24/2/24

CI/CD :-

* A CI/CD pipeline is a pipeline concept central to software
* It includes a whole field of processes, testing & tooling all facilitatinged by Git code versioning process.

Ex :- Building a toy train truck

Continuous Integration :- Every time we add a new

track piece, we have to test immediately by running toy train (data)

* It means adding didn't create any problems.

Continuous Deployment :-

once new piece fits & train runs we'll show it to everyone and use it.

* It means as soon as changes are verified, they are made live & functional in production environment.

Technically

CI ⇒ checks & tests every new piece of code (or logic) you add to data pipeline

CD ⇒ ensures once tested & approved, this code gets added to live system without manual intervention.

In data pipelines

* CI/CD in context of data pipeline deployment focuses on automating data operations and transformations.

* This merges development, testing and operational workflows into unified, automated process, ensuring data sets are consistently high quality.

* It has different applns like

(i) training ML models
(ii) supporting data science team
(iii) large-scale data analysis
(iv) BI or Data visualization
(v) unstructured data collection

CI in data pipelines

1) Automated Testing => check integrity & quality
2) Version Control => pipeline code is stored in repositories like Git
3) Consistent Environment =>
4) Data Quality checks => (null values, data type mismatches)

CD in datapipelines

1) Automated Deployment => automates deployment to production
2) Monitoring & Alerts => keep track of performance, quality
3) Rollbacks => used in case of issues
4) Infrastructure as Code (IaC) => cloud resources like storage can be provisioned automatically as part of deployment process.

Tools to support CI/CD process

1) Jenkins
2) GitLab CI/CD
3) Travis CI
4) Circle CI

GIT

* distributed version control system that facilitates collaborative s/w development by tracking changes across multiple contributors.
* can be paired with data orchestration tools & integrated into CI/CD workflows.

ETL pipelines

* Process that pull data from sources (databases, APIs) transform into usable format & then load into destination like databases, warehouses etc.,

Deploying a ETL script to Git

1) Testing
2) Deployment
3) Notifications (in case of success / failure)

Datapipeline deployment

* Git's primary function is version control, it's integration with CI/CD solutions makes it powerful deployment tool.
* This means when we push code to Git repository it can automatically be deployed to production environment directly.
* Data professionals integrate Git and CI/CD into their workflows to automate repetitive tasks,

ensure data quality, focus on optimizing data pipelines

Common workflows

1) Data Validation
2) Scheduled data jobs
3) Catching anomalies & failures
4) Novel workflows.

Git Best Practices

1) Handling large data files ⇒ use (i) versioning cloud storage buckets
   (ii) dedicated data versioning tools
2) Pull Requests
3) Code Reviews
4) Commit often
5) Commit with clear messages
6) Branch deployments.