데이터 워크플로우

워크플로우와 파이프라인

"워크플로우"와 "파이프라인"은 종종 상호 교환이 되어 사용되지만, 세부적인 차이점이 있습니다. 그들이 가리키는 개념은 서로 비슷하지만, 주로 사용되는 문맥이나 의미하는 바에 약간의 차이가 있습니다.

- **워크플로우(Workflow)**:

  ▶ 세부내용

  워크플로우는 일련의 작업들을 정의하며, 이들 작업은 특정 결과를 달성하기 위해 순서대로 또는 병렬로 수행될 수 있습니다. 워크플로우는 더 넓은 의미를 가지며, 여러 분야에서 사용되는 개념입니다. 예를 들어, 소프트웨어 개발, 사업 프로세스 관리, 데이터 분석 등에서 워크플로우 개념이 사용됩니다. 워크플로우는 작업의 실행 순서, 병렬 처리, 오류 처리, 재시도 로직 등을 정의할 수 있습니다.

- **파이프라인(Pipeline)**:

  ▶ 세부내용

  파이프라인은 일련의 데이터 처리 단계를 나타냅니다. 각 단계는 독립적으로 동작하며, 한 단계의 출력은 다음 단계의 입력이 됩니다. 데이터 파이프라인은 주로 데이터 처리, 변환, 저장을 목적으로 사용되며, ETL(Extract, Transform, Load)이 대표적인 예입니다. 파이프라인은 보통 일련의 순차적인 작업으로 구성되며, 각 단계는 이전 단계의 출력에 의존합니다.

[파이프라인 도구 소개]

파이프라인 도구는 대부분의 경우 데이터를 전처리하고, 이를 분석하거나 저장하기 위한 과정을 자동화하는데 사용됩니다. 데이터 파이프라인 도구는 ETL(Extract, Transform, Load) 도구라고도 불립니다. 아래는 몇 가지 주요 파이프라인 도구에 대한 설명입니다.

1. **Apache Airflow**:

   ▶ 세부내용

   Airflow는 Python으로 작성된 오픈 소스 워크플로우 관리 플랫폼으로, 복잡한 계산을 설계, 구성, 실행하고 모니터링하는 데 사용됩니다. 데이터 파이프라인을 자동화하고 스케줄링하는 데 강점을 가지고 있습니다.

2. **Apache Beam**:

   ▶ 세부내용

Beam은 배치 및 스트리밍 데이터 처리 작업을 캡슐화하고 이를 실행하는 일관된 프로그래밍 모델을 제공합니다. Beam 파이프라인은 런타임에 특정 실행 엔진(예: Apache Flink, Apache Samza, Google Cloud Dataflow 등)에 대한 구체적인 지식 없이 작성할 수 있습니다.

3. **Apache NiFi**:

▶ 세부내용

NiFi는 실시간 데이터 플로우를 자동화하고 제어하는데 사용되는 시스템입니다. GUI를 통해 쉽게 데이터 플로우를 만들고 모니터링할 수 있습니다.

4. **Luigi**:

▶ 세부내용

Luigi는 Spotify에서 만든 파이프라인 도구로, 복잡한 배치 작업을 구성하고 실행할 수 있습니다. 파이프라인의 여러 단계 간의 의존성을 관리하는 데 특히 유용합니다.

5. **Prefect**:

▶ 세부내용

Prefect는 최근에 개발된 파이프라인 도구로, Airflow의 기능과 유사하나 몇 가지 주요 차이점이 있습니다. Prefect는 동적인 워크플로우를 지원하며, 파이프라인 실패 시 자동 복구 메커니즘이 뛰어납니다.

6. **Apache Flink**:

▶ 세부내용

Apache Flink은 스트리밍 데이터를 처리하는데 특화된 오픈 소스 데이터 처리 엔진입니다. 배치 데이터 처리도 가능하지만, 주로 실시간 데이터 스트리밍 처리에 초점을 두고 있습니다. Flink는 분산 데이터 처리에 사용되며, 빅 데이터를 높은 처리 속도와 저지연으로 처리할 수 있는 강력한 기능을 제공합니다. 또한, Flink는 '정확한 시간 처리'를 지원하기 때문에, 시간에 따른 이벤트 처리와 같은 복잡한 스트리밍 애플리케이션을 구현할 수 있습니다.

[워크플로우 도구 소개]

워크플로우 도구는 일련의 작업을 자동화하고, 관리하며, 모니터링하는데 도움을 주는 도구들입니다. 워크플로우 도구는 다양한 분야에서 사용되며, 데이터 처리부터 CI/CD까지 다양한 환경에서 활용됩니다. 아래는 주요 워크플로우 도구들의 예시입니다:

1. **Apache Airflow:**

▶ 세부내용

Airflow는 복잡한 계산을 설계, 구성, 실행하고 모니터링하는 데 사용되는 Python으로 작성된 오픈 소스 워크플로우 관리 플랫폼입니다. 데이터 파이프라인을 자동화하고 스케줄링하는 데 강점을 가지고 있습니다.

2. **Argo**:

▶ 세부내용

Argo는 쿠버네티스 기반의 워크플로우 엔진으로, 일련의 태스크를 조정하고 실행하는 데 사용됩니다. Argo는 CI/CD, ML 워크플로우 등 다양한 쿠버네티스 기반 워크플로우를 지원합니다.

3. **Jenkins**:

▶ 세부내용

Jenkins는 CI/CD 파이프라인을 구축하고 관리하기 위한 오픈 소스 도구입니다. 소프트웨어 개발에서 사용되며, 빌드, 테스트, 배포 등의 과정을 자동화합니다.

4. **Luigi**:

▶ 세부내용

Luigi는 Spotify에서 만든 파이프라인 도구로, 복잡한 배치 작업을 구성하고 실행할 수 있습니다. 파이프라인의 여러 단계 간의 의존성을 관리하는 데 특히 유용합니다.

5. **Zapier**:

▶ 세부내용

Zapier는 클라우드 기반의 워크플로우 자동화 도구로, 다양한 웹 애플리케이션 간에 작업을 자동화하는 데 사용됩니다.

6. **Prefect**:

▶ 세부내용

Prefect는 최근에 개발된 파이프라인 도구로, Airflow의 기능과 유사하나 몇 가지 주요 차이점이 있습니다. Prefect는 동적인 워크플로우를 지원하며, 파이프라인 실패 시 자동 복구 메커니즘이 뛰어납니다.

파이프라인 및 워크플로우 도구 비교
> Argo, Apache Airflow, Apache Flink

"Argo", "Apache Airflow", 그리고 "Apache Flink"는 모두 데이터 처리와 워크플로우 관리에 사용되는 도구이지만, 각각의 특징과 사용 사례는 매우 다릅니다. 아래에서 각 도구의 장점, 단점, 그리고 차이점에 대해 살펴보겠습니다.

Argo

장점

- Argo는 쿠버네티스(Kubernetes)의 네이티브 워크플로우 엔진으로, 쿠버네티스 클러스터 내에서 컨테이너화된 워크플로우를 실행하도록 설계되었습니다.
- Argo는 여러 서브 프로젝트를 제공하여 CI/CD, GitOps, ML 워크플로우 관리 등 다양한 사용 사례를 지원합니다.

단점

- Argo는 쿠버네티스 위에 구축되었으므로, 쿠버네티스 없이 Argo를 실행하는 것은 불가능합니다. 따라서 쿠버네티스를 사용하지 않는 환경에서는 Argo의 사용이 제한될 수 있습니다.

차이점

- Argo는 쿠버네티스 환경에서 워크플로우와 파이프라인을 관리하는데 특화되어 있습니다.

Apache Airflow

장점

- Airflow는 데이터 처리 파이프라인을 자동화하고 스케줄링하는 데 강점을 가지고 있습니다.
- Python으로 작성된 Airflow는 사용자가 복잡한 워크플로우를 프로그래밍적으로 정의하고 시각화할 수 있게 해주며, 다양한 데이터 소스와 목적지와의 통합을 지원합니다.

단점

- Airflow의 설정과 배포는 다소 복잡할 수 있습니다.
- 또한, 실시간 데이터 처리에는 덜 적합한 구조를 가지고 있습니다.

차이점

- Airflow는 ETL 작업과 데이터 파이프라인의 작성 및 스케줄링에 초점을 맞추고 있습니다.

Apache Flink

장점

- Flink는 스트림 처리에 초점을 맞춘 데이터 처리 엔진입니다.
- 대량의 데이터를 실시간으로 처리하는 데 강점을 가지고 있으며, 높은 처리량과 낮은 지연 시간을 지원합니다.
- 또한, 일괄 처리와 스트림 처리를 동일한 엔진에서 처리할 수 있습니다.

단점

- Flink는 스트림 처리에 최적화된 엔진이므로, 단순한 배치 작업이나 ETL 작업에 비해 과도하게 복잡하고 무거울 수 있습니다.

차이점

- Flink는 대용량 실시간 데이터 처리에 특화되어 있습니다.