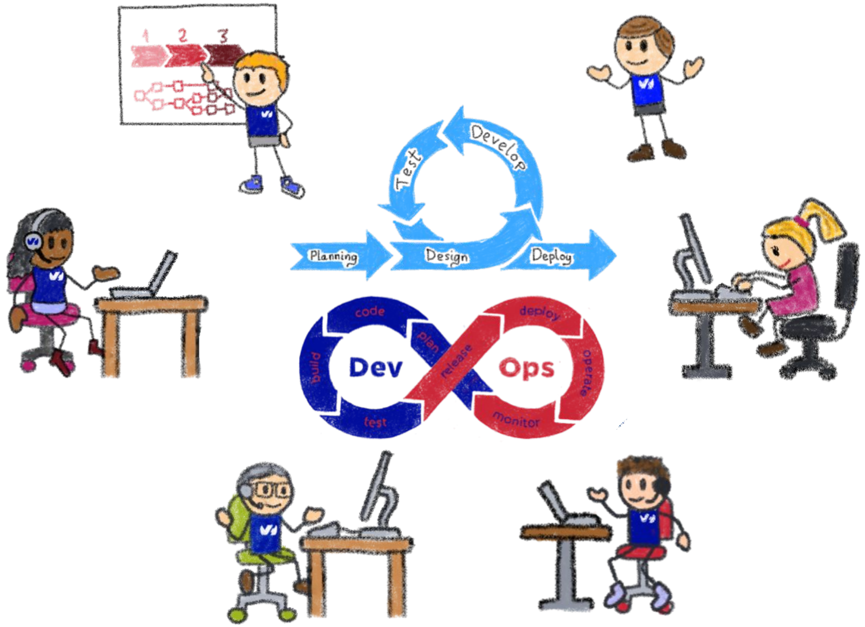
You cannot have DevOps without agility





DevOps is about shorter development sprints, increased focus on testing, increasing automation

DevOps comes with Agility



DevOps is an extension of Agile
that includes systems and operations

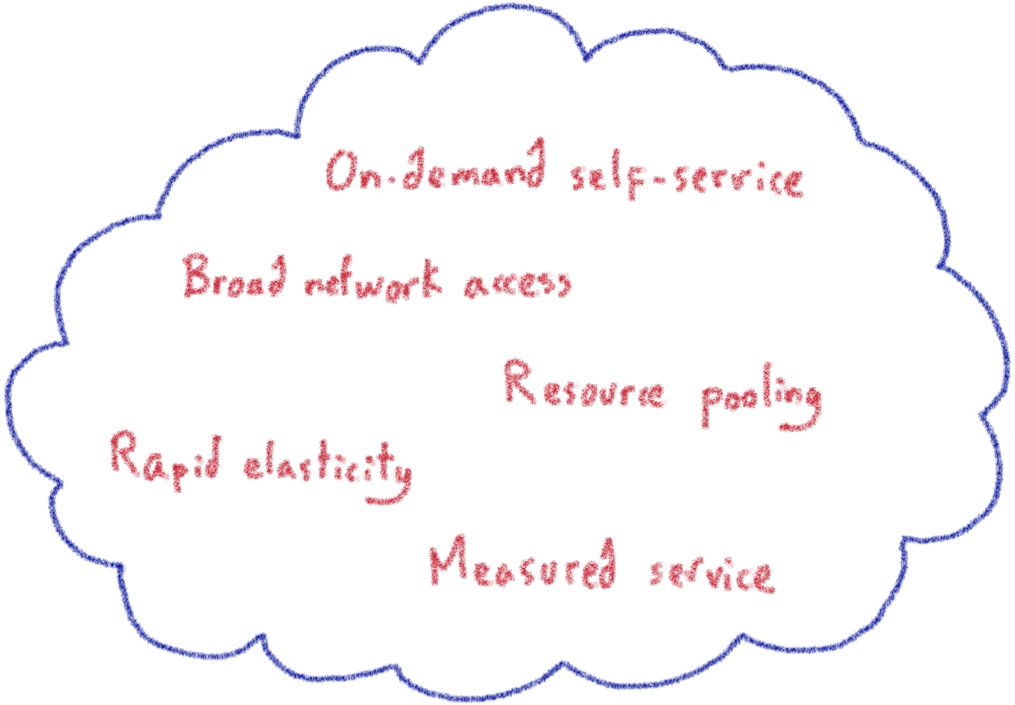
Enter the Cloud

Renting server time in other's people infra

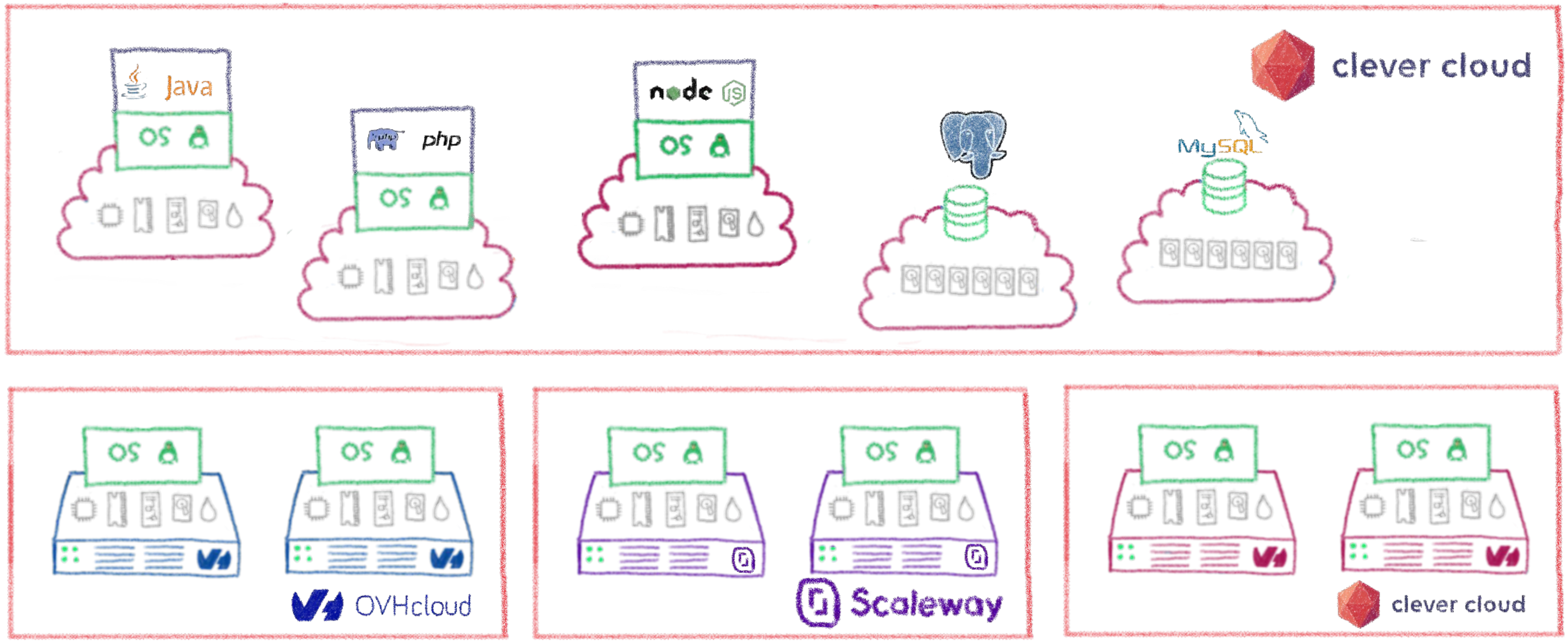
From virtualisation to the cloud

How to use the infrastructure at its full capacity

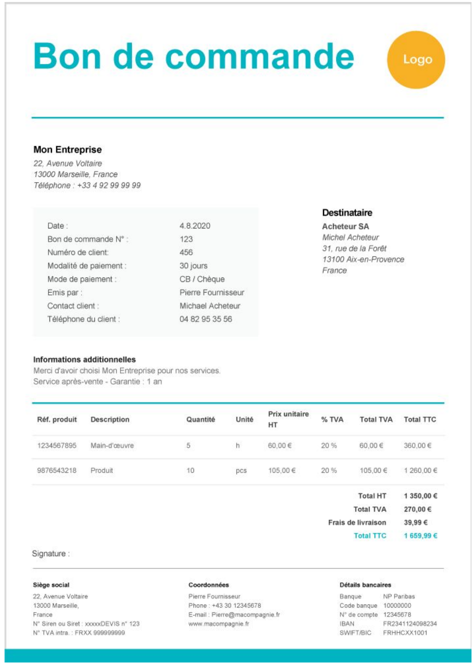
The five pillars of the Cloud



Cloud demands automation



It changes the way how IT works



Bon de commande

Logo

Mon Entreprise
22, Avenue Voltaire
13000 Marseille, France
Téléphone : +33 4 92 99 99 99

Date :	4.8.2020
Bon de commande N° :	123
Numéro de client:	456
Modalité de paiement :	30 jours
Mode de paiement :	CB / Chèque
Emis par :	Pierre Fournisseur
Contact client :	Michael Acheteur
Téléphone du client :	04 82 95 35 56

Destinataire
Acheteur SA
Michel Acheteur
31, rue de la Forêt
13100 Aix-en-Provence
France

Informations additionnelles
Merci d'avoir choisi Mon Entreprise pour nos services.
Service après-vente - Garantie : 1 an

Réf. produit	Description	Quantité	Unité	Prix unitaire HT	% TVA	Total TVA	Total TTC
1234567895	Main-d'œuvre	5	h	60,00 €	20 %	60,00 €	360,00 €
9876543218	Produit	10	pcs	105,00 €	20 %	105,00 €	1 260,00 €
						Total HT	**1 350,00 €**
						Total TVA	**270,00 €**
						Frais de livraison	**39,99 €**
						Total TTC	**1 659,99 €**

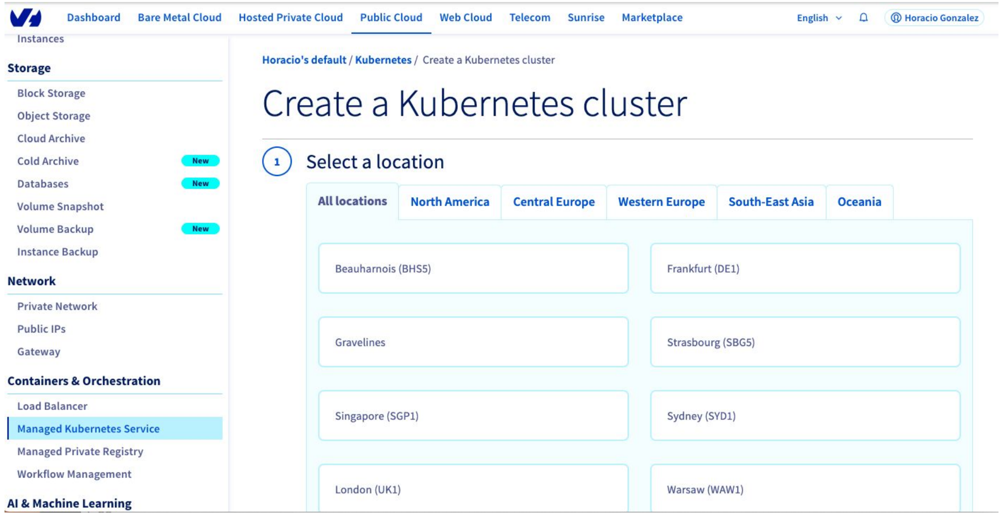
Signature :

Siège social
22, Avenue Voltaire
13000 Marseille,
France
N° Siren ou Siret : xxxxxDEVIS n° 123
N° TVA intra. : FRXX 999999999

Coordonnées
Pierre Fournisseur
Phone : +43 30 12345678
E-mail : Pierre@macompagnie.fr
www.macompagnie.fr

Détails bancaires

Banque	NP Paribas
Code banque	10000000
N° de compte	12345678
IBAN	FR2341124098234
SWIFT/BIC	FRHHCX1001



Dashboard | Bare Metal Cloud | Hosted Private Cloud | Public Cloud | Web Cloud | Telecom | Sunrise | Marketplace | English | Horacio Gonzalez

Instances

Storage
- Block Storage
- Object Storage
- Cloud Archive
- Cold Archive New
- Databases New
- Volume Snapshot
- Volume Backup New
- Instance Backup

Network
- Private Network
- Public IPs
- Gateway

Containers & Orchestration
- Load Balancer
- Managed Kubernetes Service
- Managed Private Registry
- Workflow Management

AI & Machine Learning

Horacio's default / Kubernetes / Create a Kubernetes cluster

Create a Kubernetes cluster

1 Select a location

All locations | North America | Central Europe | Western Europe | South-East Asia | Oceania

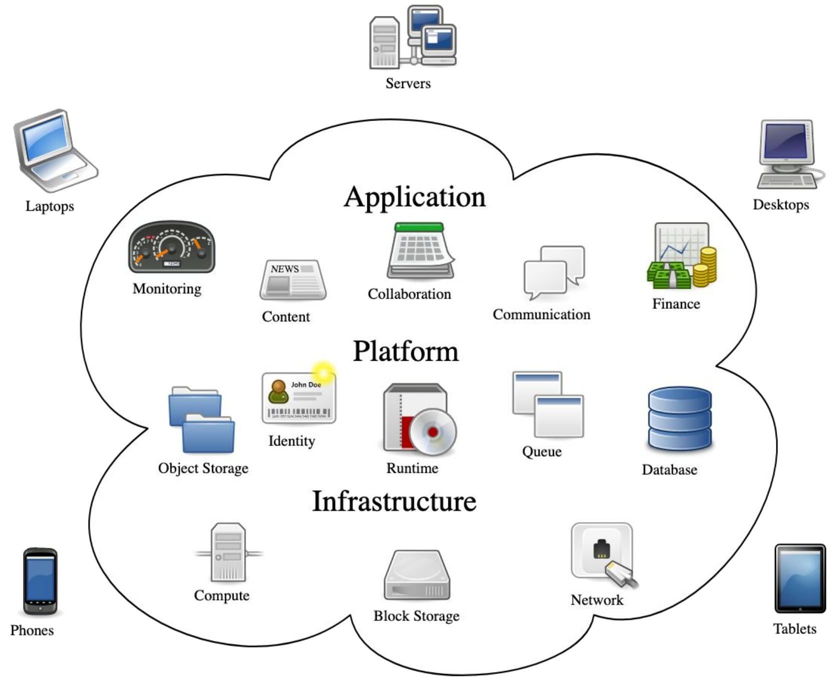
Beauharnois (BHS5)	Frankfurt (DE1)
Gravelines	Strasbourg (SBG5)
Singapore (SGP1)	Sydney (SYD1)
London (UK1)	Warsaw (WAW1)

And it demands a mentality change

Empowering developers

Infrastructure is only a click away

Distributed is the new black



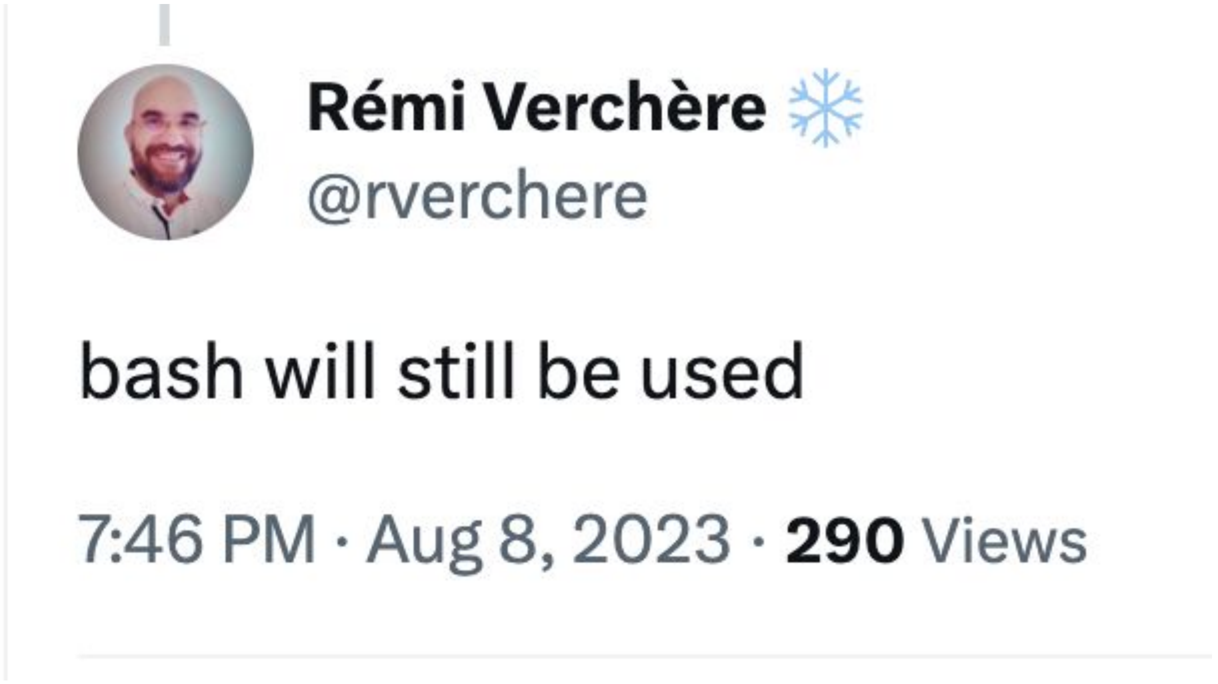
Cloud Native architectures and services

Sysadmins who code

Creating tools: automation, monitoring, observability…

New roles appear: SRE

WTF is a System Reliability Engineer?



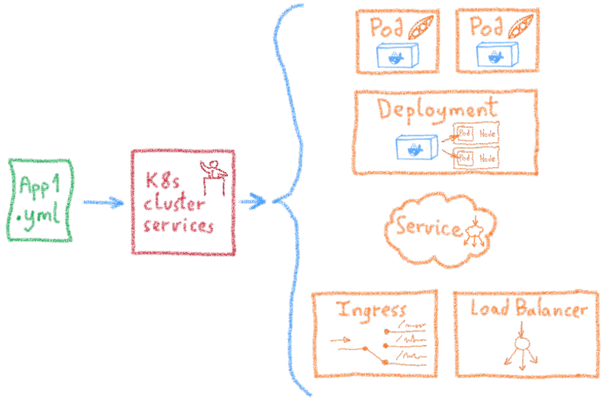
Rémi Verchère ❄️
@rverchere

bash will still be used

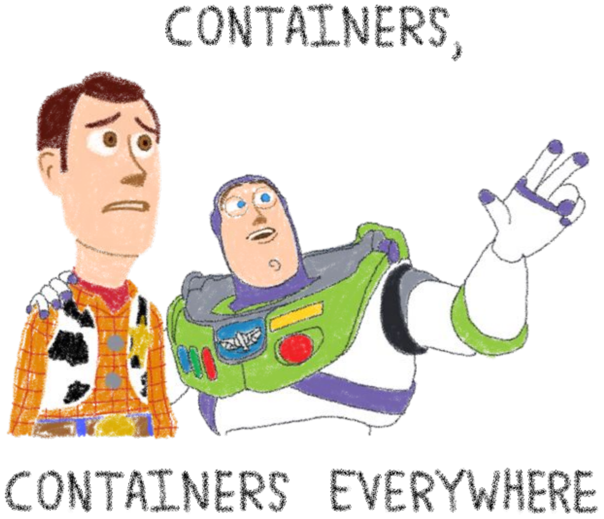
7:46 PM · Aug 8, 2023 · **290** Views

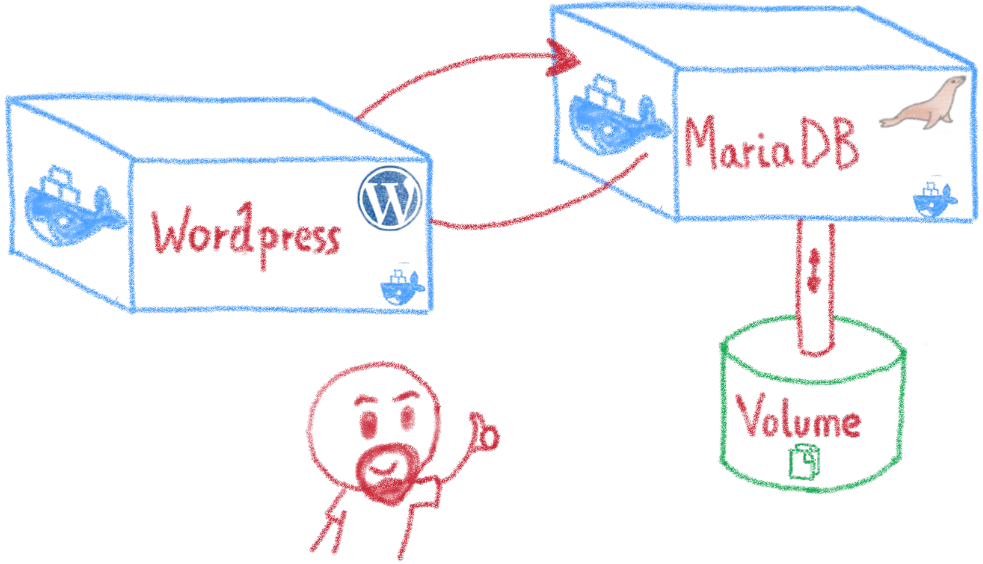
Declarative Infrastructure

The intern metaphor

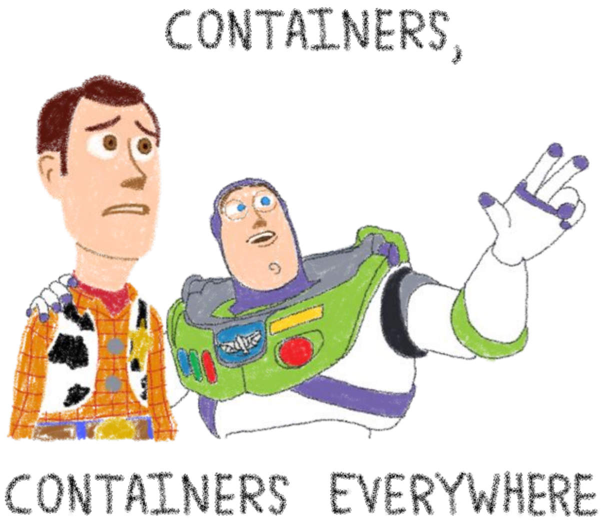


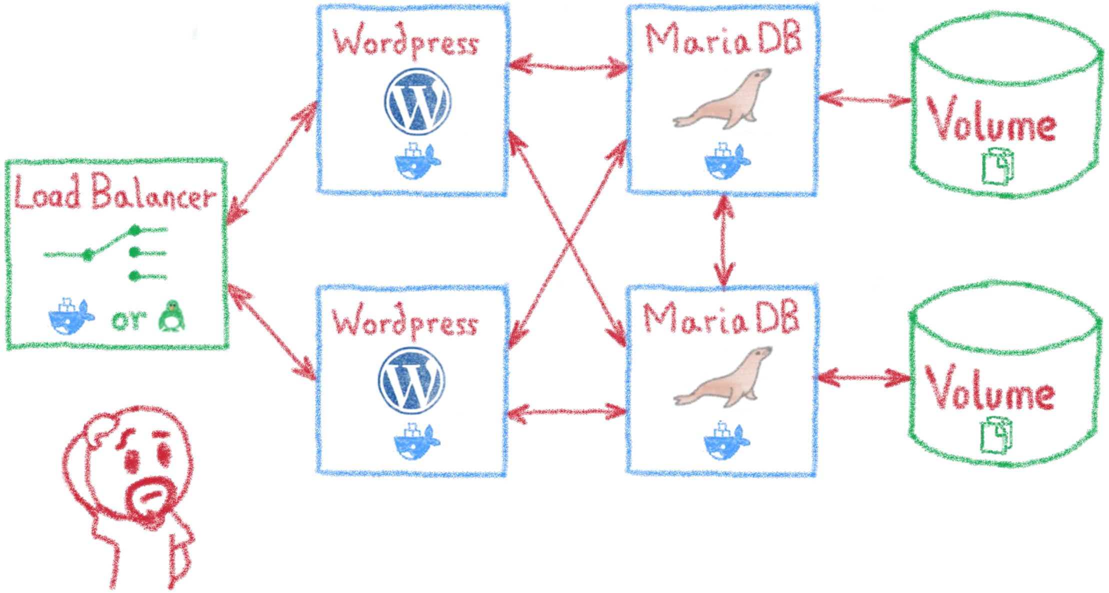
Containers make dev life easier





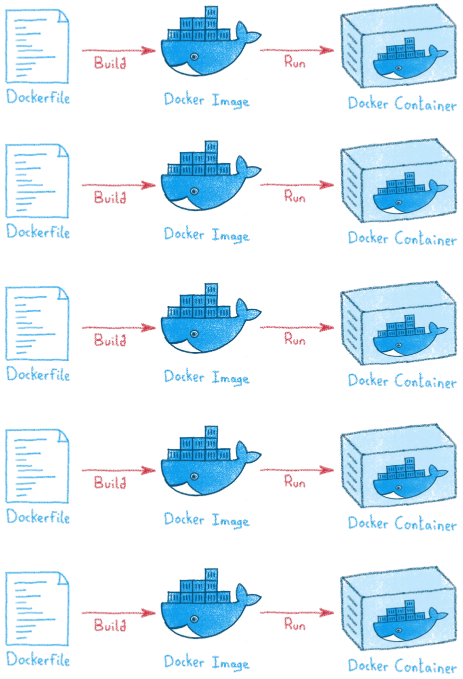
Less simple if you must operate them





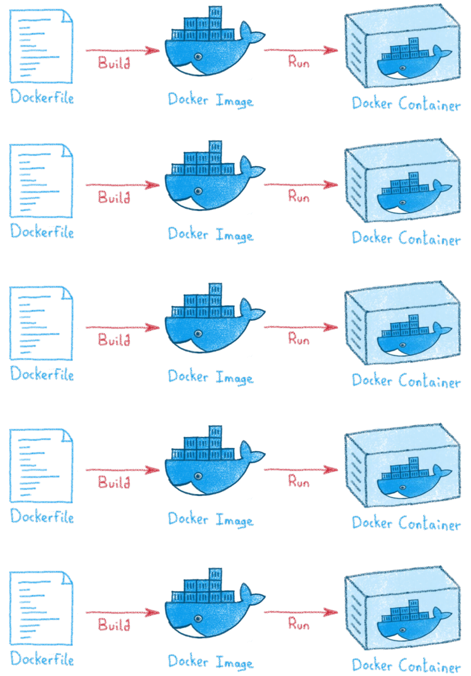
Like in a production context

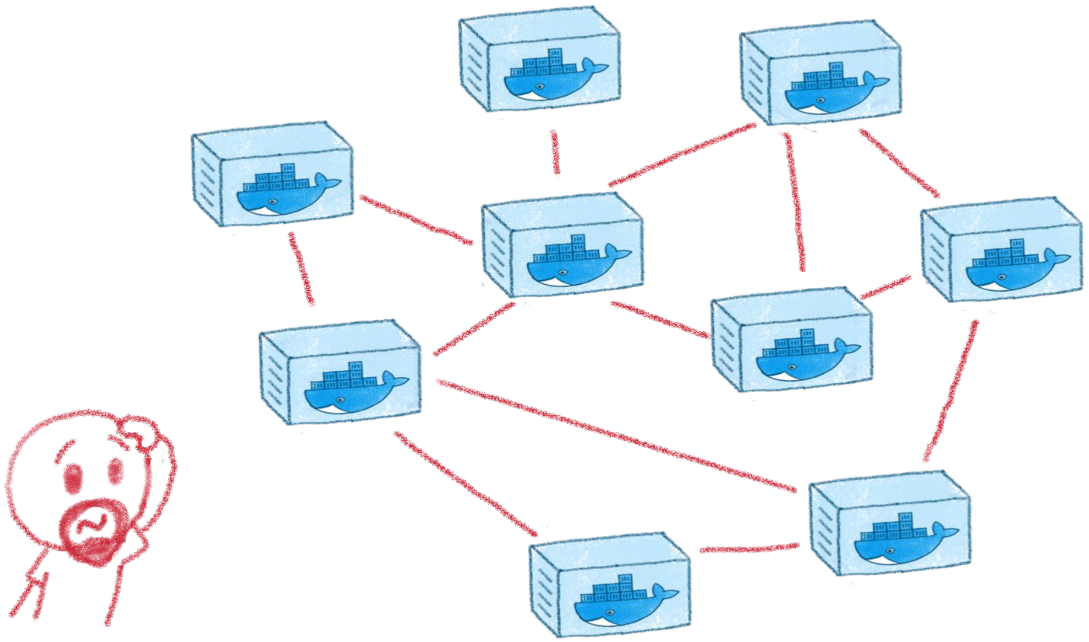
And what about microservices?



Are you sure you want to operate them by hand?

And what about microservices?



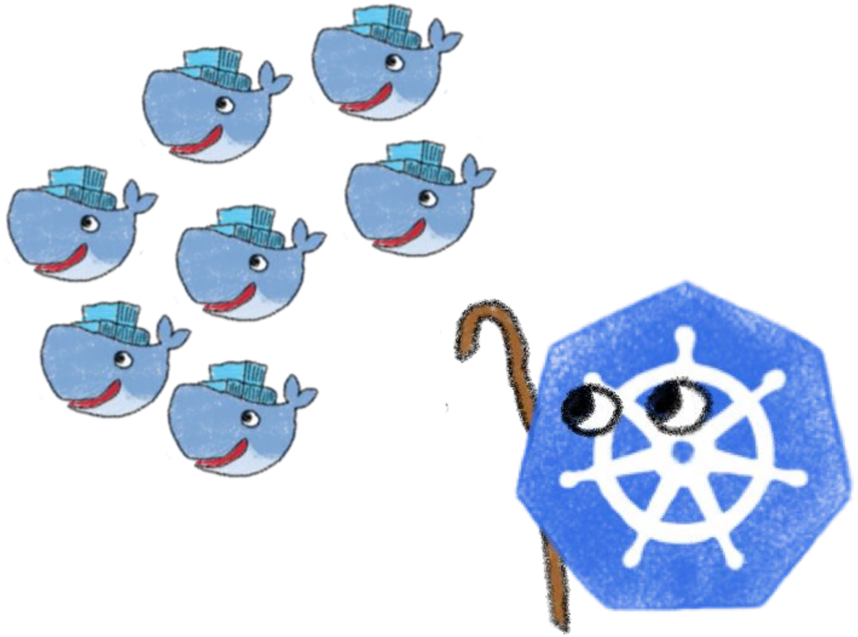


Are you sure you want to operate them by hand?

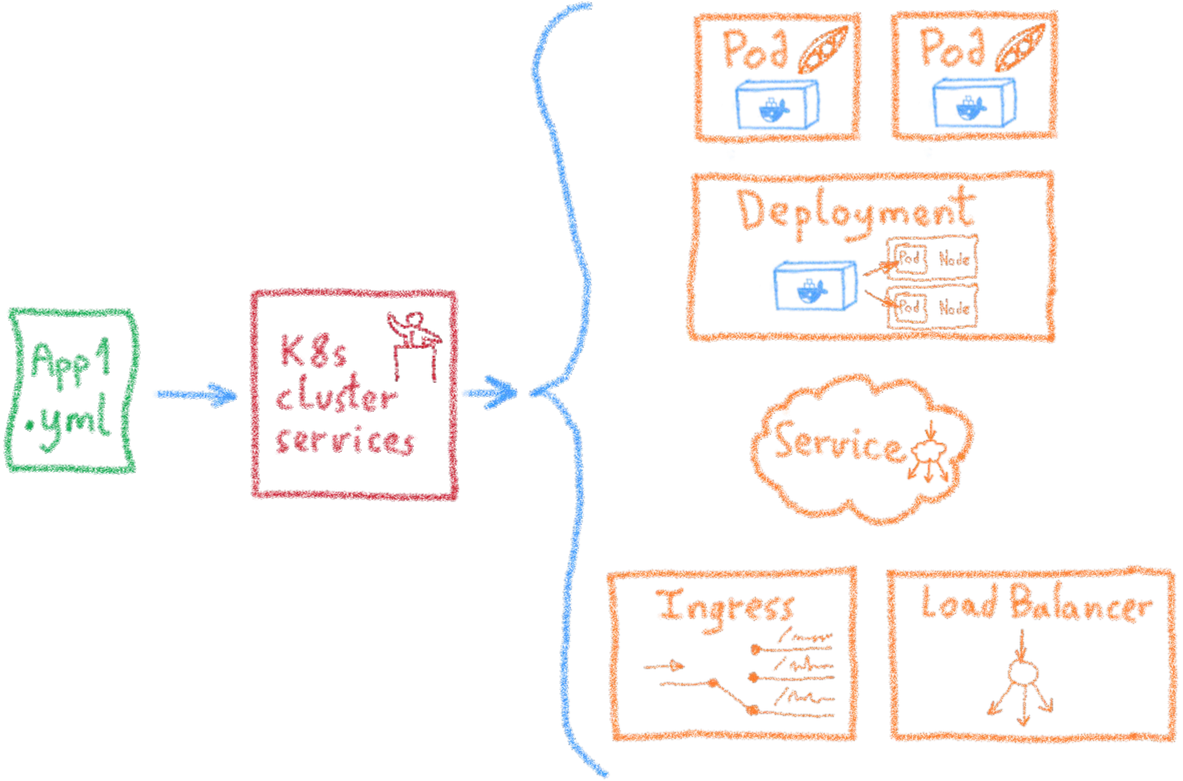
Kubernetes: a full orchestrator

Takes care of:

- Deployment
- Scaling
- Monitoring
- Repairing
- Securing
- ...



Kubernetes - Desired State Management



Ingress

Services

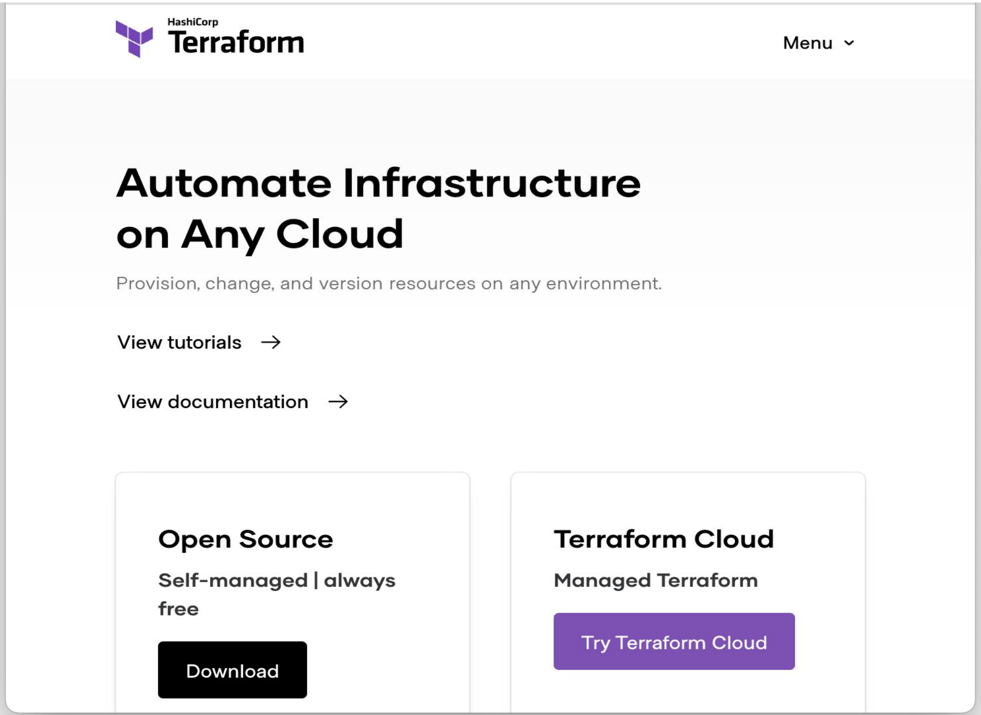
Deployments

Pods

Sidecars

Replica Sets

Terraform - Declarative infra as code



HashiCorp Terraform

Menu

Automate Infrastructure on Any Cloud

Provision, change, and version resources on any environment.

View tutorials →

View documentation →

Open Source

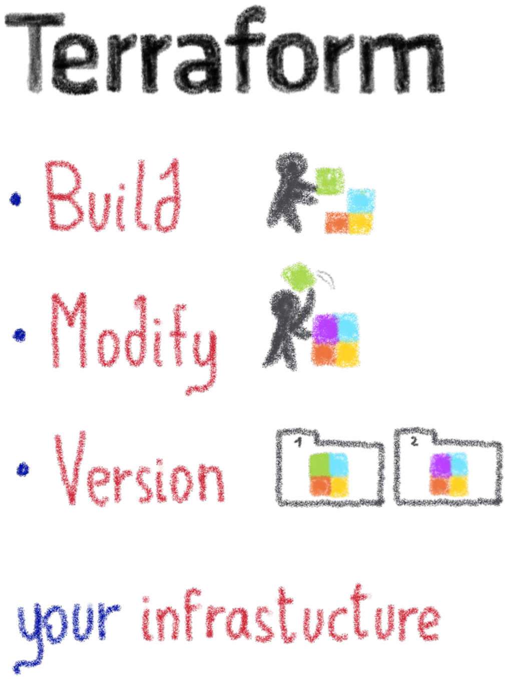
Self-managed | always free

Download

Terraform Cloud

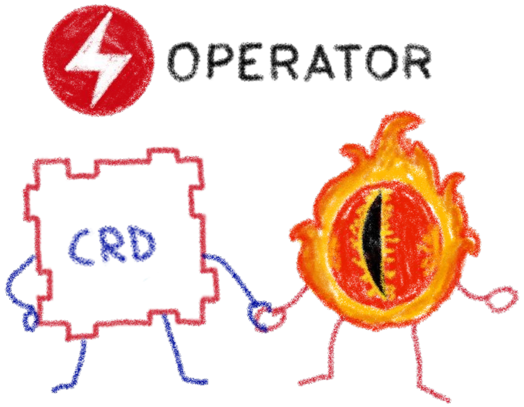
Managed Terraform

Try Terraform Cloud

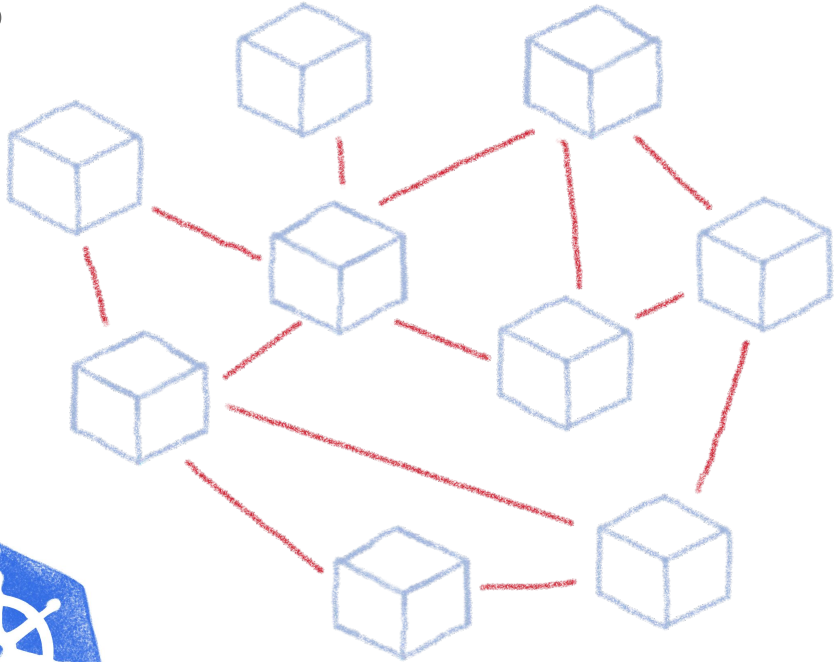


Operators to the rescue

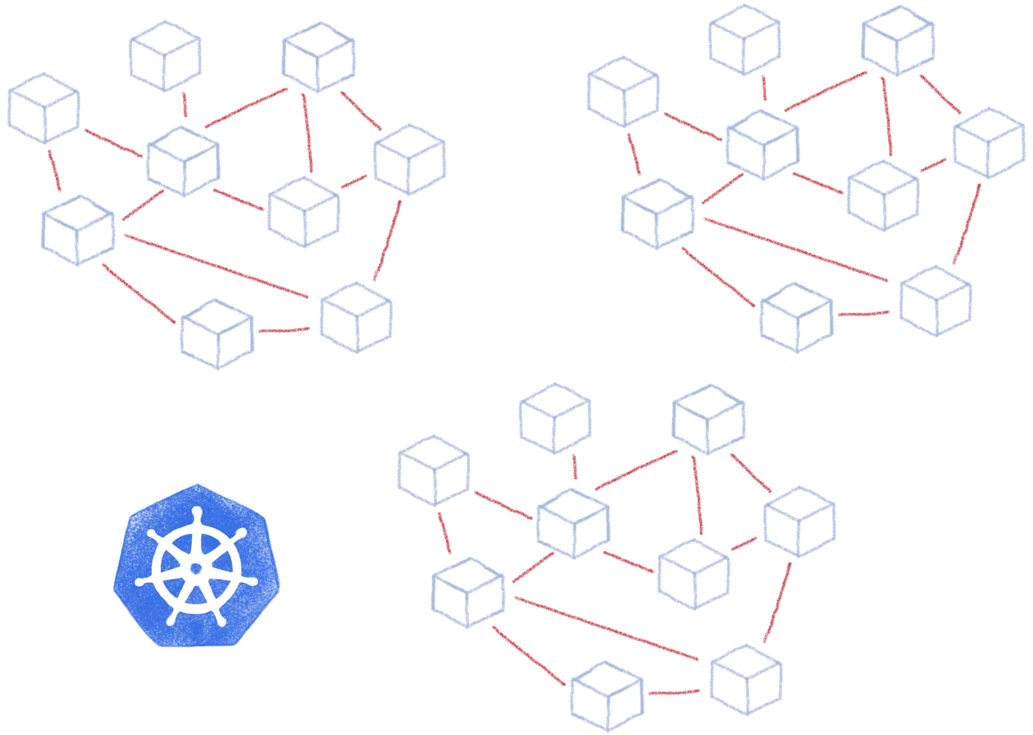
Helping to tame the complexity of K8s and using K8s in heterogeneous systems



Taming microservices with Kubernetes

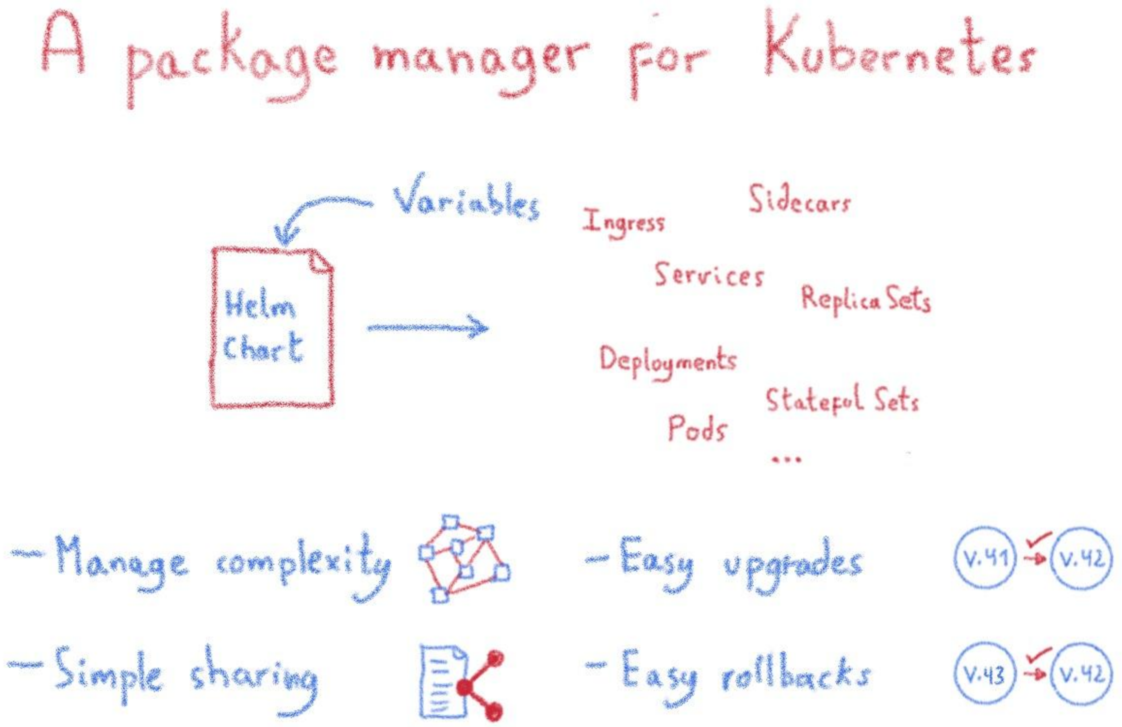


What about complex deployments

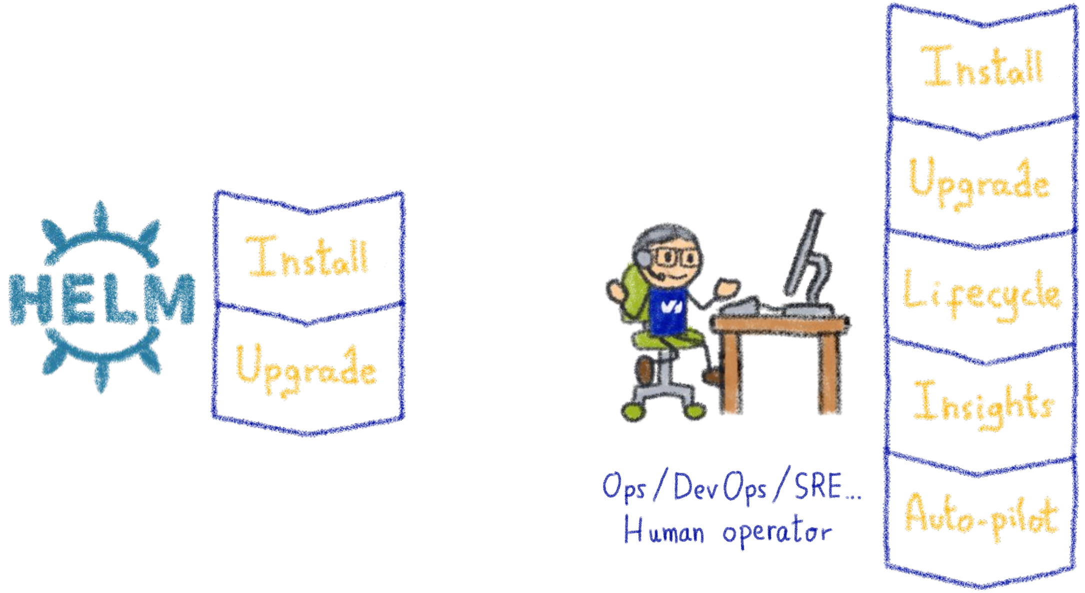


Ingress
Services
Deployments
Pods
Sidecars
ReplicaSets
Stateful Sets

Tools like Helm helps with complexity

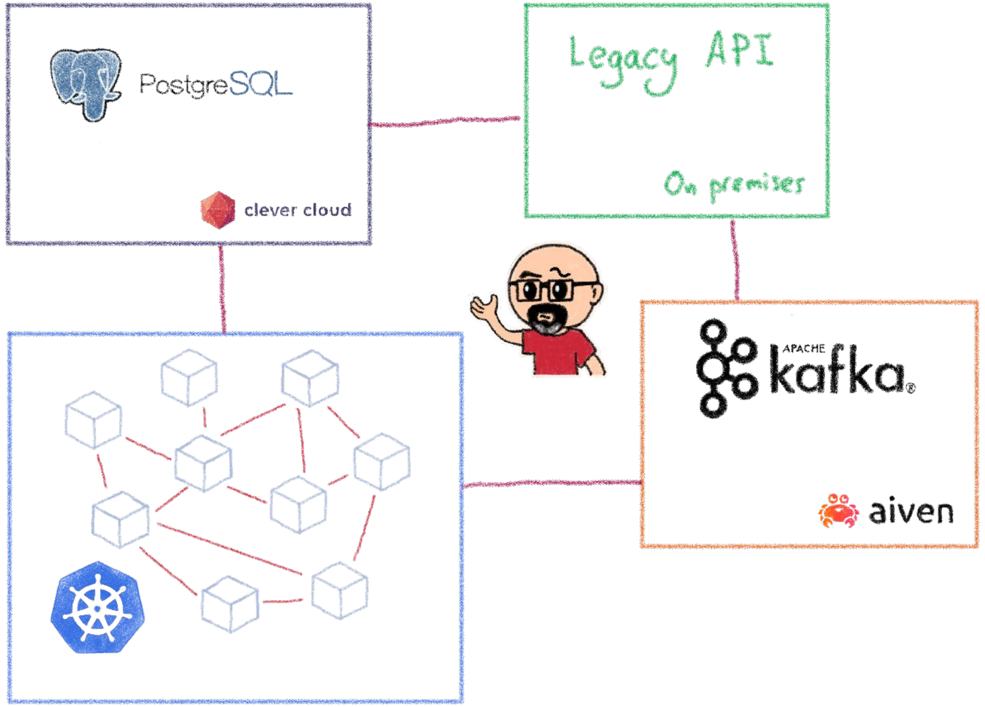


Helm Charts are configuration



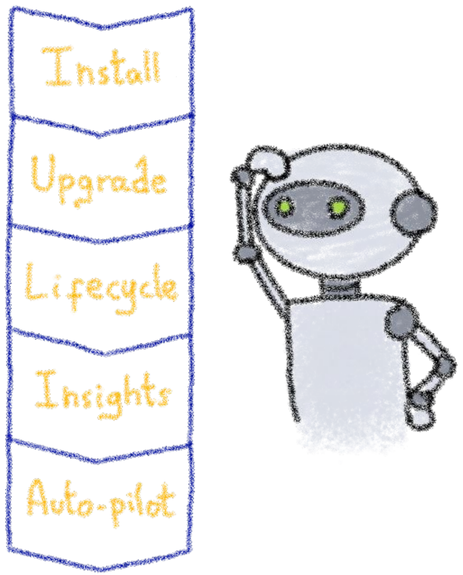
Operating is more than installs & upgrades

What about legacy?



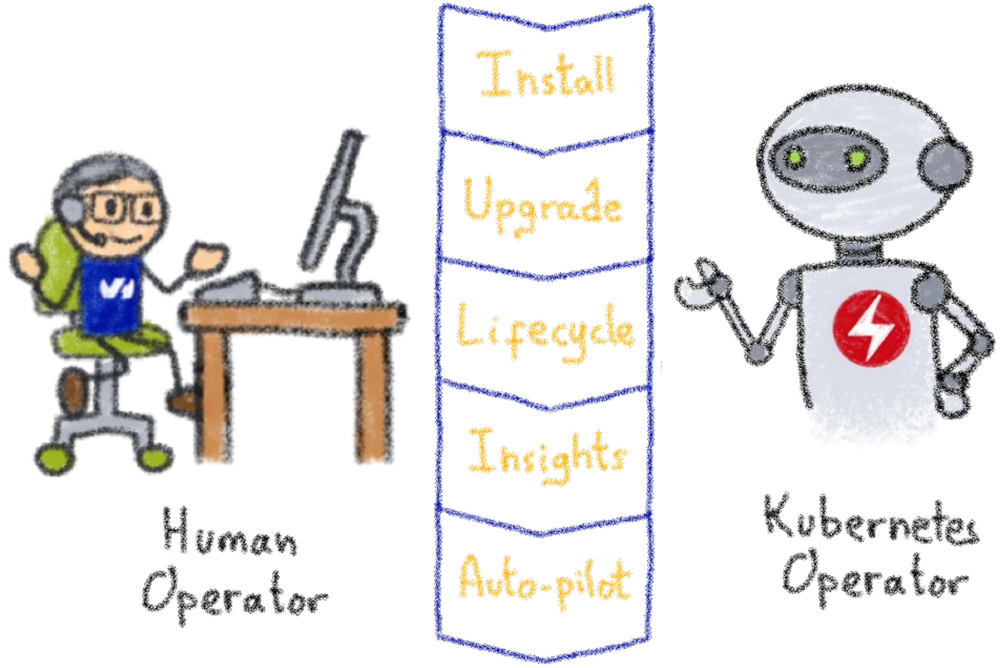
Because not everything needs/wants to be in Kubernetes

Kubernetes is about automation



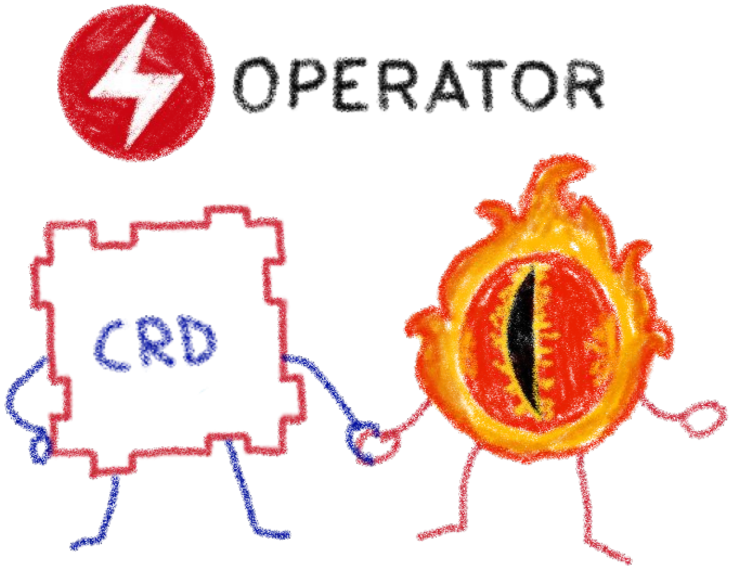
How about automating human operators?

Kubernetes Operators



A Kubernetes version of the human operator

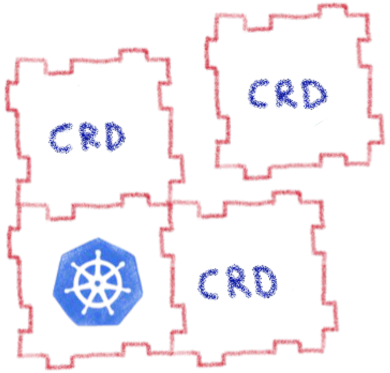
Building operators



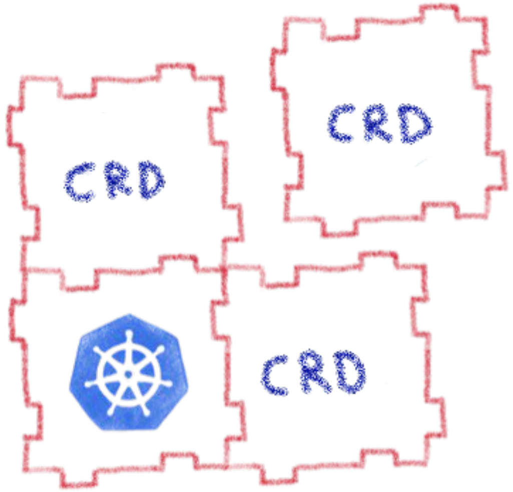
Basic K8s elements: Custom Resources & Controllers

Custom Resource Definitions

Extending Kubernetes API

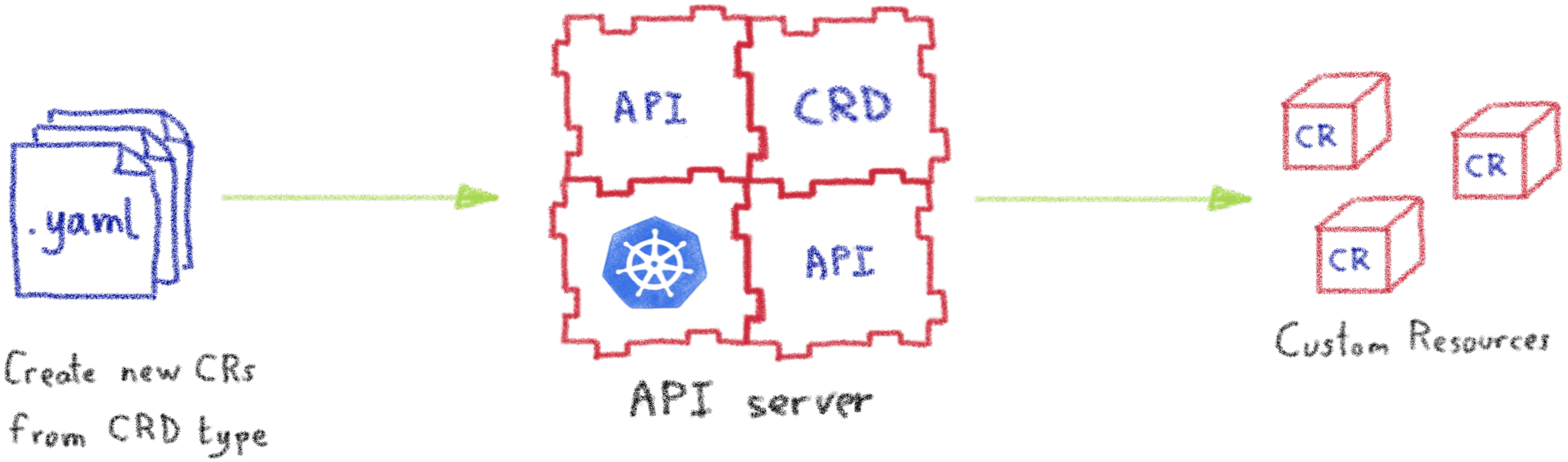


Extending Kubernetes API



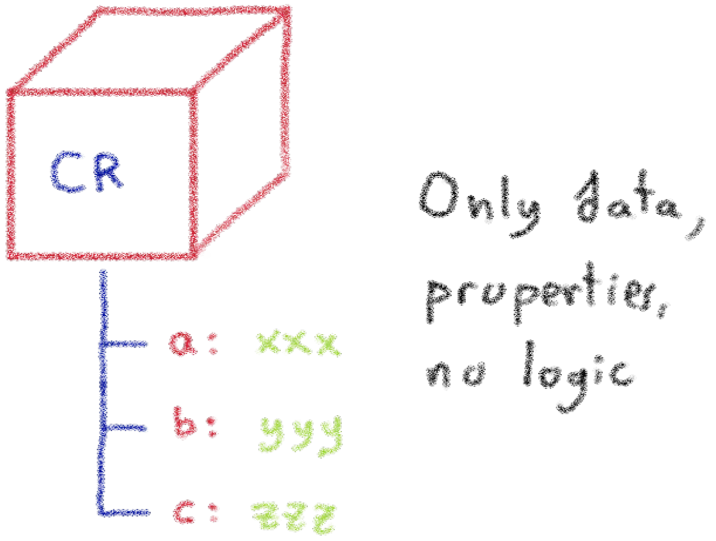
By defining new types of resources, internal or external to the cluster

With a CRD you can create CR in the cluster



They are the blueprints of the Custom Resources

Custom Resources are simply data

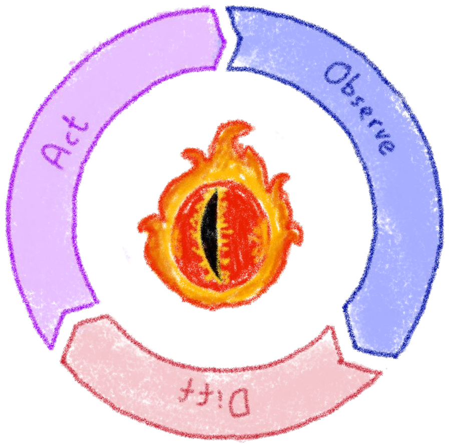


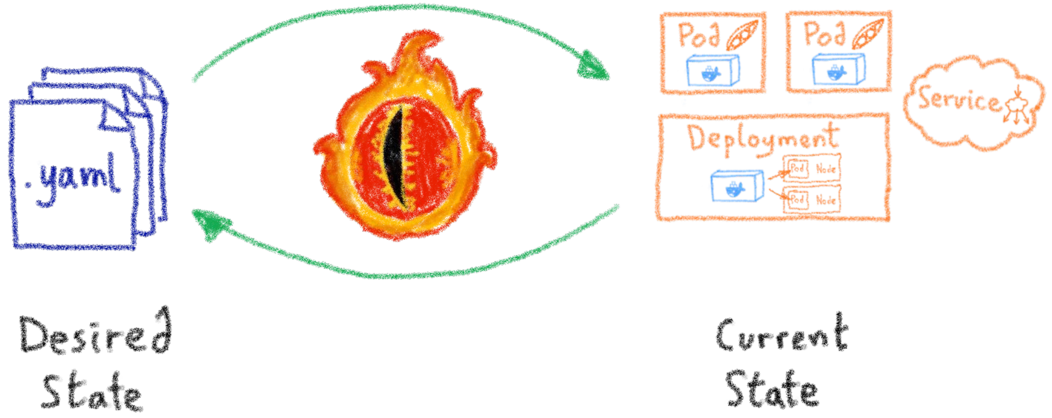
All the logic must be in the Controller

Kubernetes Controllers

Keeping an eye on the resources

A reconcile loop





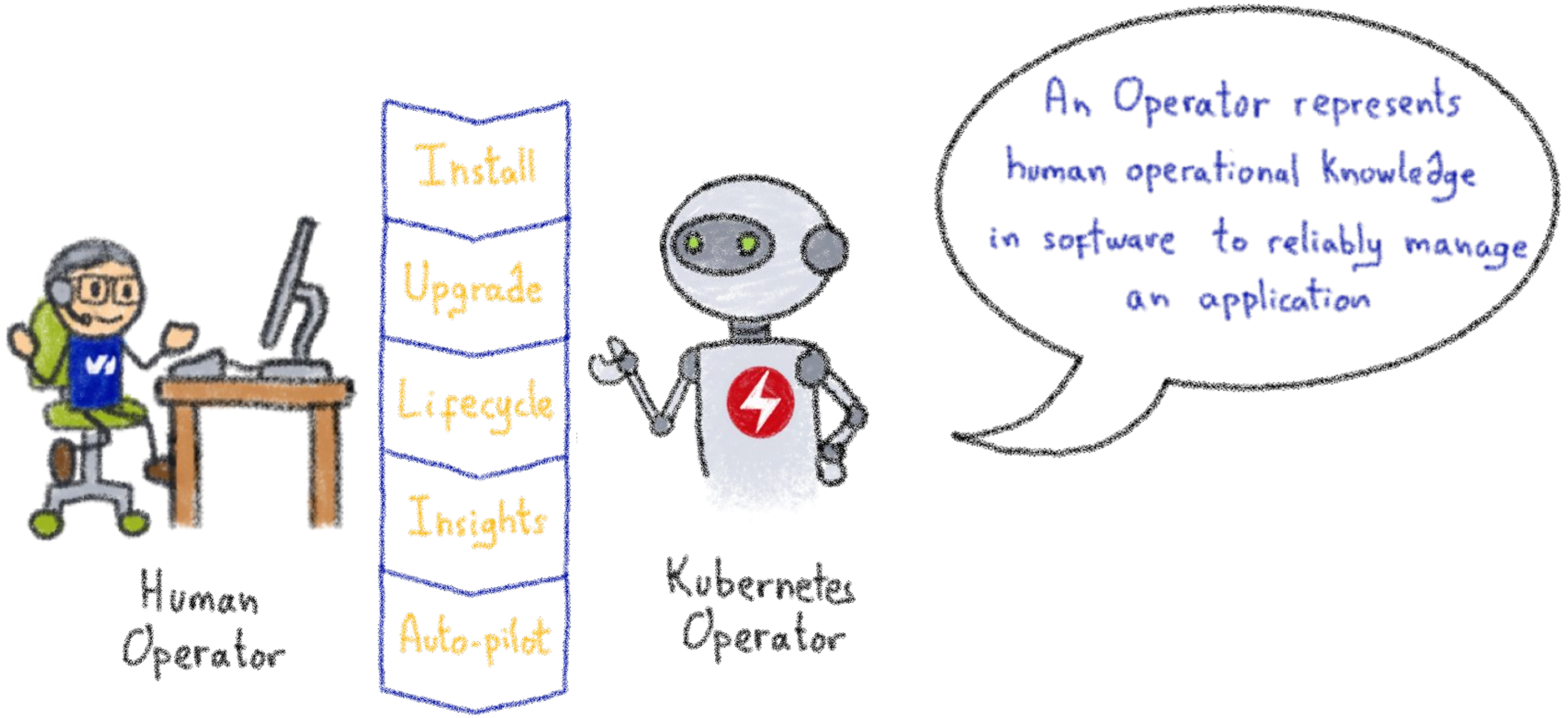
Controllers watch the state of the cluster, and make or request changes where needed

Kubernetes Operator

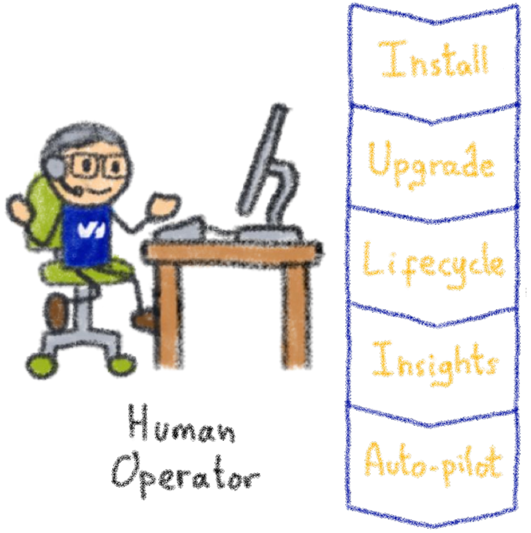
Automating operations

OPERATORS

What's a Kubernetes Operator?

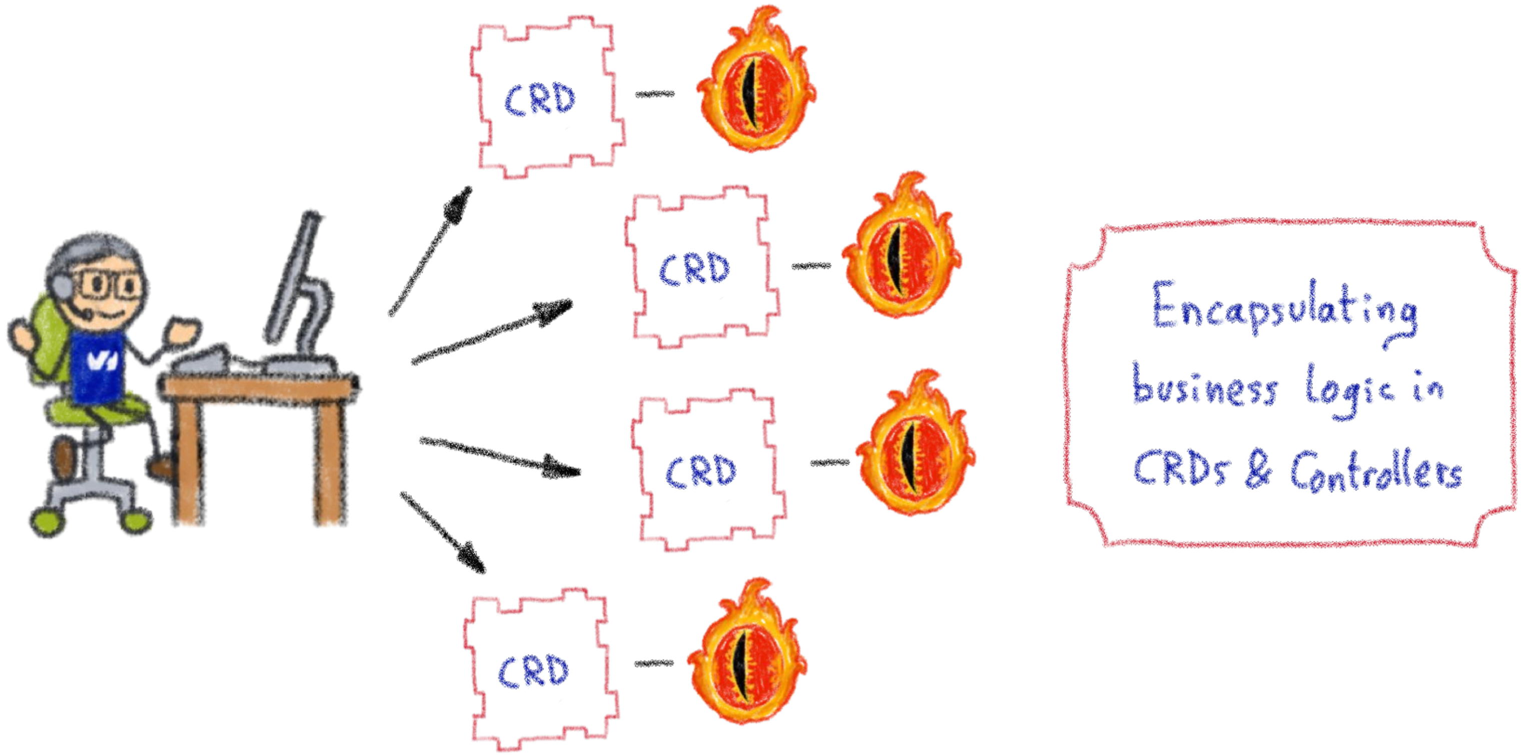

Install
Upgrade
Lifecycle
Insights
Auto-pilot
Human Operator
Kubernetes Operator
An Operator represents human operational knowledge in software to reliably manage an application

Example: databases

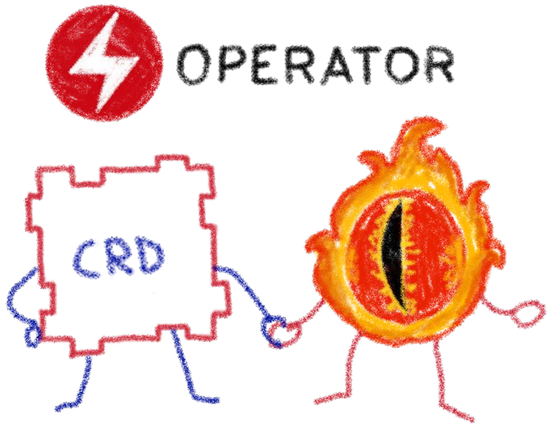


Things like adding an instance to a pool, doing a backup, sharding...

Knowledge encoded in CRDs and Controllers

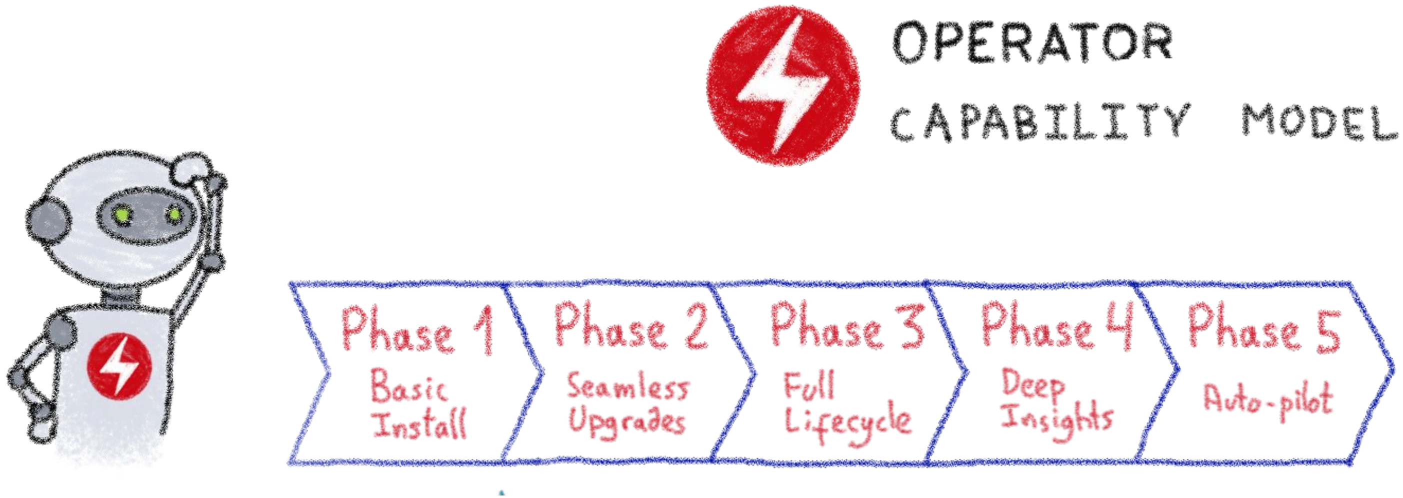


Custom Controllers for Custom Resources



Operators implement and manage Custom Resources using custom reconciliation logic

Operator Capability Model



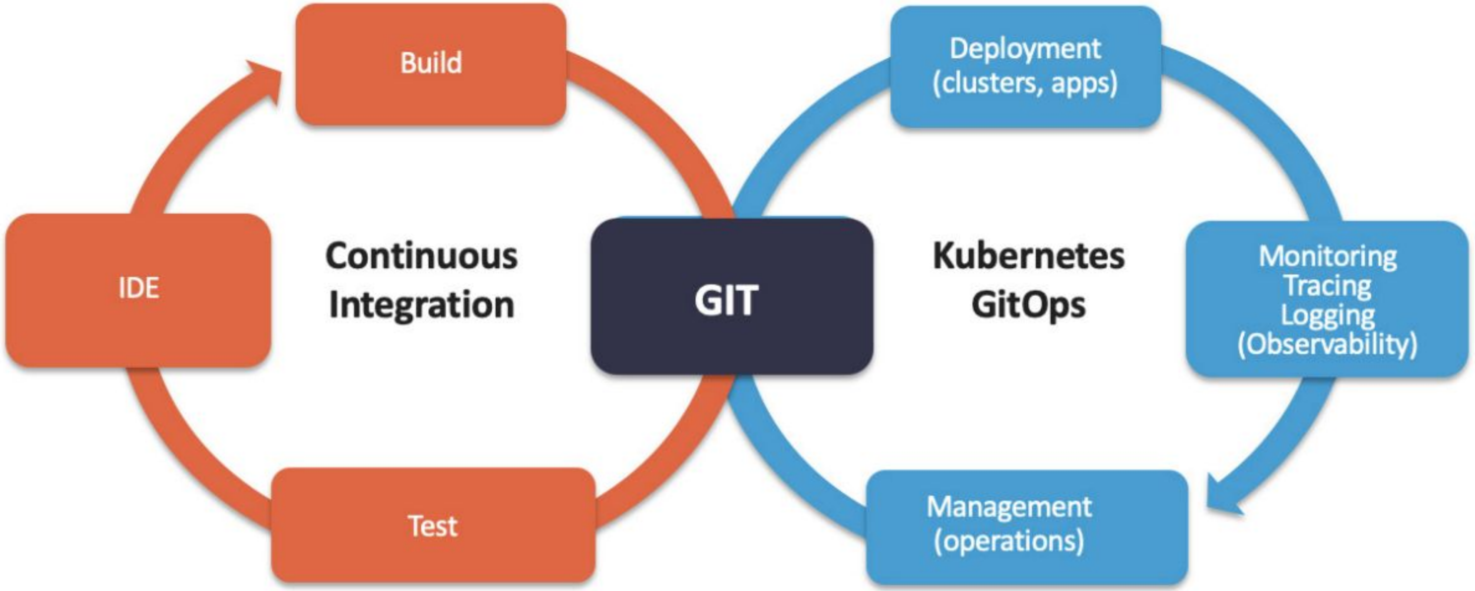
Gauging the operator maturity

GitOps

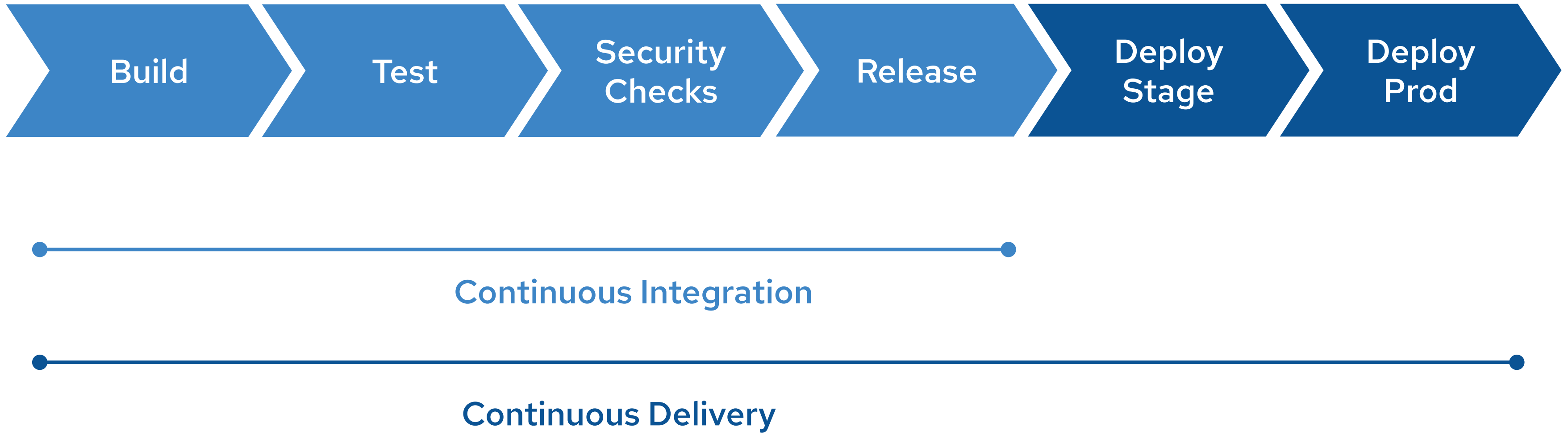
Les devs n'ont pas vocation à avoir accès au cluster
Comment faire ?

Les devs utilisent déjà Git, et c'est leur source de vérité. Utilisons Git comme source de vérité pour l'infra déclarative

A central source of truth



Continuous Integration(CI) & Continuous Delivery (CD)



What is GitOps?

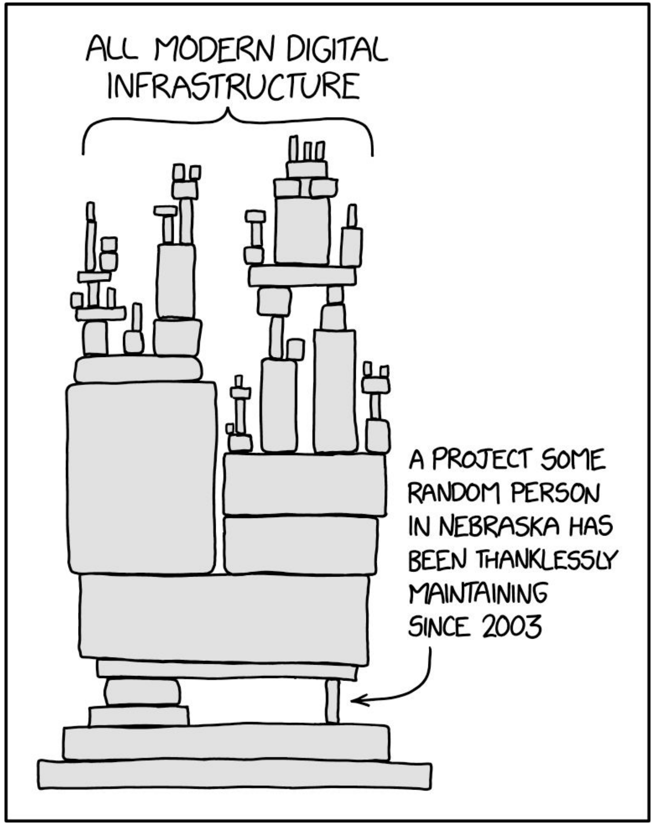
Git is the single source of truth → Treat everything as code → **Operations through Git workflows**

Platform

What is it? Why is it useful?

Why do we need Platform Engineers?

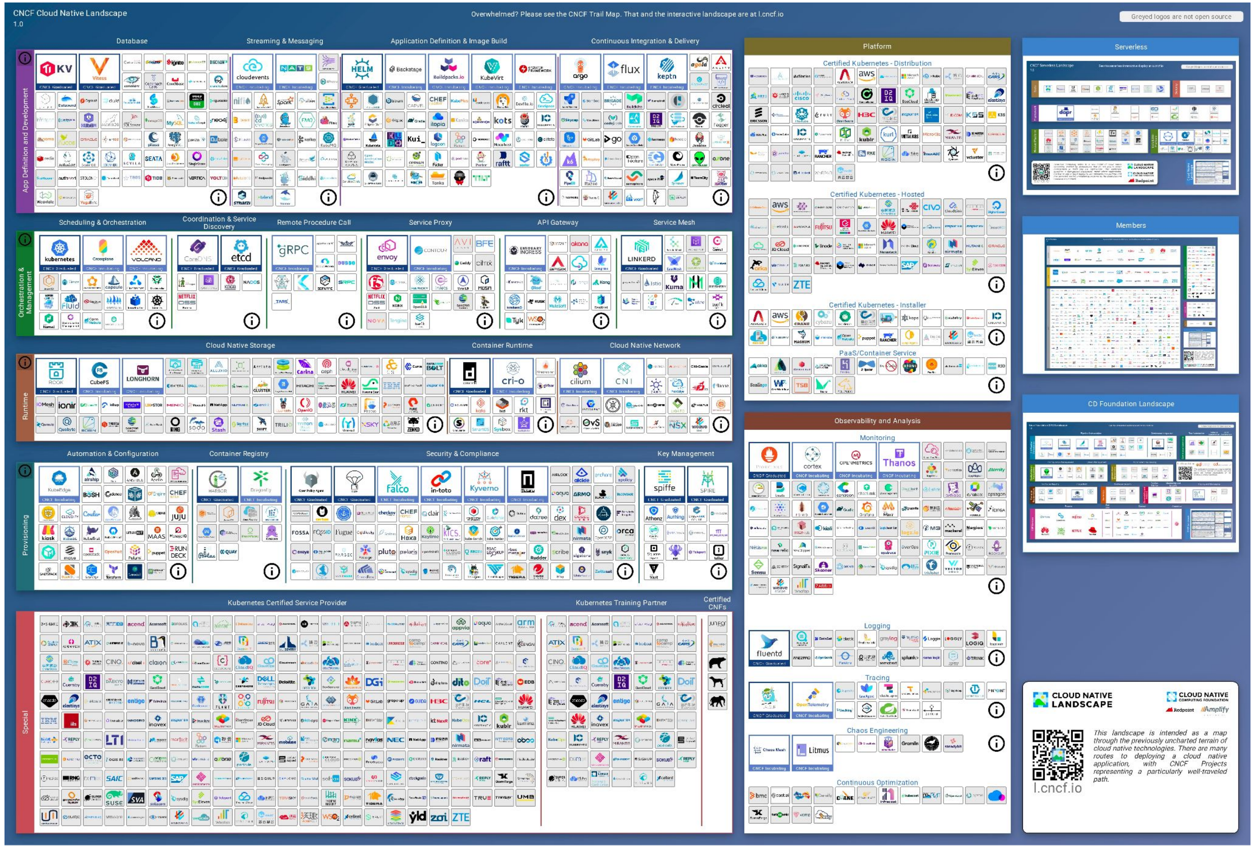
A fancy name for something already there


ALL MODERN DIGITAL INFRASTRUCTURE
A PROJECT SOME RANDOM PERSON IN NEBRASKA HAS BEEN THANKLESSLY MAINTAINING SINCE 2003

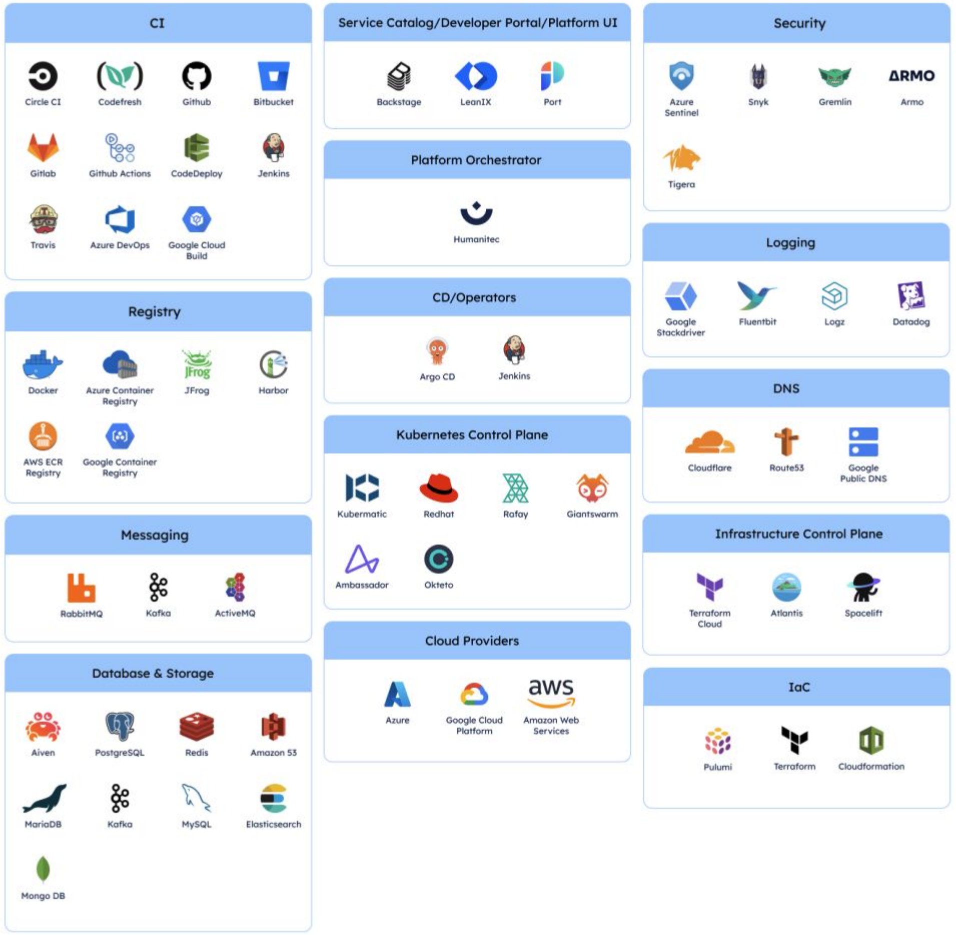
Most companies already have some kind of platform

Often homemade…

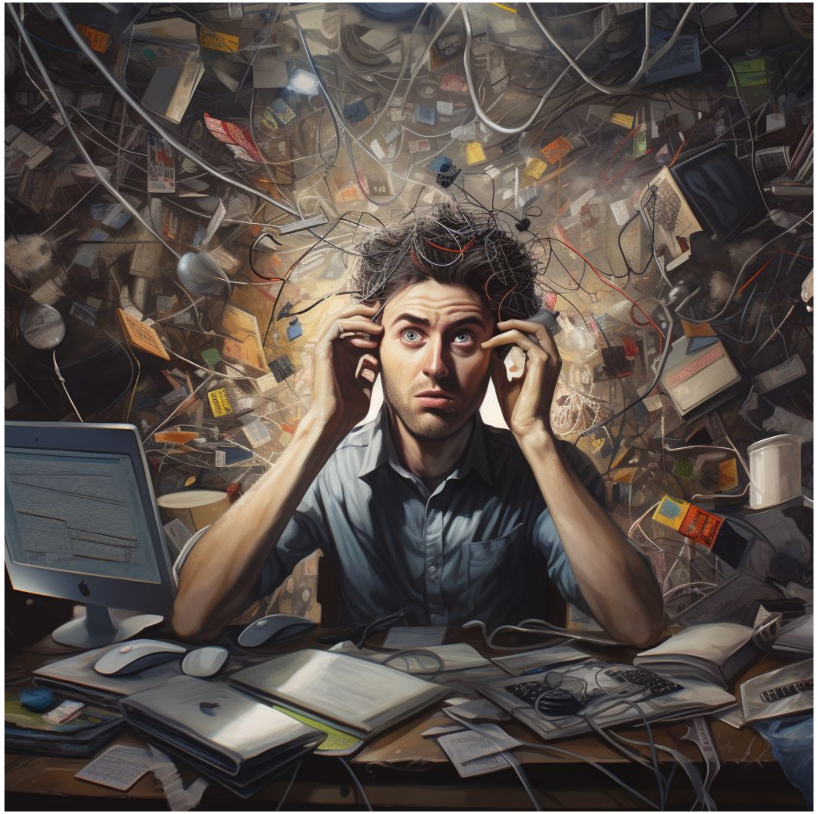
So many options ...


CNCF Cloud Native Landscape
Overwhelmed? Please see the CNCF Trail Map. That and the interactive landscape are at l.cncf.io
Greyed logos are not open source
App Definition and Development
Database
Streaming & Messaging
Application Definition & Image Build
Continuous Integration & Delivery
Orchestration & Management
Scheduling & Orchestration
Coordination & Service Discovery
Remote Procedure Call
Service Proxy
API Gateway
Service Mesh
Runtime
Cloud Native Storage
Container Runtime
Cloud Native Network
Provisioning
Automation & Configuration
Container Registry
Security & Compliance
Key Management
Special
Kubernetes Certified Service Provider
Kubernetes Training Partner
Certified CNFs
Platform
Certified Kubernetes - Distribution
Certified Kubernetes - Hosted
Certified Kubernetes - Installer
PaaS/Container Service
Observability and Analysis
Monitoring
Logging
Tracing
Chaos Engineering
Continuous Optimization
Serverless
Members
CD Foundation Landscape
CLOUD NATIVE LANDSCAPE
This landscape is intended as a map through the previously uncharted terrain of cloud native technologies. There are many routes to deploying a cloud native application, with CNCF Projects representing a particularly well-traveled path.
l.cncf.io

So many options ...



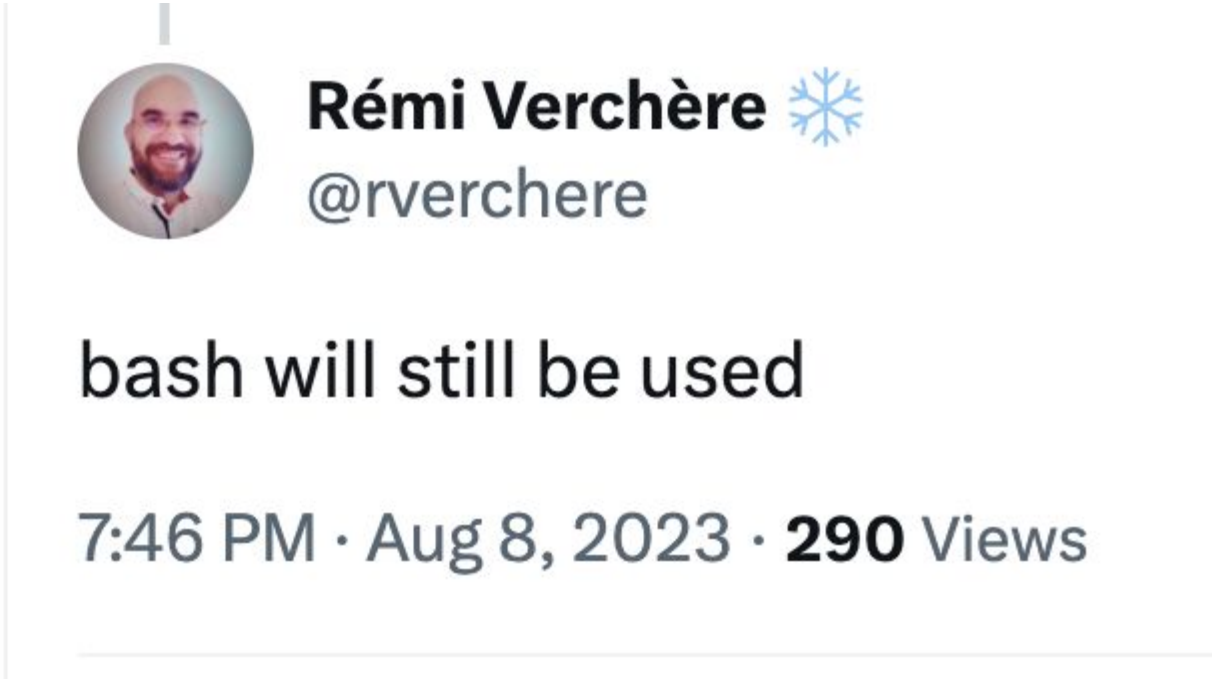
Shift left and Cognitive Load



Managing the self-service commodity

“The discipline of designing and building toolchains and workflows that enable self-service capabilities for software engineering organizations in the cloud-native era.
Platform engineers provide an ***integrated product*** *most often referred to as an “Internal Developer Platform” covering the operational necessities of the entire lifecycle of an application.”*

Lucas Galante

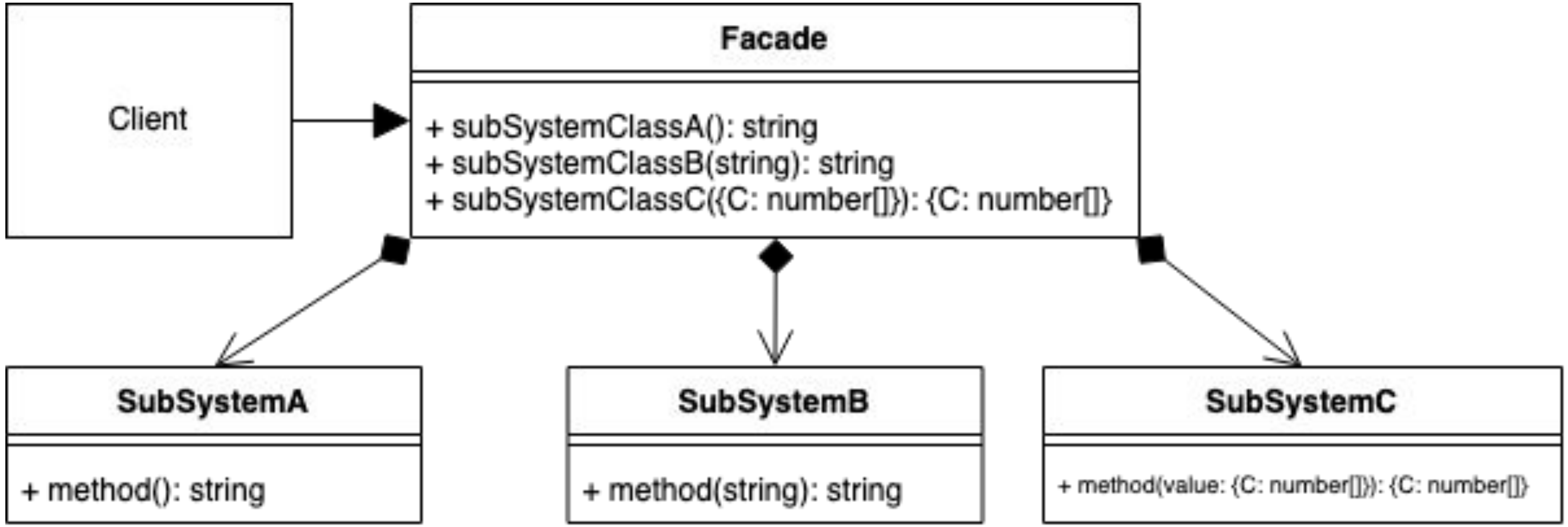


Rémi Verchère ❄️
@rverchere

bash will still be used

7:46 PM · Aug 8, 2023 · **290** Views

IDP is the new Facade Pattern

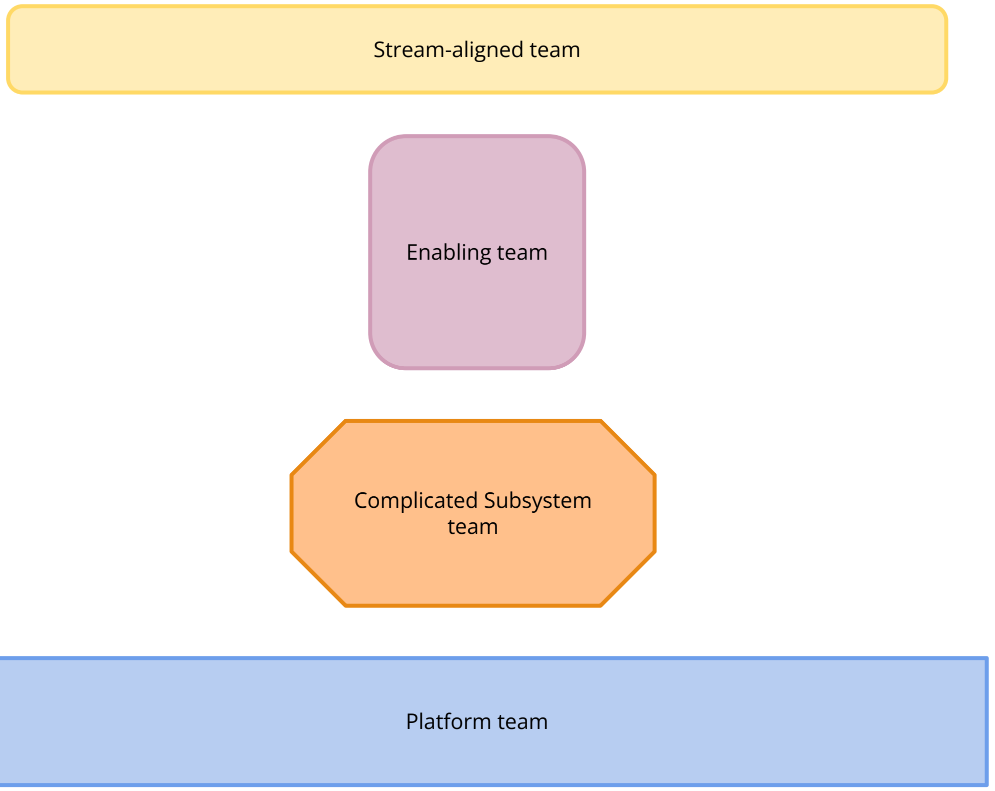

Client
Facade
+ subSystemClassA(): string
+ subSystemClassB(string): string
+ subSystemClassC({C: number[]}): {C: number[]}
SubSystemA
+ method(): string
SubSystemB
+ method(string): string
SubSystemC
+ method(value: {C: number[]}): {C: number[]}

Team Topologies

“Organizations that consider establishing such a platform team should be very cautious not to accidentally create a separate DevOps team, nor should they simply relabel their existing hosting and operations structure as a platform.”

TechRadar, October 2021

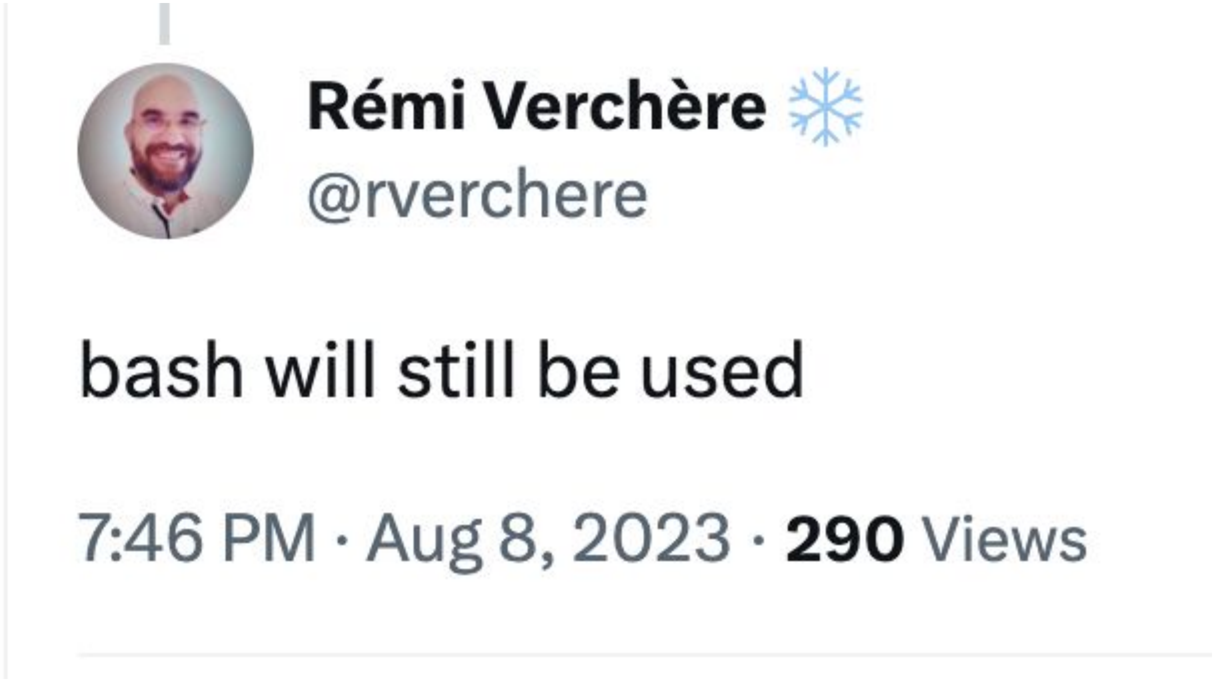
Team Topologies



Platform as a Product

The Internal Dev Platform is the **Product**

- Conduct user research
  - Run friction logs
  - Empathy meetings
- Create a roadmap
- …



Rémi Verchère ❄️
@rverchere

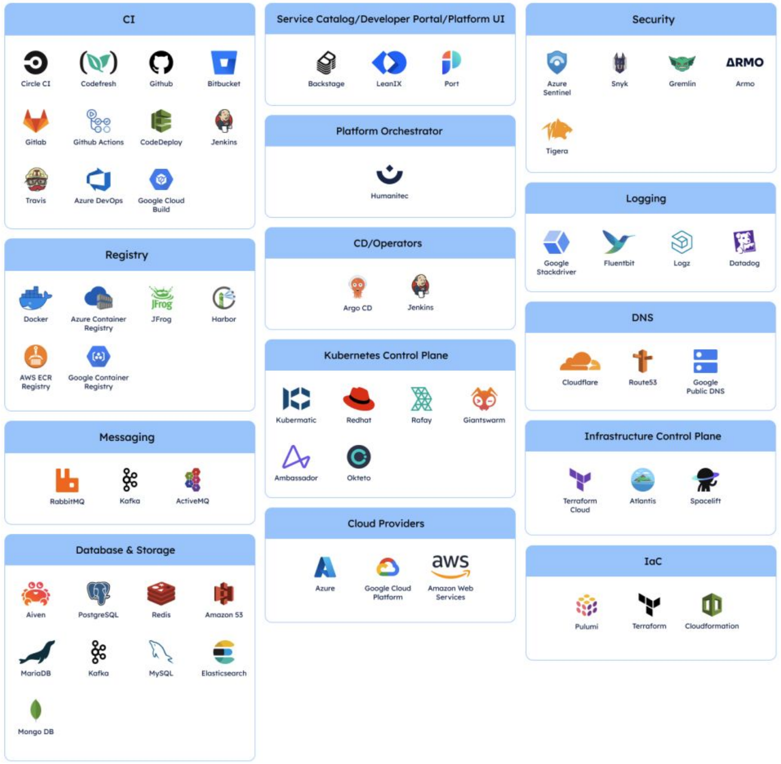
bash will still be used

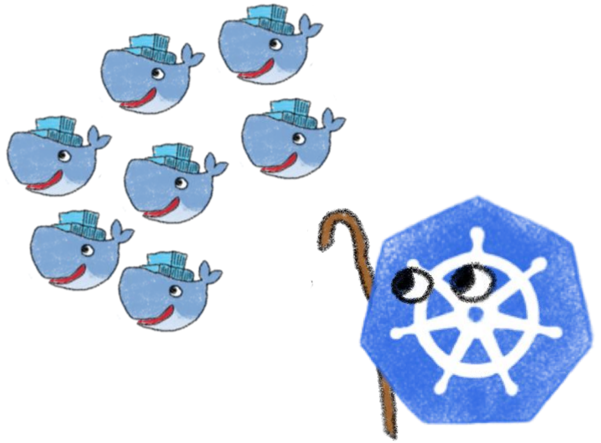
7:46 PM · Aug 8, 2023 · **290** Views

Build your own Platform

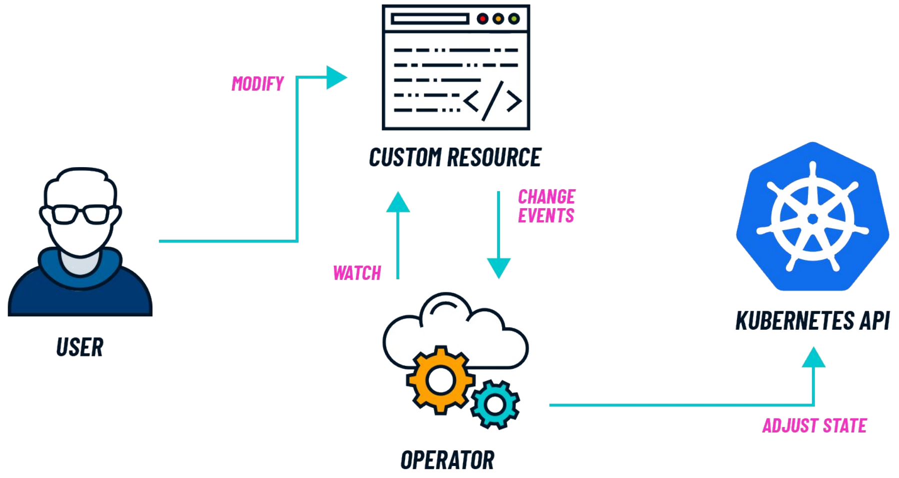
And becoming Platform Engineer

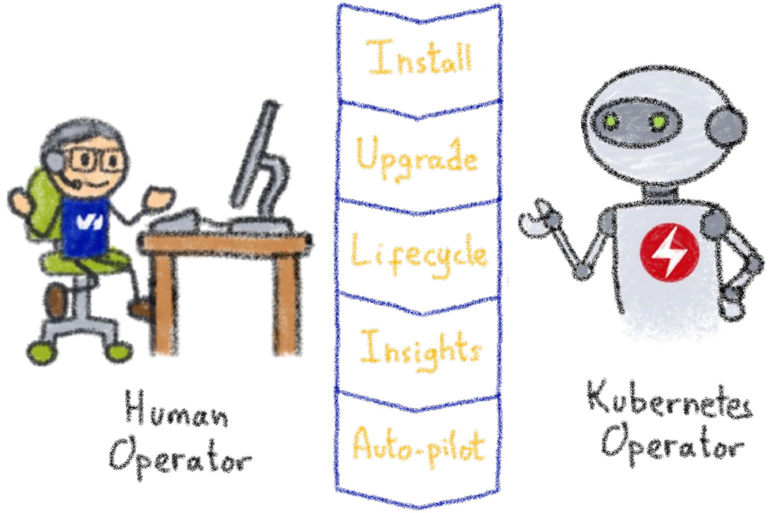
How to glue those all together ?





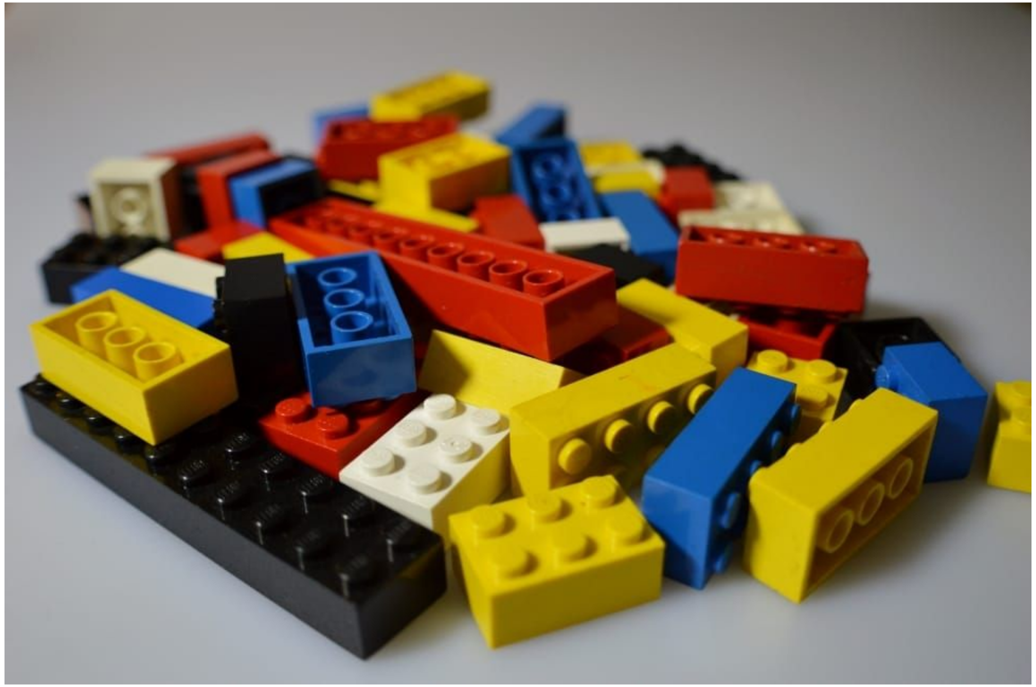
Kubernetes Operators





A Kubernetes version of the human operator

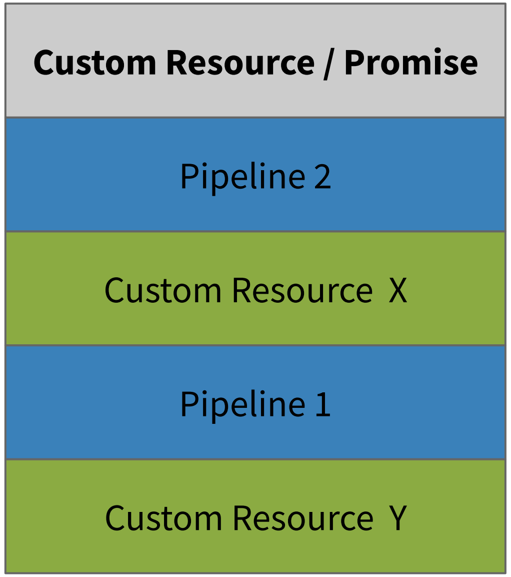
How to assemble all those building bricks ?

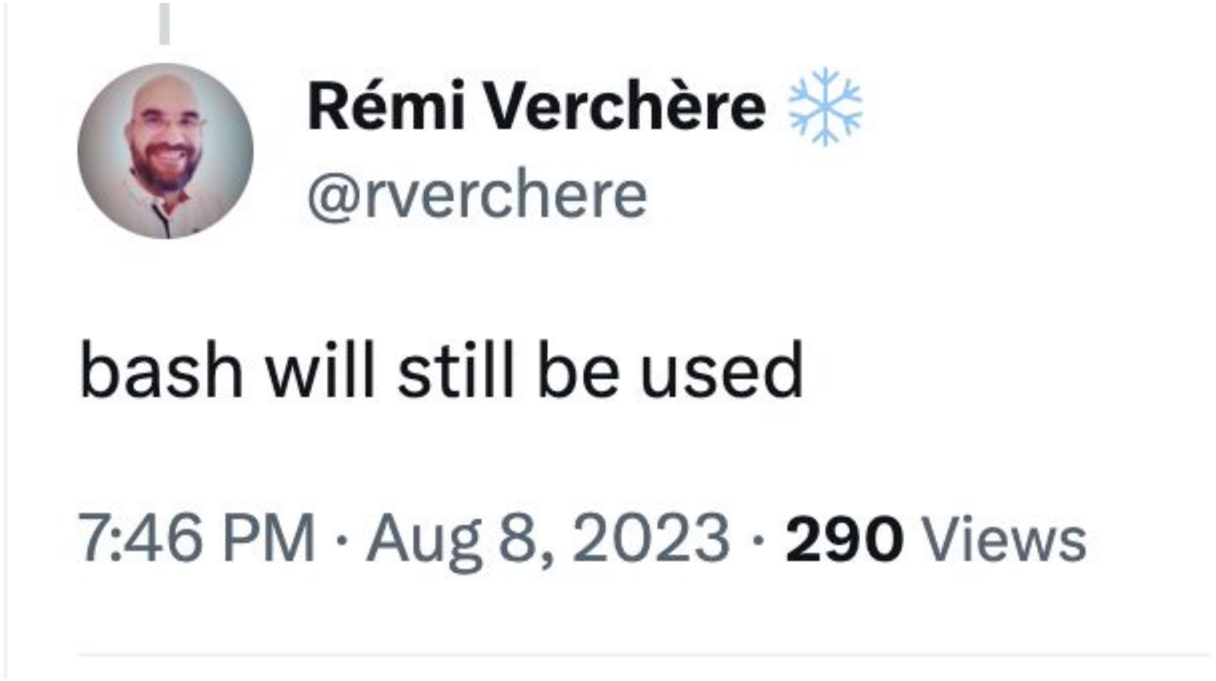


Platform Building Frameworks

KRATIX

“A framework for building platforms”





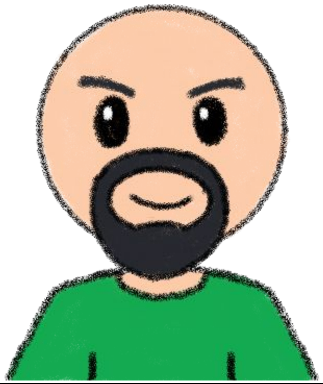
Rémi Verchère ❄️
@rverchere

bash will still be used

7:46 PM · Aug 8, 2023 · **290** Views

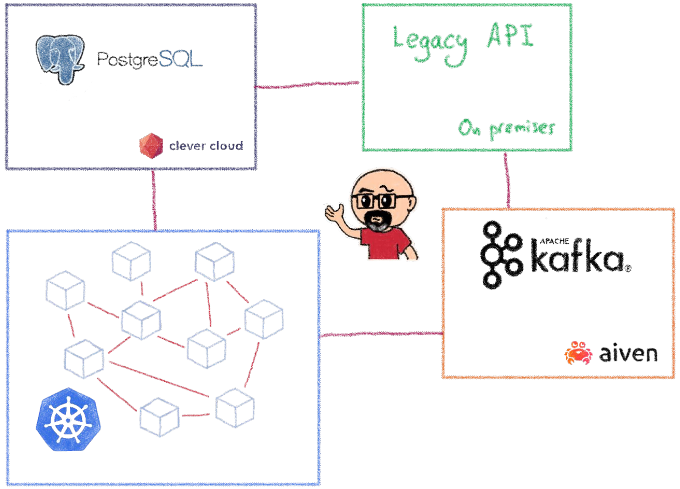
Example: How to build an operator

Because they are easier than you think...



Example: Using several operators

They are made to work together



Example: Building a K8s-based Platform

If you're already a Kubernetes user

KRATIX

Example: Using a PaaS provider

There is a world outside Kubernetes

clever cloud