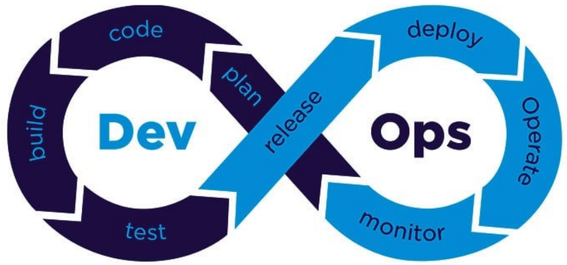
DevOps



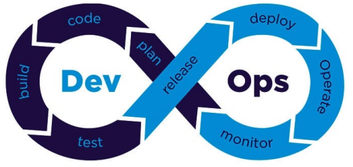
Sonia BEN AISSA

https://learn.microsoft.com/en-us/training/modules/get-started-with-devops/

TEK-UP University

Plan

1. What is DevOps?
2. Difference between Dev and Ops?
3. Key Principles of DevOps
4. Difference between DevOps and DevSecOps?
5. Common DevOps Practices
6. DevOps Lifecycle

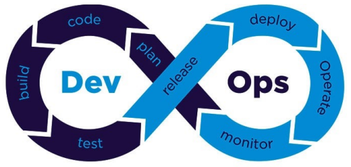


1. What is DevOps?

DevOps The difference between IT development and IT operations lies in their focus, responsibilities, and objectives

DevOps is an umbrella term that describes:

- The operation of a team collaborating an entire programming production
- A combination of tools and philosophies that increase a team's capability to produce results at high efficiency.



2. Difference between Dev and Ops?

The difference between **IT development** and **IT operations** lies in their focus, responsibilities, and objectives

Development : It’s more about creating software product or application. This domain can have roles like developer , tester , system analyst, business analyst etc.

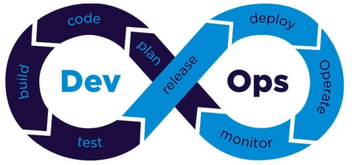
Example: doing coding in C++ or .Net is development

Operations : It’s more about providing required infrastructure necessary of Development and testing. This domain can have roles like IT Technician, system Admin, Help desk engineer etc.

Example : setting up office network or creating new machine , installation etc.

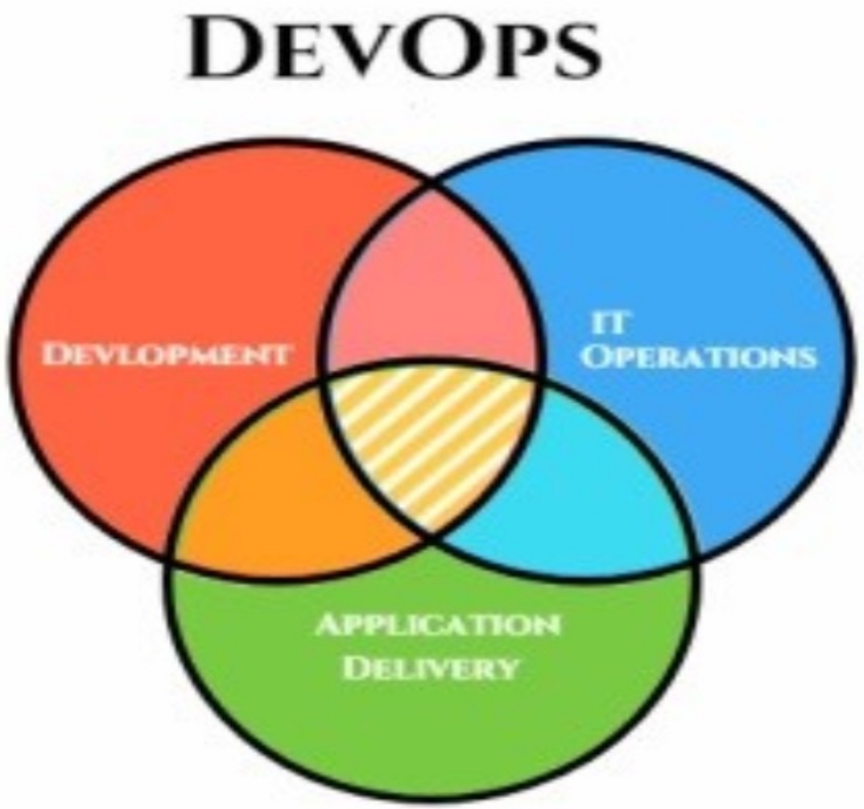
Devops : This is new area recently emerged in IT. Devops act as intermediate between Development and operation team. mainly works with developers and the IT staff for the code releases to various environments.

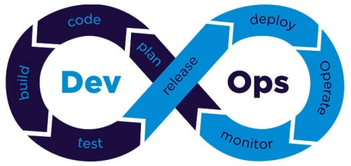
Example: Continuous Integration, Continuous Deployment, Continuous Monitoring etc.



2. Difference between Dev and Ops?

The difference between **IT development** and **IT operations** lies in their focus, responsibilities, and objectives





3. Key Principles of DevOps

1.Collaboration and Communication

- Breaking down silos between development and operations teams
- Fostering a culture of shared responsibility
- Encouraging open and frequent communication

2.Automation

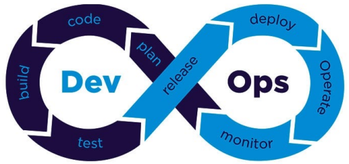
- Automating repetitive tasks to reduce errors and increase efficiency
- Implementing continuous integration and continuous delivery (CI/CD)
- Leveraging infrastructure as code (IaC)

3.Continuous Improvement

- Embracing a mindset of ongoing learning and optimization
- Regularly reviewing and refining processes
- Encouraging experimentation and innovation

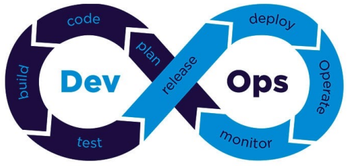
4.Customer-Centric Approach

- Focusing on delivering value to end-users
- Gathering and acting on user feedback
- Aligning development goals with business objectives



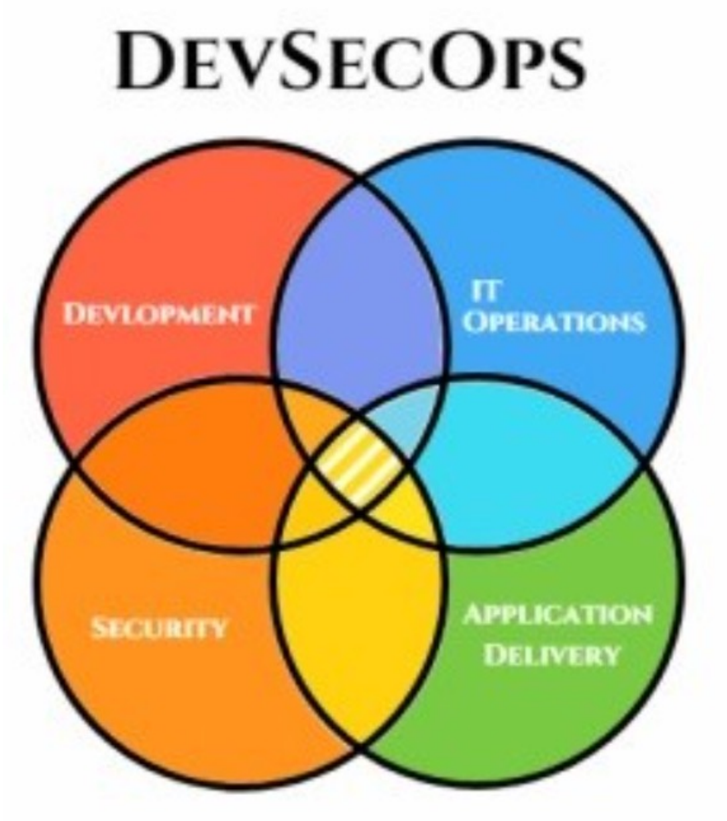
3. Key Principles of DevOps

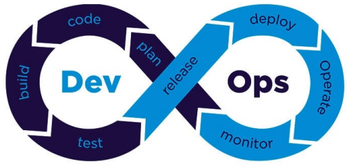
5. **Rapid and Frequent Delivery**
   - Implementing smaller, more frequent releases
   - Reducing time-to-market for new features
   - Enabling faster response to market changes and user needs
6. **Monitoring and Feedback**
   - Implementing comprehensive monitoring and logging
   - Using data-driven decision making
   - Creating feedback loops for continuous improvement
7. **Security Integration (DevSecOps)**
   - Incorporating security practices throughout the development lifecycle
   - Automating security tests and compliance checks
   - Fostering a security-minded culture across teams



4. Difference between DevOps and DevSecOps?

The primary distinction is that **DevSecOps** takes a more thorough approach to security. Both approaches emphasize **automation**, but DevSecOps goes one step further by incorporating security testing and secure coding techniques.





5. Common DevOps Practices

1.Continuous Integration (CI)

- Regularly merging code changes into a central repository
- Automating build and test processes
- Detecting and addressing integration issues early

2.Continuous Delivery (CD)

- Automating the release process
- Ensuring code is always in a deployable state
- Enabling frequent and reliable releases

3.Infrastructure as Code (IaC)

- Managing and provisioning infrastructure through code
- Using tools like Terraform, Ansible, or CloudFormation
- Ensuring consistent and reproducible environments

4.Microservices Architecture

- Breaking down applications into smaller, independent services
- Enabling easier scaling and maintenance
- Allowing teams to work on different services independently

5. Common DevOps Practices

5. Containerization

- Using container technologies like Docker
- Ensuring consistency across different environments
- Facilitating easier deployment and scaling

6.Configuration Management

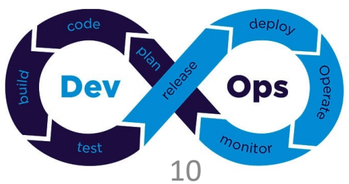
- Automating the management of software configurations
- Using tools like Puppet, Chef, or Ansible
- Maintaining consistency across environments

7.Automated Testing

- Implementing various types of automated tests (unit, integration, performance)
- Ensuring code quality and reducing manual testing efforts
- Enabling faster feedback on code changes

8.Continuous Monitoring

- Implementing real-time monitoring of applications and infrastructure
- Using tools for log analysis and performance metrics
- Enabling proactive issue detection and resolution



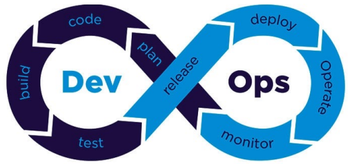
5. Common DevOps Practices

9. Version Control

- Using Git or other version control systems
- Implementing branching strategies (e.g., GitFlow)
- Facilitating collaboration and code review processes

10.Agile Project Management

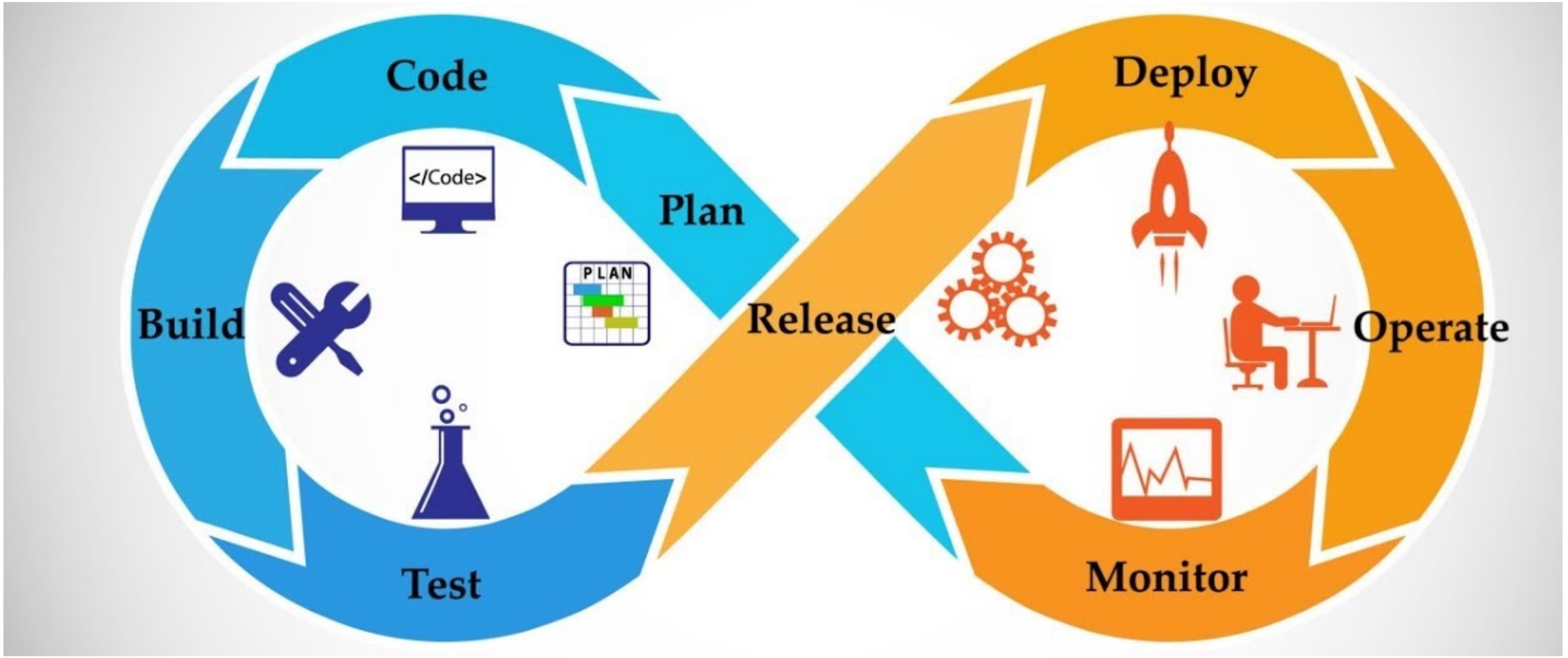
- Adopting agile methodologies (Scrum, Kanban)
- Breaking work into smaller, manageable chunks
- Promoting flexibility and rapid iteration

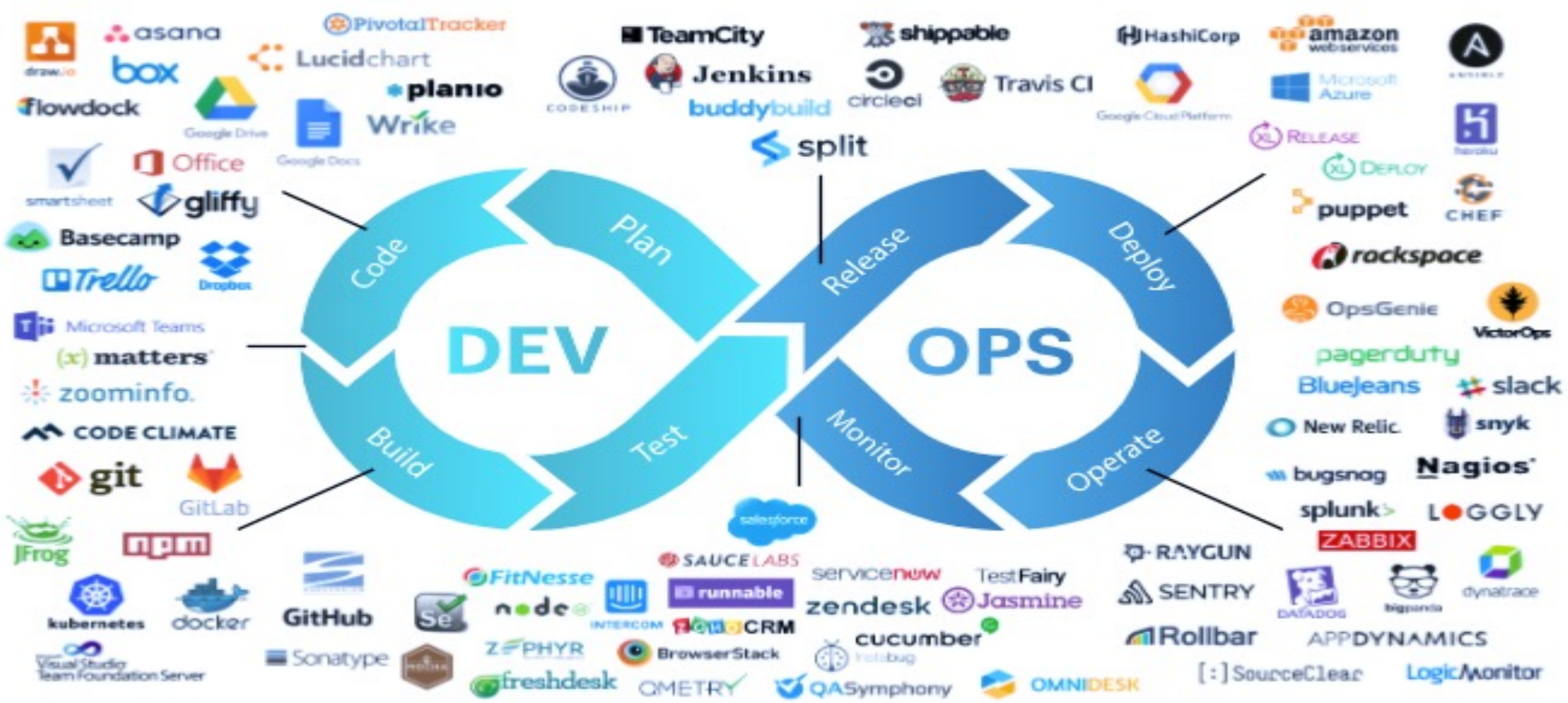


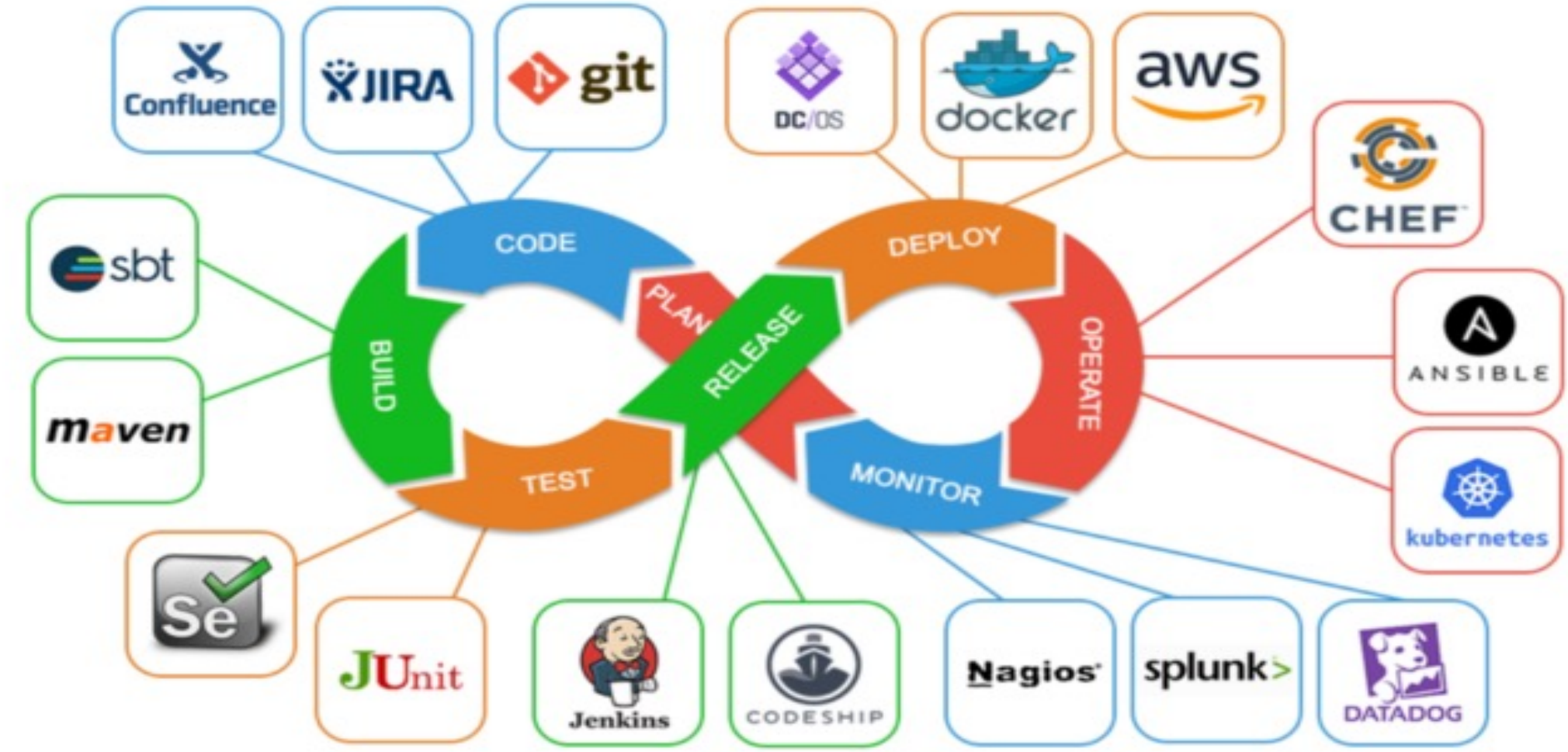
6. DevOps Lifecycle

A standard DevOps Lifecycle consists of 7 phases:

1. Continuous Development
2. Continuous Integration
3. Continuous Testing
4. Continuous Monitoring
5. Continuous Feedback
6. Continuous Deployment
7. Continuous Operations


Code
</Code>
Plan
PLAN
Build
Release
Deploy
Operate
Test
Monitor


DEV
OPS
Code
Plan
Build
Test
Release
Deploy
Monitor
Operate
asana
box
flowdock
Google Drive
PivotalTracker
Lucidchart
planio
Wrike
Google Docs
Office
smartsheet
gliffy
Basecamp
Trello
Dropbox
Microsoft Teams
matters
zoominfo.
CODE CLIMATE
git
GitLab
JFrog
npm
kubernetes
docker
Visual Studio Team Foundation Server
GitHub
Sonatype
TeamCity
Jenkins
CODESHIP
buddybuild
shippable
circleci
Travis CI
split
FitNesse
node
Se
INTERCOM
SAUCE LABS
runnable
CRM
ZEPHYR
BrowserStack
freshdesk
QMETRY
salesforce
servicenow
zendesk
TestFairy
Jasmine
cucumber
QASymphony
OMNIDESK
HashiCorp
Google Cloud Platform
amazon web services
Microsoft Azure
RELEASE
DEPLOY
puppet
CHEF
rackspace
OpsGenie
VictorOps
pagerduty
BlueJeans
slack
New Relic
snyk
bugsnag
Nagios
splunk>
LOGGLY
ZABBIX
RAYGUN
SENTRY
DATADOG
Rollbar
APPDYNAMICS
[:]SourceClear
LogicMonitor
dynatrace


Confluence
JIRA
git
DC/OS
docker
aws
CHEF
sbt
Maven
CODE
PLAN
BUILD
TEST
RELEASE
DEPLOY
OPERATE
MONITOR
ANSIBLE
kubernetes
Se
JUnit
Jenkins
CODESHIP
Nagios
splunk>
DATADOG