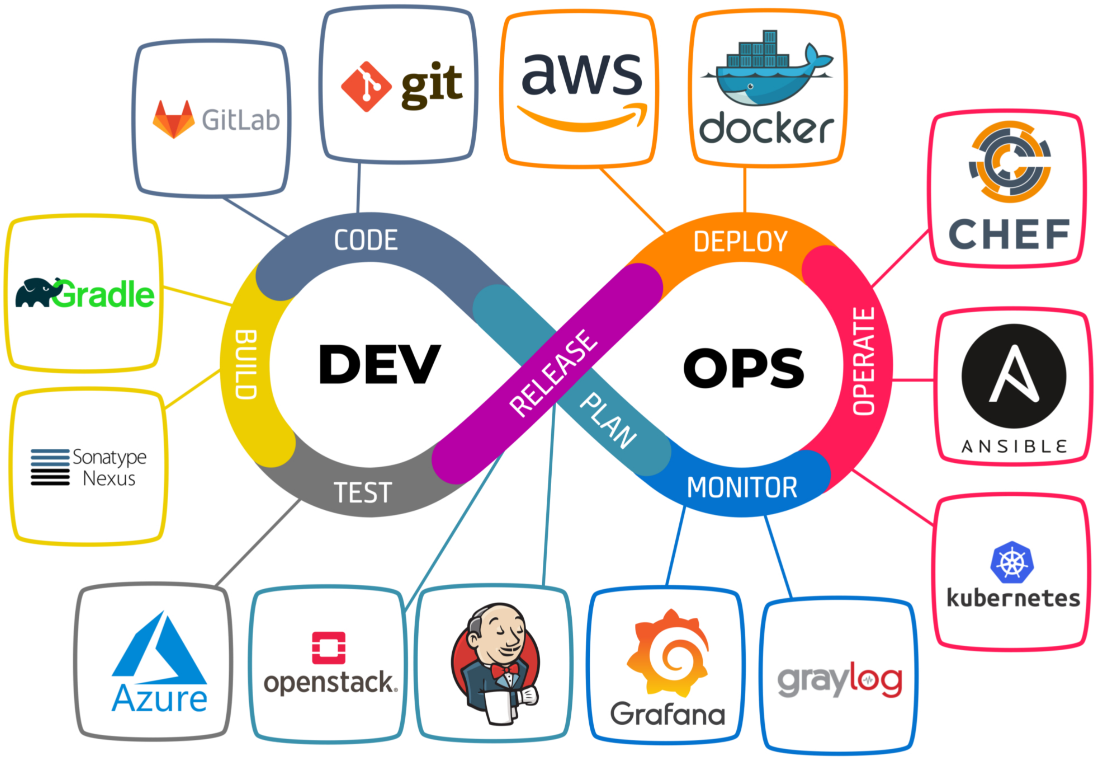
Overview of DevOps and CI/CD Processes

DevOps

- **Definition** : DevOps is a set of practices that combines software development (Dev) and IT operations (Ops) to shorten the systems development lifecycle and provide continuous delivery with high software quality.



- **Core Principles** :

- **Collaboration** : Breaks down silos between development and operations teams, fostering a culture of collaboration and communication.
- **Automation** : Emphasises automating repetitive tasks in building, testing, and deploying software to increase efficiency and reduce human errors.
- **Continuous Improvement** : Encourages iterative improvements in processes and tools to adapt to changing requirements and objectives.
- **Feedback Loops** : Improves decision-making and response times through constant feedback across all stages of the development lifecycle.

- **Benefits** :
  - Faster delivery of features and updates.
  - Improved deployment frequency and reduced failure rates.
  - Enhanced collaboration between teams leading to higher productivity.
  - Increased reliability and security of applications due to automated testing and monitoring.

Continuous Integration (CI)

- **Definition** : Continuous Integration is the practice of integrating code changes into a shared repository frequently, preferably multiple times a day. Each integration is verified by an automated build and test process to detect errors quickly.
- **Key Practices** :
  - **Version Control** : Code is stored in a version control system like Git, enabling collaboration and tracking of changes.
  - **Automated Builds** : Automated tools compile the code into executable artifacts to ensure the build process is consistent and quick.
  - **Automated Testing** : Unit tests and integration tests are run automatically to verify that code changes do not break the existing functionality.
- **Benefits** :
  - Early detection of integration bugs and errors.

    - Reduced time spent on manual testing and debugging.
    - Faster feedback loops, allowing developers to address issues immediately.

Continuous Delivery (CD)

- **Definition** : Continuous Delivery is an extension of CI and aims to automate the release process. It ensures that code is always in a deployable state and can be released into production at any time with minimal manual intervention.
- **Key Practices** :
    - **Automated Deployment Processes** : Scripts and tools automatically deploy updates to staging and production environments.
    - **Environment Consistency** : Ensures consistent environments across development, testing, and production stages.
    - **Frequent Releases** : Software is released in small, incremental updates, reducing risks associated with large releases.
- **Benefits** :
    - Reduced deployment risk and easier rollbacks.
    - Faster and more frequent delivery of features to customers.
    - Improved product quality due to consistent testing and validation in each stage.

DevOps vs CI/CD

Now that we have understood the basics of DevOps and CI/CD, let’s try to dive deep into DevOps vs CI/CD. These are two software development methodologies that have gained popularity in recent years. Both aim to improve the speed and quality of software development and deployment, but they do so in different ways. Here is a comparison of CI/CD and DevOps.

CI/CD	DevOps

Involves the use of automated testing and continuous integration	Involves the use of infrastructure as code, containerisation, and automation of infrastructure provisioning
Focuses on creating a rapid feedback loop for developers	Focuses on creating a culture of collaboration and shared responsibility between development and operations teams
Emphasizes the use of automation to reduce human error and improve consistency	Emphasizes the use of monitoring and logging to identify and resolve issues quickly
Tools include Jenkins, Travis CI, and CircleCI	Tools include Ansible, Puppet, and Chef
Involves the use of version control systems such as Git	Involves the use of agile methodologies and continuous testing
Focuses on streamlining and automating the software release process	Focuses on improving communication and collaboration between development and operations teams to achieve faster and more reliable software releases.

It’s worth noting that CI/CD and DevOps are not mutually exclusive, and often work together to improve the software development and deployment process.

Jenkins

Tekton