DATA ORCHESTRATION: THE CORE PILLAR FOR DATAOPS

Introduction

Organizations are drowning in data and thirsty for knowledge, wisdom, and insights. Data engineering teams in aspiring data-driven organizations are overwhelmed with fast-changing technology and organizational complexities as they look to move from proof of concept (POC) to proof of value (POV) and establish a sustainable operating model with continuous improvement.

For most organizations today, data unification and data integration challenges are growing overwhelmingly complex as they gravitate toward best-of-breed tools in a disaggregated data ecosystem. Data engineering that leverages DataOps and data orchestration is the foundational pillar on which organizations should build their next-generation data platforms in an ever-evolving data ecosystem to scale data teams with all the inherent process variability.

Why do Organizations Need DataOps

All organizations want to be data-driven but there's a huge disconnect between wanting to be data-driven and getting it done. Bleeding and cutting-edge technologies are immature and not battle-tested and will not get organizations there. It is the operationalization process of technologies that is the key for organizations to become data-driven.

Most data teams do not think about "Day 2" which begins when product teams have completed

development and successfully deployed to production. Do they have an end-to-end process to deploy artifacts? Have they tested what they are about to deploy with functional performance, load, and stress tests? Are they ready to roll back production changes if problems happen in production but keep the lights running?

There is a disconnect between doing POCs and POVs with emergent technologies and leveraging them to build and successfully deploy real-life use cases to production. There are a few reasons for this disconnect and most of them can be addressed by the missing component in the data economy: DataOps. Many organizations do DataOps, but it is ad hoc, fragmented, and built without guidelines, specifications, and a formalized process.

Data infrastructures today, spanning ingestion, storage, and processing, are deployed on distributed systems that include on-premises, public and private cloud, hybrid, and edge environments. These systems are a complex mix of servers, virtual machines, networking, memory, and CPU where failures are inevitable. Organizations need tools and processes in place that can quickly do root cause analysis and reduce the mean time to recovery (MTTR) from failures.

DataOps eliminates gaps, inefficiencies, and misalignments across the different set of steps from data production to consumption. It coordinates and orchestrates the development and operationalization processes in a collaborative, structured, and agile manner, enabling organizations to streamline data delivery and improve productivity through multiple process integrations and automations, delivering the velocity to build and deploy data-driven applications with trust and governance.

What is DataOps?

DataOps streamlines and automates data processes and operations to inform and speed the building of products and solutions and help organizations become data-driven. The goal of DataOps is to move from ad hoc data practices to a formalized data engineering approach with a controlled framework for managing processes and tasks.

To become data-driven, organizations need tools and processes that automate and manage end-to-end data operations and services. DataOps allows organizations to deliver these data products and services with velocity, reliability and,efficiency, with automation, data quality and trust.

DataOps is accomplished through a formal set of processes and tools that detect, prevent, and mitigate issues that arise when developing data pipelines and deploying them to production. This improves operational efficiency for building data products and data-driven services. DataOps applies ideas of continuous integration and continuous delivery (CI/CD) to develop agile data pipelines and promotes reusability with data versioning to collaboratively manage data integration from data producers to consumers. It also reduces the end-to-end time and costs of building, deploying, and troubleshooting the building of data platforms and services.

For organizations investing in digital transformation with analytics, artificial intelligence (AI), and machine learning (ML), DataOps is an essential practice to manage and unlock data assets to yield better insights and improve decision-making and operational efficiency. DataOps enables innovation through decentralization while harmonizing domain activities in a coherent end-to-end pipeline of workflows. It handles global orchestration, in a shared infrastructure with inter-domain dependencies enabling policy enforcements.

Why organizations need data orchestration

The data ecosystem today is flooded with a rich ecosystem of tools, frameworks, and libraries with no "one tool to rule them all." Some tools are programmatically accessible with well-defined APIs, while others are invoked through APIs or command lines to integrate with other processes in the ecosystem. With the disaggregated data stack, enterprises must stitch together a plethora of different tools and services to build end-to-end data driven systems.

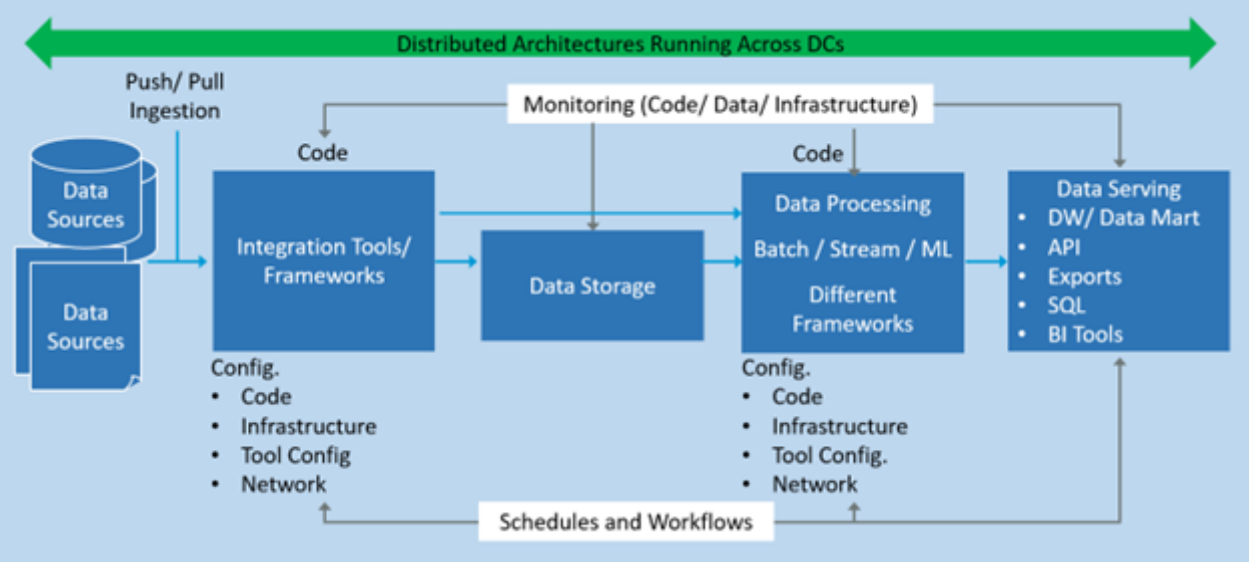

Figure 1. A cross-sectional view of a simple data pipeline.

Most organizations do not have handful number of data pipelines; they have tens and hundreds of them, with complex deployment models that span from edge to on-premises and across multiple cloud data centers. Components are deployed on different data centers and each component is built with distributed architecture and sub-components running across different servers and virtual machines. There are multiple handshaking points, each being a failure point—from network to server outages.

When architecting and building these pipelines, data engineers need to guarantee that the pipeline code executes as intended and that it is built with data engineering best practices and operational rigor. Handling both good cases and bad cases when things go wrong is critical. The goal is to coordinate all the pieces and make sure that the entire pipeline is resilient, can handle failover, and recover from the point of failure.

Production data pipelines are complex with multiple forks and dependencies and different mix of trigger logic for the tasks. All this is managed by the orchestrator's scheduling and workflow engine. Data orchestration sits at the center of increasingly complex data operations. It controls and coordinates data operations in workflows that involve multiple stakeholders across the technical spectrum. It keeps track of what has happened and when and what still needs to happen for a successful pipeline execution. The orchestration engine triggers jobs and coordinates multiple tasks within the job to enforce dependencies. It logs actions and maintains traces for audits to provide a comprehensive view of the status of troubleshooting tasks.

Data and ML engineers leverage data orchestration for use cases like data ingestion/data integration, data processing, and data delivery. The most important data orchestration capabilities requested by data engineers and data scientists include ease of use, monitoring, and easy debugging and observability of their data pipelines. The end goal is to enhance productivity and to

architect, develop, and deploy well-engineered data pipelines with monitoring, testing, and quick feedback loops.

What is data orchestration?

Modernizing an organization's data infrastructure can become increasingly difficult and error-prone without a data orchestrator in a data engineering team's toolkit. A long list of data engineering tasks needs to be accomplished before one can start working on the real business problem and build valuable data products. These tasks include provisioning infrastructure for data ingestion, storage, processing, consumption, testing, and validation, handling failures, and troubleshooting.

Data orchestration is the glue between these tasks across the data stack that manages dependencies, coordinates the tasks in the defined workflow, schedules the tasks, manages the outputs, and handles failure scenarios. It is a solution that is responsible for managing execution and automation of steps in the flow of data across different jobs. The orchestration process governs data flow according to the orchestration rules and business logic. It schedules and executes the workflow that is associated with the data flow.

This flow structure is labelled as a dependency graph, also called a DAG. Data orchestration is this process of knitting and connecting the tasks into a chain of logical steps to build an end-to-end data pipeline that is well coordinated.

Data orchestrators can be of two types - task-driven or data-driven. The former is when the data orchestrator doesn't care about what's the input or output of a step in a pipeline; its only focus is orchestrating a workflow. Data-driven orchestrators not only orchestrate the workflows, but are aware of the data that flows between tasks and their artifact outputs that can be version-controlled and allow tests to be associated with them.

A good orchestrator lets data teams quickly isolate errors and failures. Data orchestrators can be triggered according to a time-based trigger or custom-defined logic. The infrastructure required by data domains can be unified into a self-service infrastructure-as-a-platform managed using a DataOps framework.

Some of the best practices with data orchestrators include decoupling the DAGs to break them down to the simplest of dependencies with sub-DAGs. Making the DAG fault-tolerant so that if one of the sub-DAG breaks, they can be easily re-executed.

Data orchestration should be configuration-driven to ensure portability across environments and can provide repeatability and reproducibility. Other best practices include making data orchestration process single-click, with checkpoints to enable recovery of broken data pipelines from the point of failure and ensuring the ability to retry failed tasks with a configurable backoff process.

Conclusion

Enterprises are cautioned against jumping into building data platforms for data-driven decision-making without incorporating principles of data engineering and DataOps. These synergistic capabilities will provide organizations with the necessary process formality and velocity and reduce data and technical debt in the long run.