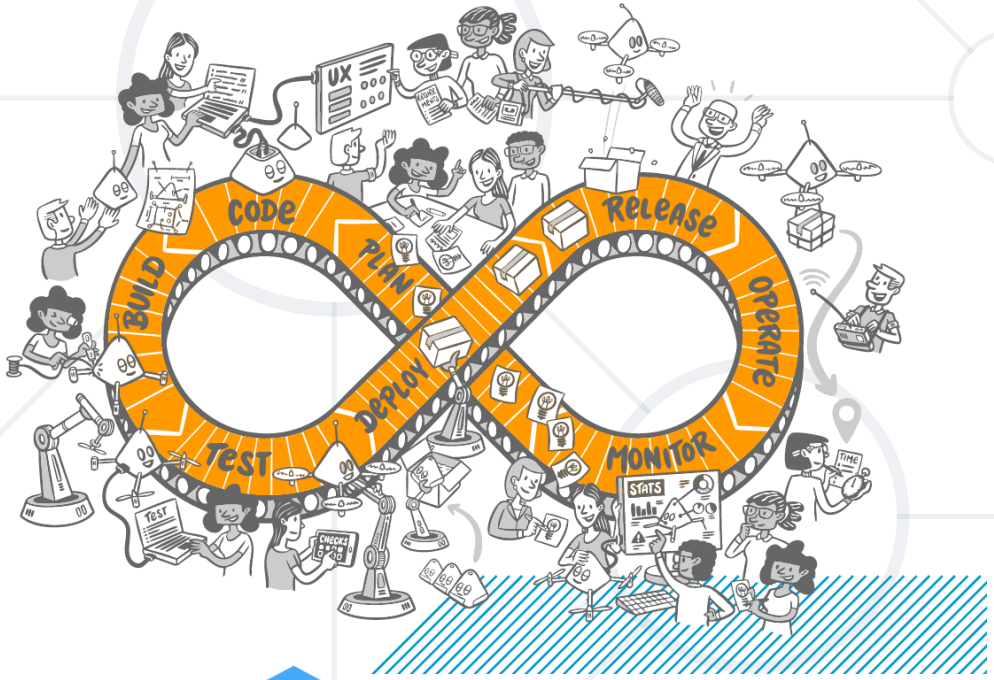
DevOps Overview

What Is It, Practices, Tools, Trends



SoftUni Team

Technical Trainers

Software University

SoftUni

Software University

https://softuni.bg

Have a Question?

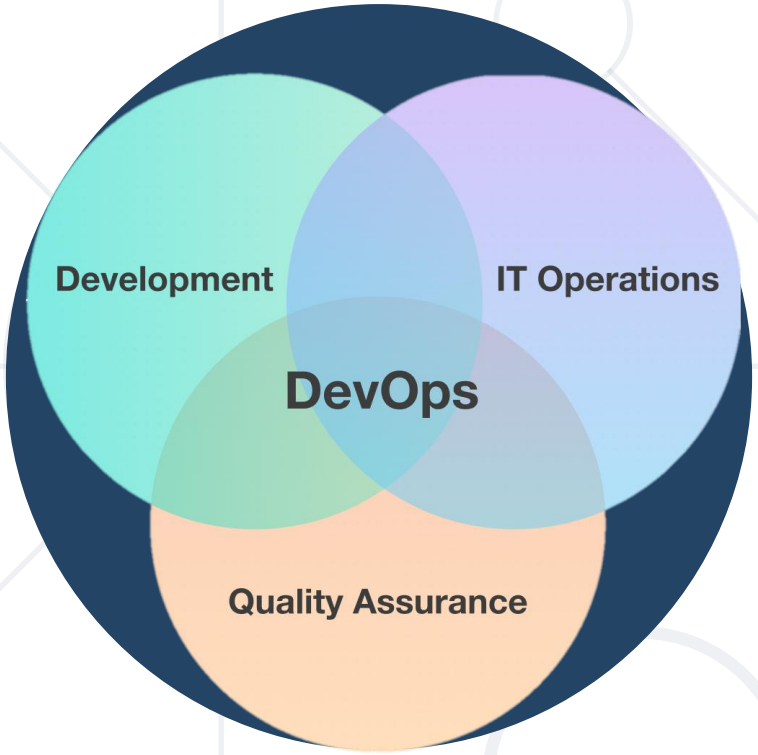
Software University

sli.do

#Dev-Ops

Table of Content

1. What is DevOps?
2. DevOps Practices
3. DevOps Trends



What is DevOps?

Combining Software Development and IT Teams

What is DevOps?

- **DevOps** is a set of practices, tools, and philosophy that combines **development** (**Dev**) and **operations** (**Ops**) into one, continuous process
- Unites **people**, **process**, and **technology** in application **planning**, **development**, **delivery**, and **operations**
  - Enables coordination and collaboration between isolated roles like development, IT operations, quality engineering, and security

DevOps Lifecycle

- **DevOps lifecycle** (or **pipeline**) is a series of automated development processes or workflows within an **iterative development lifecycle**
- Represents the processes, capabilities, and tools for **development** (left side) and **operations** (right side)
- Follows a continuous approach

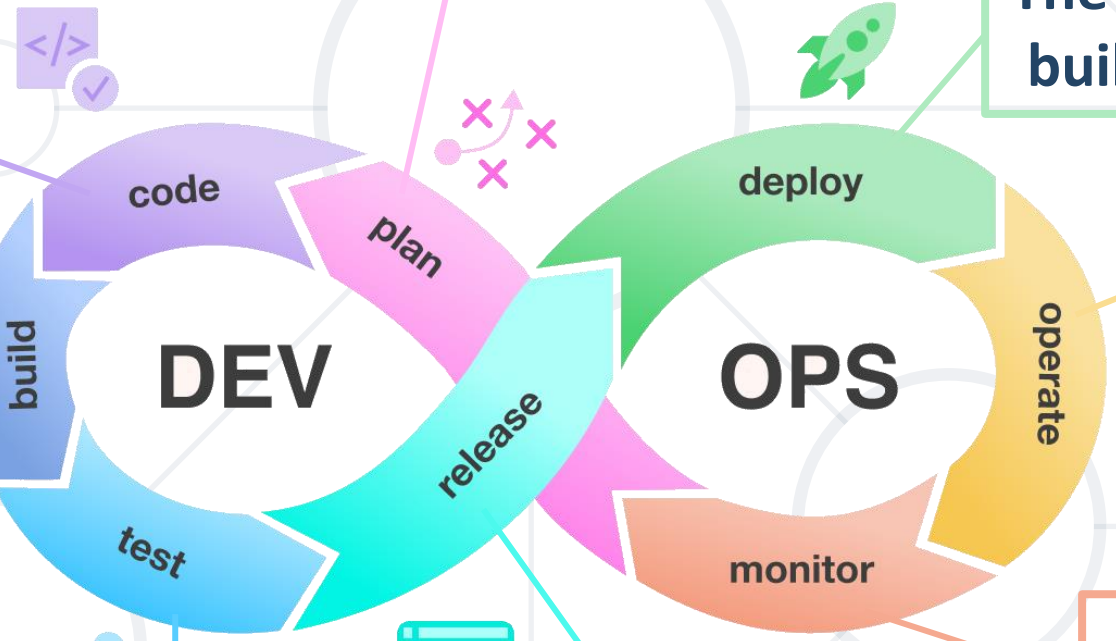
DevOps Lifecycle Stages

Software University

Identify business requirements and collect end-user feedback

Code development

The production environment is built and the build is released



Release is live now. Operations team takes care of server configuring and provisioning

Finished code is committed to a shared repository

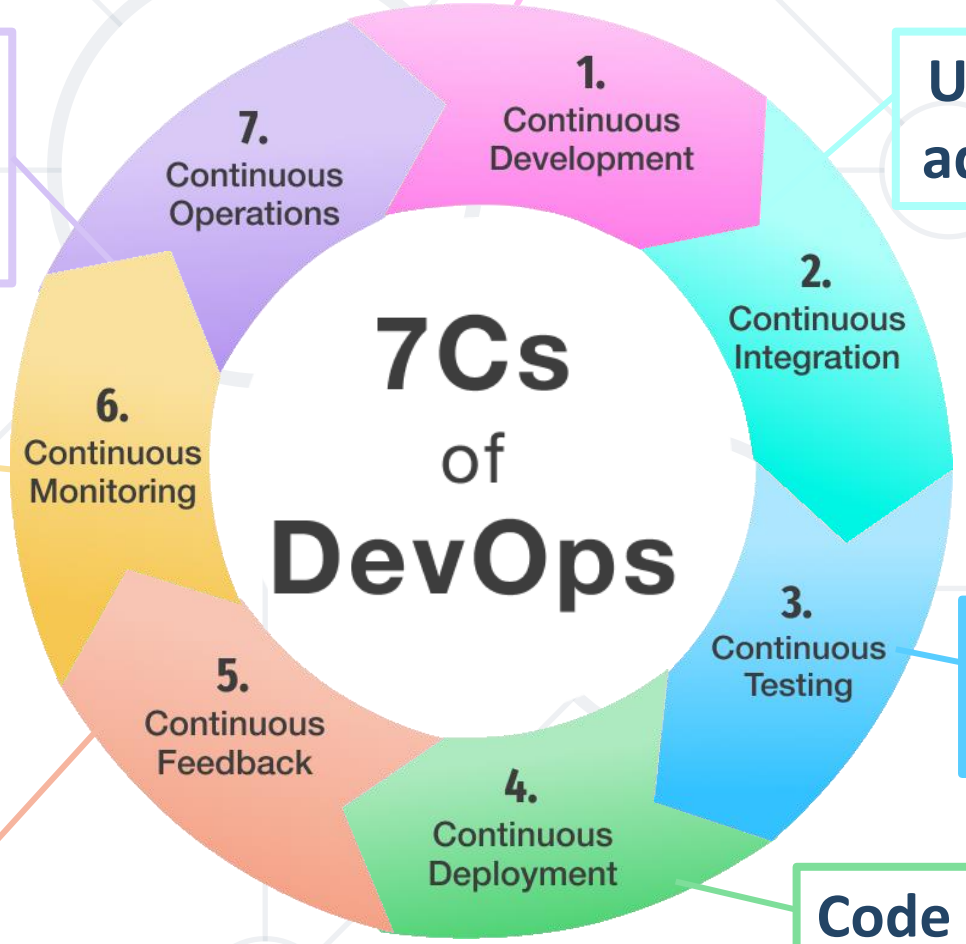
DevOps pipeline is monitored to find problems / bottlenecks

Build is deployed to a test environment and tests are performed

Operations team schedules the releases or deploys multiple releases to production

DevOps Pipeline

- 7 DevOps **pipeline phases**



Plan and code

Update code and add new features

Run automated or manual tests

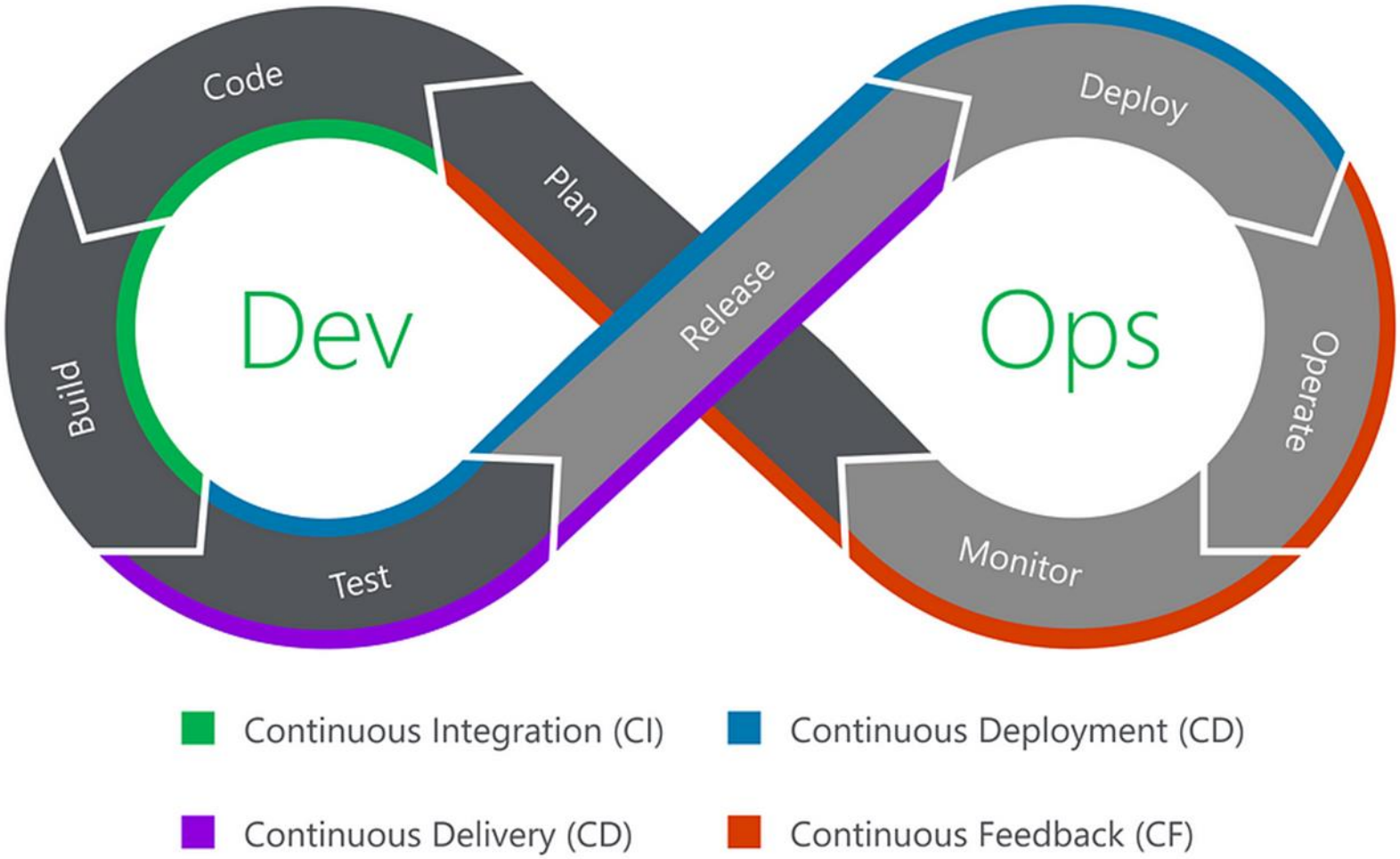
Code is deployed on production servers

Evaluate user experience to improve future releases

Monitor for system errors or performance issues

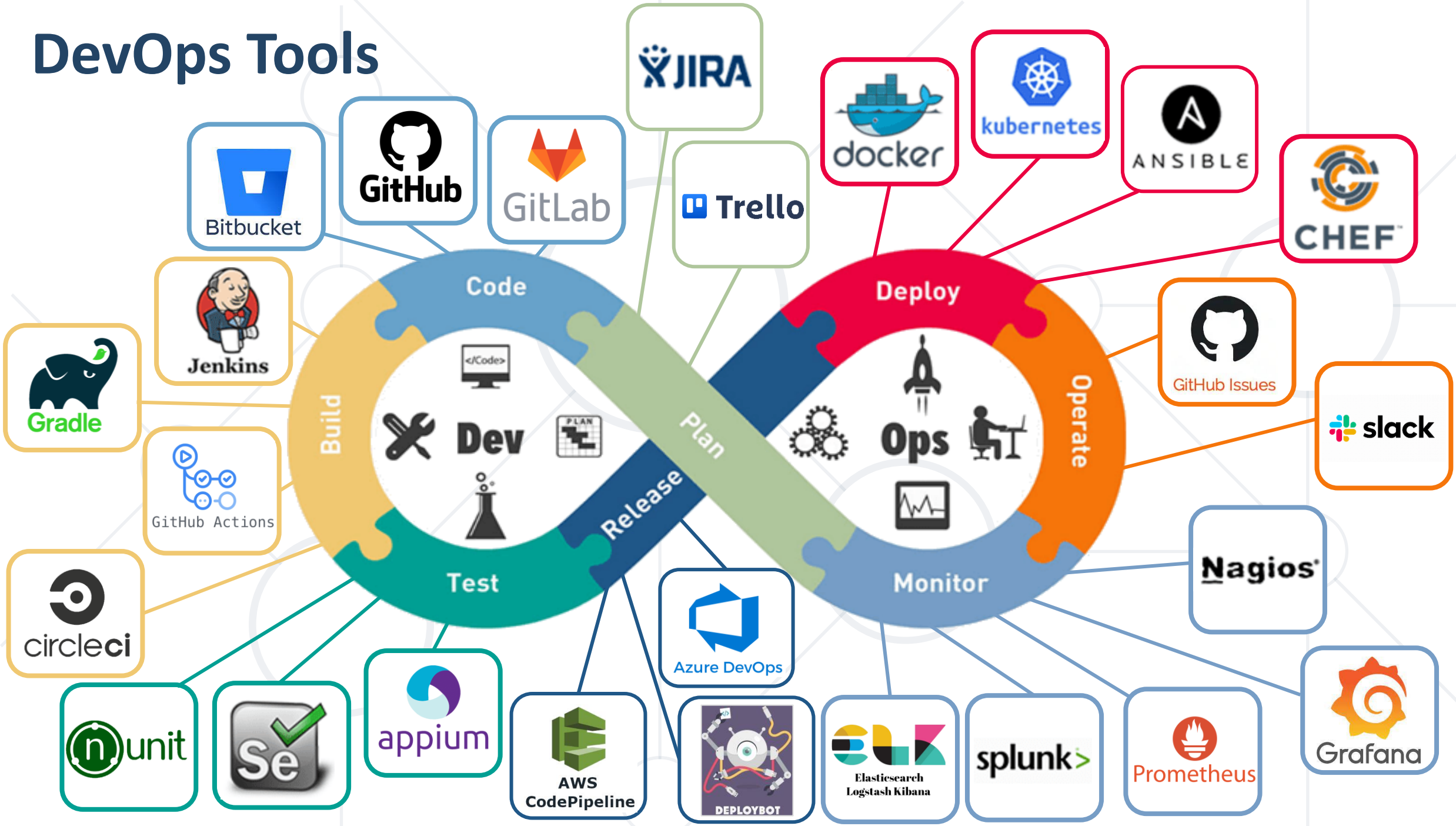
Automate launching the app and its updates

Continuous Everything



Source: https://medium.com/taptuit/the-eight-phases-of-a-devops-pipeline-fda53ec9bba

DevOps Tools



DevOps Culture

- **DevOps culture** is a collaborative approach to software development and delivery that emphasizes **communication**, **automation**, and **improvement**
- **Collaboration** is crucial – all teams should communicate about DevOps processes, priorities, and concerns together
- As teams align, they take **ownership** and **become involved in other lifecycle phases**, not just the ones central to their roles
- DevOps teams remain agile by **releasing software in short cycles**
- Teams strive to **learn** and **continuously improve**

DevOps Engineers

Software University

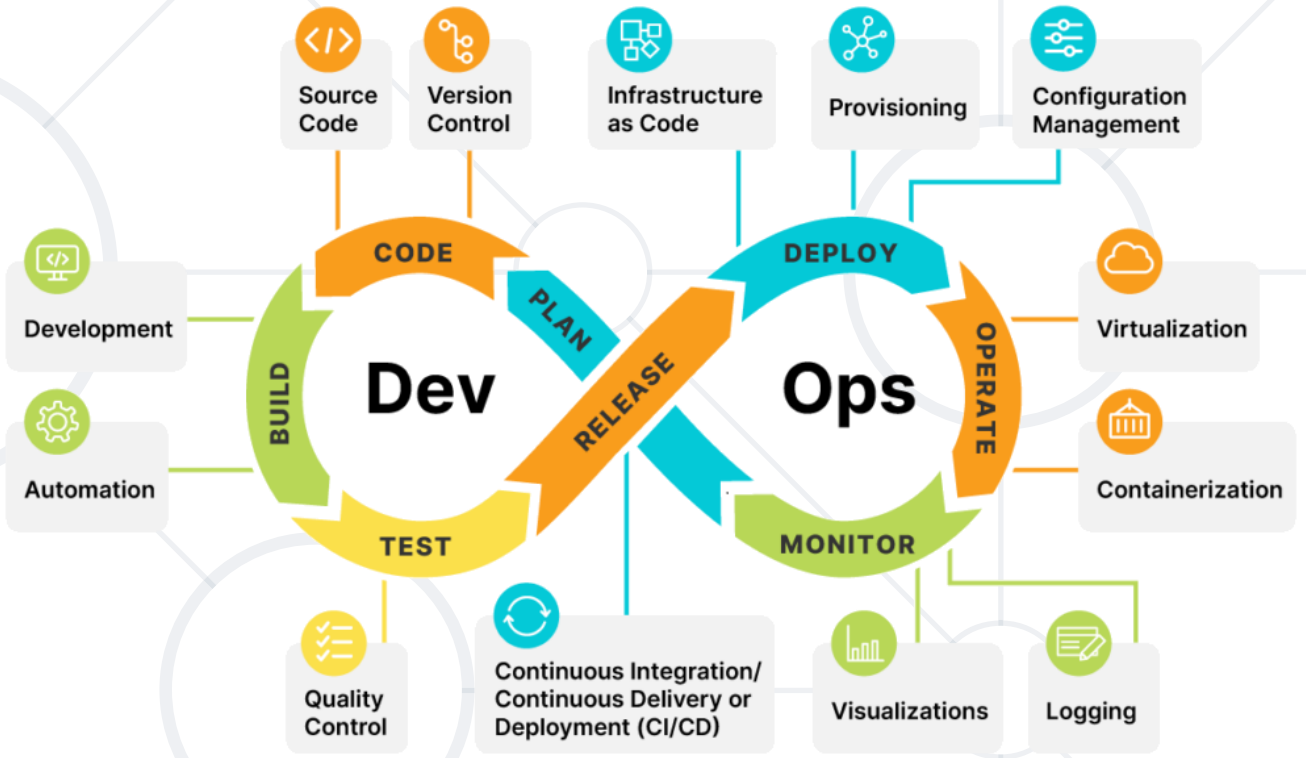
- **DevOps engineers** are responsible for the **deployment**, and **maintenance** of software applications
  - **Collaborate** with development and operations teams
  - Their job includes automating processes, managing infrastructure, monitoring performance, and ensuring the reliability and security of the software
- They understand **development lifecycles**, **DevOps culture**, **practices** and **tools**

DevOps Practices

Helpful Throughout the Application Lifecycle

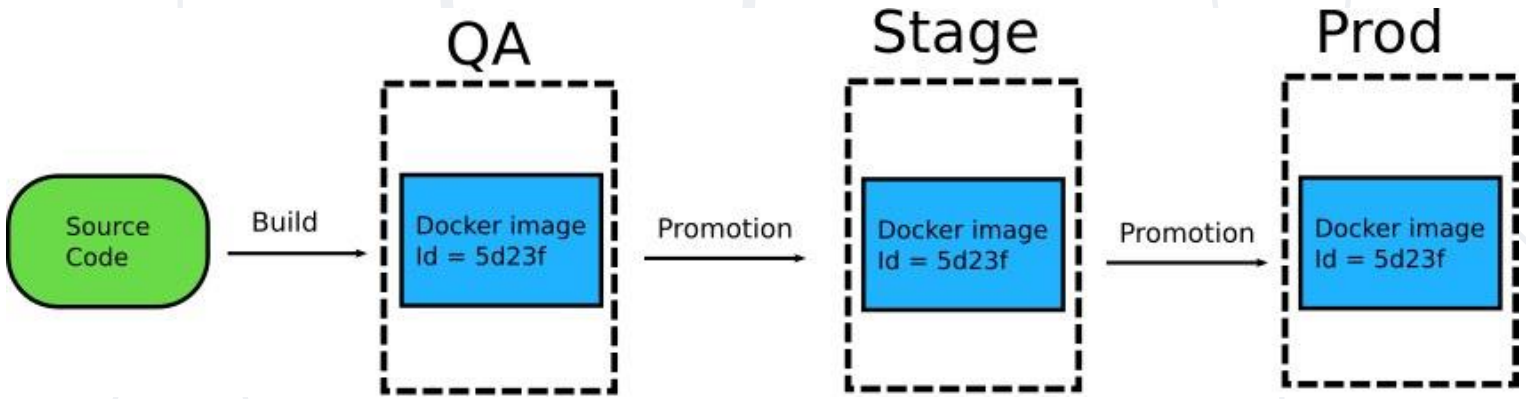
DevOps Practices

- Many **DevOps practices**
  - Varying on the specific context and organization



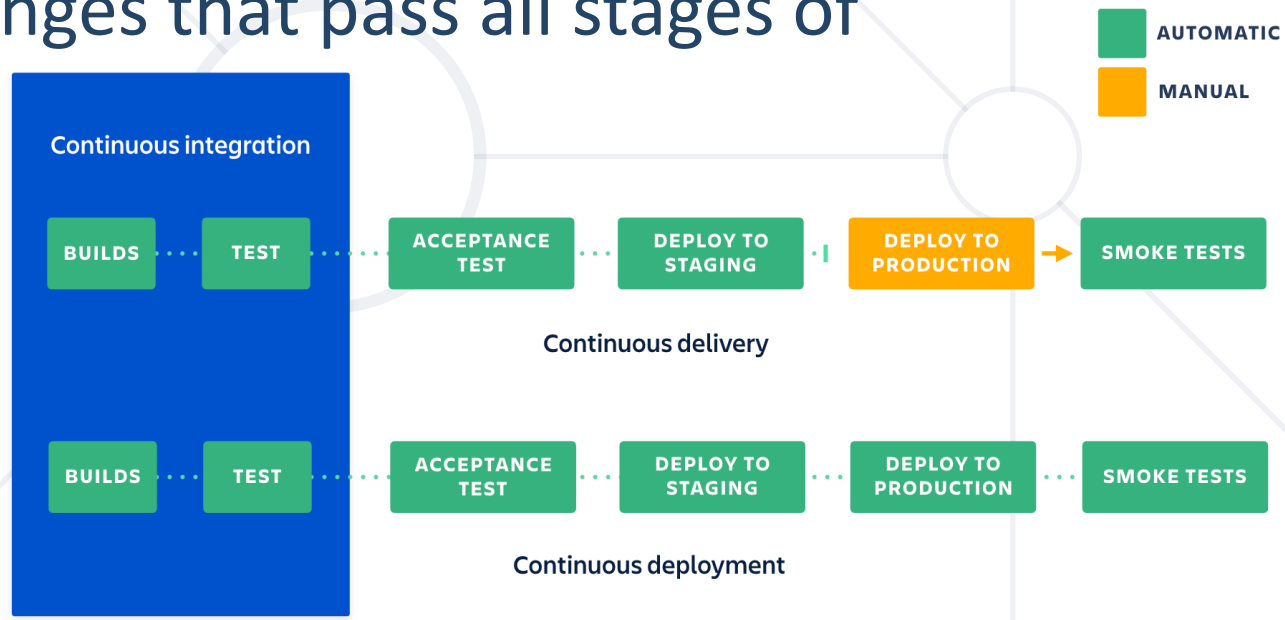
- Some practices are
  - **CI/CD**
  - **Infrastructure as code (IaC)**
  - **Version control**
  - **Monitoring** and **logging**
  - **Automation**
  - **Agile software development**

CI/CD Pipeline (1)



CI/CD Pipeline (2)

- **CI/CD** allows organizations to ship software quickly and efficiently
  - **Continuous integration** – developers regularly **merge code changes** into a central repository, which are validated by **automated tests**
  - **Continuous delivery** – code changes are automatically **prepared for a release** to production (and can be manually deployed)
  - **Continuous deployment** – changes that pass all stages of production pipeline are **released automatically** (optional)
- Tools: **GitHub Actions**, Jenkins, CircleCI, etc.



Infrastructure as Code (IaC)

- **Infrastructure as Code** (**IaC**) == managing and provisioning of infrastructure through code instead of through manual processes
- Used to **automatically manage infrastructure resources**
  - Servers
  - Operating systems
  - Software platforms
  - Storage
  - Networking
  - Etc.

IaC Tools

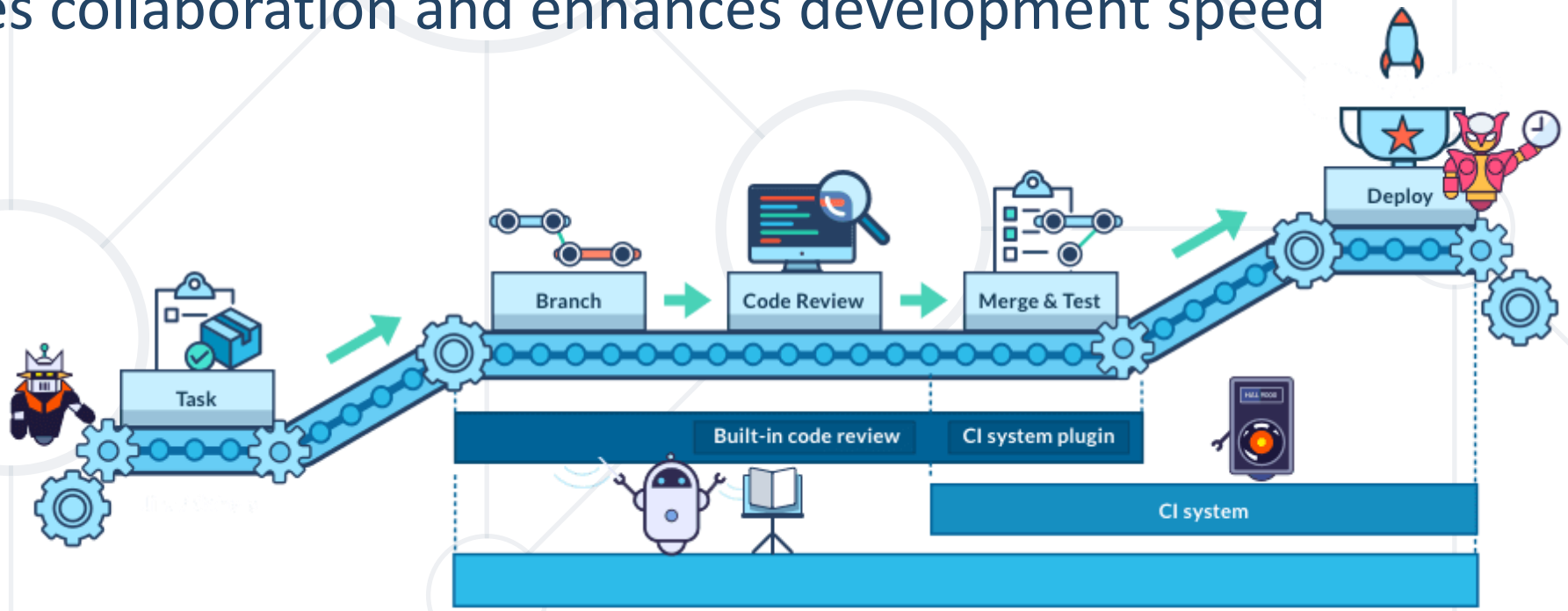
- **IaC tools** define **infrastructure resources** using **code / config files**
  - Can be version controlled, tested, and deployed automatically
- Tools: Ansible, Puppet, Chef, Saltstack, Terraform, etc.
- **Approaches** to Iac
  - **Declarative** – defines the desired state of the system, i.e. resources you need and their properties
  - **Imperative** – defines the specific commands for the desired configuration

Version Control (1)

- **Version control** (**source control**) == the practice of **managing code in versions** to make code easy to review and recover
  - Includes tracking revisions and change history
    - Saves each individual changes in a special database
  - Necessary for CI/CD and IaC
    - Helps enhance efficiency
    - Allows preserving agility when a team grows larger
- Tools: **Git**, SVN, Mercurial, etc.

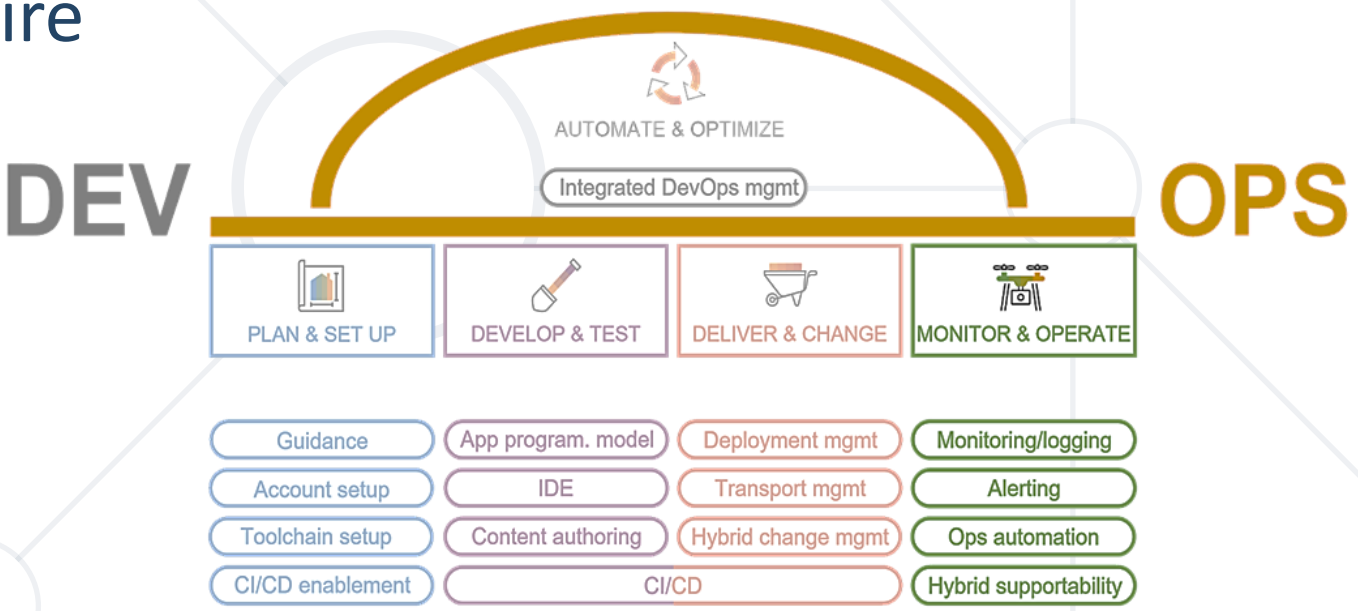
Version Control (2)

- **Essential** for software development
  - Serves as a **safety net** to protect code
  - Allows several people to work on a project **simultaneously**
    - Improves collaboration and enhances development speed



Monitoring and Logging

- **Monitoring** means having **full, real-time visibility** into the health and performance of the entire application stack
  - **App metrics**, **event data**, **logs**, traces, etc. are collected and analyzed
  - Actionable and meaningful **alerts** are set for failures in the entire deployment pipeline
    - Thus, DevOps team can mitigate issues in real time
- Tools: ELK Stack, Splunk, Prometheus, Nagios, etc.



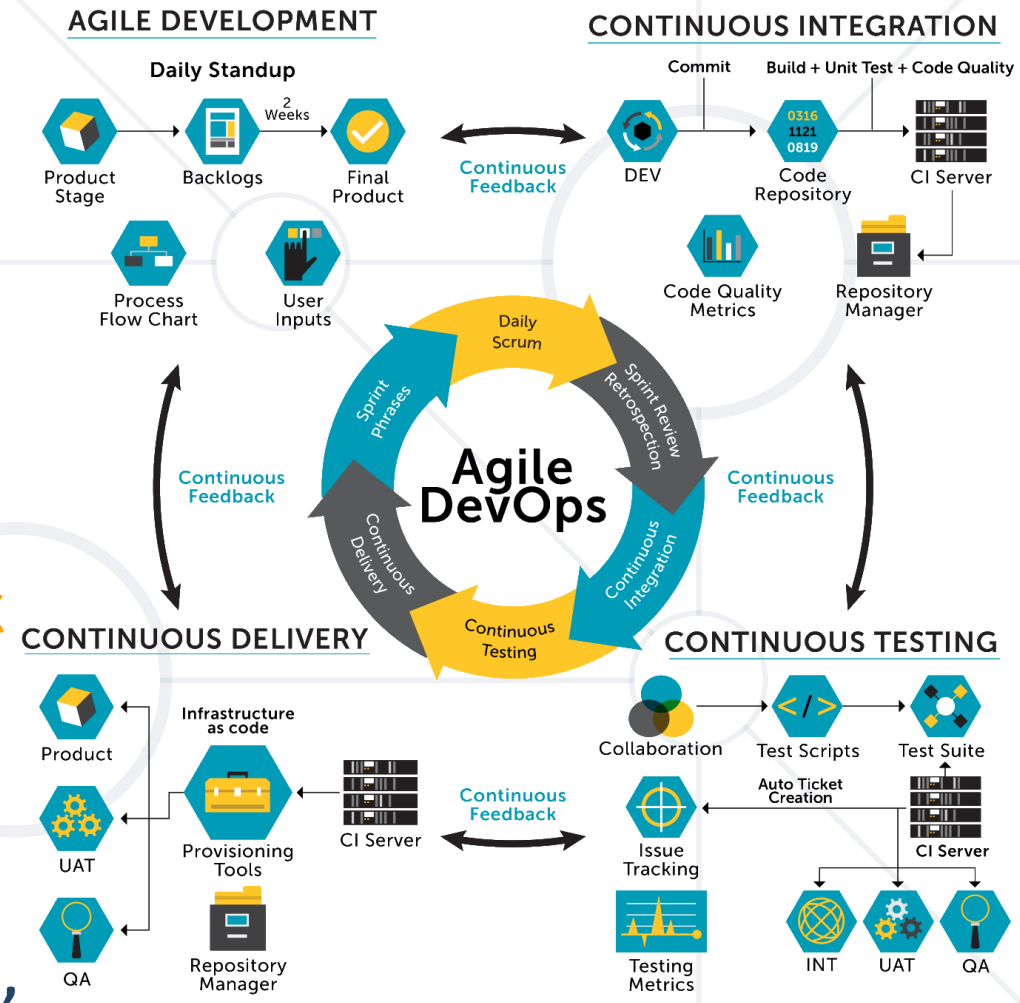
Automation

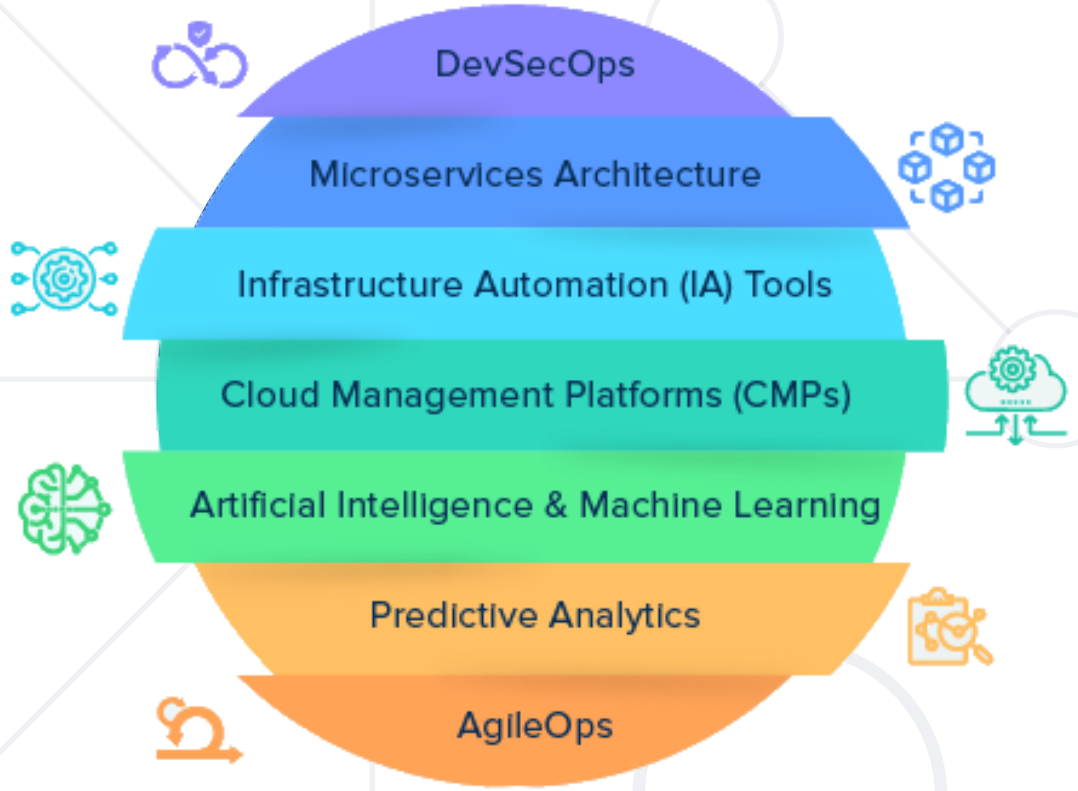
- DevOps teams aim to **automate** as much of the **software lifecycle** as possible to have more time for writing code and developing features
  - With **automation** the simple act of pushing code changes to a source code repository can trigger a build, test, and deployment process
  - Pros: **software delivery** is **faster**, **processes** are **consistent**, **predictable** and **scalable**, teams don't perform tedious manual tasks
- **Tools** are different for each step of the DevOps process

Agile Software Development

Software University

- **Agile** == modern software development approach
- It emphasizes on
  - **High adaptability to change** through **short release cycles**
  - **Customer** and **user feedback**
  - **Team collaboration**
- In **DevOps**, **Agile practices** include increased automation, improved collaboration, etc.



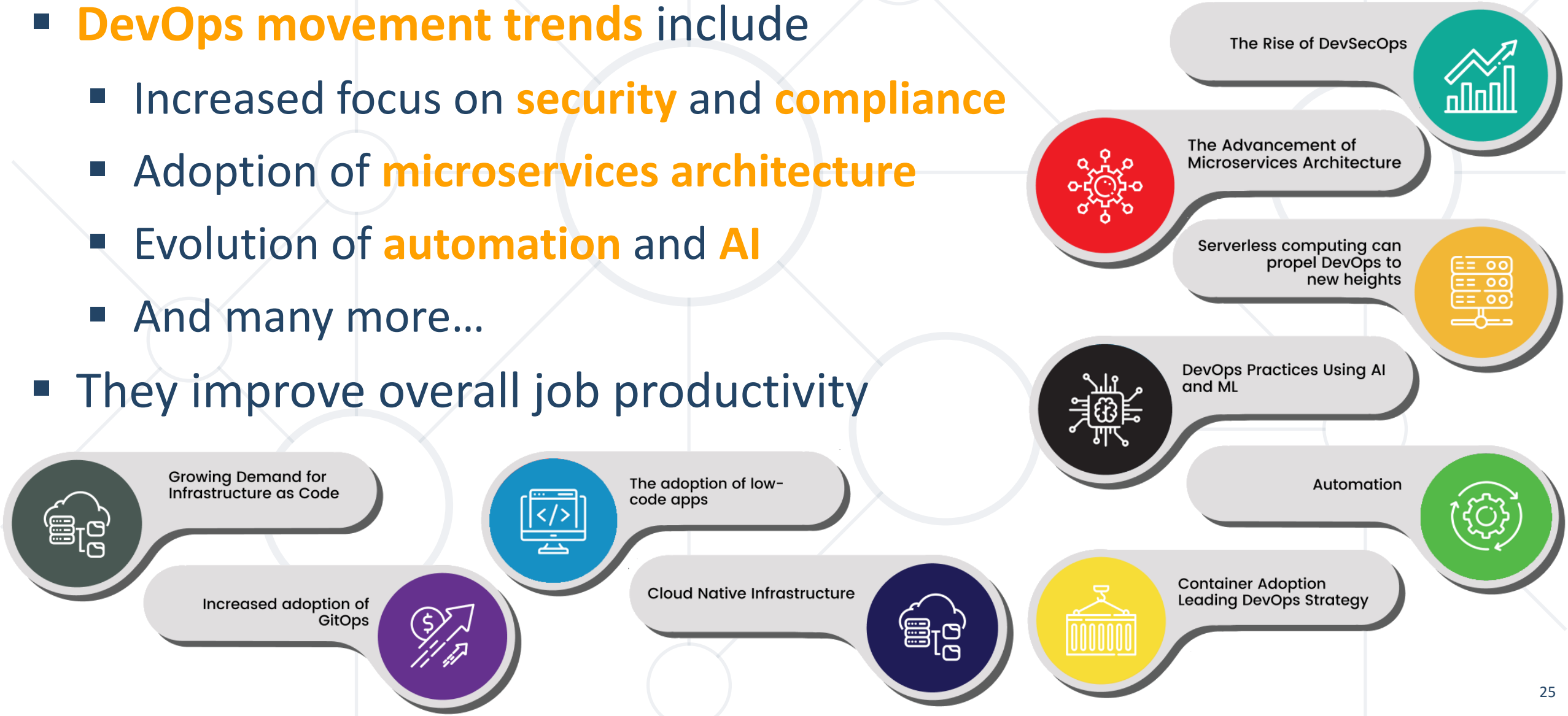


DevOps Trends

Additional DevOps Practices for Improved Lifecycle

DevOps Trends

- **DevOps movement trends** include
  - Increased focus on **security** and **compliance**
  - Adoption of **microservices architecture**
  - Evolution of **automation** and **AI**
  - And many more…
- They improve overall job productivity

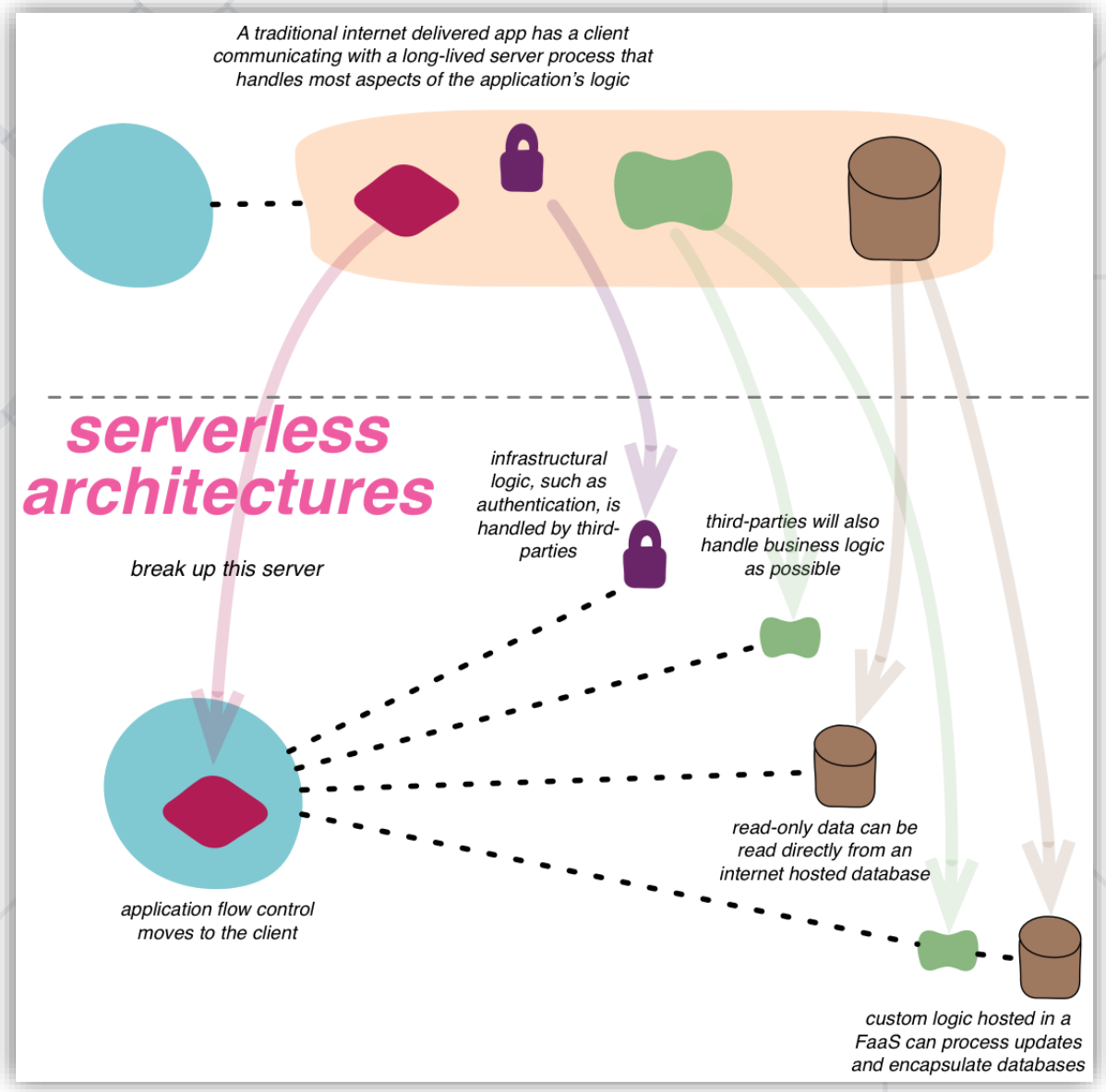


DevSecOps

- **DevSecOps** = **development** + **security** + **operations**
- Includes **DevOps framework** with **security** as a shared responsibility
- Its mindset is to **integrate security practices** into applications and infrastructure from the start
- Identifying security vulnerabilities
  - Static Analysis
    - Catch potential security issues without executing the code
  - Dynamic Analysis
    - Detect security issues at runtime

Serverless Computing (1)

- **Serverless computing** refers to **outsourcing back-end cloud** infrastructure and operations tasks to a **cloud provider**
  - Developers **don't manage servers** but build and run **apps in containers**
  - A **cloud provider** handles the routine tasks of provisioning, maintaining, and scaling the server infrastructure



Serverless Computing (2)

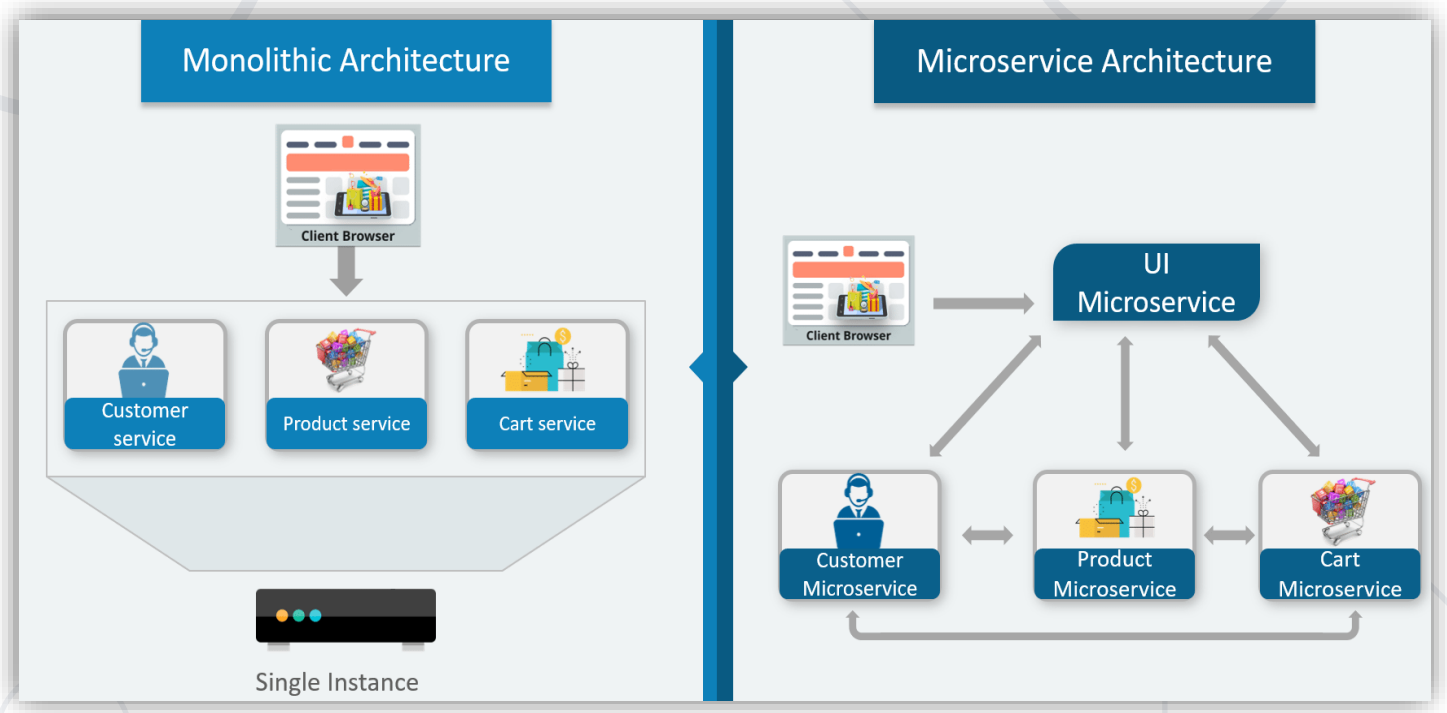
- **Serverless computing** == **Function-as-a-Service** (**Faas**)
- Based on **event-driven execution**
- **Stateless nature**
  - Serverless functions are designed to be stateless
- Wide range of tools
  - Frameworks
  - SDKs
  - CLIs

Microservices Architecture (1)

- **Microservices** == architectural approach to development that **breaks the application** into different **loosely coupled services**
  - Each service focuses on a specific business capability
    - Can be independently developed, deployed and scaled
- As everything is broken down into separate services, **development teams** can also be **divided** to **tackle each service**
  - Makes the development process more flexible

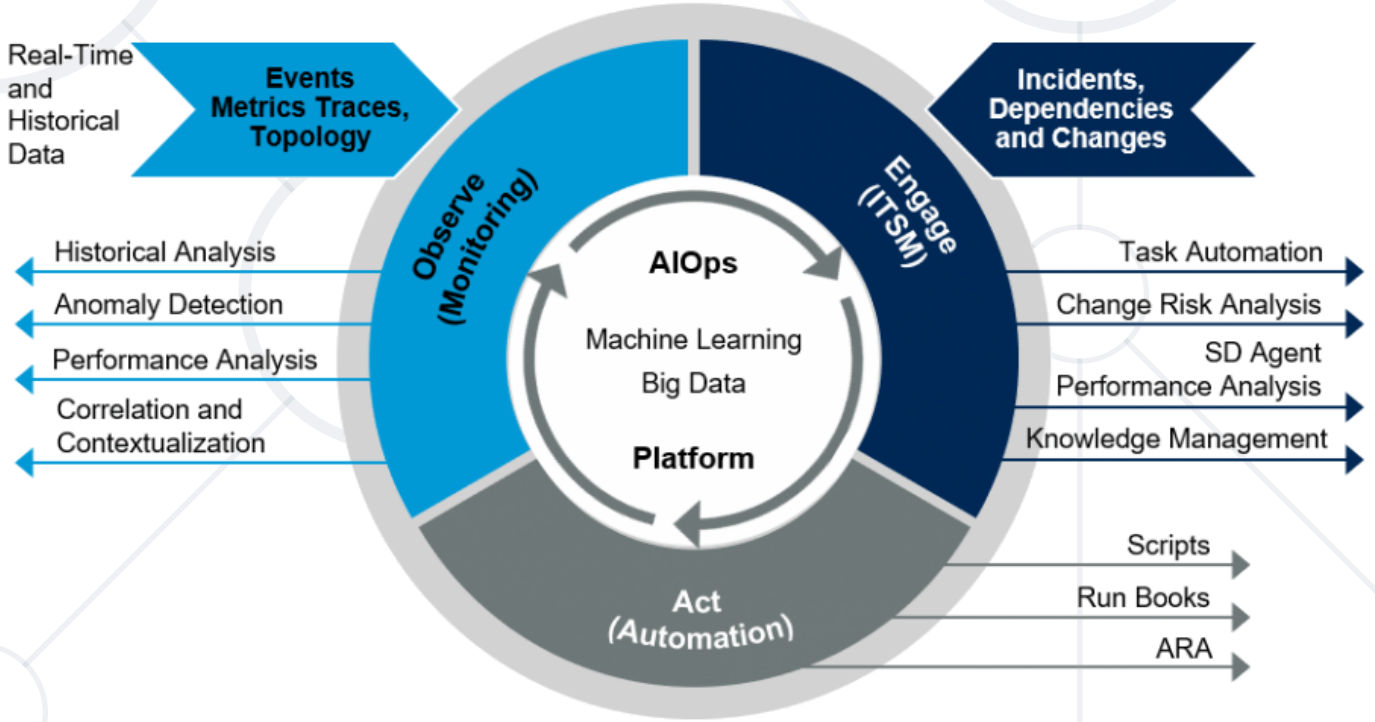
Microservices Architecture (2)

- **Communication** between services is typically achieved through **lightweight protocols**, e.g., HTTP/REST
- Each microservice can have its own technology stack, programming language and db
  - These depend on the specific business requirements



AIOps and MLOps

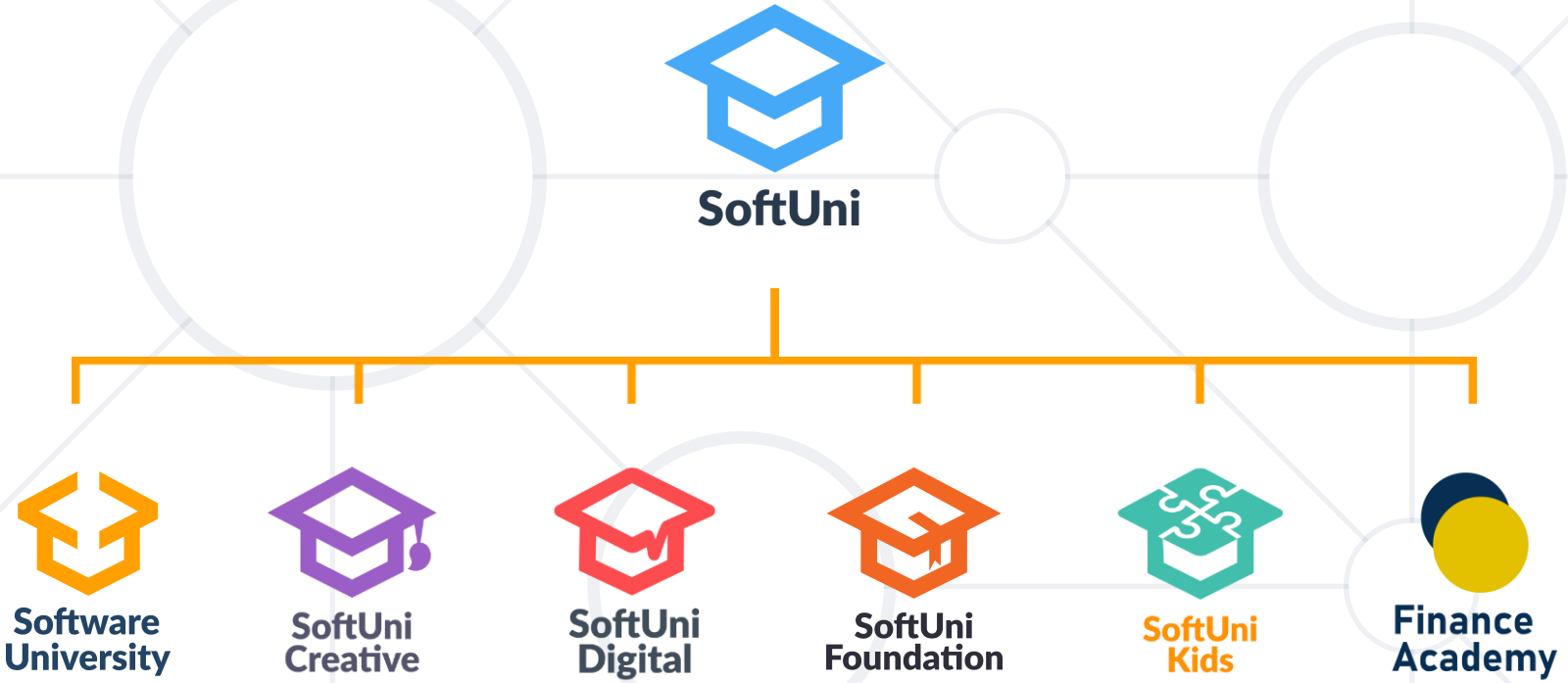
- **AIOps** (Artificial Intelligence for IT Operations) refers to the use of artificial intelligence (**AI**) and machine learning (**ML**) technologies to automate and enhance various IT operations and processes
- **AIOps** helps with identifying the main cause of the problems that hamper operational productivity
- **MLOps** helps with optimizing operations and enhancing productivity



Summary

- **DevOps** == a set of **practices**, **tools**, and a **cultural philosophy** that automate and integrate the processes between **software development** and **IT operations teams**
- 8 **DevOps lifecycle stages** and 7 **pipeline phases**
- **DevOps practices** include **CI/CD**, **Infrastructure as Code**, **Version Control**, **Monitoring** and **Logging**, **Automation**, **Agile Software Development**, etc.
- DevOps trends include **DevSecOps**, **Microservices**, **Serverless Computing** and **AIOps**

Questions?



SoftUni Diamond Partners

Software University

SUPER HOSTING .BG

Coca-Cola HBC Bulgaria

POKERSTARS
POKER | CASINO | SPORTS
a Flutter International brand

INDEAVR
Serving the high achievers

AMBITIONED

DRAFT KINGS

SOFTWARE GROUP

createX

Postbank
Решения за твоето утре

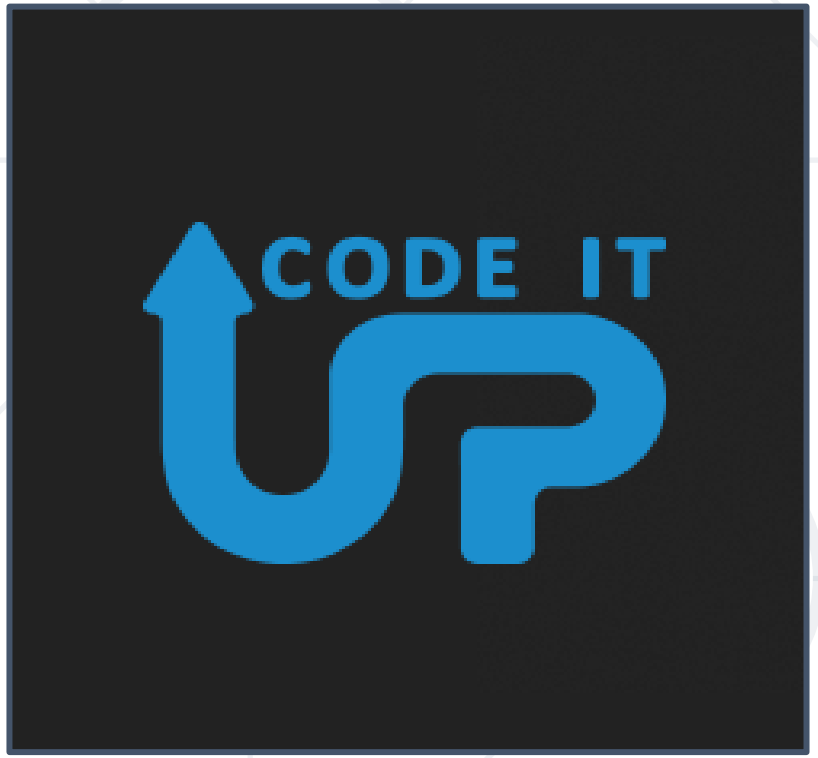
BOSCH

DXC TECHNOLOGY

SmartIT

Educational Partners

Software University



License

- This course (slides, examples, demos, exercises, homework, documents, videos and other assets) is **copyrighted content**
- Unauthorized copy, reproduction or use is illegal
- © SoftUni – https://about.softuni.bg/
- © Software University – https://softuni.bg

Trainings @ Software University (SoftUni)

- Software University – High-Quality Education, Profession and Job for Software Developers
  - softuni.bg, about.softuni.bg
- Software University Foundation
  - softuni.foundation
- Software University @ Facebook
  - facebook.com/SoftwareUniversity
- Software University Forums
  - forum.softuni.bg