



Intro to DevOps and Beyond

Ravindu Nirmal Fernando





Ravindu Nirmal Fernando

<https://ravindunfernando.com>

About Me

- STL – DevOps @ Sysco LABS – Sri Lanka
- MSc in Computer Science specialized in Cloud Computing (UOM)
- AWS Certified Solutions Architect – Professional
- Certified Kubernetes Administrator (CKA)
- AWS Community Builder



The Era before DevOps



Developers
Focused on Agility



Operators
Focused on Stability

"Destructive downward spiral in IT" – Gene Kim



Act 01 – Operations teams maintaining large fragile applications

Doesn't have any visibility on the application, whether or not its working as expected



Act 02 – The product managers

Larger, unrealistic commitments made to the outside world (client/investors) without understanding the complexities behind development and operations



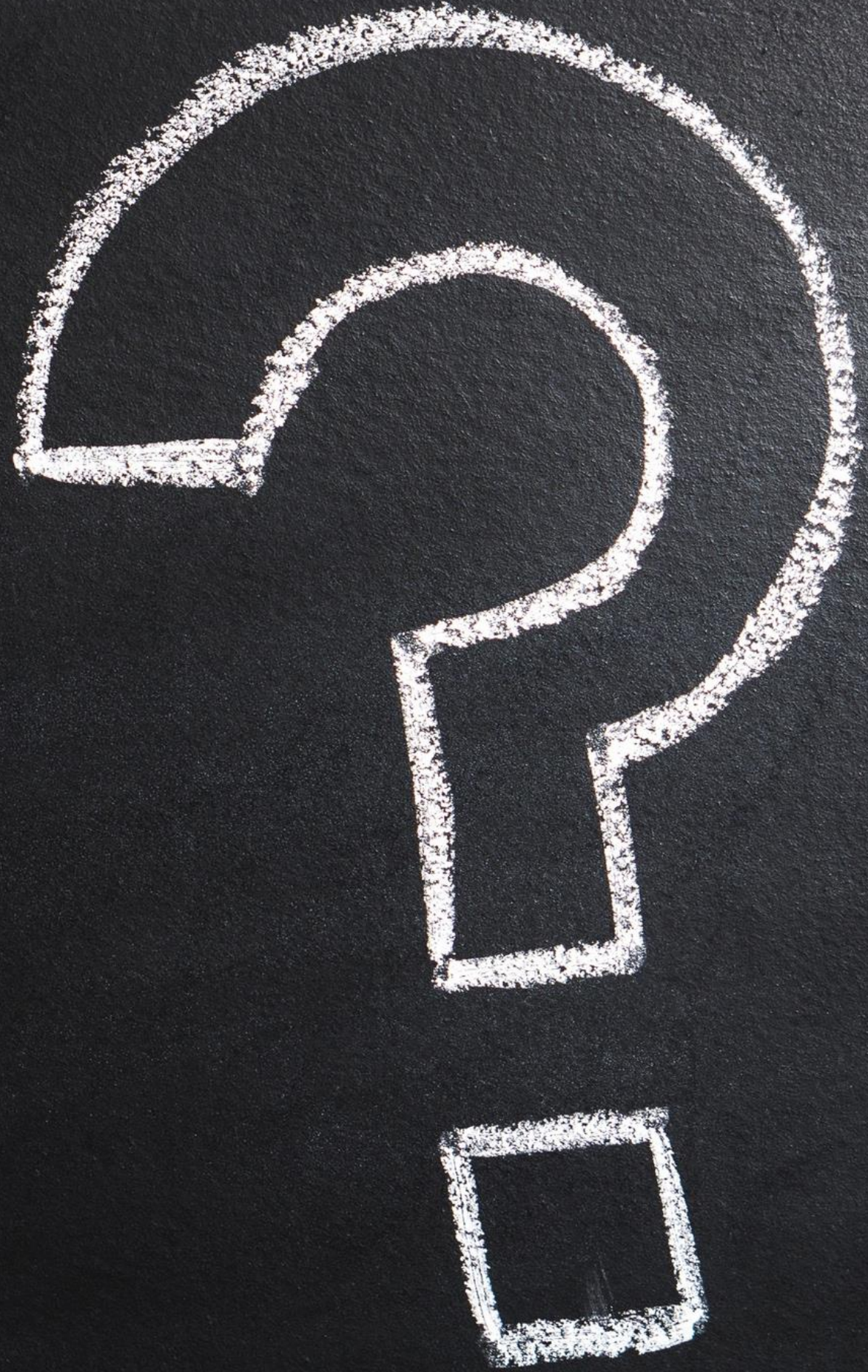
Act 03 – The Developers

Developers taking shortcuts and putting more and more fragile code on top of existing ones



Act 04 – Dev and Ops at war

"It worked on my machine" phenomenon



**How can we
overcome
these issues?**

“DevOps is the combination of cultural philosophies, practices, and tools that increases an organization’s ability to deliver applications and services at high velocity”

– What is DevOps? [AWS] –

“A compound of development (Dev) and operations (Ops), DevOps is the union of people, process, and technology to continually provide value to customers.”

– What is DevOps? [Azure] –

DevOps allows evolving and improving products at a faster pace than businesses using traditional software development and infrastructure management processes. This speed allows businesses to serve their customers better and compete effectively.

Key Areas in DevOps



Reduce Organizational Silos

Everyone's shares the ownership of production and information is shared among everyone



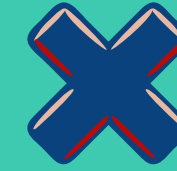
Implement Gradual Changes

Frequent deployments, frequent deterministic releases in small chunks which can be rolled back



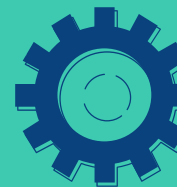
Measure Everything

Application, systems monitoring and metrics etc...



Accept Failure as Normal

Blameless PMs/ RCA. Risk taking mindset.



Leverage Tooling and Automation

Automate and reduce manual work as much as possible



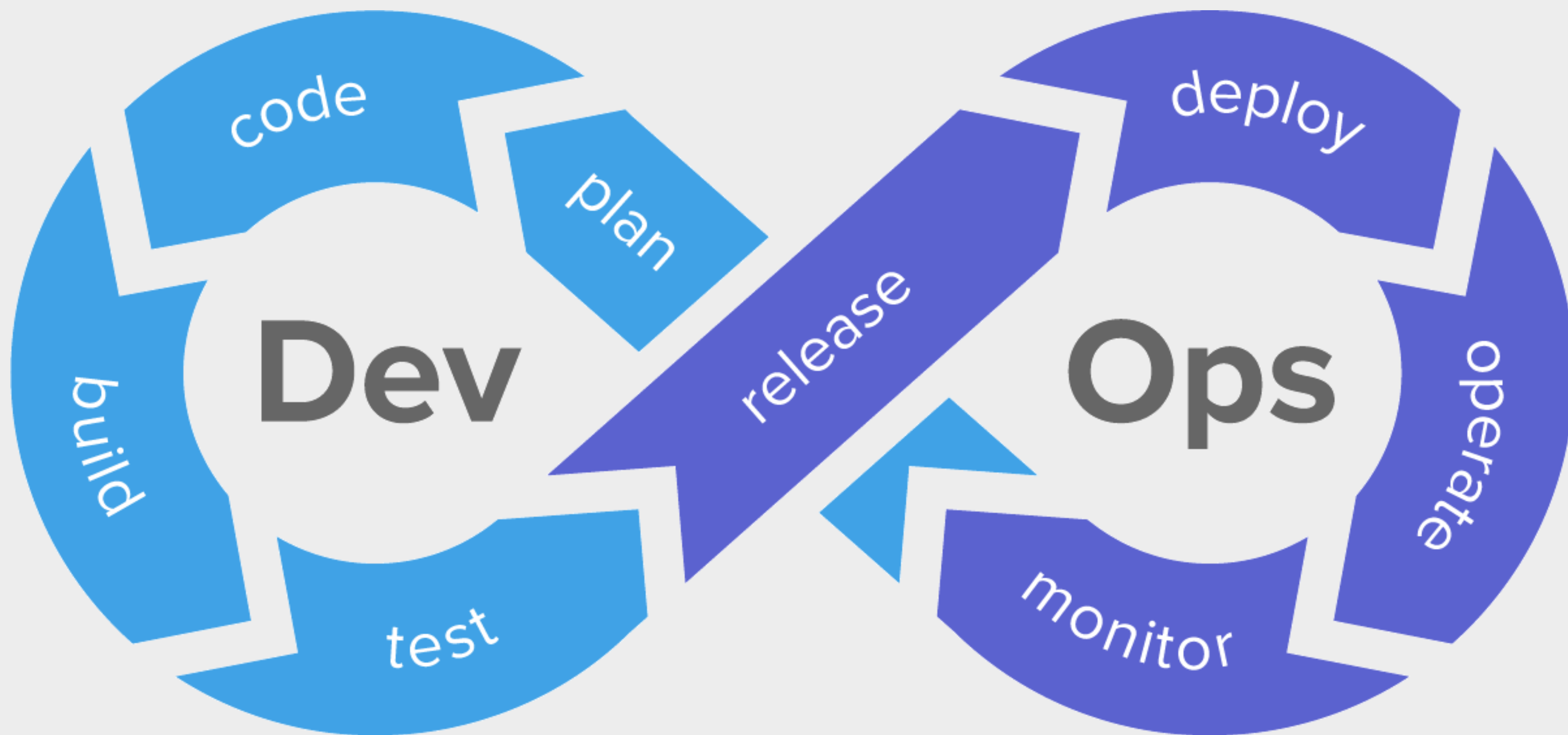
DevOps Practices

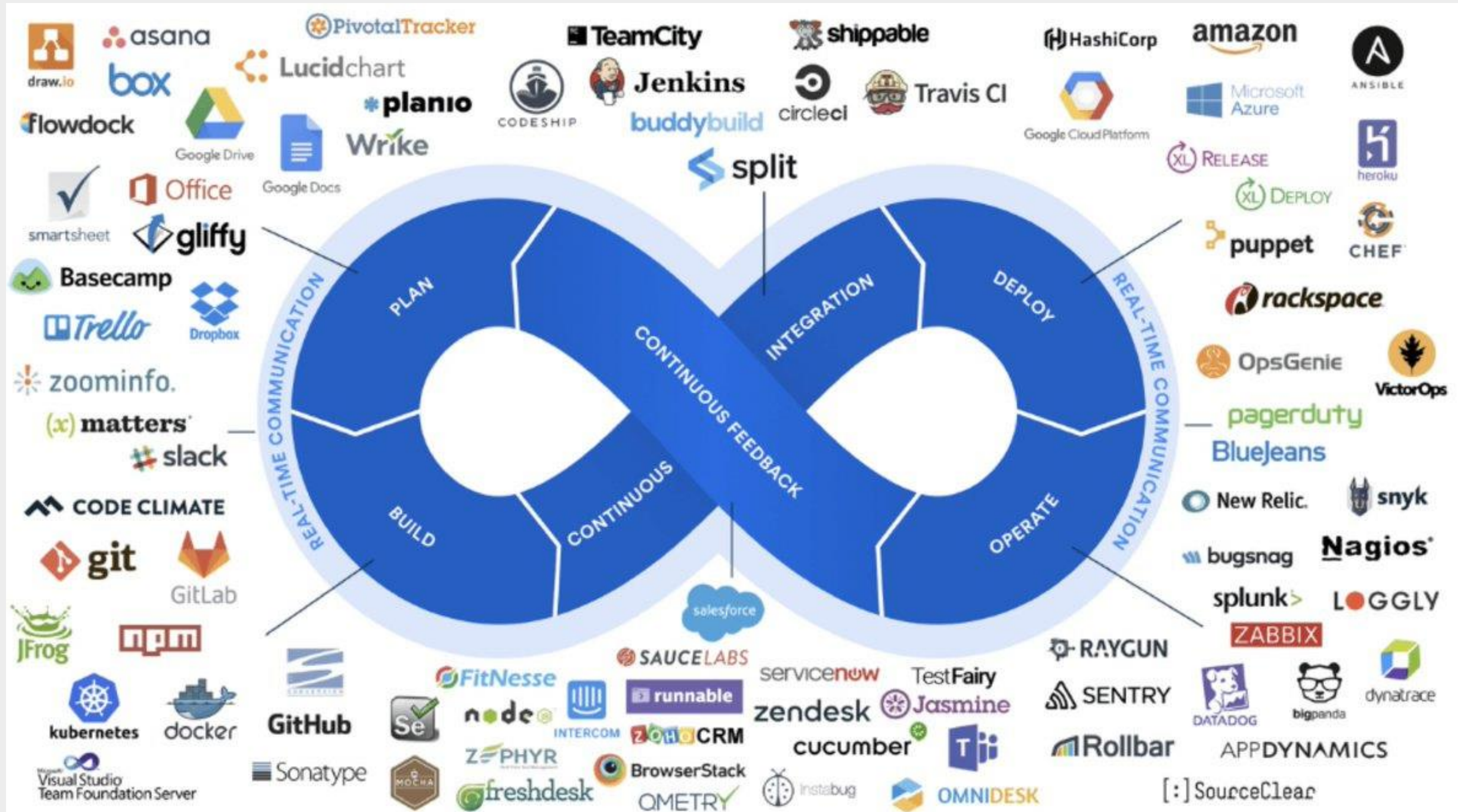
- Continuous Integration (CI) – Software development practice where developers regularly merge their code changes into a central repository, after which automated builds and tests are run.
- Continuous Delivery (CD) – Software development practice where code changes are automatically built, tested, and prepared for a release to production (automated code change deployment to staging/ pre-production system).
- Continuous Deployment (CD) – Every change that passes all stages of the pipeline will be deployed into production (released to customers). This practice fully automates the whole release flow without human intervention and only a failed test will prevent a new change being deployed.
- Microservices – The microservices architecture is a design approach to build a single application as a set of small services with each focusing on SRP. Each service can be created, deployed and run independently.

- Infrastructure as Code – A practice in which infrastructure is provisioned and managed using code and software development techniques, such as version control and continuous integration.
 - Configuration Management
 - Policy as Code
- GitOps – builds on the concept of IaC, incorporating the functionality of Git repositories, merge requests (MRs) and CI/CD to further unify software development and infrastructure operations. GitOps incorporates managing both infrastructure and applications as code.
- Cloud Infrastructure – Cloud provides more flexibility, scalability and toolsets for organizations to implement DevOps culture and practices. Serverless architecture in cloud brings down the efforts of DevOps teams as it eliminates server management operations.
- Continuous Monitoring, Logging and Alerting – Organizations monitor metrics and logs to see how application and infrastructure performance impacts the experience of their product's end user. Combined with real time alerts organizations can do a real time analysis on the application status.



DevOps Tools and Technologies



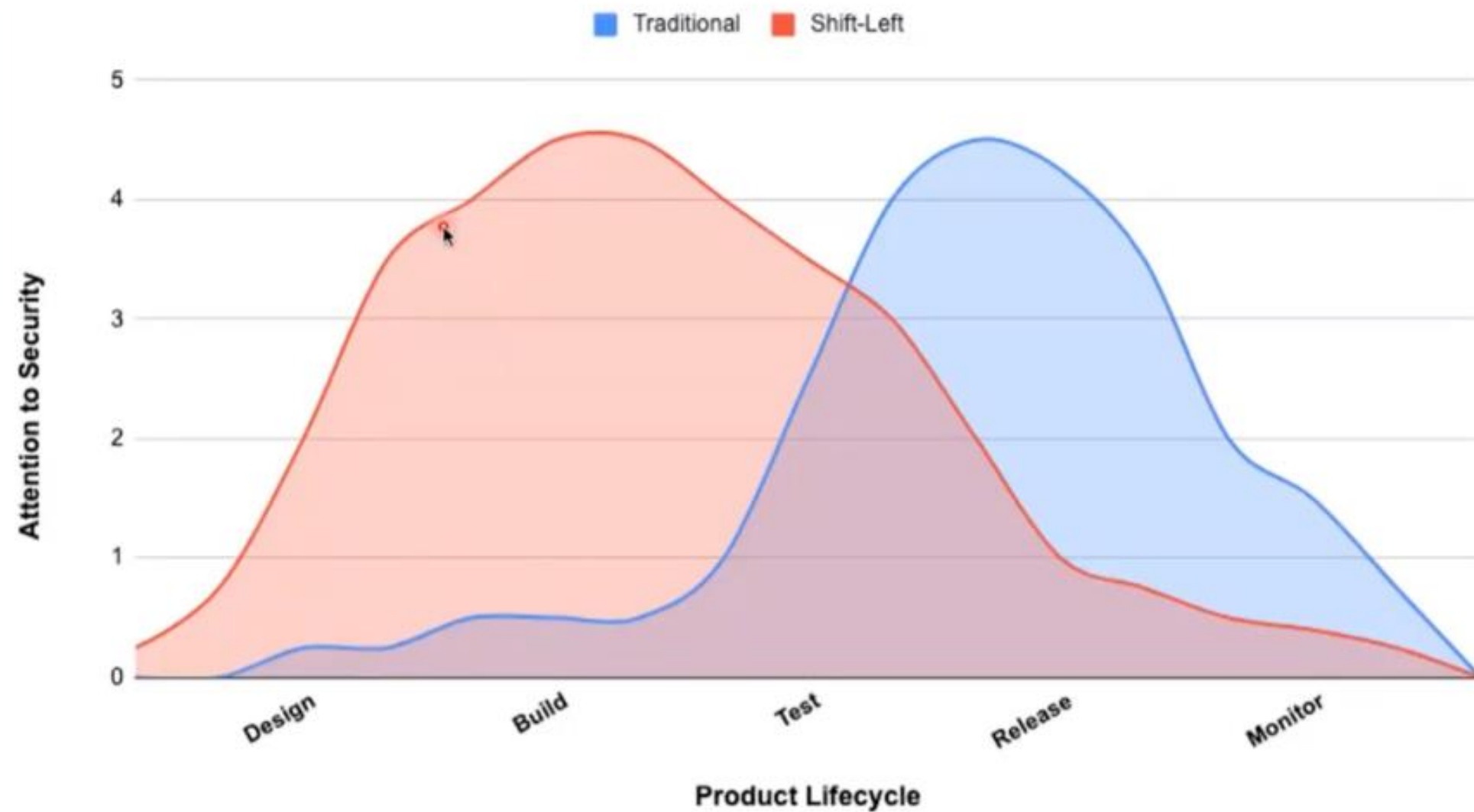




Beyond DevOps

DevSecOps

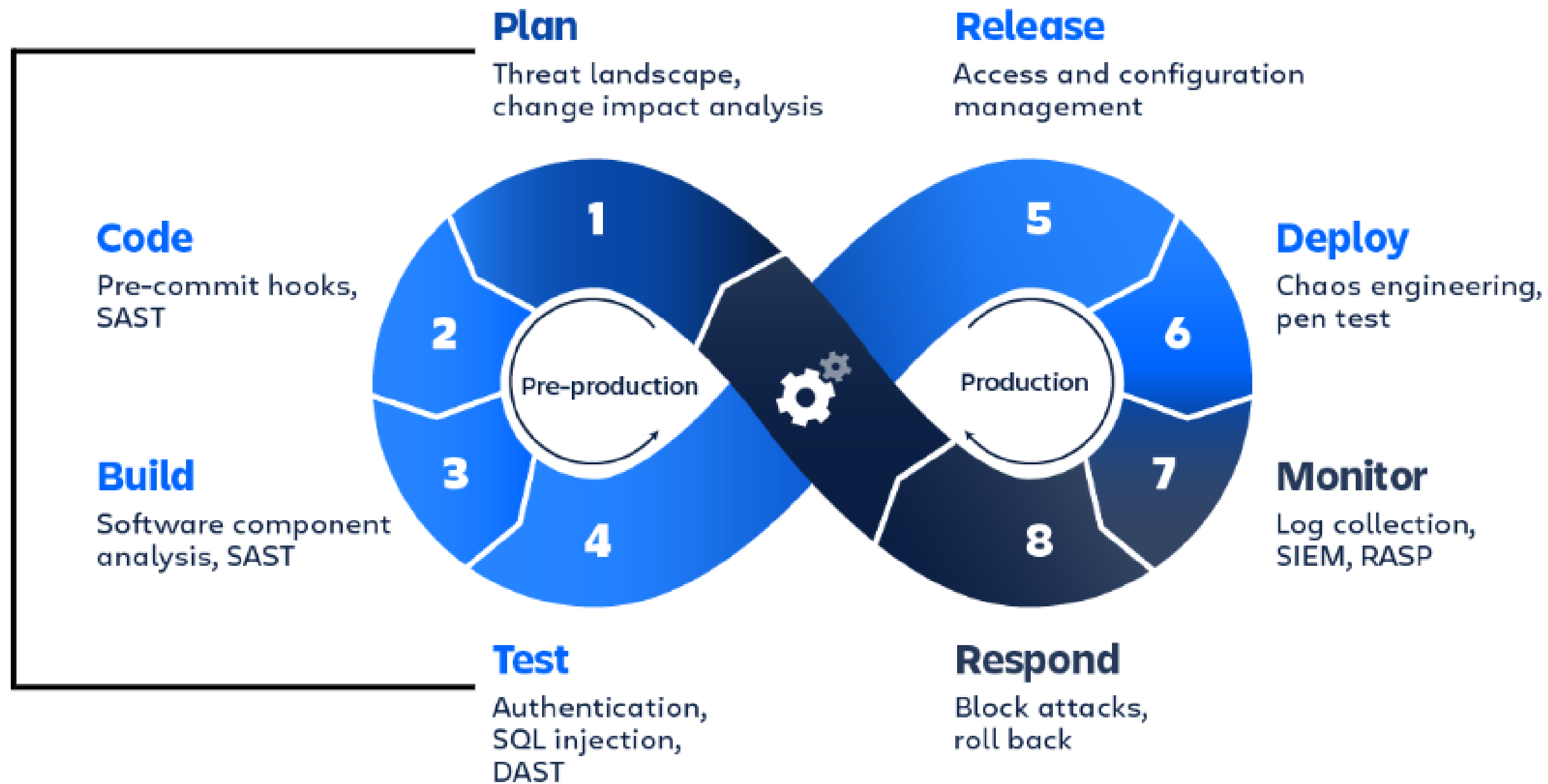
Traditional vs Shift-Left Security Model



Idea of moving Security in the early stages of the SDLC pipeline

"the practice of integrating security into a continuous integration, continuous delivery, and continuous deployment pipeline"

DevSecOps



SRE (Site Reliability Engineering)

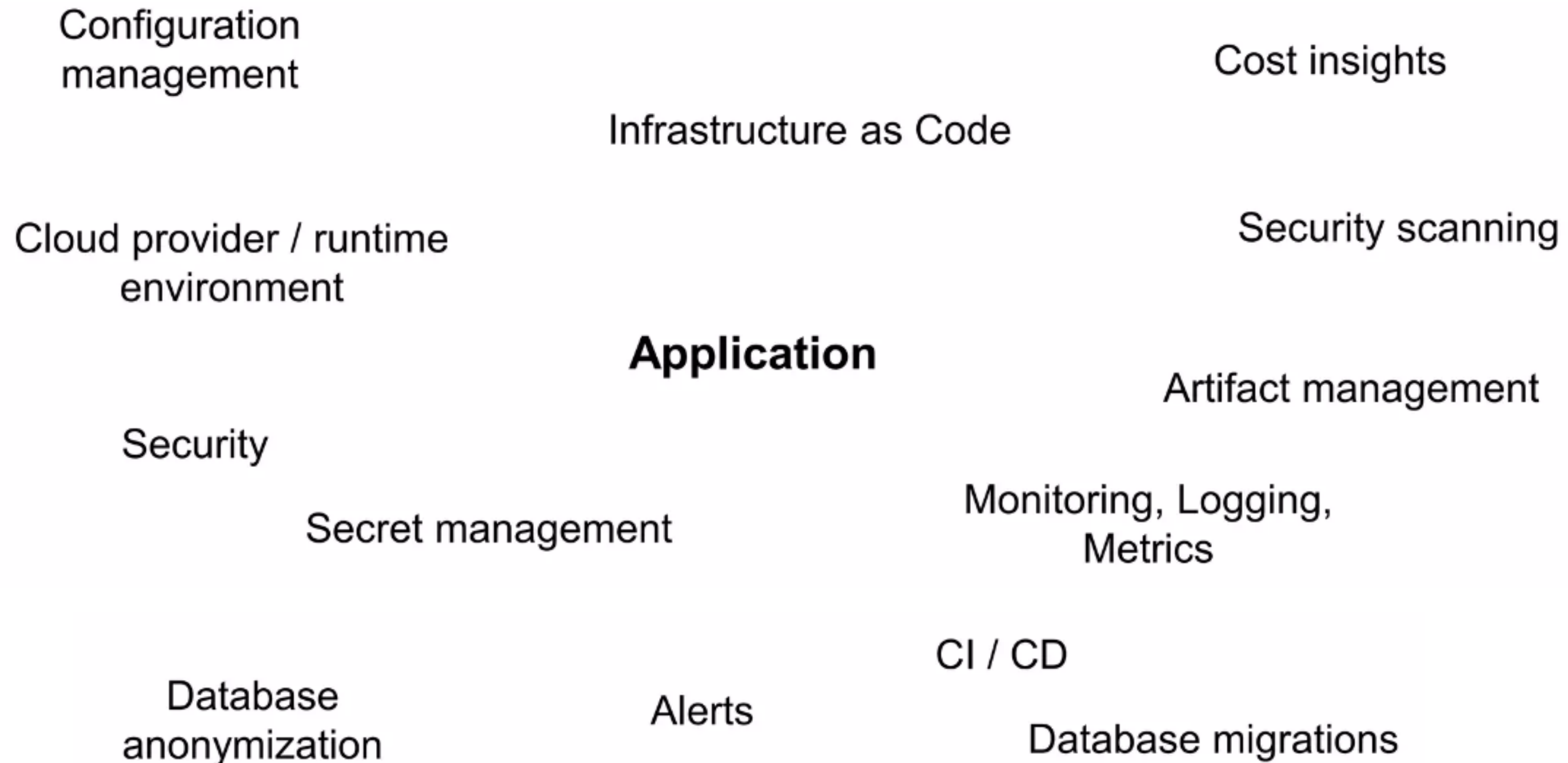
- Not competing with DevOps
- Think that Class SRE implements Interface DevOps
- SRE is a part of the DevOps umbrella

SRE Practices

- Identify and measure **SLIs**, define **SLOs** and agree/ commit to **SLA** for product and service
- Chaos Engineering
- Removing toil
- System designing (DR, Multi-Region, Mult-Cloud)
- Postmortems/ Root Cause Analysis
- Observability

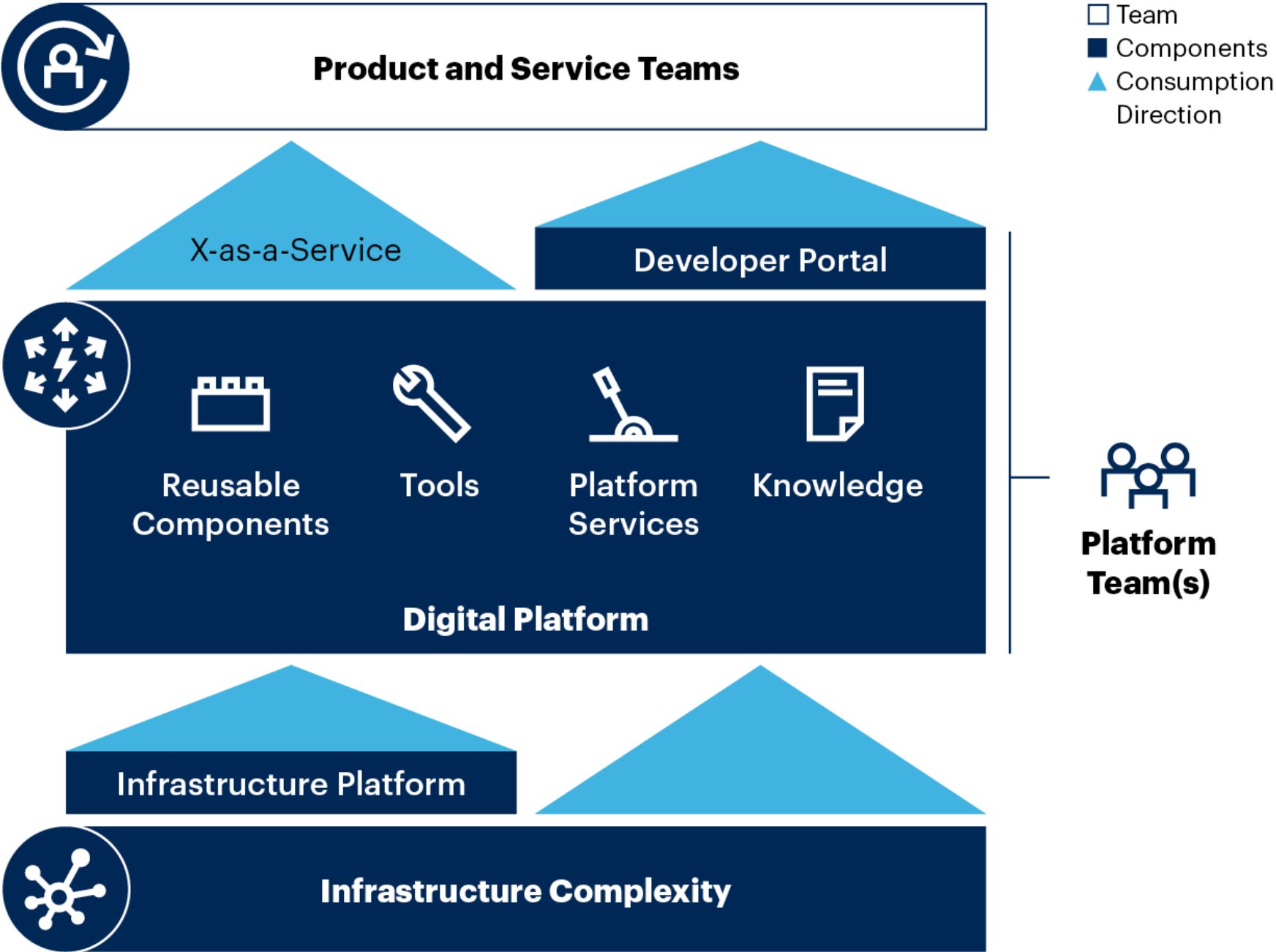
Platform Engineering

Before jumping to definition, let's understand the problem...



“The composition and integration of a set of processes, tools and automation (components) to build a coherent platform with the goal of empowering developers to be able to easily build, maintain and deploy their business logic”

Diagram of Platform Engineering



gartner.com

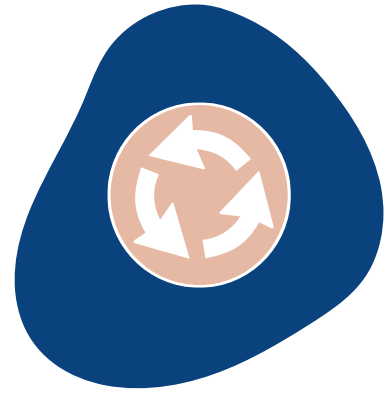
Source: Gartner
© 2023 Gartner, Inc. and/or its affiliates. All rights reserved. CM_GTS_2479487





Carrier as a DevOps Engineer

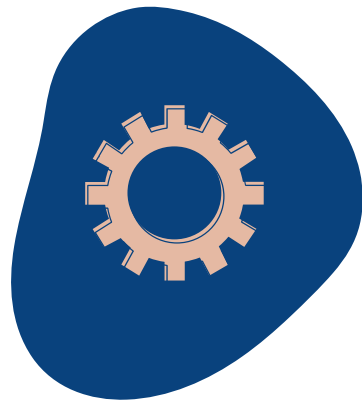
DevOps Engineer Role



CI/ CD Management & Automation



Cloud Deployment and Management



Infrastructure Management



Performance Assessment and Monitoring



Writing Specifications and Documentation



Assisting with DevOps culture adoption

References

- <https://sre.google/sre-book/table-of-contents/>
- <https://www.gartner.com/en/articles/what-is-platform-engineering>
- https://youtu.be/uTEL8Ff1Zvk?si=5QT_LrzedX-BMezt



LinkedIn

<https://www.linkedin.com/in/ravindufernando/>

X (Twitter)

@ravindunf

Thank You!