CI/CD Pipelines

⇒ A CI/CD pipeline is a concept central to software. It spans a whole field of processes, testing methods and tooling, all facilitated by git code versioning process.

⇒ Continuous Integration (CI): Every time you add a new code or change, you immediately have to test it by running the data through it (pipeline).
This ensures new addition doesn't cause any problems. If there is an issue, you can resolve it and fix it before it becomes a bigger problem.

⇒ ~~Contiguor~~

⇒ Continuous Deployment (CD):- Once you're confirmed that the new code/addition or change fits perfectly, you can made the changes live and functional on the main track.

⇒ Use Toy Train eg. for easy explaination

⇒ CI/CD, in the context of pipelines deployment, focuses on automating data operations and transformations.

⇒ CI in Data Pipelines.

- Automated Testing :- They check the integrity and quality of data transformations, ensuring that data is processed ~~as~~ expected and any error is spotted early.

- Version Control :- Data pipeline code is stored in repositaries like Git, allowing tracking & managing changes

- Consistent Environment :-
- Data Quality Checks :- Include checks for null values, data range violations, data type mismatches, or other custom quality rules.

⇒ Continuous Data Pipeline Deployment (CD)

- Automated Deployment
- Monitoring and Alerts
- Rollbacks
- Infrastructure as Code (IaC)

⇒ While CI/CD is a concept, there are various tools and frameworks developed to implement and support CI/CD practices, such as Jenkins, GitLab CI/CD, Travis CI, Circle CI, and many others.