What is DevOps?

DevOps is a combination of "Development" and "Operations" it's a culture, a set of practices, and tools that bring together software development (Dev) and IT operations (Ops) teams to

- Increase collaboration and communication
- Automate processes
- Deliver software faster and more reliably

DevOps is not a tool or a single methodology , it's a philosophy that promotes agility, automation, continuous feedback, and cross-functional collaboration throughout the software development lifecycle (SDLC).

Core Principles of DevOps

1. Collaboration: Breaks down silos between Dev and Ops.
2. Automation: Automates repetitive tasks like testing, deployment, and infrastructure provisioning.
3. Continuous Integration (CI): Code is integrated and tested frequently (often multiple times a day).
4. Continuous Delivery (CD): Software is always in a deployable state.
5. Monitoring & Feedback: Real-time monitoring to detect and resolve issues quickly.
6. Infrastructure as Code (IaC): Manage infrastructure through code for consistency and repeatability.
7. Lean and Agile Thinking: DevOps borrows heavily from Agile and Lean methodologies to focus on customer value and iterative progress.

Advantages of DevOps

1. Faster Time to Market
   - Rapid and continuous delivery of new features and updates.
   - Automation reduces manual intervention and speeds up deployment.
2. Improved Deployment Frequency

    - Frequent, smaller, incremental updates instead of large releases.
    - Easier to test and roll out new changes.

3. Reduced Failure Rate of Releases
    - Automated testing and validation help catch bugs early.
    - Teams catch errors before reaching production.
4. Faster Recovery from Failures
    - Continuous monitoring allows rapid detection of failures.
    - Rollbacks or hotfixes can be deployed quickly.
5. Improved Collaboration & Communication
    - Shared responsibilities across Dev and Ops.
    - Everyone works toward the same goals using the same tools.
6. Continuous Feedback and Improvement
    - Real-time feedback from production environments.
    - Allows teams to make data-driven improvements quickly.
7. Cost Efficiency
    - Automation reduces manual work and operational overhead.
    - Efficient use of infrastructure through containerization and cloud-native tools.
8. Enhanced Security
    - Automated security checks (DevSecOps).
    - Early identification of vulnerabilities during the CI/CD process.
9. Scalability and Flexibility
    - Easily scale applications using orchestration tools like Kubernetes.
    - Infrastructure as Code allows replicable, scalable infrastructure.

SCRUM EVENTS AND ARTIFACTS

To support the Scrum process, several key **events (ceremonies)** and **artifacts** are used regularly during each Sprint:

Scrum Events

1. **Sprint**
   - A fixed-length development cycle (usually 2–4 weeks).
   - The team commits to delivering a set of features or work items.
2. **Sprint Planning**
   - Held at the beginning of each Sprint.
   - The team and Product Owner decide what work will be done in the Sprint.
   - The team creates a **Sprint Goal** – a clear objective for the Sprint.
3. **Daily Scrum (Stand-up)**
   - A short (15-minute) daily meeting.
   - Team members share what they did yesterday, what they'll do today, and any blockers.

Scrum Artifacts

1. **Product Backlog**
   - A list of all desired work on the project, managed by the Product Owner.
   - Includes features, bugs, technical work, and improvements.
   - Items are prioritized based on business value.
2. **Sprint Backlog**
   - A subset of the Product Backlog selected for the current Sprint.
   - Includes all tasks the team commits to completing in the Sprint.
3. **Increment**
   - The usable and potentially shippable product created at the end of a Sprint.