Software Maintenance: A DevOps Practitioners Perspective

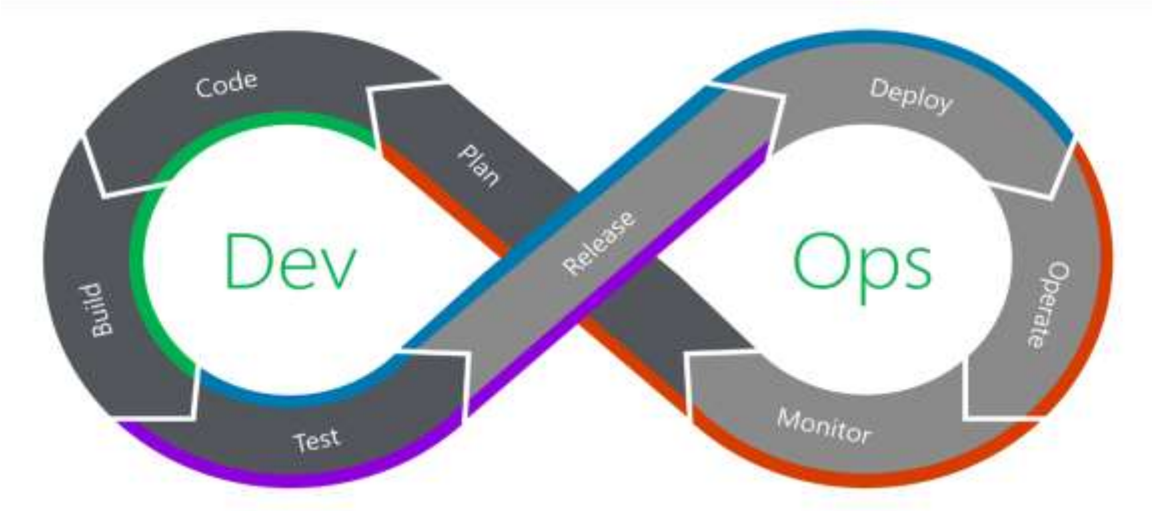
Rafed Muhammad Yasir

Today We Will Discuss

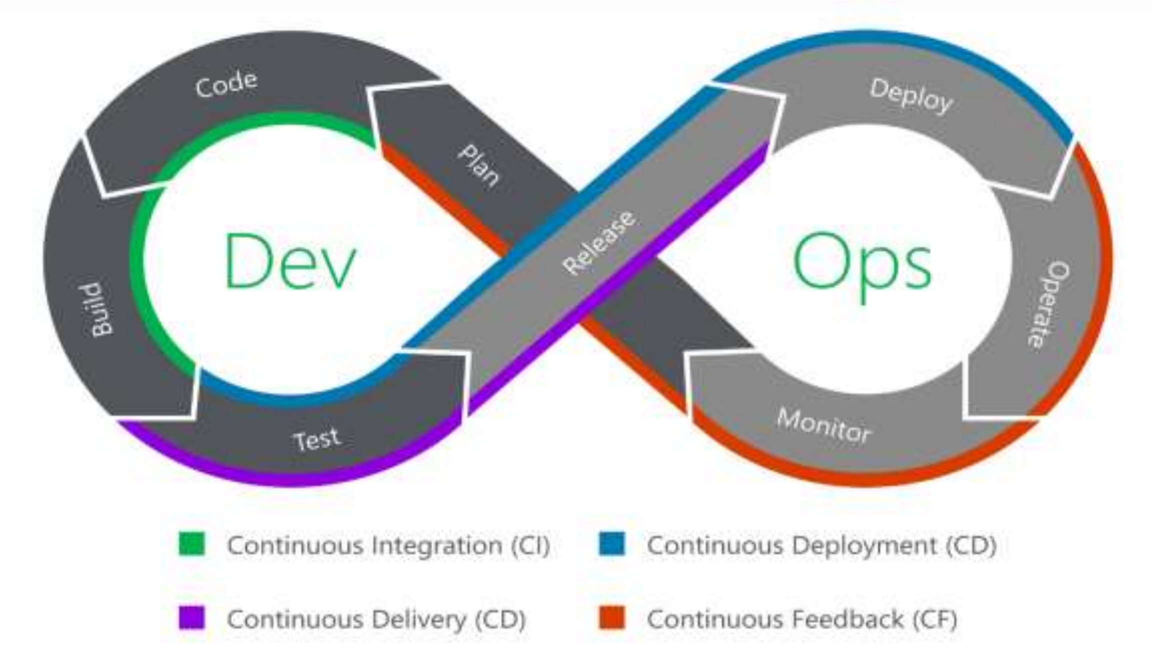
1. What is DevOps
2. What is the DevOps lifecycle
3. Fundamentals of code maintenance: Git branching strategies
4. Production maintenance: Observability

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

The DevOps Lifecycle



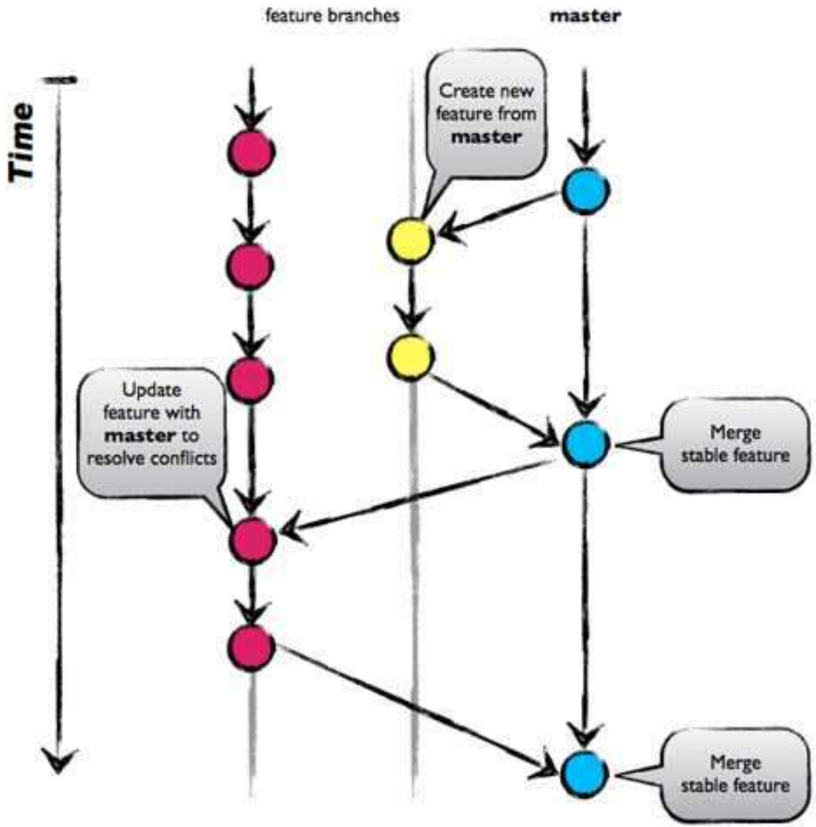
The DevOps Lifecycle: Continuous Everything!


Code
Plan
Build
Dev
Test
Release
Deploy
Ops
Operate
Monitor
🟩 Continuous Integration (CI)
🟦 Continuous Deployment (CD)
🟪 Continuous Delivery (CD)
🟥 Continuous Feedback (CF)

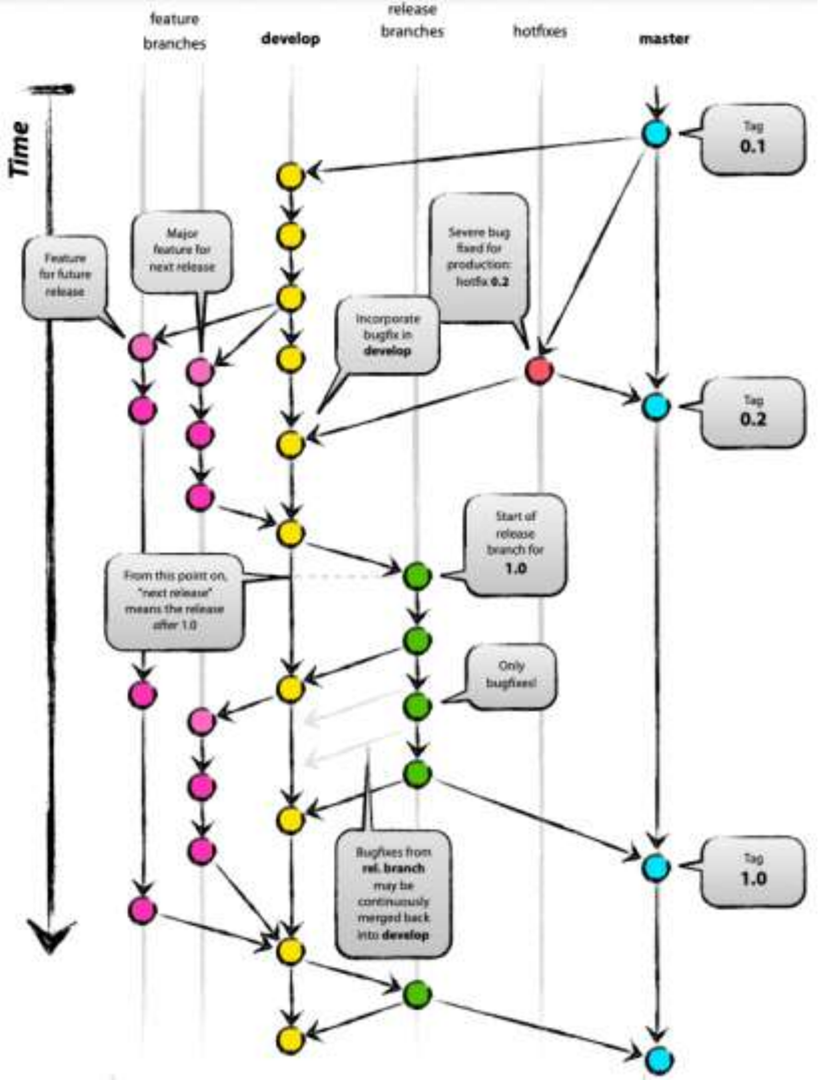
Git Branching Strategies

1. Github flow
2. Git flow
3. Trunk-based development

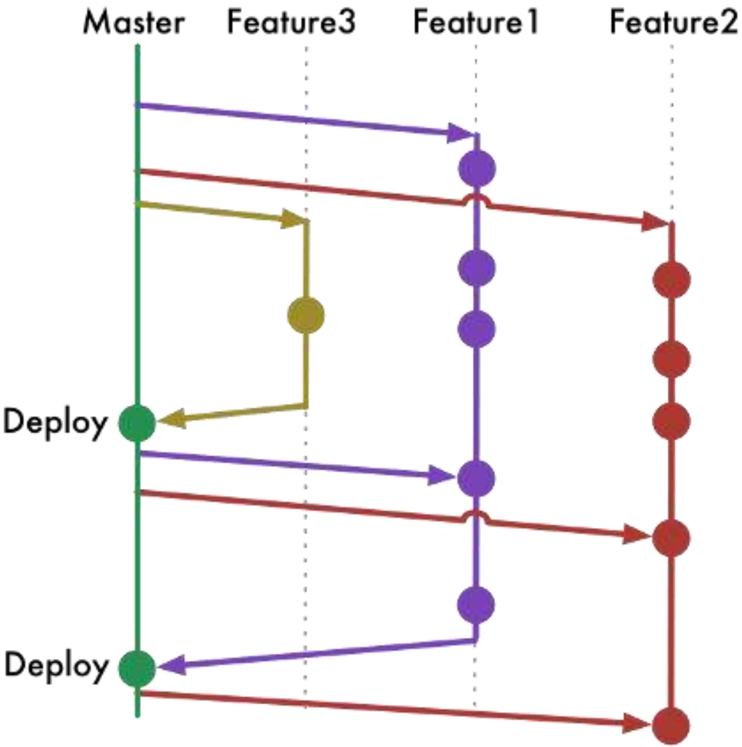
1. Github Flow


feature branches
master
Time
Create new feature from **master**
Update feature with **master** to resolve conflicts
Merge stable feature
Merge stable feature

2. Git Flow


feature branches
develop
release branches
hotfixes
master
Time
Tag 0.1
Feature for future release
Major feature for next release
Severe bug fixed for production: hotfix 0.2
Incorporate bugfix in develop
Tag 0.2
Start of release branch for 1.0
From this point on, "next release" means the release after 1.0
Only bugfixes!
Bugfixes from rel. branch may be continuously merged back into develop
Tag 1.0

3. Trunk-based Development



Production maintenance: Observability

Observability is the **ability to measure** the state of a system by **examining its outputs**. A system is considered “observable” if the current state can be estimated by only using information from outputs, namely sensor data.

The three pillars are observability are:

1. Metrics
2. Logs
3. Traces

In short known as MLT. Another popular version is MALT (MLT with Alerts)

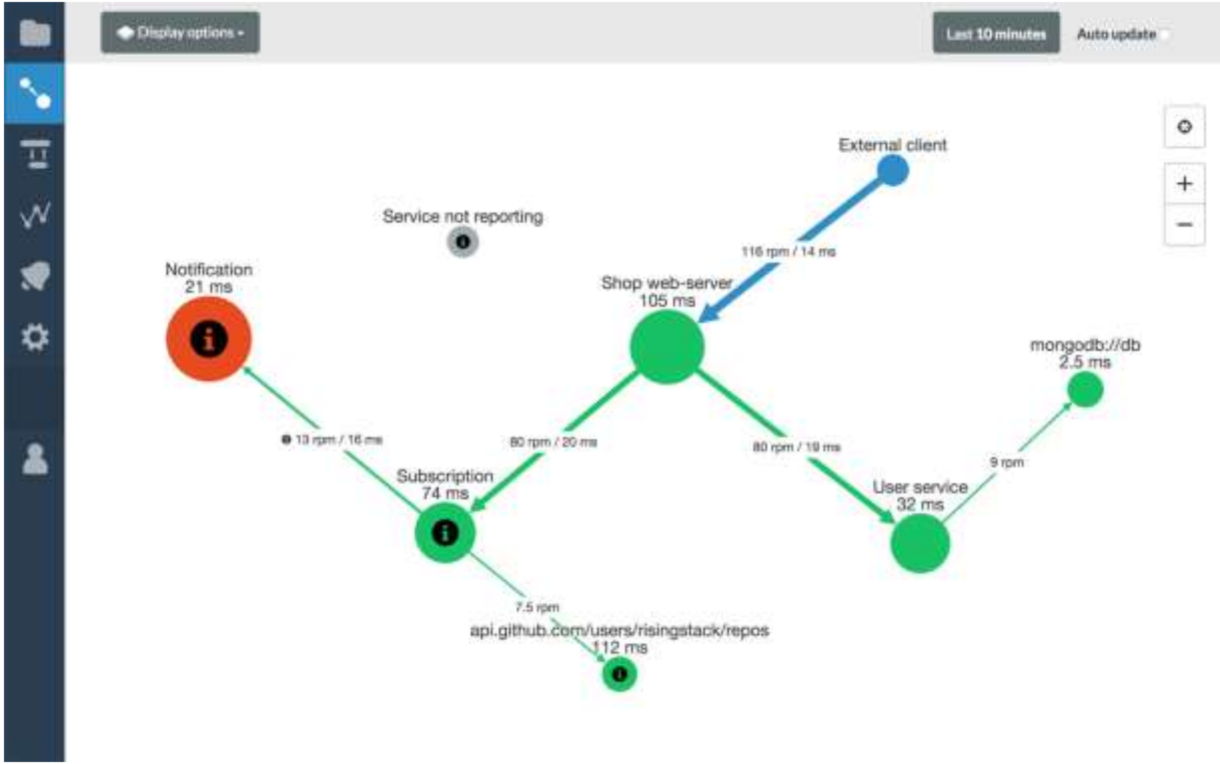
MLT Observability: Metrics

- How much CPU is being utilized?
- Is the RAM full?
- How much disk space is used?
- How many times did a process crash/restart?
- How many users are accessing the system?
- Are there non 20X responses? (40X, 50X)

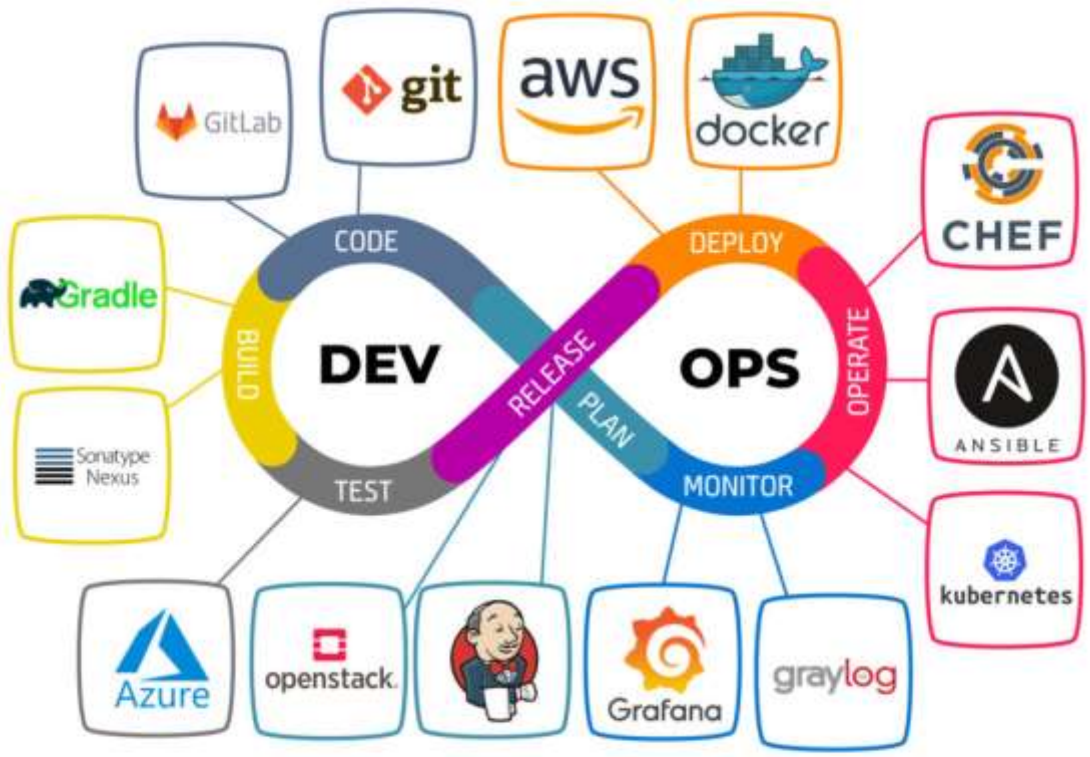
MLT Observability: Logs

- Record exceptions
- Common log levels:
  - DEBUG
  - INFO
  - WARN
  - ERROR
  - CRITICAL

MLT Observability: Tracing



The Heart of DevOps is Automation



Thanks Everyone!