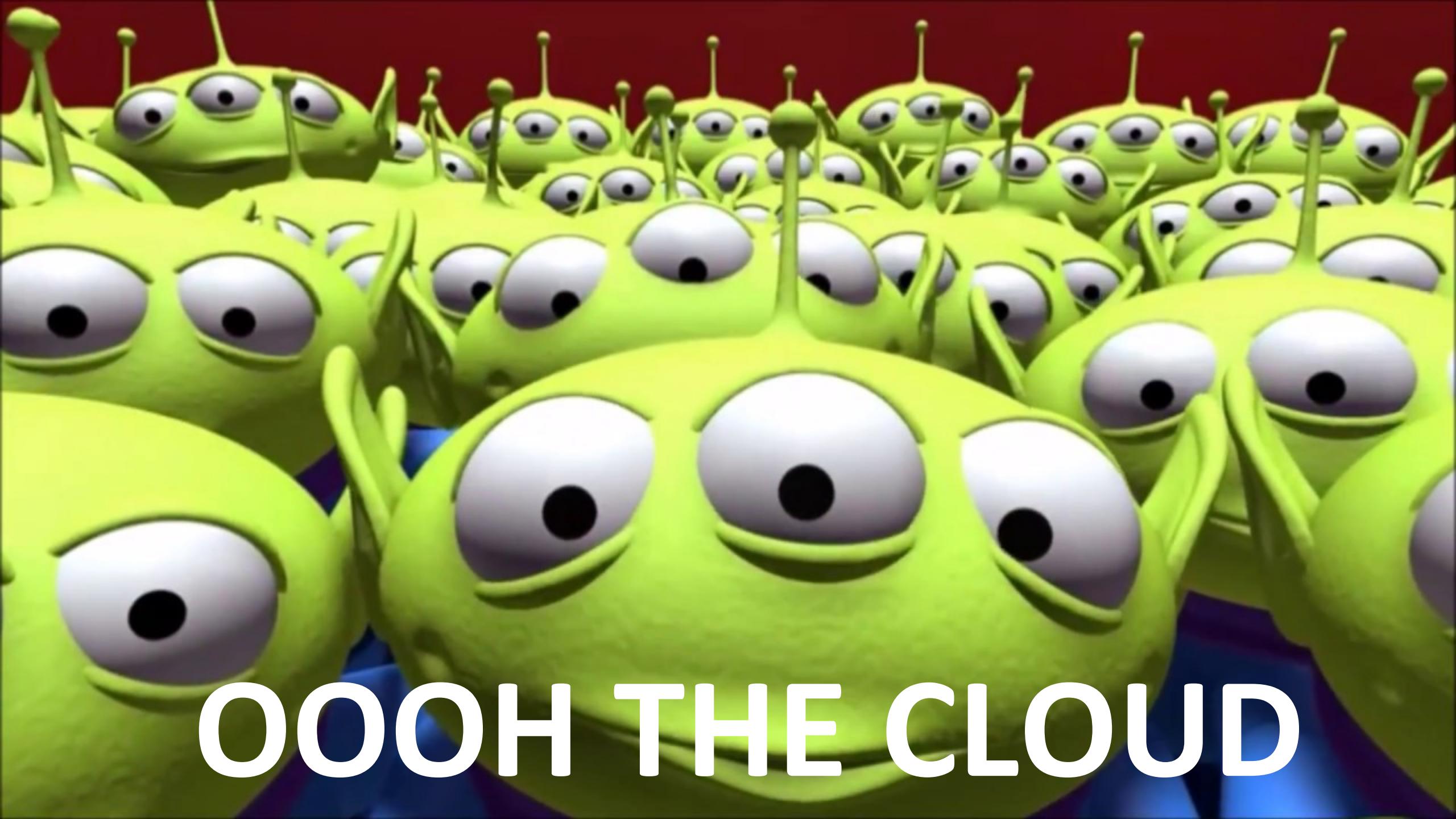




# Platform Engineering 101: Building Internal Developer Platforms

Maarten  
*Red Hat*



OOOOh THE CLOUD

$H_2 = E$

$H_2 = E \left( t_0, 0^{\alpha}, \dots \right)$

$H_2 = E \left( t_0, 1^{\alpha}, \dots \right)$

$\cdot E \left( t_0, \dots \right)$

$H_2 = P E \left( t_0, 0^{\alpha}, \dots \right)$

$\cdot E \left( t_0, \dots \right)$

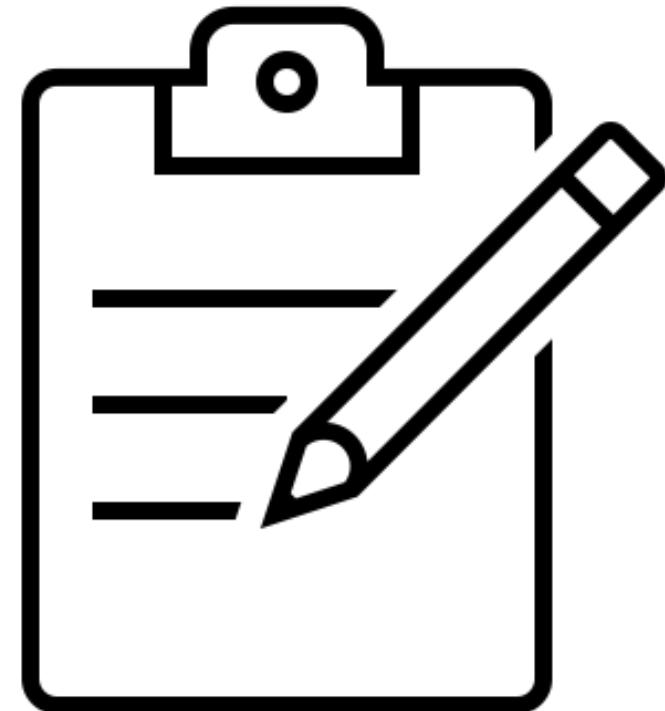
$\cdot P E \left( t_0, \dots \right)$

$G \cdot PLS$



# Agenda

- How Did We Get Here?
- Developer Struggles
- Platform Engineering
- Internal Developer Platform (IDP)
- Internal Developer Portal
- Benefits for Developers
- Tools to Help
- Summary
- Resources



What  
Happened?!



Shift Left:  
Increasing  
cognitive  
load



It's not just  
app development  
anymore

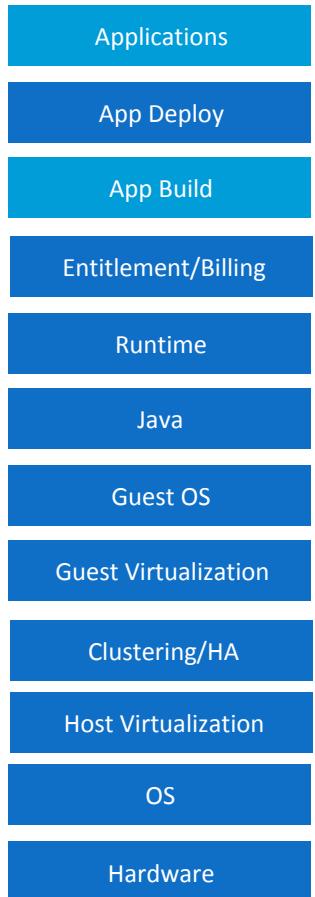
Dev  
Ops  
DevOps  
FinOps  
MLOps  
GitOps  
DataOps  
DevSecOps  
BizDevOps  
BizDevSecOps  
BizDevSecFinMLOps

# Organizational Impact & Cognitive Load

## Traditional App Delivery



Dev



Ops

**Key:**

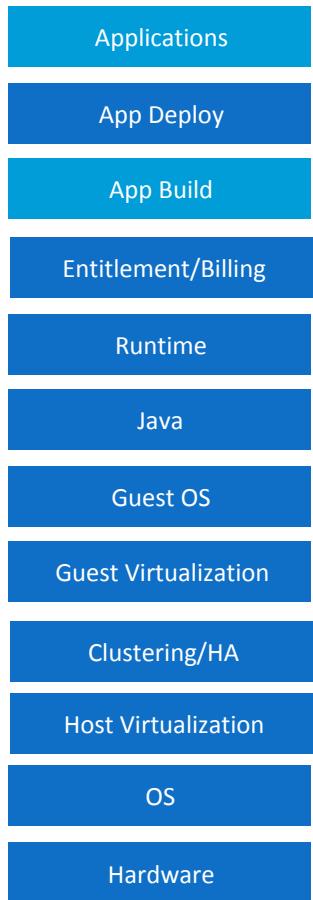
Dev Responsible
Ops Responsible

# Organizational Impact & Cognitive Load

## Traditional App Delivery



Dev



Ops

Key:

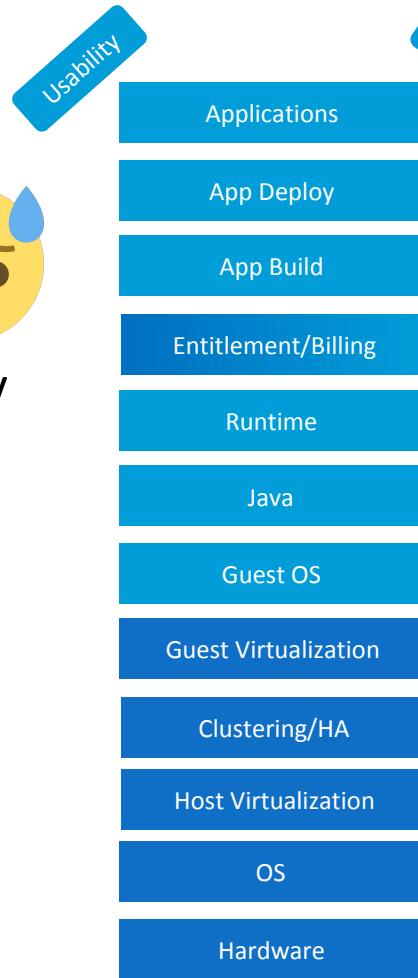
Dev Responsible

Ops Responsible

## Modern App Delivery

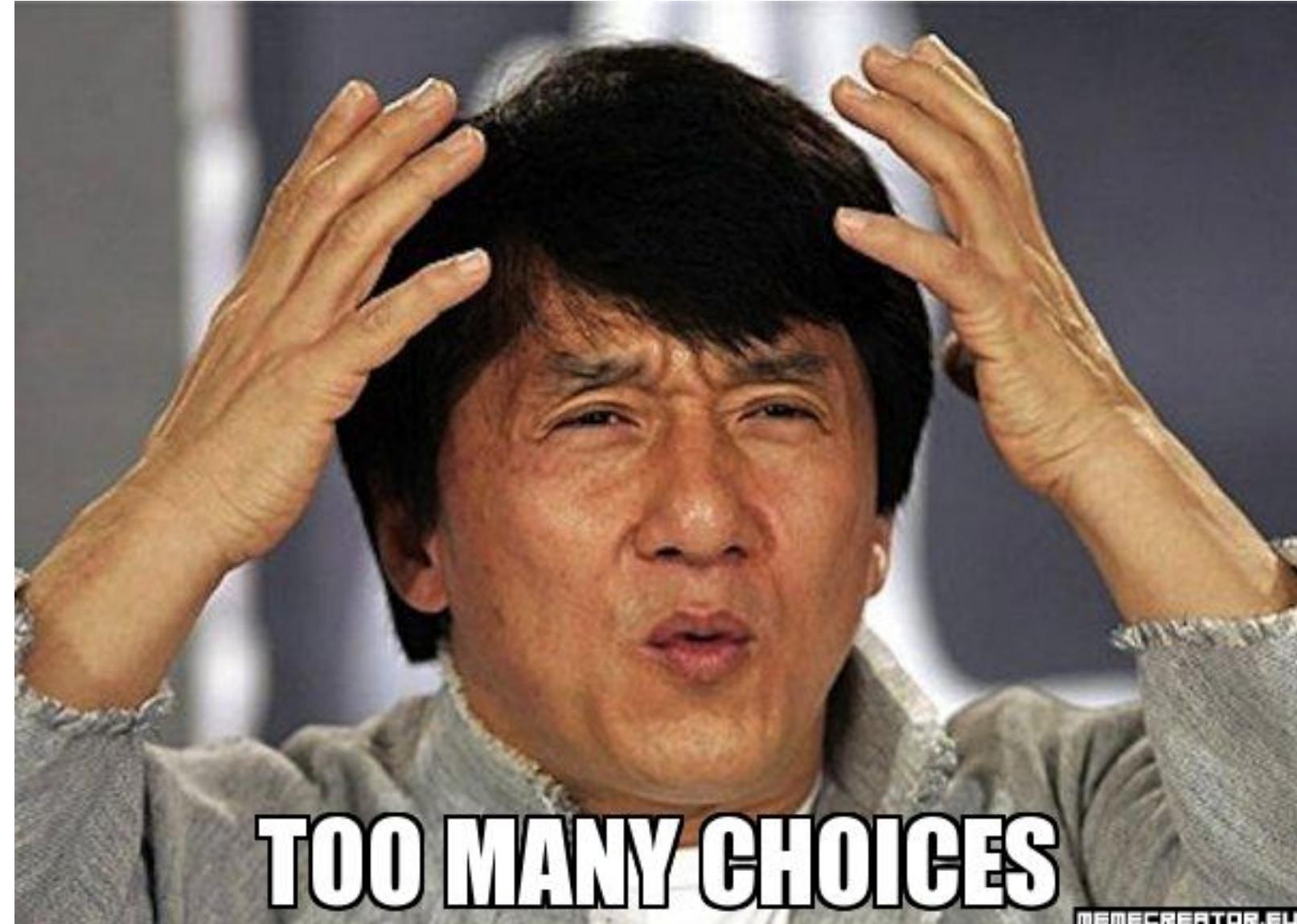


Dev



Ops

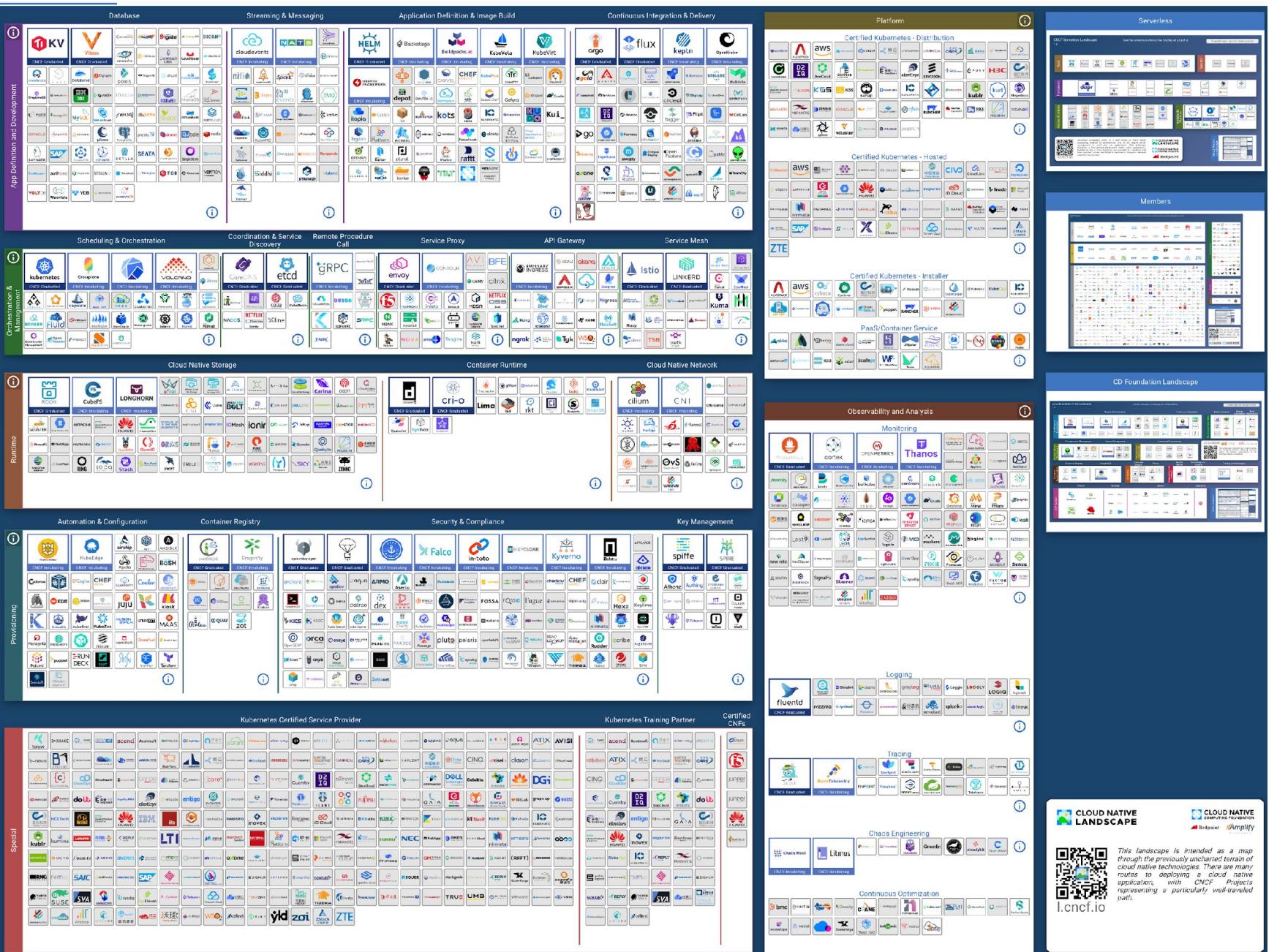
Tools:  
Too much  
choice



# The Cloud-Native Grocery Store

# Where's Waldo?

or Wally?

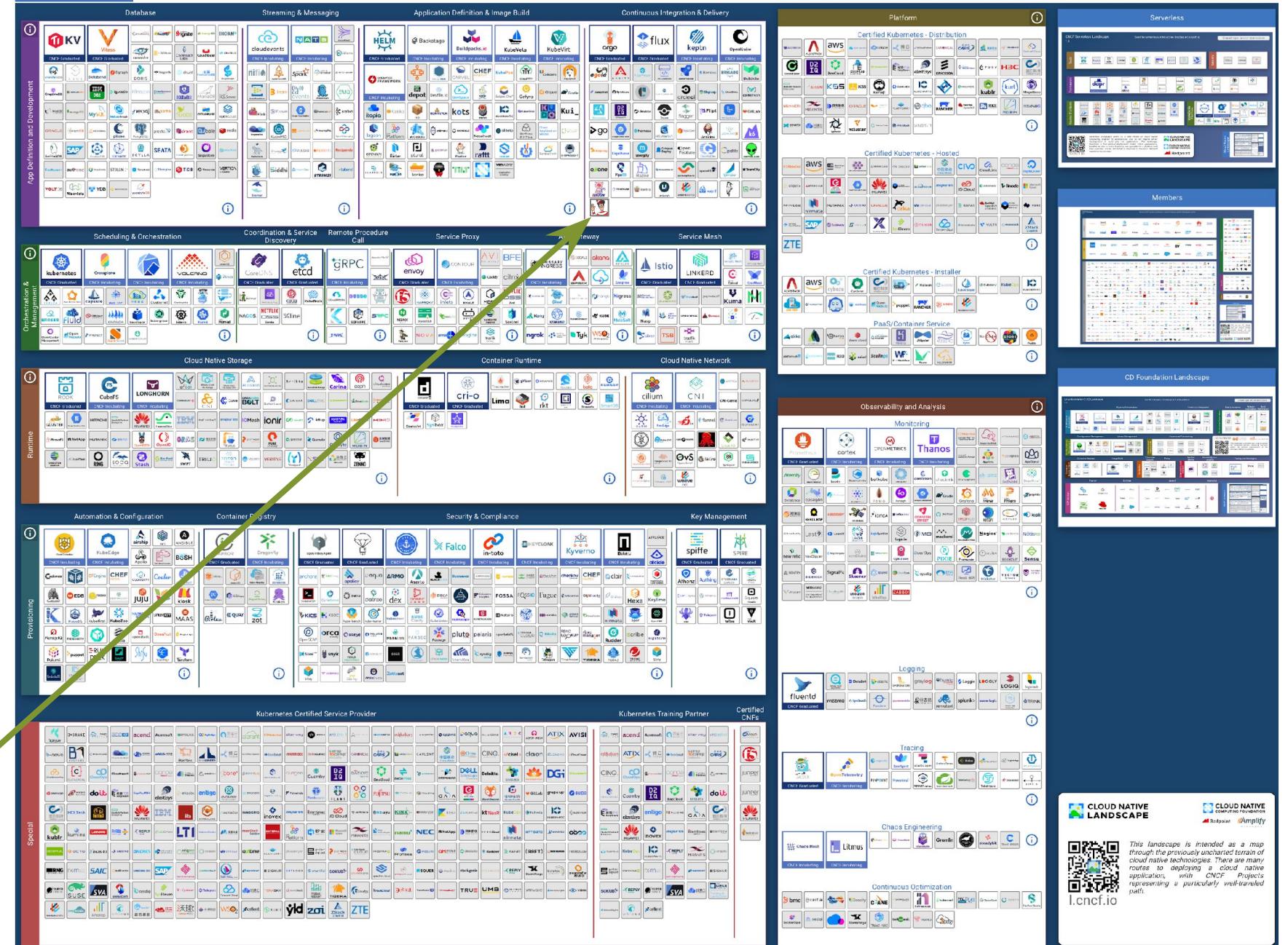


Source: <https://landscape.cncf.io/>

# The Cloud-Native Grocery Store

Where's Waldo?

or Wally?



Source: <https://landscape.cncf.io/>

**CLOUD NATIVE LANDSCAPE**  
SAP | Red Hat | IBM | Amplify  
  
[l.cncf.io](https://landscape.cncf.io)

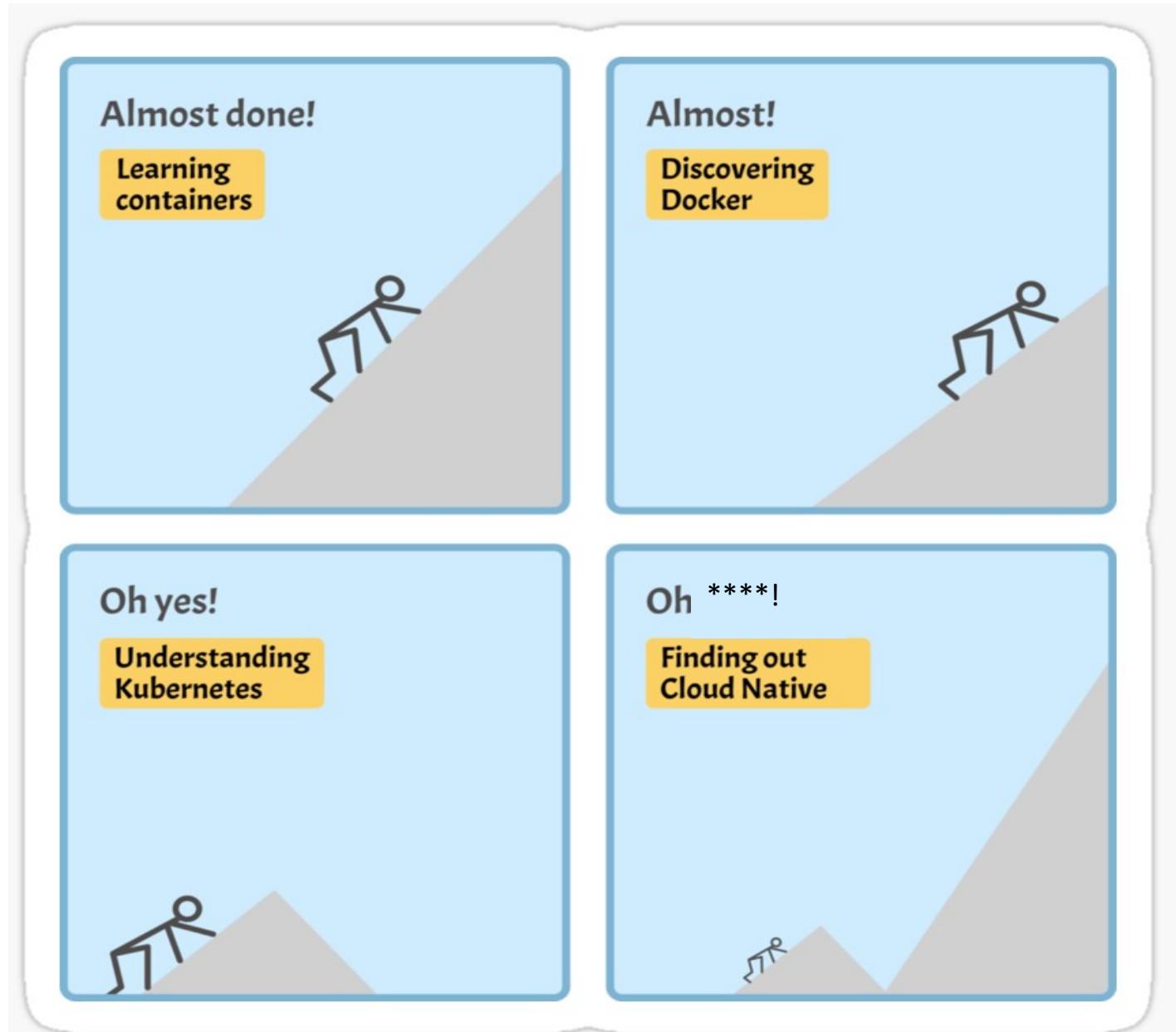
This landscape is intended as a map through the previously uncharted terrain of cloud native technologies. There are many routes to deploying a cloud native application, with CNCF projects representing a particularly well-traveled path.

# Developer Struggles



# 1. CI/CD Tooling

- Steep learning curve – takes time
- Too many options
- OSS standards – still too many!
- Observability



Source: [https://www.reddit.com/r/ProgrammerHumor/comments/lgmi45/adopting\\_cloud\\_native/](https://www.reddit.com/r/ProgrammerHumor/comments/lgmi45/adopting_cloud_native/)

## 2. Container Management

- Offers portability and flexibility
- Many decisions needed
  - Which tools to use?
  - How to configure environments?
  - How to build our apps?
- Challenging to remain consistent and stick to best practices
- Operators & Cloud Native Buildpacks can help



Source: <https://semaphoreci.com/blog/kubernetes-deployment>

### 3. Security & Regulatory Requirements

- Compliance & regulatory awareness
- SBOM –what's in your application?
- Vulnerability awareness
  - Source code
  - Images
  - External libraries
- Trusted software supply chain
- Compromised credentials?
- Authentication & Authorization



Source: <https://www.asylas.com/asylas/security-memes-our-year-end-humor-break/>

# Other Struggles

- Likely drowning in scripts and maintenance costs
- Can't reach degree of automation wanted
- Key-person dependency
- Creating a new environment can be loooooong
- Environments can get blocked testing



# What developers are saying....

Top three causes of developer burnout:

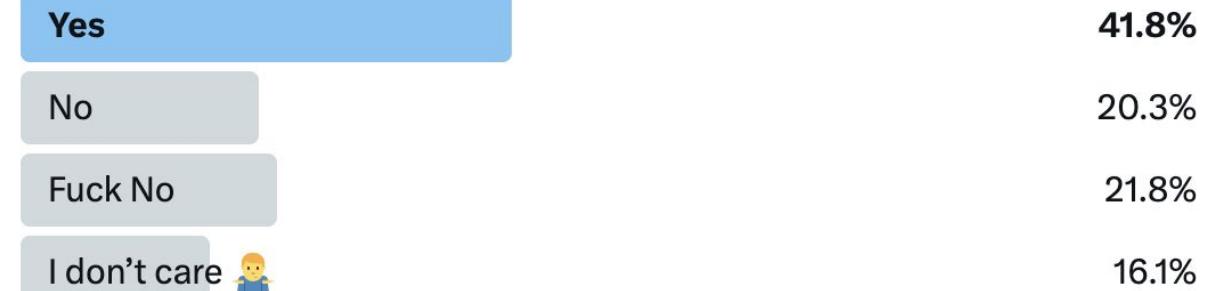
1. Increasing workload/demand from other teams (39%)
2. Pressure of digital transformation (37%)
3. Requirement to learn new skills (35%)

76% of organizations say cognitive load is so high it causes angst and low productivity for developers.



Luca ✅ @luca\_cloud · Aug 24, 2022

Devs... be real with me. Do y'all want to do Ops? 🧵



330 votes · Final results

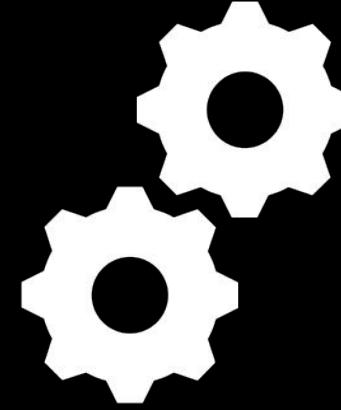
Replies to @luca\_cloud

[https://twitter.com/intent/tweet?ref\\_src=twsrctf%7Ctwcamp%5Etweetembed%7Ctwterm%5E1562349679660122112%7Ctwgr%5Ea1005affb9784baa8ed6ee07c455995f8ad267df%7Ctwcon%5Es1\\_&ref\\_url=https%3A%2F%2Fblog.gitguardian.com%2Fplatform-engineering-and-security-a-very-short-introduction%2F&in\\_reply\\_to=1562349679660122112](https://twitter.com/intent/tweet?ref_src=twsrctf%7Ctwcamp%5Etweetembed%7Ctwterm%5E1562349679660122112%7Ctwgr%5Ea1005affb9784baa8ed6ee07c455995f8ad267df%7Ctwcon%5Es1_&ref_url=https%3A%2F%2Fblog.gitguardian.com%2Fplatform-engineering-and-security-a-very-short-introduction%2F&in_reply_to=1562349679660122112)

# But... How to overcome these struggles?



# Platform Engineering



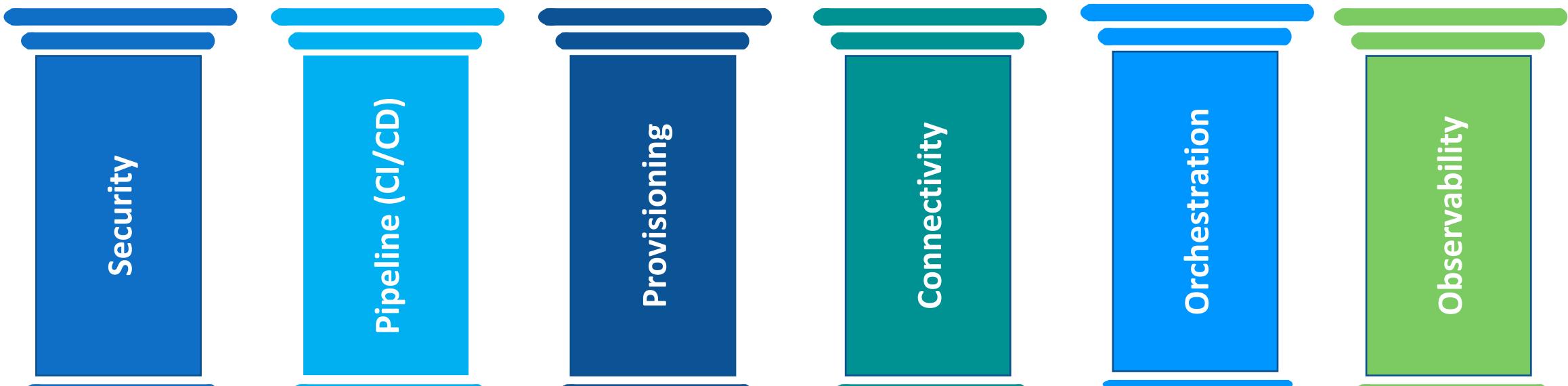
# What is Platform Engineering?

**Platform engineering** is 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.

- platformengineering.org

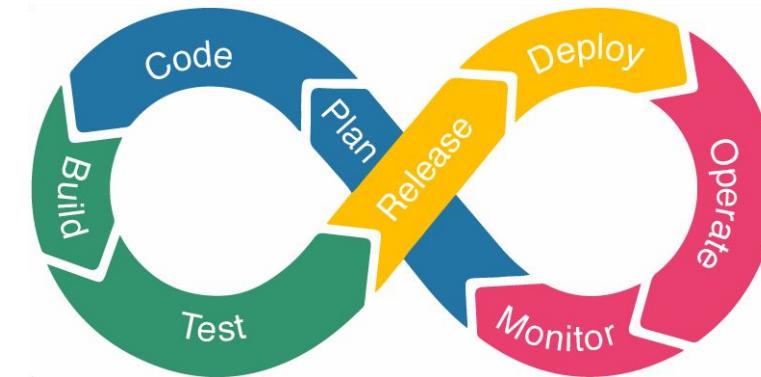


# Platform Engineering Pillars

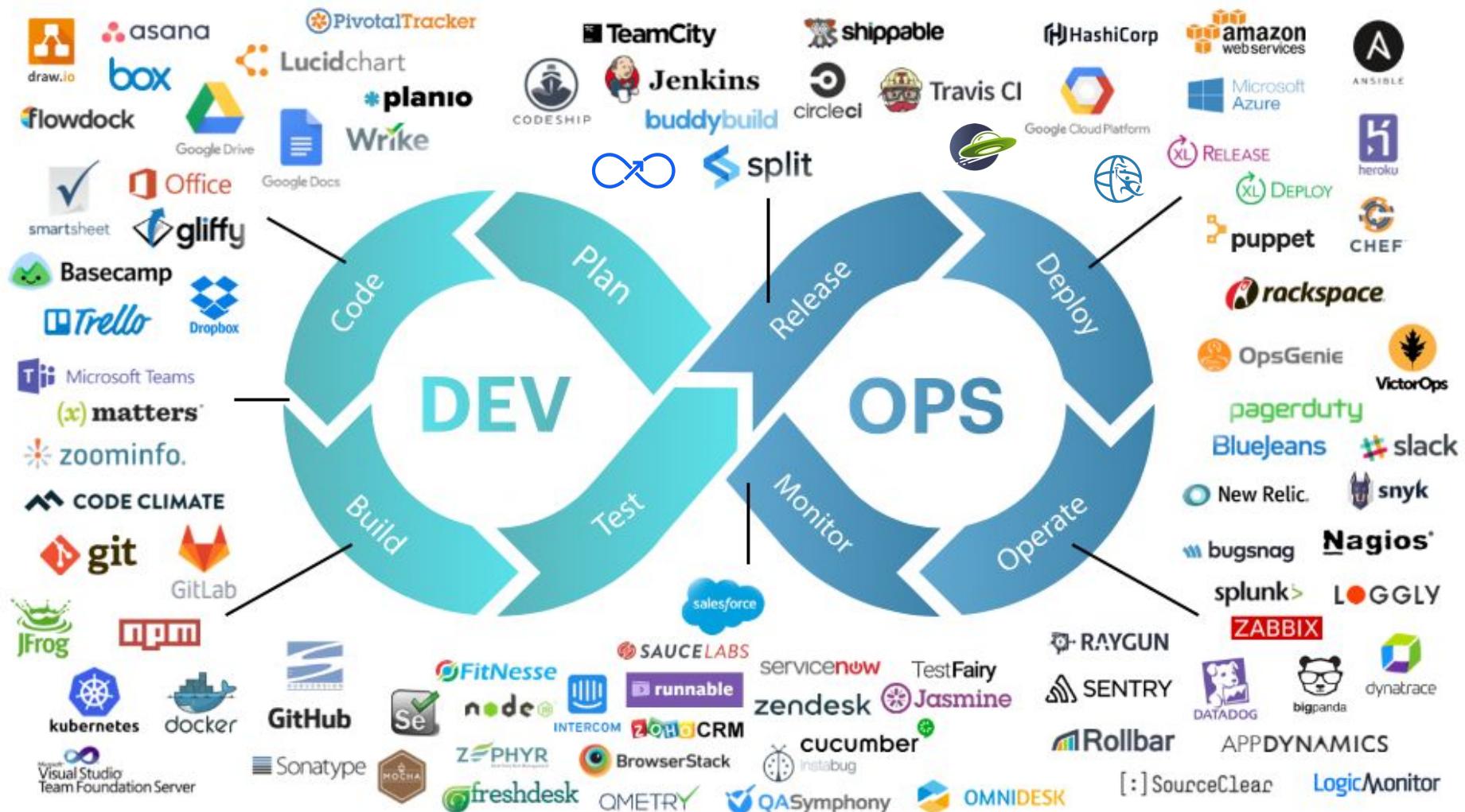


# The Platform Engineering recipe - Ingredients

- Integrated Development Environments
- Version Control Systems
- Resources - containers, databases, storage, compute, transit, etc.
- Runtimes and frameworks
- Artefact, container registry
- Kubernetes
- Observability
- Vulnerability scanners
- CI tools
- GitOps/CD tools
- Infrastructure as Code tools

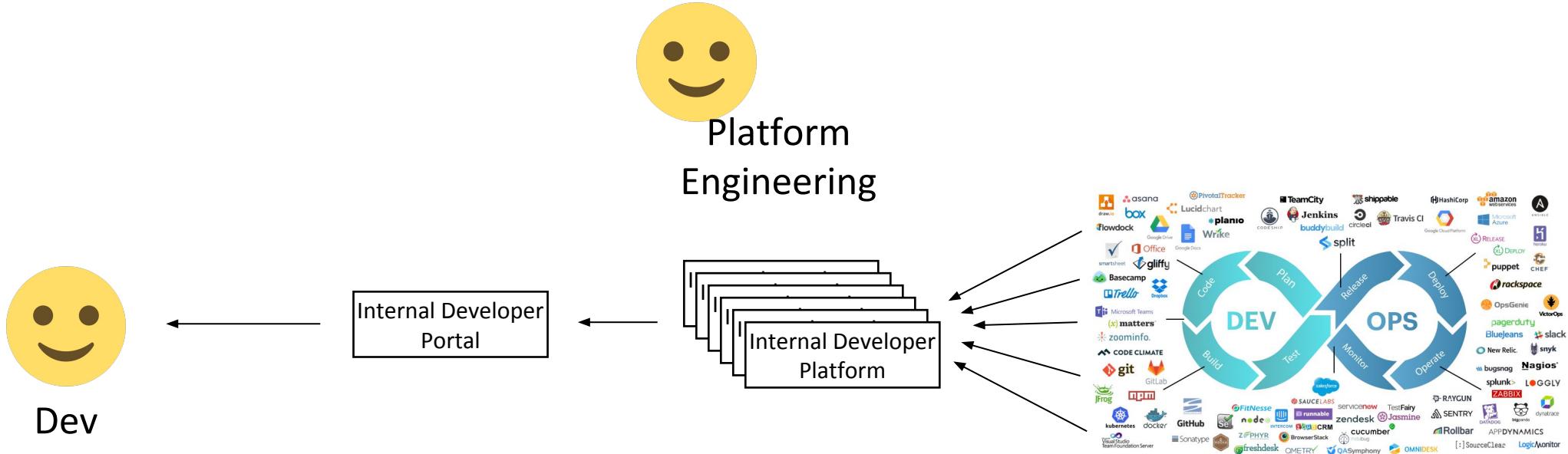


# Bake your own Platform

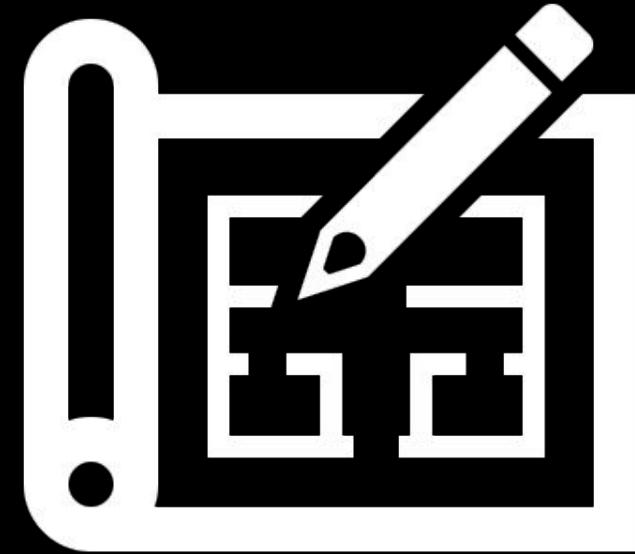


Source: <https://www.openxcell.com/devops/>

So, what does this look like?



Internal  
Developer  
Platform (IDP)



## What is an IDP?

An Internal Developer Platform (IDP) is a **central collection of tools, services and automated workflows** that enable development teams to be able to rapidly deploy and deliver software. These platforms **provide a service layer** that abstracts away the complexities of infrastructure management and application configuration.



# Internal Developer Portal



## What is an Internal Developer Portal?

An Internal Developer Portal is the **singular interface** to your developer platform - serves as the "face" of the Internal Developer Platform.

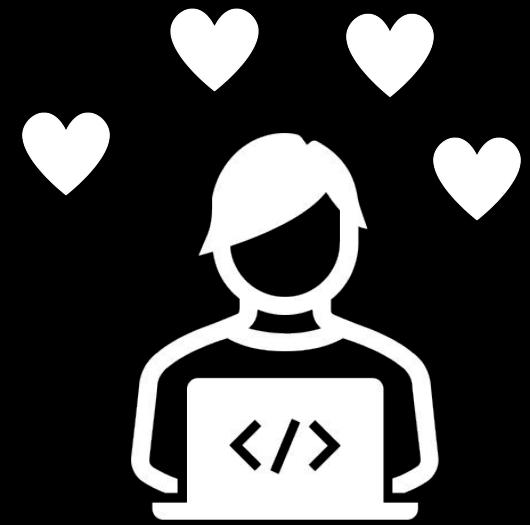
It allows developers to perform self-service operations and actions, and offers visibility into the infrastructure.



Internal Developer Platform  
≠  
Internal Developer Portals



# Benefits for Devs



# Organizational impact



Dev

Applications



Platform Engineering

App Build  
App Deploy  
Entitlement/Billing  
Runtime  
Java  
Guest OS



Ops

Entitlement/Billing  
Guest Virtualization  
Clustering/HA  
Host Virtualization  
OS  
Hardware

Internal Developer  
Platform  
IDP

# Quantitative Dev Benefits

Impact	Before	After
Waiting times due to blocked environments decreased by 90%	4 hours/week per developer	24 minutes/week per developer
Average Deployment Frequency (up an average of 4X)	1.5 per week	6 per week
Visibility and transparency across teams, services and environments decreases transactional communication	Limited to your 15 minute daily scrum	Direct reaction based on activity of colleagues.
Onboarding time for new developers to delivery setup in hours	30 hours	4 hours
Lead time	13 days	4 days

<https://humanitec.com/blog/impact-of-internal-developer-platforms>

## Good

- Increased developer autonomy
- Reduced time to market
- Improved developer satisfaction

## Challenges

- Precise user authentication
- Building a scalable team
- Building self-serving solutions
- Ensuring alignment and standardization
- Compliance with security regulations

## Bad

- Strain on platform engineering teams
- Difficulty managing standardization of configurations/tools

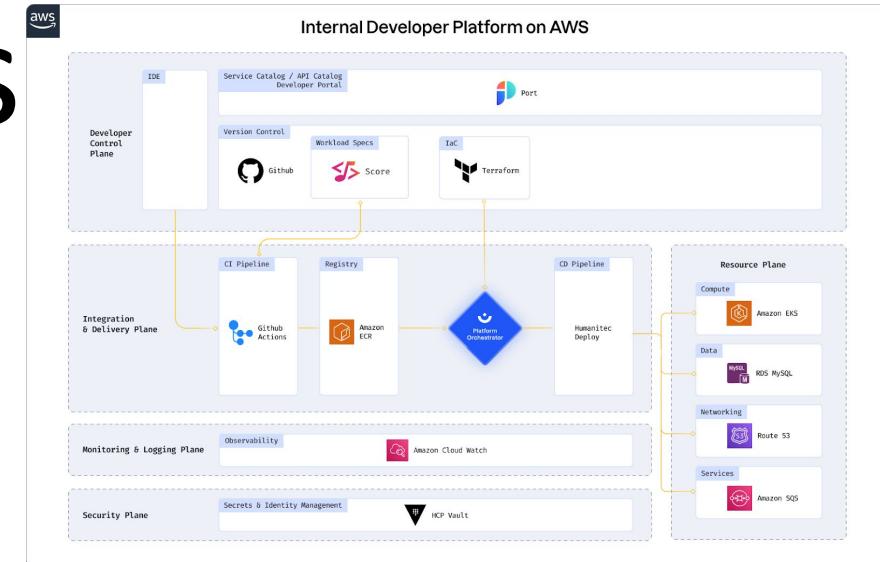
# Tools to Help



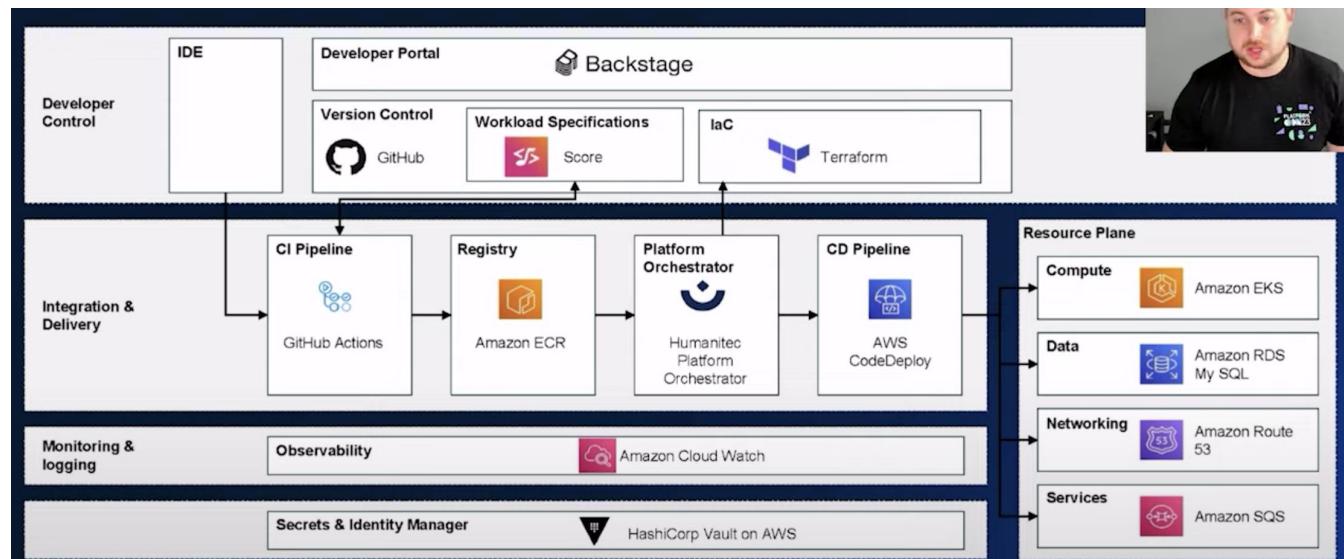
# Reference architectures

Architecture is divided into subgroups

- Developer control plane
- Integration and delivery
- Monitoring and logging
- Security plane
- Resource plane

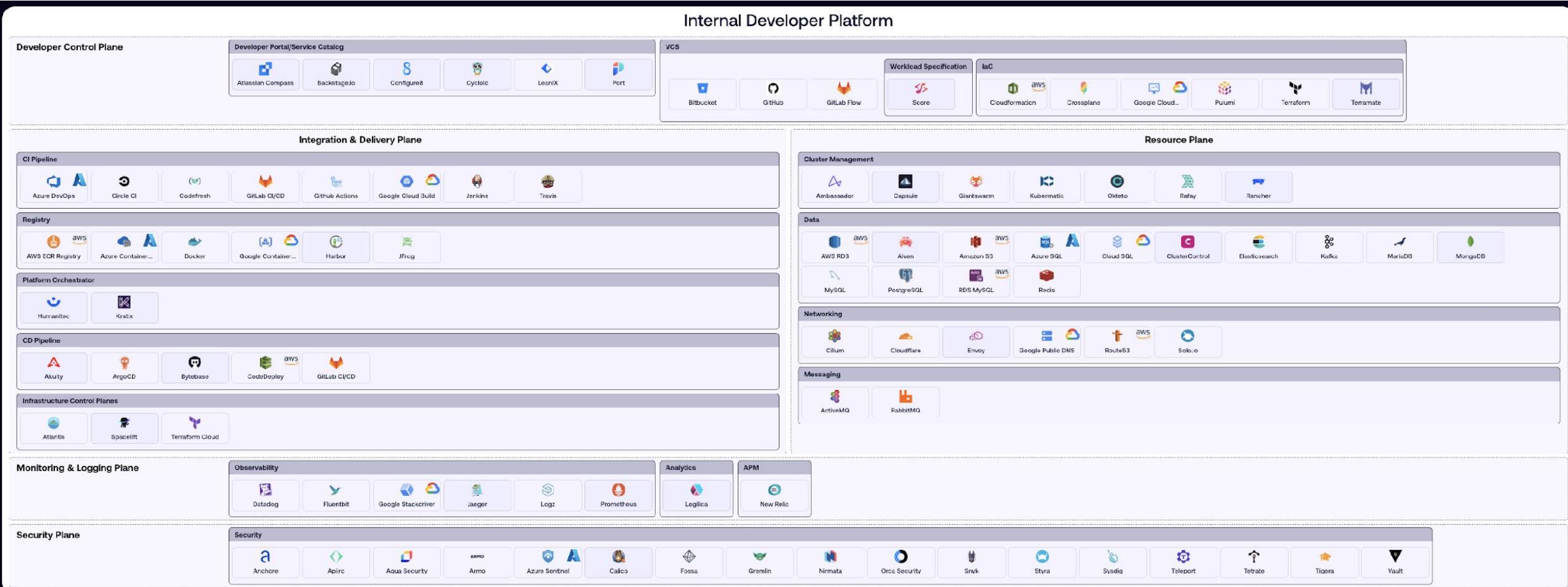


<https://humanitec.com/reference-architectures>



McKinsey's Internal Developer Platform reference architecture

# IDP Tooling Landscape

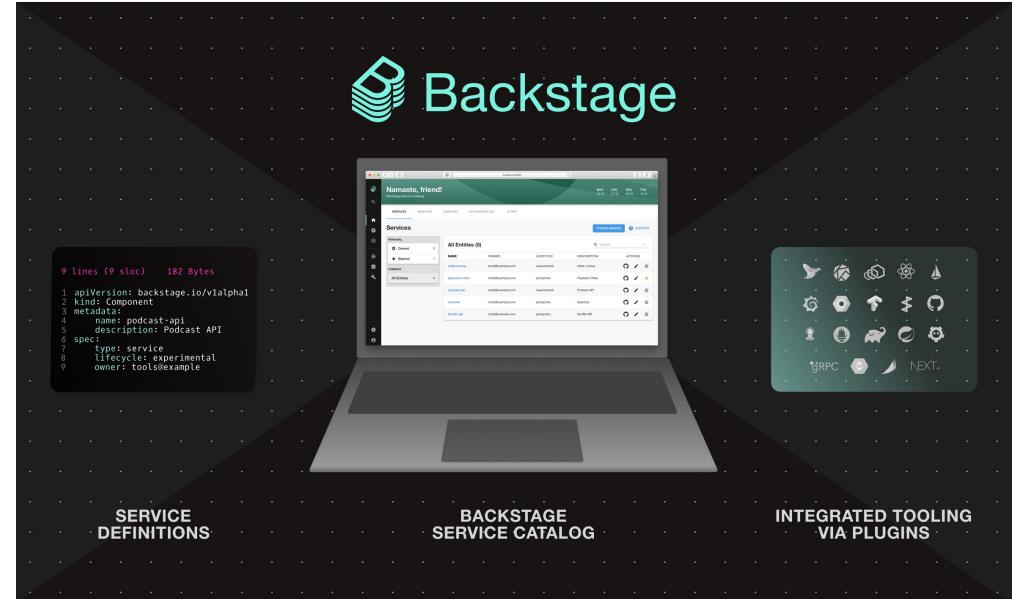


# Backstage



<https://backstage.io/>

- CNCF incubating project
- Donated by Spotify
- An open platform for building developer portals
- Powered by a centralized software catalog
- Can customize with plugins

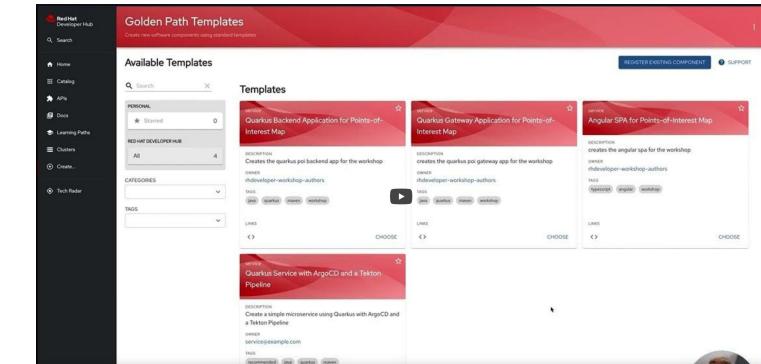


# Red Hat Developer Hub

- Enterprise-grade, open developer platform for building developer portals
- Contains a supported and opinionated framework
- ~~Built on the Janus open-source project (which extends Backstage)~~
- Extends functionality through a set of predefined supported dynamic plug-ins



<https://developers.redhat.com/rdh>



[https://www.youtube.com/watch?v=tVOC0mFR\\_4&t=182s](https://www.youtube.com/watch?v=tVOC0mFR_4&t=182s)

Examples: <https://github.com/maarten-vandeperre/developer-hub-documentation>

# Demo

# Let IDPs guide you toward the future



# Summary



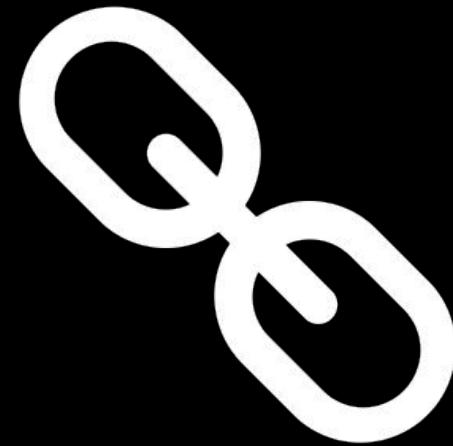


# Summary



- Cognitive load and demands on developers are ever-increasing as we move towards cloud-native development
- Developers can be supported and helped through...
  - Platform Engineering
  - Internal Developer Platforms (IDP)
  - Internal Developer Portals
  - Golden/Pathed Paths
- What this looks like will be specific to your organization
- Use the tools, technologies and examples already out there

# Resources



# Examples

- Netflix - <https://www.youtube.com/watch?v=36FcxlPerdQ>
- Humanitech - <https://www.youtube.com/watch?v=Ky8LWwU2tt8>
- Wise -  
<https://platformengineering.org/talks-library/adopting-backstage-as-developer-portal>
- Spotify -  
<https://engineering.atspotify.com/2020/04/how-we-use-backstage-at-spotify/>
- Project Polaris at Snyk -  
<https://www.getambassador.io/kubernetes-expert-interviews/crystal-hirschorn> or <https://getdx.com/podcast/snyk-developer-experience>
- Twilio and Snyk -  
<https://www.getport.io/webinars/how-twilio-and-snyk-improve-the-developer-experience>
- Veterans United Home Loans

# Resources

- Platform Engineering Guide (InfoQ)
  - <https://www.infoq.com/minibooks/platform-engineering-guide/>
- PlatformCon (previous edition's recordings available)
- Platform Engineering Slack channel
  - <https://platformengineering.org/slack-rd>
- Platform Weekly newsletter
  - <https://platformweekly.com/>



# Resources

- Benefits -  
<https://digital-platform.playbook.ee/introduction/benefits-of-a-digital-platform>
- Internal Developer Platform vs Internal Developer Portal -  
<https://www.configure8.io/blog/internal-developer-portal-vs-internal-developer-platform-whats-the-difference-and-why-both-matter>
- Tools for IDP creation -  
<https://medium.com/contino-engineering/creating-your-internal-developer-platform-part-2-65ff217cecd6>
- Why IDPs matter -  
<https://developers.redhat.com/articles/2024/09/25/why-internal-developer-portals-matter#>



# Resources

- Trying it out yourself (for free)

- <https://developers.redhat.com/>
- <https://developers.redhat.com/learn/openshift/deploy-red-hat-openshift-using-helm-charts>
- <https://openliberty.io/start/>
- <https://github.com/OpenLiberty/liberty-backstage-template/tree/main>
- <https://github.com/maarten-vandeperre/developer-hub-documentation/tree/main>
- <https://github.com/grace-maarten/platform-engineering-101>

# THANK YOU



<https://www.linkedin.com/in/grace-jansen/>

<https://www.linkedin.com/in/maarten-vandepperre-8780743b/>

