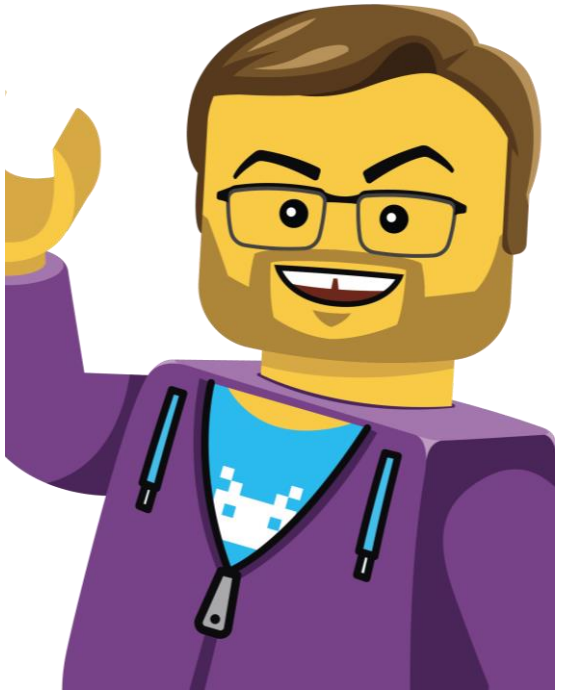
plain concepts

DevOps

2020.06.30

EL CAMINO DE LA ENTREGA DE VALOR EN SOFTWARE

plain concepts



Luis Fraile

CONSULTOR ALM-DEVOPS

Digamos que ayudo a equipos de desarrollo de software a poder hacerlo mejor mediante DevOps y ... oh .... Ah ... *practices ágiles*

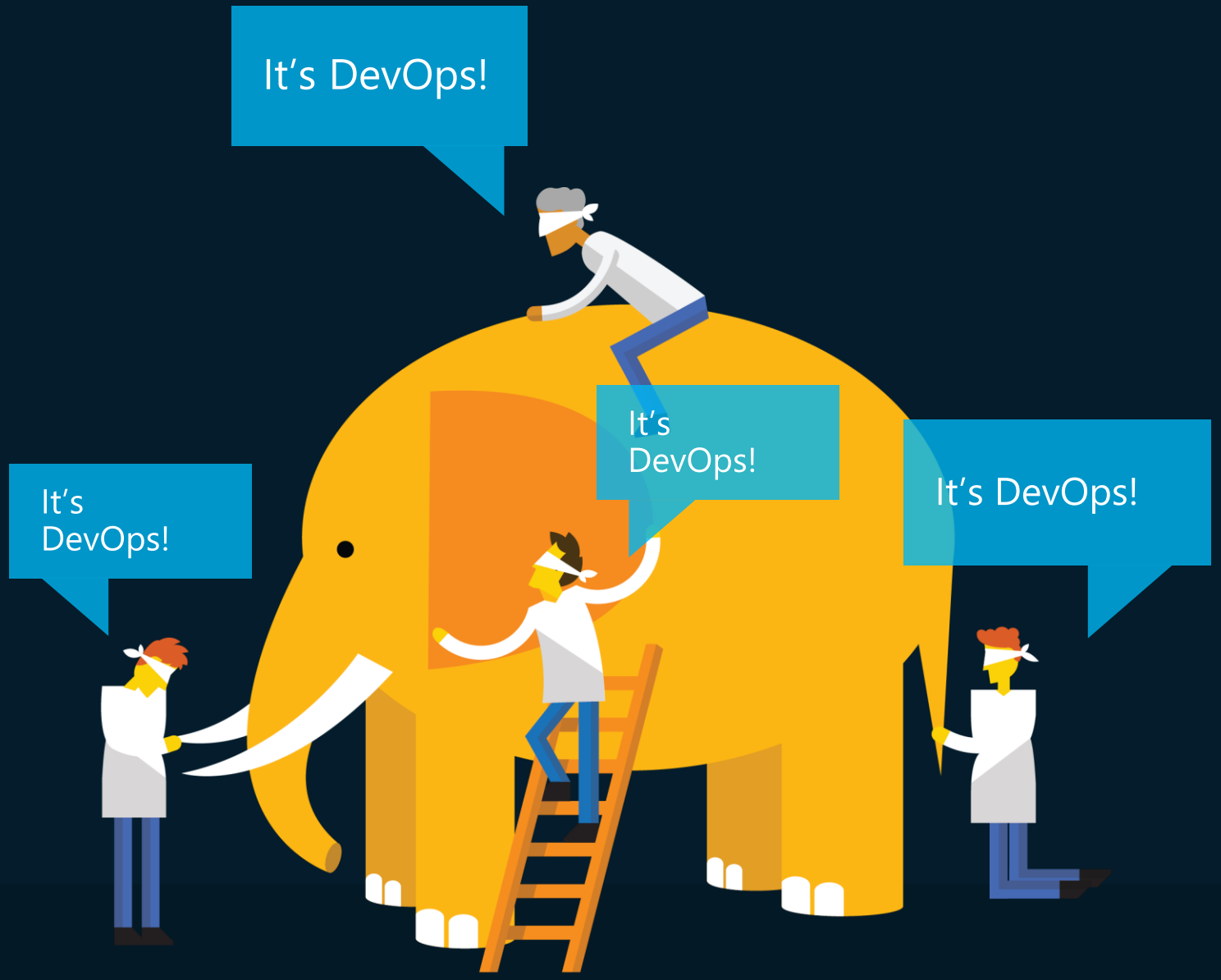
@lfraile

plain concepts

"**DevOps** is development and operations collaboration"

"**DevOps** is using automation"

"**DevOps** is small deployments"



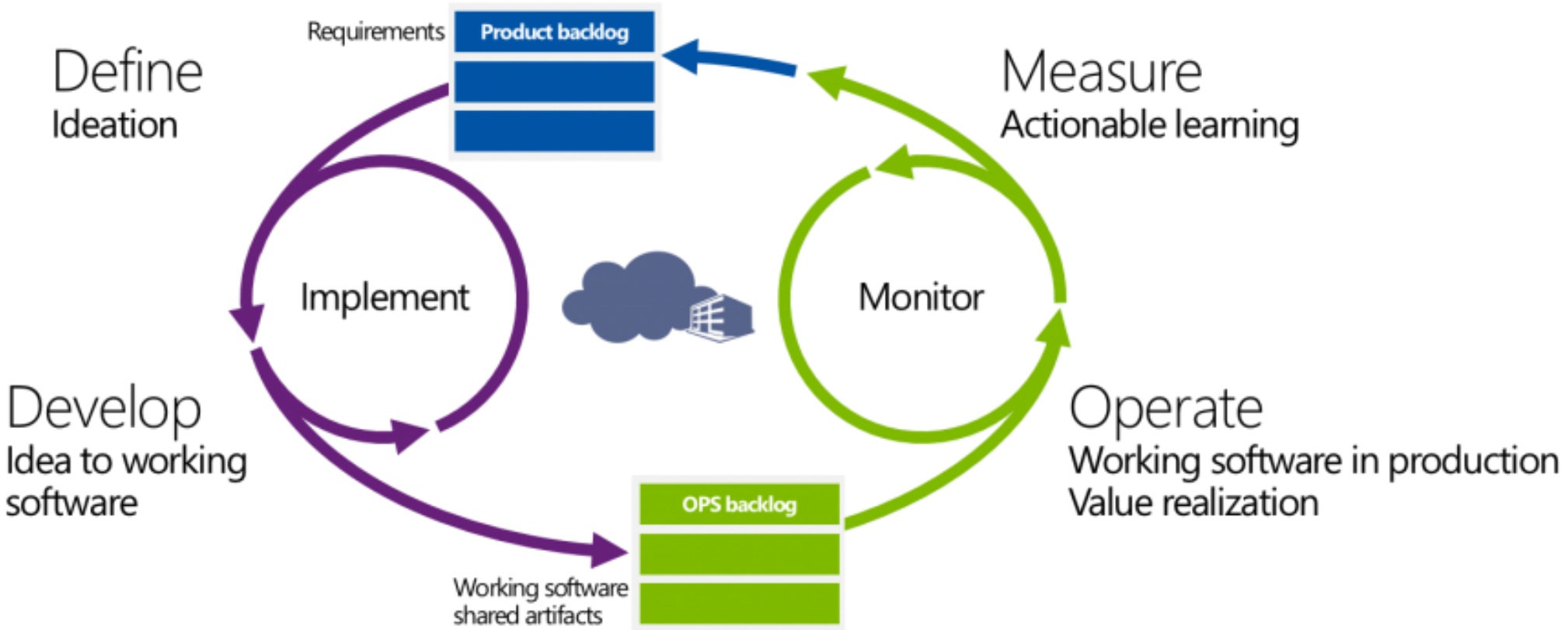
"**DevOps** is treating your infrastructure as code"

"**DevOps** is feature switches"

"Scrum for Ops?"

plain concepts

Waste elimination | Cycle time reduction | Integration & visibility

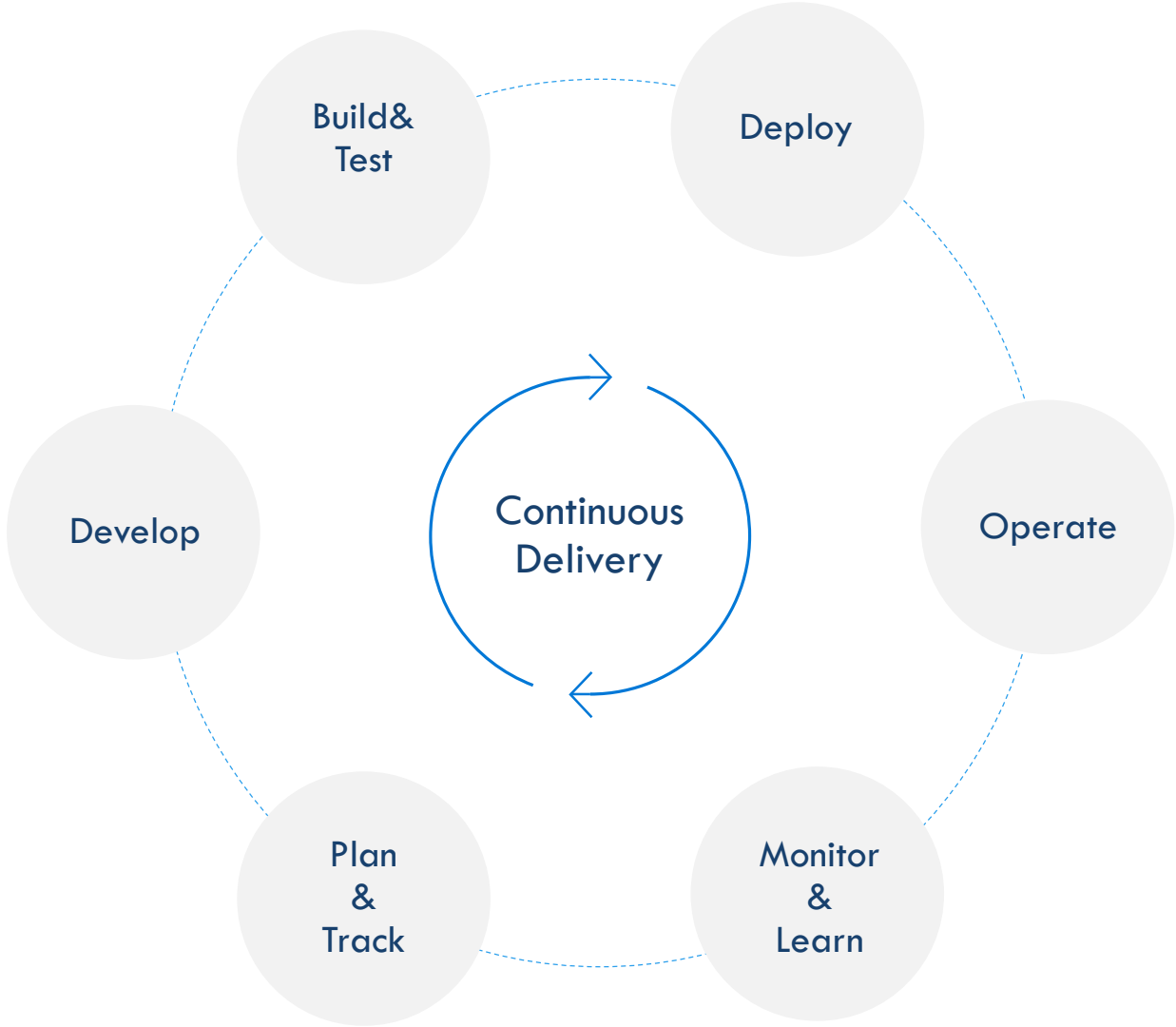


Continuous feedback | Continuous quality | Continuous delivery

plain concepts

> DevOps is the union of **people**, **process**, and **products** to enable continuous delivery of value to your end users.

Donovan Brown



plain concepts

Pero ¿por qué hacemos DevOps?

Agilidad (¿agile?)

Validar pronto y continuadamente

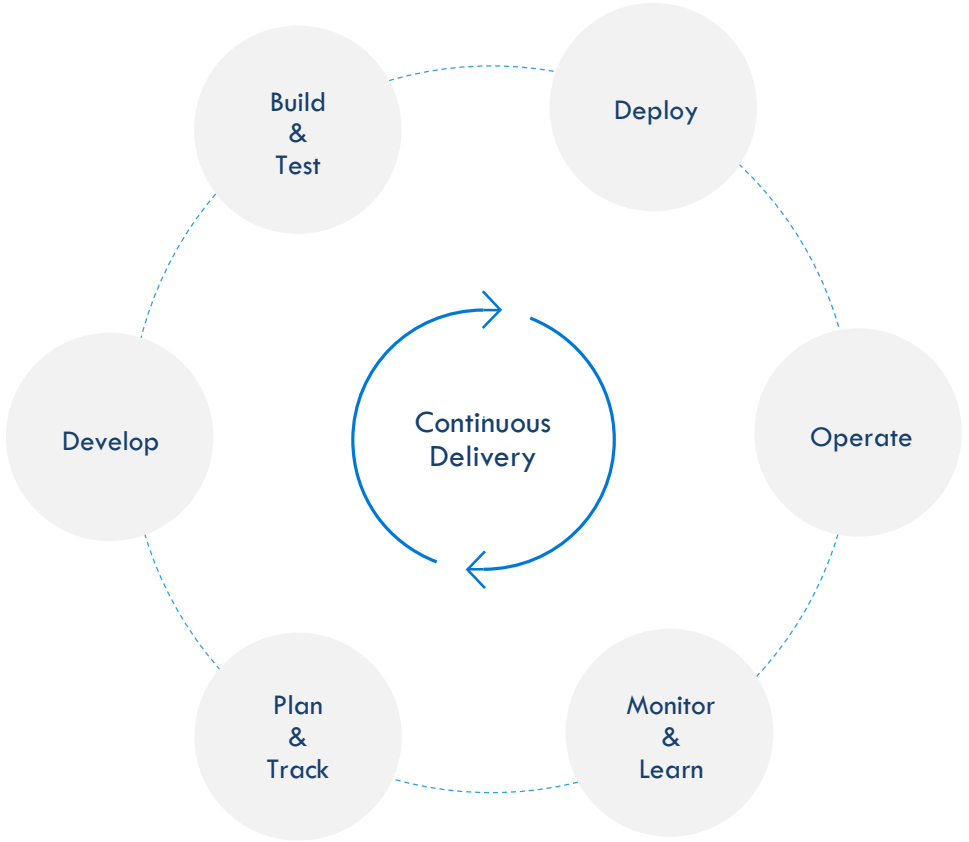
Rapidez en la entrega

Y en la respuesta a desastres

Aseguramiento de calidad

Entregar más no significa entregar cualquier cosa

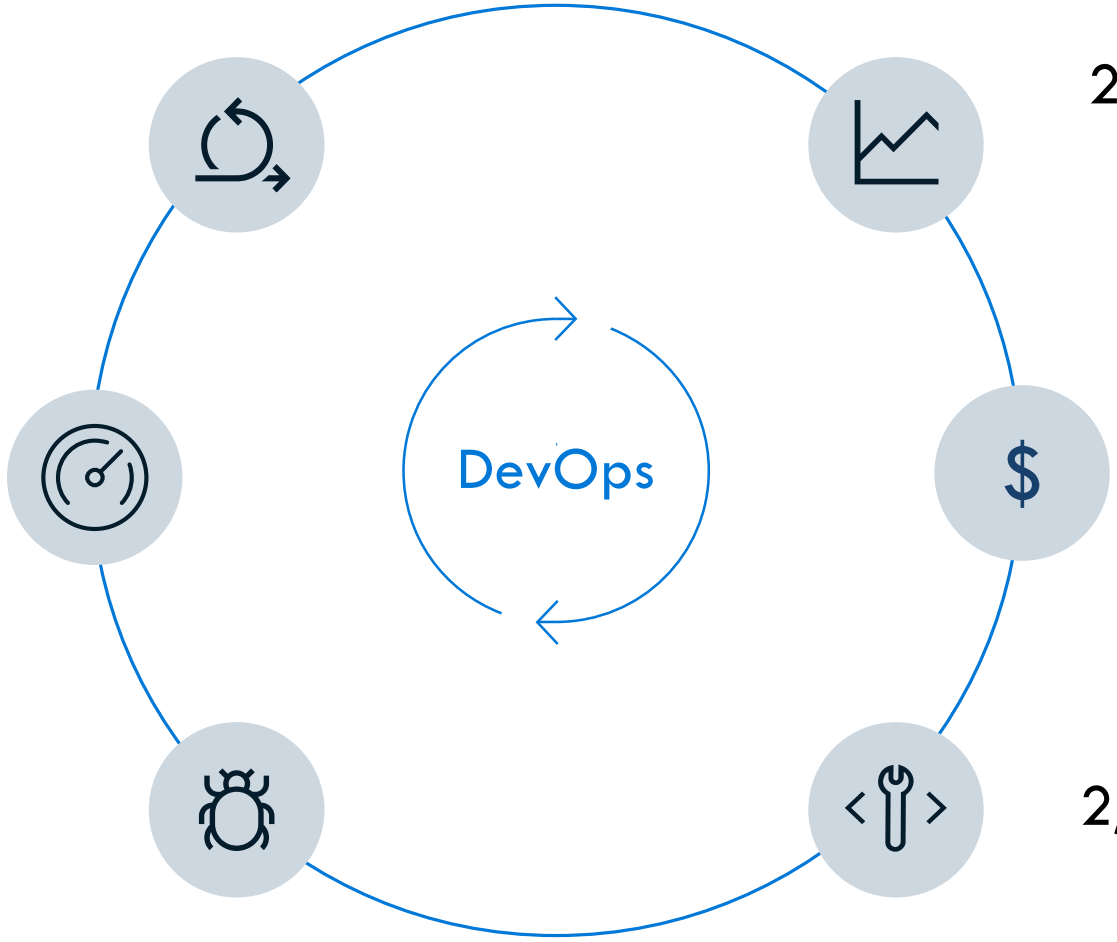
… simplemente vivir mejor desarrollando software.



DevOps...

46x Deployment Frequency

2,555x Faster Lead Time For Changes



Faster Time to Market

Increased Revenue

7x Lower Change Failure Rate

2,604x Faster Mean Time to Recover

Source: 2018 Accelerate: State of DevOps: Strategies for a New Economy." N. Forsgren, J. Humble, G. Kim. DevOps Research and Assessment (DORA)

plain concepts

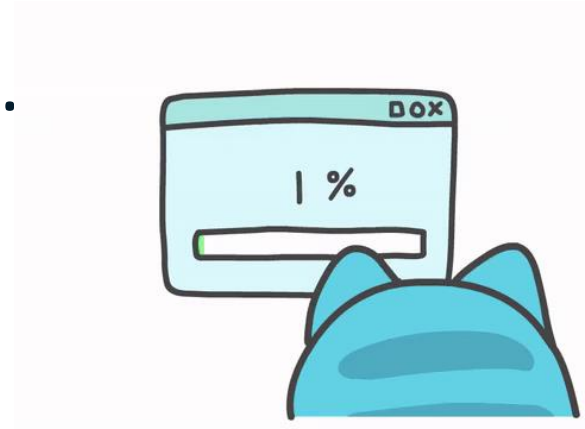
¿Qué es lo que queremos?

Working software

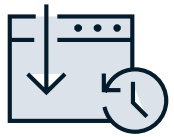
No queremos simples “features”

Productos que resuelven “problemas”

Y que no generen otros ..



Some practices



Continuous Integration (CI)

- Improve software development quality and speed.
- When you use Azure Pipelines or Jenkins to build apps in the cloud and deploy to Azure, each time you commit code, it's automatically built and tested and bugs are detected faster.

Continuous Deployment (CD)

- By combining continuous integration and infrastructure as code (IaC), you'll achieve identical deployments and the confidence to deploy to production at any time.
- With continuous deployment, you can automate the entire process from code commit to production if your CI/CD tests are successful.

Continuous Learning & Monitoring

- With Azure Application Insights you can identify how your applications are performing and test if the recent deployment made things better or worse.
- Using CI/CD practices, paired with monitoring tools, you'll be able to safely deliver features to your customers as soon as they're ready.

plain concepts

Azure DevOps

Azure Boards

Gestión del trabajo

Azure Pipelines

CI / CD

Azure Repos

Grepositorios Git

Azure Test Plans

Gestión de planes de pruebas

Azure Artifacts

Artefactos (nuget, npm, maven, …)

https://azure.com/devops

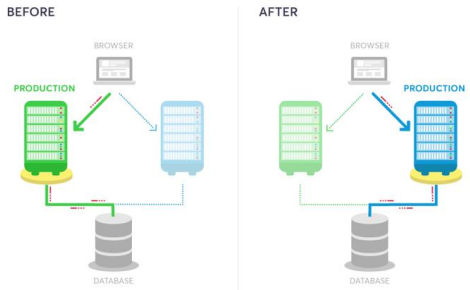
Estrategias de ~~despliegue~~ entrega

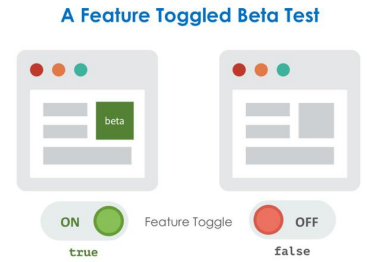
Unas cuantas

- Blue/Green
- Toggles
- Rolling updates
- Por entornos …

O todas a la vez

Pero todas comparten algo





plain concepts

Empieza todo en el desarrollo

Hay que cambiar la mentalidad de desplegar a **entregar**

Necesitamos cambiar el modo de pensar en el desarrollo

No es sólo escribir *código limpio*

Definir estrategia de entrega, junto a negocio, junto a ops, junto a sec

plain concepts

Aspectos a tener en cuenta

- Calidad
- Criterios de entrega
- Sostenibilidad
- Monitorización
- Disaster recovery
- Service Reliability Engineering

plain concepts

Criterios de entrega

¿Sabes cuando y dónde entregar?

No es entregar software sin más

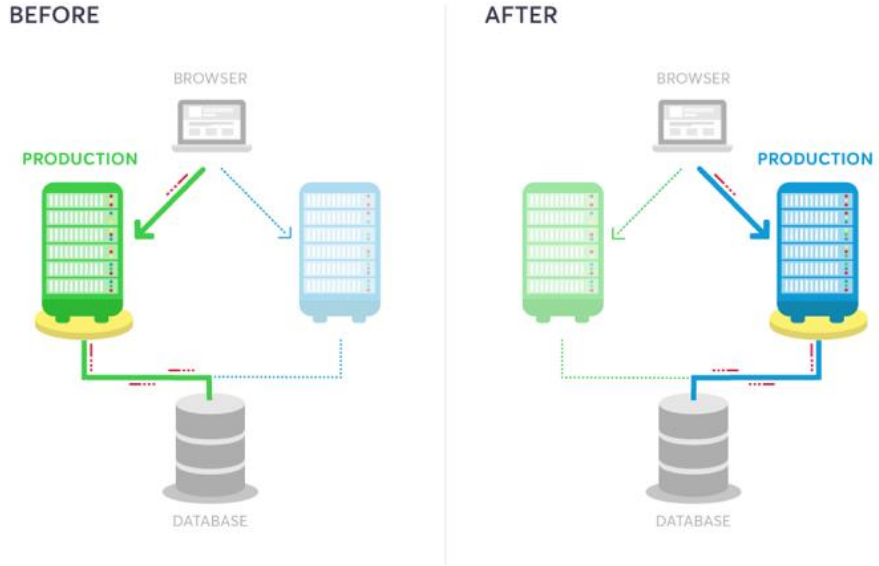
Establecer nuestros criterios de entrega

- Calidad
- Rendimiento
- Aceptación de los usuarios
- Integración en el producto

Reglas de calidad mínima de paso entre entornos

plain concepts

Blue-Green



Producción está en entorno “green”

Desplegamos a un entorno “blue”

Con mecanismos tipo balanceador cambiamos

Las peticiones de antiguo producción se terminan de ejecutar

Todas las peticiones nuevas entran a nueva versión

https://www.blazemeter.com/blog/five-blue-green-deployment-best-practices-for-a-smooth-release/

plain concepts

Rolling updates

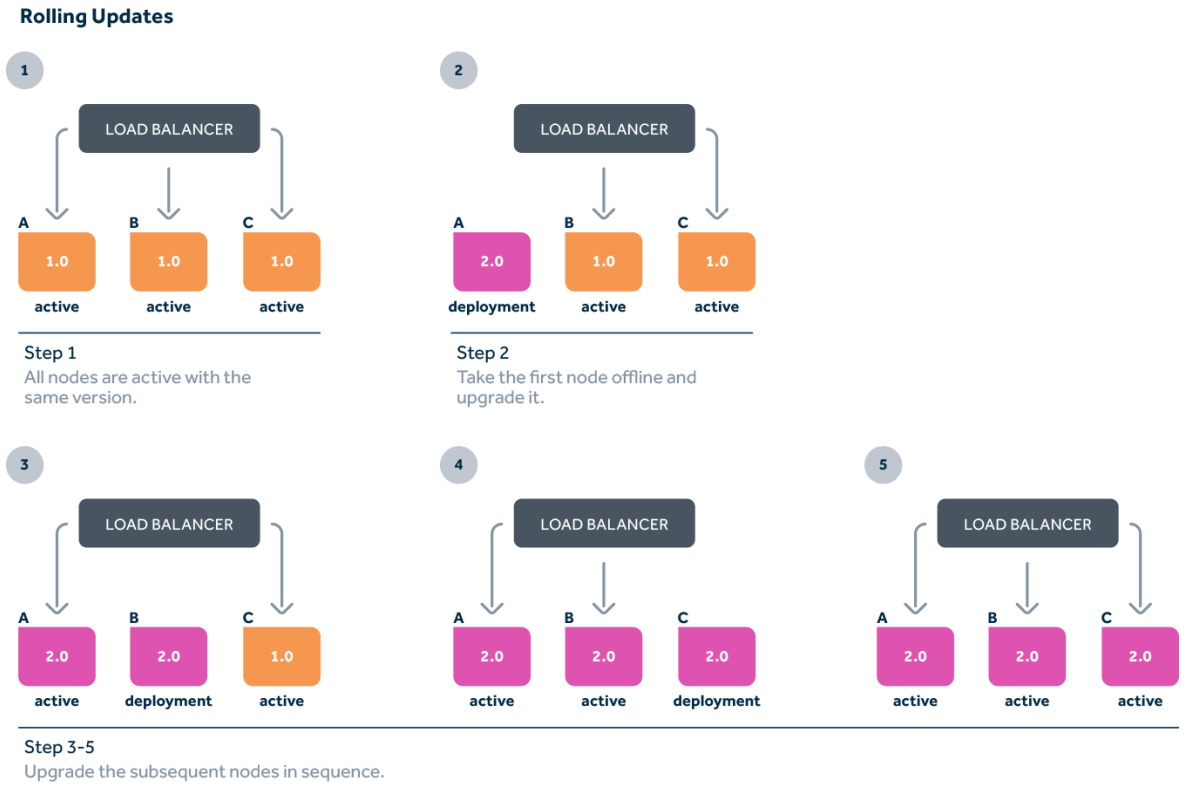
Muy común en k8s

Se desactivan nodos paulatinamente

Se despliegua a nodos desactivados

Se agregan los nodos con nueva versión

Convive durante un tiempo vieja y nueva versión



https://dzone.com/articles/take-release-automation-to-the-next-level-episode-2

plain concepts

Feature toggles

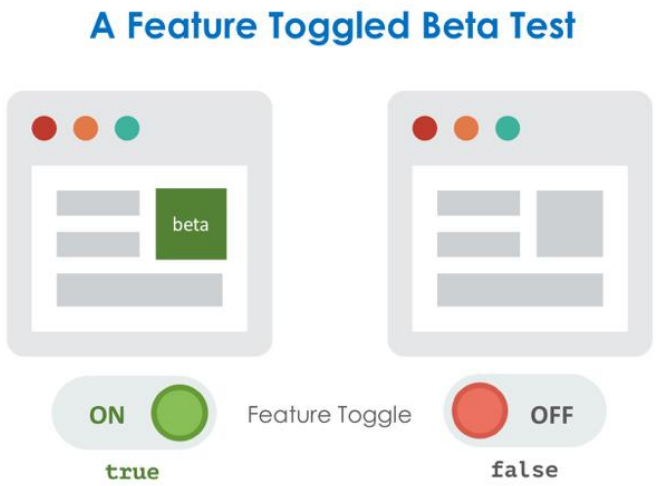
No es puramente “despliegue”

Activamos características en función a reglas

Se combina con cualquiera de las anteriores

OJO: No es solo on-off

OJO: a la limpieza de toggles olvidadas



plain concepts

Estrategia de entrega

¿Toggles?

- ¿Cómo y cuándo las vamos a activar?
- ¿cómo vamos a medir la efectividad?
- ¿Compatibilidad?

Blue-Green

- ¿Estrategia de cambio?
- ¿Compatibilidad?

Rolling updates

- ¿ciclo de update?
- ¿Compatibilidad?

plain concepts

Compatibilidad de versiones

Tenemos que prepararnos para convivir con dos versiones

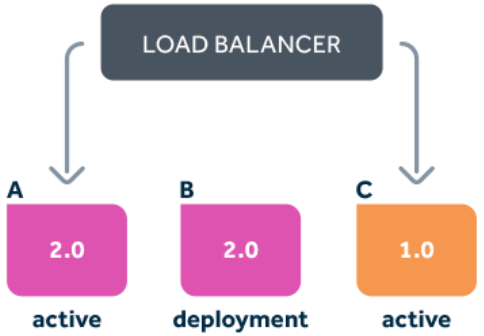
Parallel Changes (Danilo Sato -

https://martinfowler.com/bliki/ParallelChange.html)

- Expand
- Migrate
- Contract

Tanto para almacén de datos

Como para interfaces



https://www.slideshare.net/sergio_pino/commit-conf-2019-zero-downtime-deployment

plain concepts

Facebook, Netflix, etc.

Son siempre hype

- Desplegar cada commit
- Elección libre de herramientas y estrategias
- Chaos Monkey

Cada cual usa sus estrategias

Interesante aprender de ellos

Pero (salvo excepciones) NO somos ellos

plain concepts

Monitorización

¿Cómo saber cuándo hacemos rollback, desactivamos toggle, etc?

Volvemos a Dev: estrategias de monitorización en el Código

- Herramientas de monitorización (no solo logs)
- “Tracking” de eventos

Y vamos a Ops

- Alertas
- Prevención temprana

Y sobre todo: incluirlas en nuestros flujos de entrega automatizados

plain concepts

Monitorizar el negocio

No solo logs y errores

Patrones de uso

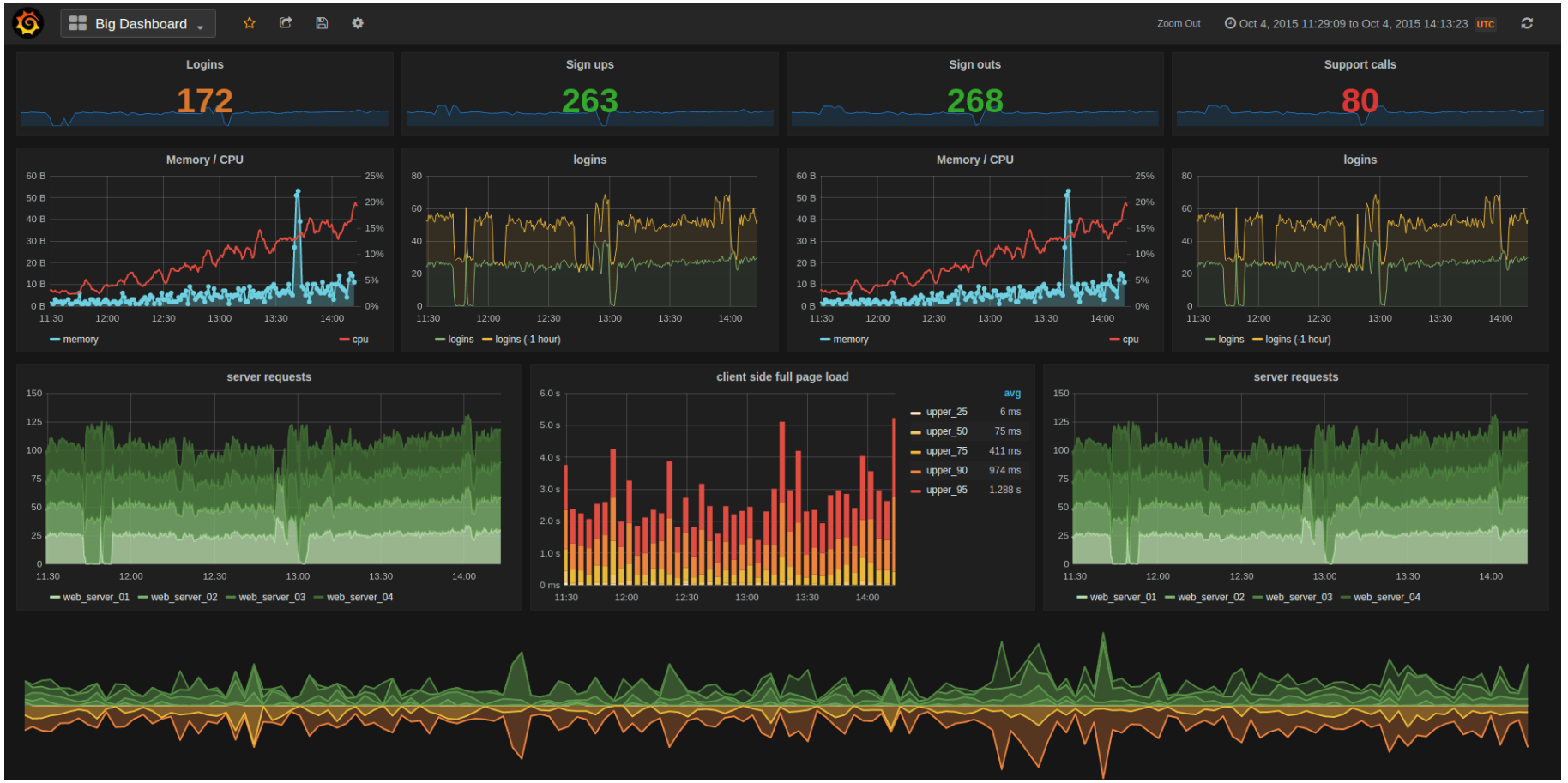
- Características
- Usuarios
- Dispositivos
- …

Definición de eventos importantes

Instrumentación de eventos (… de dominio …)

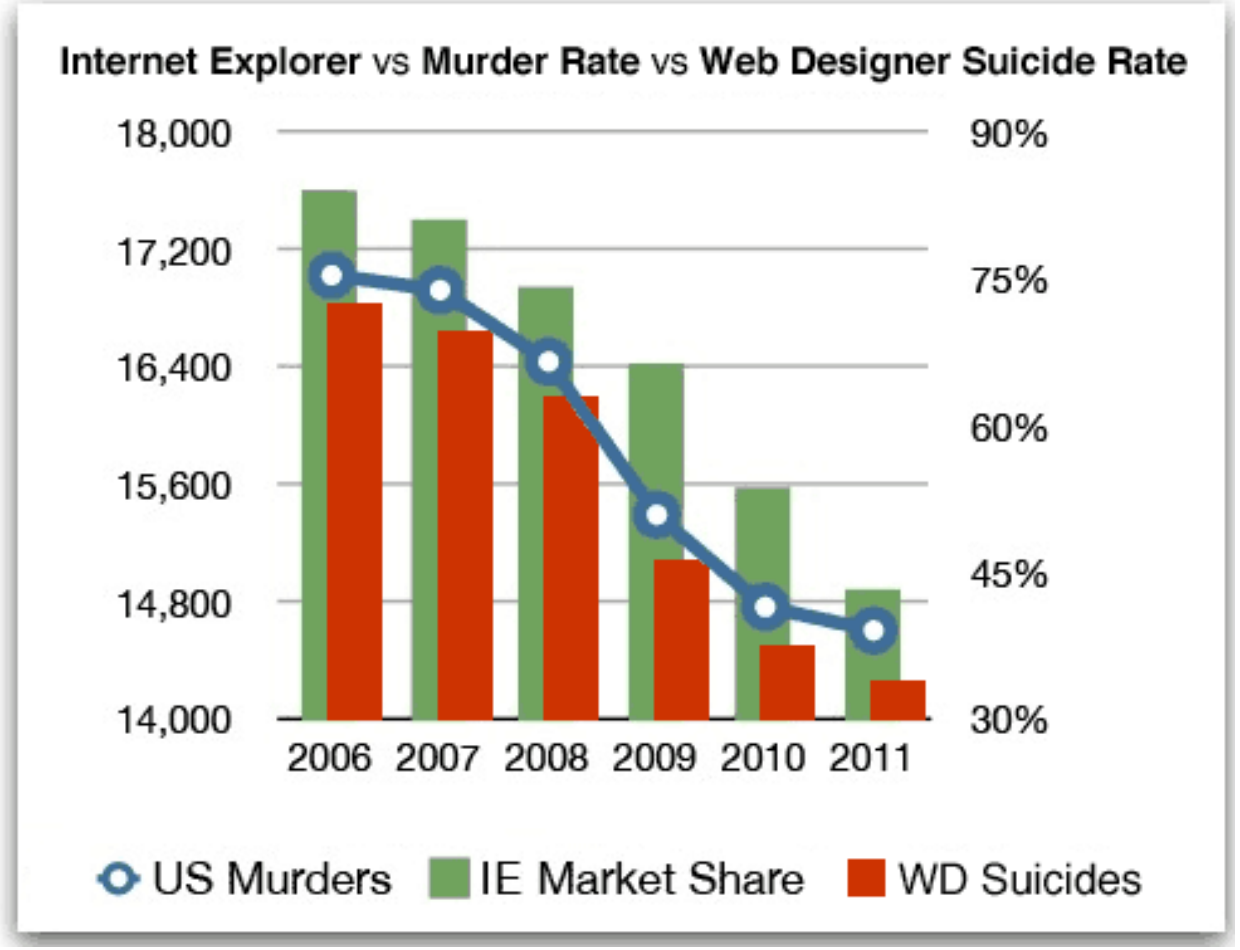
plain concepts

Data porn



plain concepts

Correlación != Causalidad



Thanks Modesto San Juan (10% royalties) …

plain concepts

Infraestructura

Forma parte de la entrega

Hay que asegurar la consistencia y estabilidad

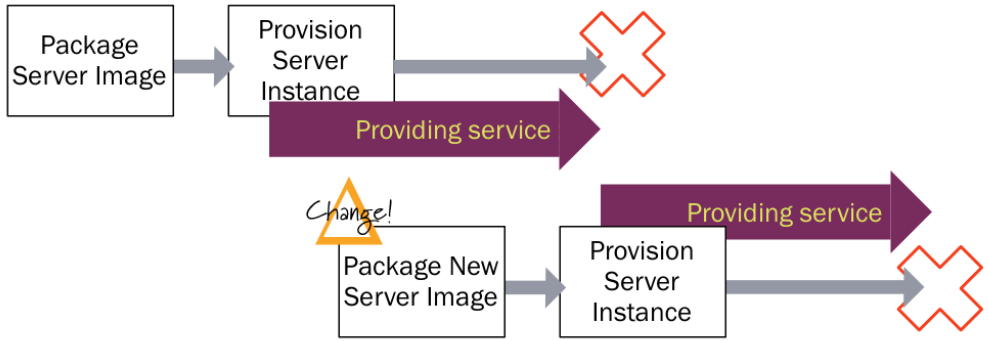
Infraestructura inmutable

https://martinfowler.com/bliki/ImmutableServer.ht

- Siempre se destruye lo antiguo y se crea nueva infraestructura

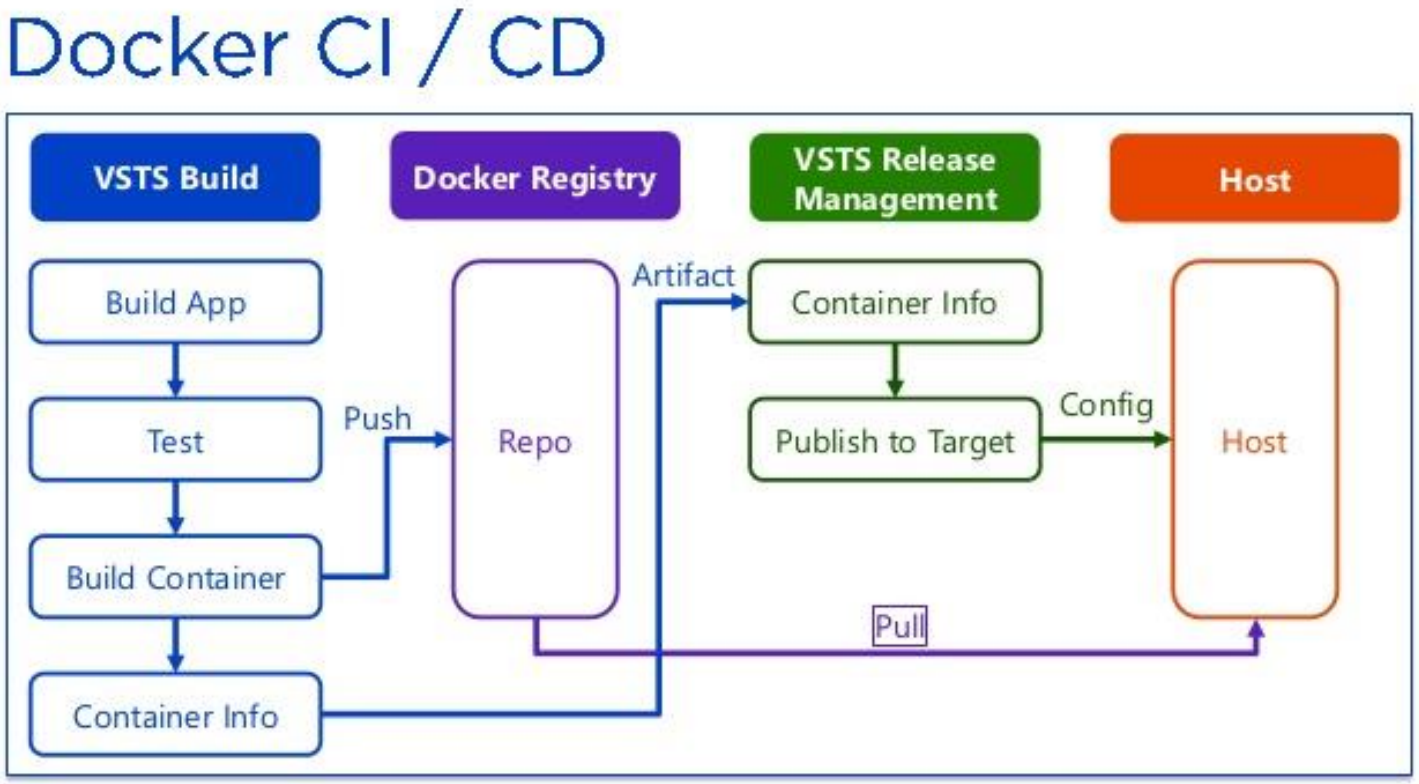
Infrastructure as Code

- Terraform, ansible, puppet, ARM, etc.
- Se desarrolla y entrega con las versions



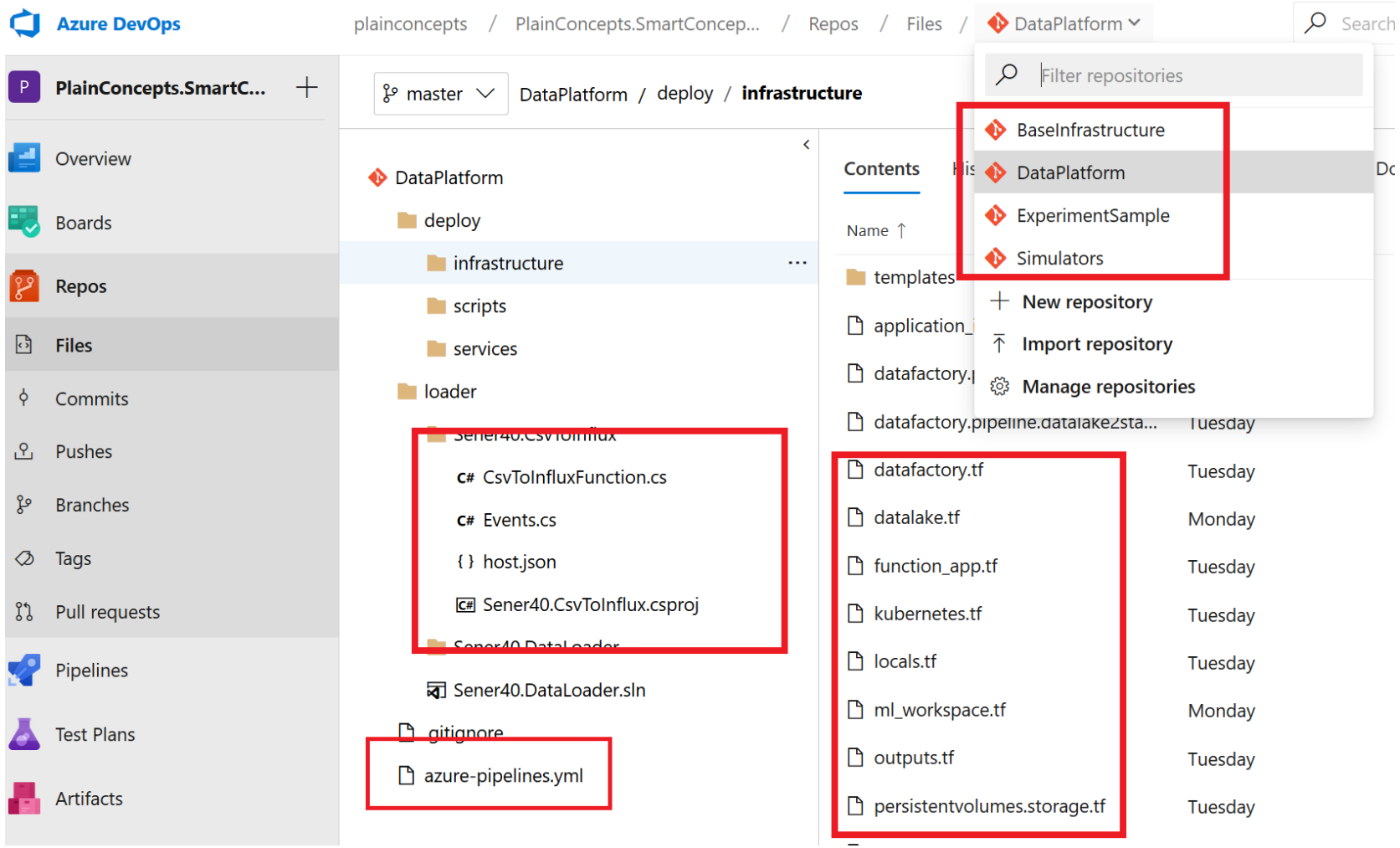
plain concepts

Immutable infrastructure con Docker



plain concepts

Version everything!!



plain concepts

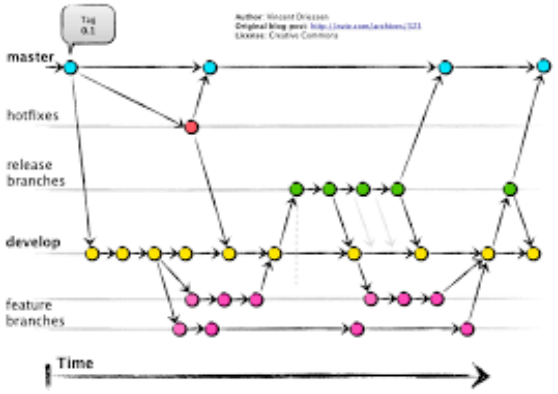
Gestión del código (aKa repositories)

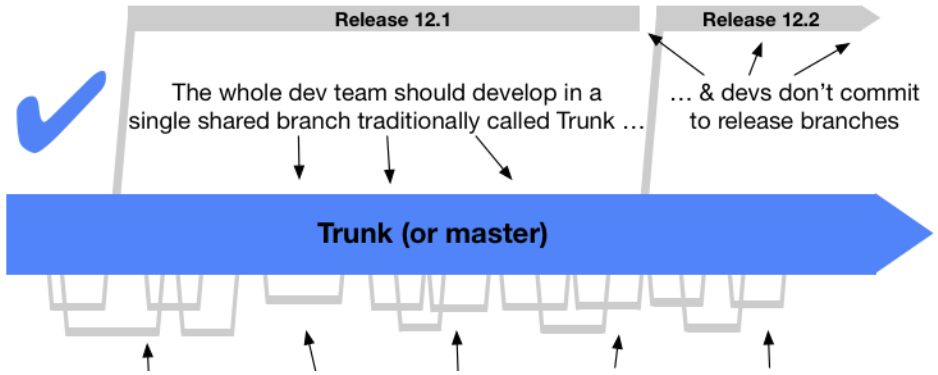
Git flow, trunk based, … da para un debate

Yo: normalmente trunk based con ramas de feature cortas (pero no siempre)

Recomendaciones:

- Un único repositorio código aplicación y CI/CD/IaC
- CI por supuesto con Testing ante cada push/PR





plain concepts

Calidad y testing

¿Acaso dudamos aún de esto? **NO**

Siempre y continuadamente hay que estar probando

“MAKE YOUR BUSINESS GREAT AGAIN”

Thank you

@lfraile

@plainconcepts

www.plainconcepts.com