

Handleiding: Data Pipelines Bouwen en Begrijpen

1. Wat is een Data Pipeline?

Een data pipeline is een geautomatiseerde reeks stappen die data ophaalt, verwerkt en verplaatst van bron naar bestemming. Denk aan het ophalen van data via een API, het transformeren naar een bruikbaar formaat, en het laden in een database of dashboard.

2. Belang van Pipelines in Data Engineering

Data pipelines zorgen voor herhaalbaarheid, schaalbaarheid en monitoring van je dataprocessen. In plaats van handmatig te exporteren en importeren, laat je de pipeline automatisch draaien, bijvoorbeeld elke nacht of elk uur.

3. Voorbeeldarchitectuur

Een typische pipeline heeft de volgende lagen: - Bron (bijv. API, database, CSV) - Extractie - Transformatie - Validatie - Laden (bijv. datawarehouse of datalake) - Monitoring

4. Tools voor Pipelines

- Azure Data Factory: visuele low-code ETL-pipelines - Apache Airflow: open-source scheduler met Python DAG's - dbt: transformaties met SQL, versioneerbaar - Prefect: moderne workflow tool - Python scripts: flexibel en krachtig

5. Voorbeeld met Python & Pandas

```
import pandas as pd
import requests

# Extract
resp = requests.get('https://api.example.com/sales')
data = pd.DataFrame(resp.json())

# Transform
filtered = data[data['amount'] > 100]
filtered['region'] = filtered['zipcode'].str[:2]

# Load
filtered.to_csv('output/sales_clean.csv', index=False)
```

6. Scheduling en Automation

Gebruik cronjobs (Linux), Task Scheduler (Windows), of cloud services (Azure Data Factory triggers, Airflow schedulers) om je pipelines automatisch te laten draaien.

7. Logging en Foutafhandeling

Implementeer try/except-blokken in Python, of gebruik logboeken (bijv. met logging-module). Log elke stap zodat je makkelijk ziet waar het fout ging.

8. Testen en Monitoring

Test je pipeline met unit tests voor scripts, en gebruik tools zoals Great Expectations voor data-validatie. Monitor de uitvoering en stel alerts in bij fouten of afwijkingen.

9. CI/CD voor Pipelines

Gebruik GitHub Actions of Azure DevOps om je pipelines automatisch te testen en uit te rollen. Versiebeheer helpt bij samenwerking en betrouwbaarheid.

10. Best Practices

- Houd je pipeline modulair (extract, transform, load gescheiden) - Hergebruik code waar mogelijk - Documenteer je pipeline met uitleg en flow-diagram - Zorg voor monitoring, logging én fallback scenario's

Conclusie

Met een goede data pipeline voorkom je handmatig werk en vergroot je de betrouwbaarheid van je dataprocessen. Begin klein, automatiseer slim, en leer al doende. Veel succes!