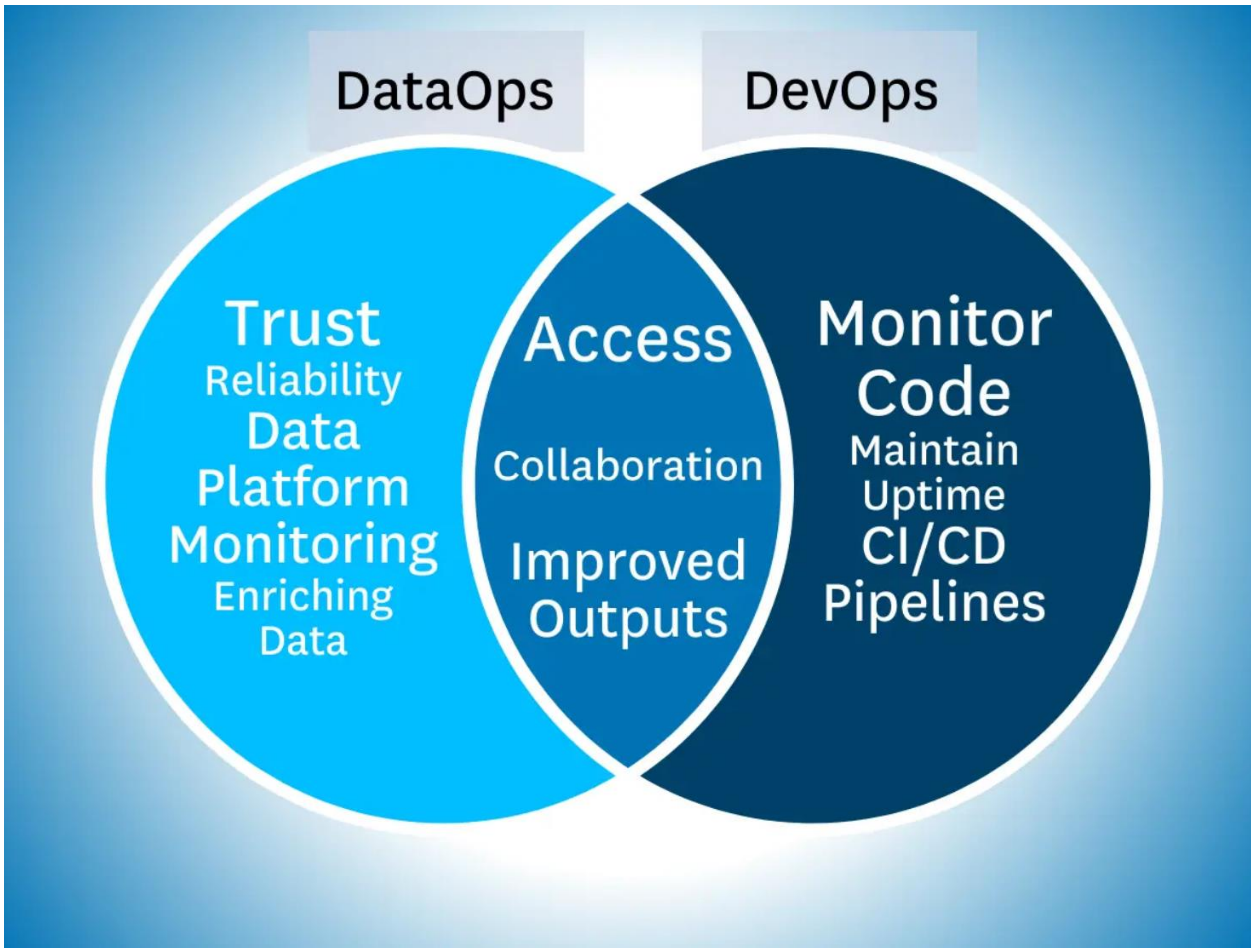
DevOps and DataOps



- **Key differences between DevOps and DataOps:**

DataOps	DevOps
Makes sure that data is reliable, trustworthy, and not broken	Makes sure that the website or software (mainly backend) is not broken with addition, alteration, or removal of code
DataOps' ecosystem consists of databases, data warehouses, schemas, tables, views, integration logs from other key systems	Here CI/CD pipelines are built, automation of code is discussed, and uptime & availability are improved constantly
DataOps is platform agnostic. It is a group of concepts that can be put into practice where there is data.	DevOps is also platform agnostic however the cloud companies have streamlined the DevOps playbook
DataOps focuses on monitoring and enriching data	DevOps focuses on the code

The Four Phases of the DevOps Lifecycle:

1. Planning:

This is the ideation phase; tasks are created and are backlogged based on priority. Multiple products will lead to multiple backlogs. The waterfall approach does not work well with DevOps tasks, therefore, agile methodologies like Scrum or Kanban are used.

2. Develop:

This phase consists of coding, writing, unit testing, reviewing, and integrating code with the existing system. Upon successful development of code, the same is prepared for deployment into various environments. DevOps teams automate mundane and manual steps. They achieve stability and confidence by incrementing in a small manner. This is where continuous deployment and continuous integration arises.

3. Delivery:

In this phase, the code is deployed in an appropriate environment. This could be prod, pre-prod, staging, etc. Regardless of where the code is deployed it is deployed in a consistent and reliable way. The Git language has made it easy to deploy code on almost all popular servers by just typing a few lines of code.

4. Operate:

This is the phase that involves monitoring, maintaining, and fixing applications in production. This is the actual place where downtime is spotted and reported. DevOps teams identify issues using tools like PagerDuty in the operational phase before their customers find out about them.

DataOps Lifecycle Look Like:

1. Planning:
Partnering with product, engineering, and business teams to set KPIs, SLAs, and SLIs for the quality and availability of data.
2. Development:
Building the data products and machine learning models that will power your data application.
3. Integration:
Integrating the code and/or data product within your existing tech and or data stack. For example, you might integrate a dbt model with Airflow so the dbt module can automatically run.
4. Testing:
Testing your data to make sure it matches business logic and meets basic operational thresholds (such as uniqueness of your data or no null values).
5. Release:
Releasing your data into a test environment.
6. Deployment:
Merging your data into production.
7. Operate:
Running your data into applications such as Looker or Tableau dashboards and data loaders that feed machine learning models.
8. Monitor:
Continuously monitoring and alerting for any anomalies in the data.

What is a data warehouse?

A data warehouse is a central repository of information that can be analyzed to make more informed decisions. Data flows into a data warehouse from transactional systems, relational databases, and other sources, typically on a regular cadence. Business analysts, data engineers, data scientists, and decision makers access the data through business intelligence (BI) tools, SQL clients, and other analytics applications.

What are the benefits of using a data warehouse?
Benefits of a data warehouse include the following:

- **Informed decision making**
- **Consolidated data from many sources**
- **Historical data analysis**
- **Data quality, consistency, and accuracy**
- **Separation of analytics processing from transactional databases, which improves performance of both systems**

What is Data Normalization?

One of the most popular methods for preparing data is Normalization, which enables us to alter the values of numerical columns in the dataset to a standard scale.

Normalization is the method used to arrange the data in a database. It is a scaling method that reduces duplication in which the numbers are scaled and moved between 0 and 1. When there are no outliers since it can't handle them, normalization is employed to remove the undesirable characteristics from the dataset.

What is Data Standardization?

Standardization, often referred to as z-score Normalization, occasionally is a method for rescaling the values that meet the characteristics of the standard normal distribution.

Standardization is crucial because it enables reliable data transmission across various systems. It would be easier for computers to exchange data and communicate with one another with standardization. Additionally, standardization makes it simpler to process, analyze, and store data in a database. Businesses can use their data to make better judgments with this method. Companies can more readily compare and evaluate data when standardized, allowing them to gain insights into how to run their businesses better.

Standardization also lacks a bounding range, in contrast to normalizing. Therefore, normalization will have no effect on any outliers you may have in your data.

Normalization	Standardization
This method scales the model using minimum and maximum values.	This method scales the model using the mean and standard deviation.
When features are on various scales, it is functional.	When a variable's mean and standard deviation are both set to 0, it is beneficial.
Values on the scale fall between [0, 1] and [-1, 1].	Values on a scale are not constrained to a particular range.
Additionally known as scaling normalization.	This process is called Z-score normalization.
When the feature distribution is unclear, it is helpful.	When the feature distribution is consistent, it is helpful.

Some effective strategies for handling missing data in a dataset:

1. **Removing rows or columns with a high proportion of missing values.**
2. **Imputing missing values using statistical methods such as mean, median, or mode.**
3. **Predictive modeling to estimate missing values based on other variables.**
4. **Using algorithms that can handle missing data, such as decision trees or random forests.**
5. **Employing techniques like multiple imputation to generate multiple plausible values for missing data.**
6. **Considering the reasons for missingness and incorporating domain knowledge to inform handling strategies.**