Course Content

Content

Section 1: Introduction

- Course Content
- Course Information
- Source Files

Content

Section 2: Introduction to Data Engineering

- The Data Problem!
- Data Engineering Definition - ETL Vs ELT
- Data Engineering Ecosystem
- Why Data Engineering Matters?
- Data Engineering Vs Other Roles
- Importance of Data Engineering-Real World Example

Content

Section 3: Required Installations

- Git Installation
- GitHub Repo & VSCode
- Virtual Environ and Python Packages
- Docker Desktop Installation

Content

Section 4: Hands-On Project Introduction

- Introduction
- Understanding the Data
- AWS Account
- Data Storage Paradigms
- AWS Credentials
- AWS S3 Bucket
- AWS Connection and Test

Content

Section 5: Hands-On Project - Data Ingestion

- Cloud Connection
- Define the Data Path
- Get the List of Data Files
- Upload Data
- Verification
- Execution, Debugging, Evaluation

Content

Section 6: Hands-On Project - ETL Pipeline Development

- ETL Pipeline – Introduction
- Extracting Function
- Transforming Part1
- Transforming Part2
- Transforming Part3
- Transforming Part4
- Business Metrics Part1
- Exercise - Product Performance Metric
- Exercise Solution
- Exercise - Sales Revenue Metric
- Exercise Solution
- Loading to S3 Part1
- Loading to S3 Part2
- Final Execution

Content

Section 7: Hands-On Project – Orchestration

- Introduction
- Flow - Simple Example
- Task - Simple Example
- Deployment - Simple Example
- Work Pool - Simple Example
- Orchestration - Hands-On Project Part1
- Orchestration - Hands-On Project Part2

Content

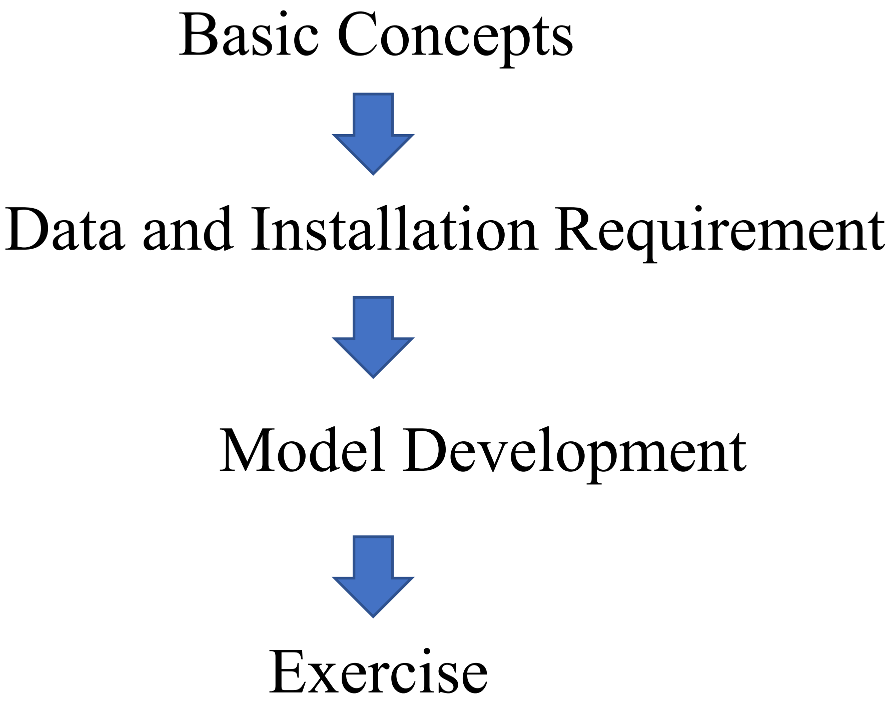
Section 8: Hands-On Project – Containerization

- Introduction to Containerization
- Docker File and Docker Compose
- Image Creation and Execution

Course Information

Course Information

Course Structure



Basic Concepts

Data and Installation Requirement

Model Development

Exercise

Course Information

Some Tips to Start!

1. Do not Skip the Basic Concept Explanation Part to Understand the Theory First!
2. During Model Development, Try to Create Your Model with Me and do Not Wait until the End
3. Do Not Use Ready idf Files!

Course Information

If any Error occurs, what should you do?

First Step → Double check (and even triple check) your model with mine (Note that most of the errors happen because of small mistakes!

Second Step → Read the error and Try to address the issue raised by E+

Third Step → Google the exact error

Forth Step → In the Q&A part of the Udemy, change the "All lecture" to "Current lecture" to find out if this error happens to other students and what solution proposed

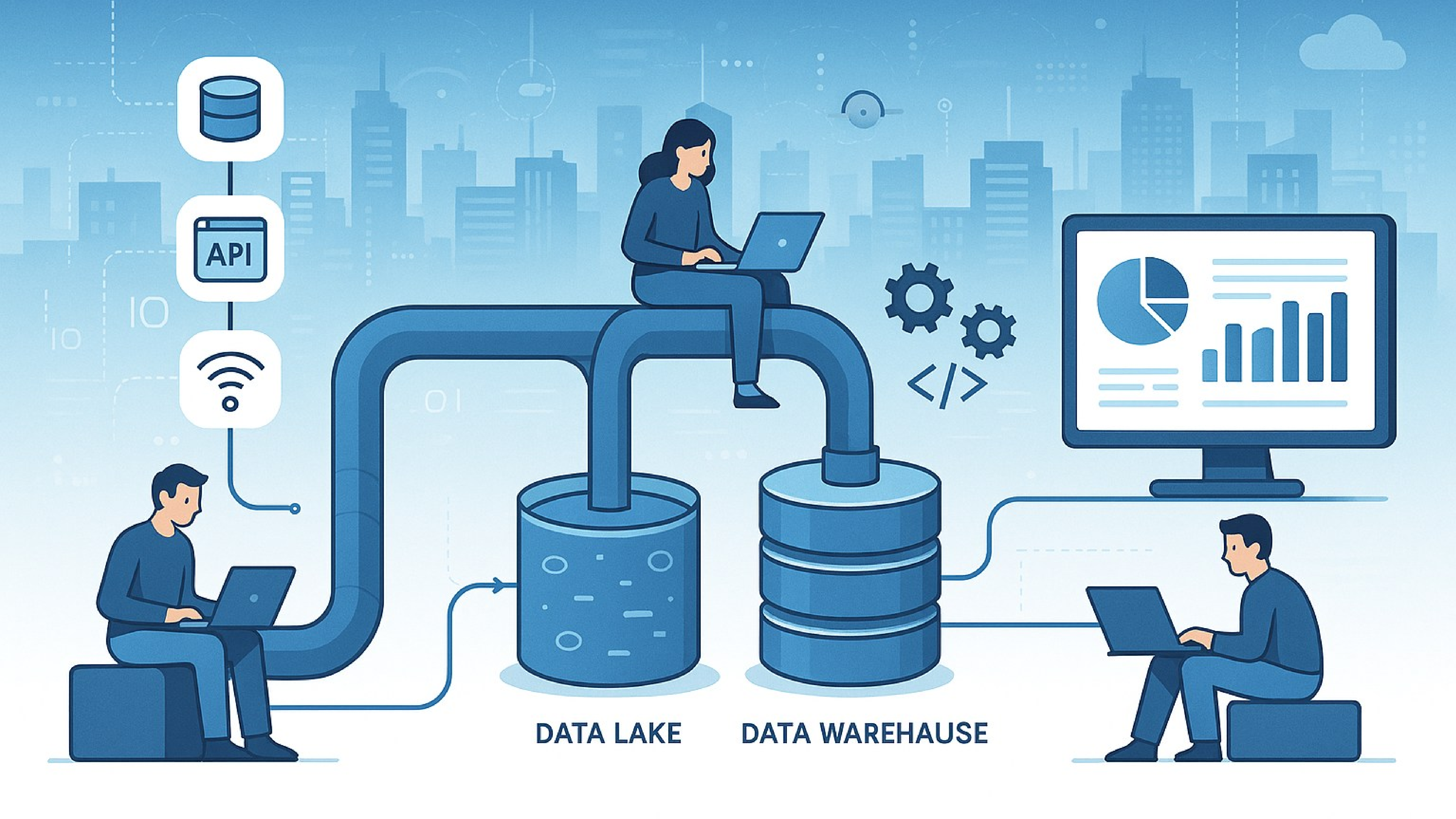
Last Step → Send the error you receive with the screenshot from the error in the Q&A part

Course Information

Udemy Platform

In below tab of videos, you can:

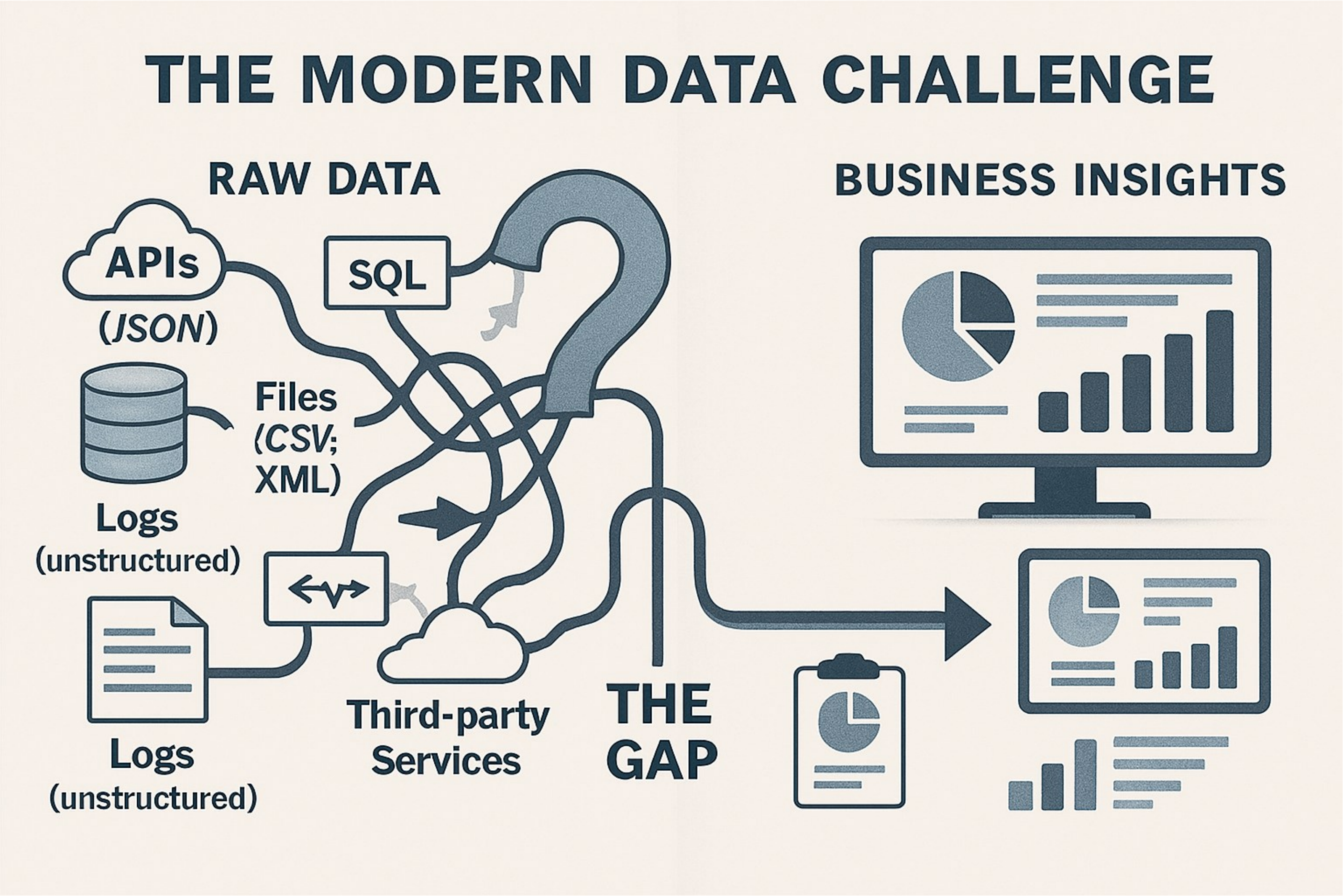
- Set a subtitle (English) from Caption Setting (CC) section
- Using setting button, make sure the video quality is set to highest quality
- If you think I am speaking too fast or too slow, you can easily change the playback rate
- If you have any problem with playing the videos and accessing to the content, please contact Udemy help.


API
IO
IO
01
</>
DATA LAKE
DATA WAREHAUSE

Introduction to Data Engineering

What is Data Engineering?

The Data Problem

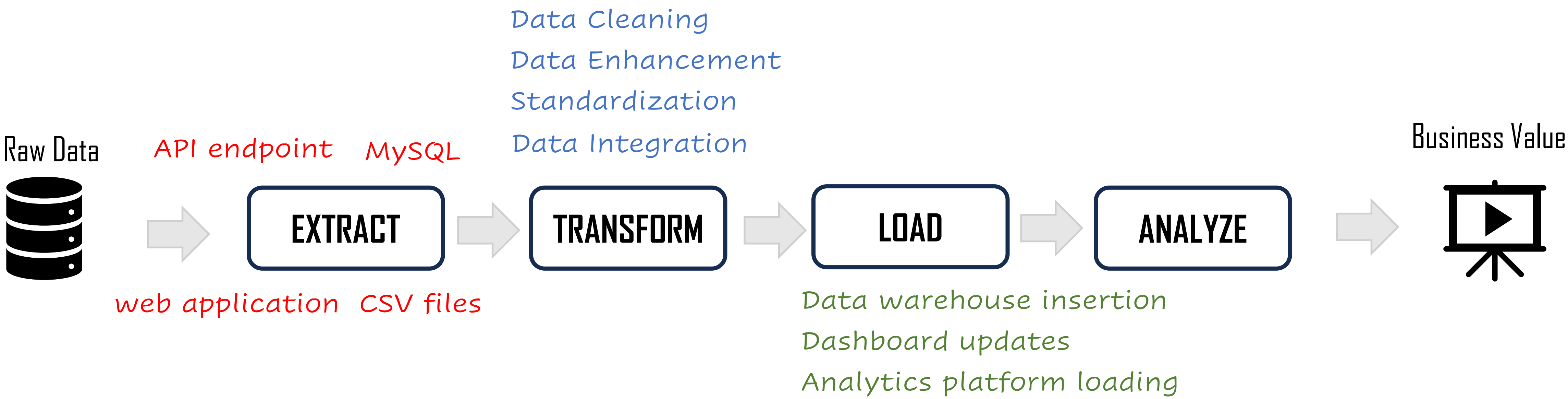


Generated by AI

Introduction to Data Engineering

What is Data Engineering?

Data Engineering Definition



Raw Data

API endpoint MySQL

web application CSV files

EXTRACT

Data Cleaning

Data Enhancement

Standardization

Data Integration

TRANSFORM

LOAD

Data warehouse insertion

Dashboard updates

Analytics platform loading

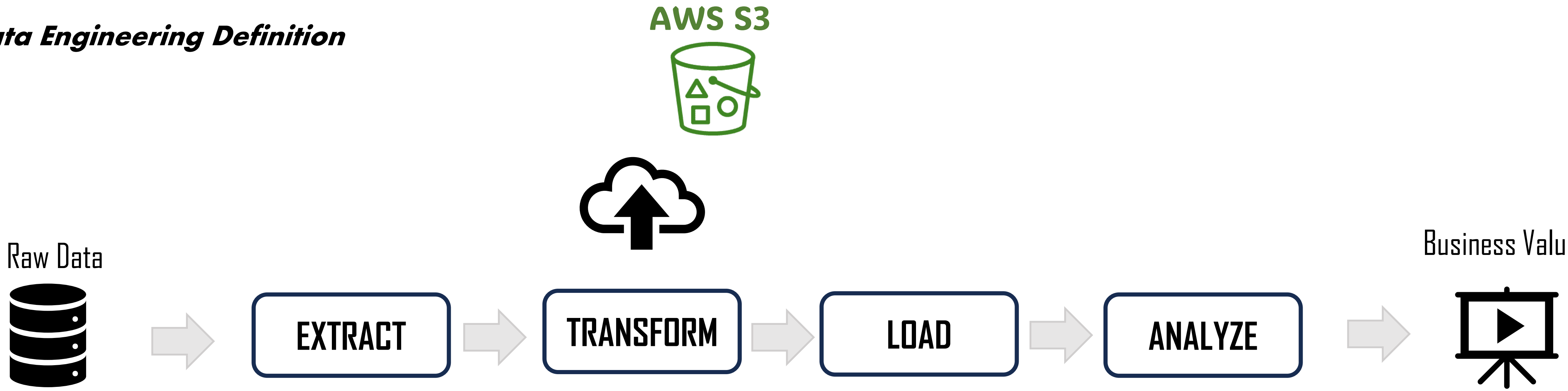
ANALYZE

Business Value

Introduction to Data Engineering

What is Data Engineering?

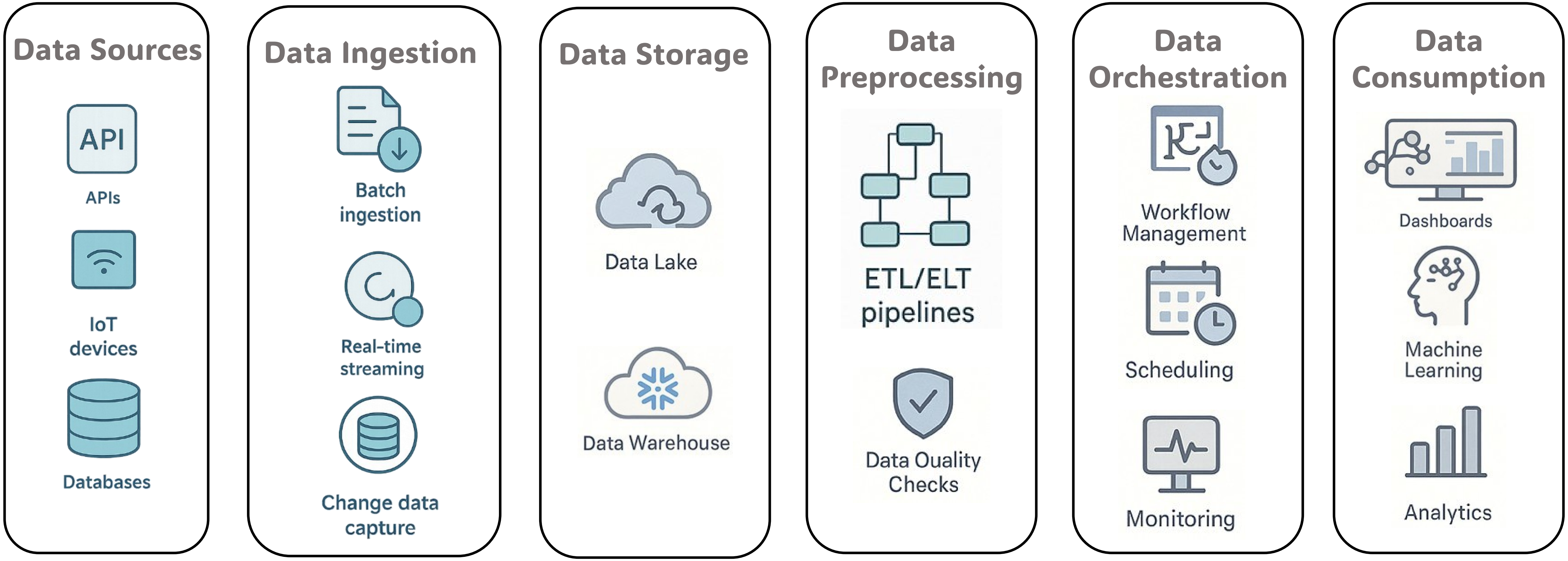
Data Engineering Definition



Introduction to Data Engineering

What is Data Engineering?

The Data Engineering Ecosystem



Data Sources

- APIs
- IoT devices
- Databases

Data Ingestion

- Batch ingestion
- Real-time streaming
- Change data capture

Data Storage

- Data Lake
- Data Warehouse

Data Preprocessing

- ETL/ELT pipelines
- Data Ouality Checks

Data Orchestration

- Workflow Management
- Scheduling
- Monitoring

Data Consumption

- Dashboards
- Machine Learning
- Analytics

Introduction to Data Engineering

What is Data Engineering?

Why Data Engineering Matters?

✓ With Data Engineering:
- Automated, reliable data pipelines
- Single source of truth
- Real-time or near real-time insights
- Data-driven decision making at scale

✗ Without Data Engineering:
- 80% of time spent on data preparation
- Inconsistent data across teams
- Manual, error-prone processes
- Delayed insights and decisions

Introduction to Data Engineering

What is Data Engineering?

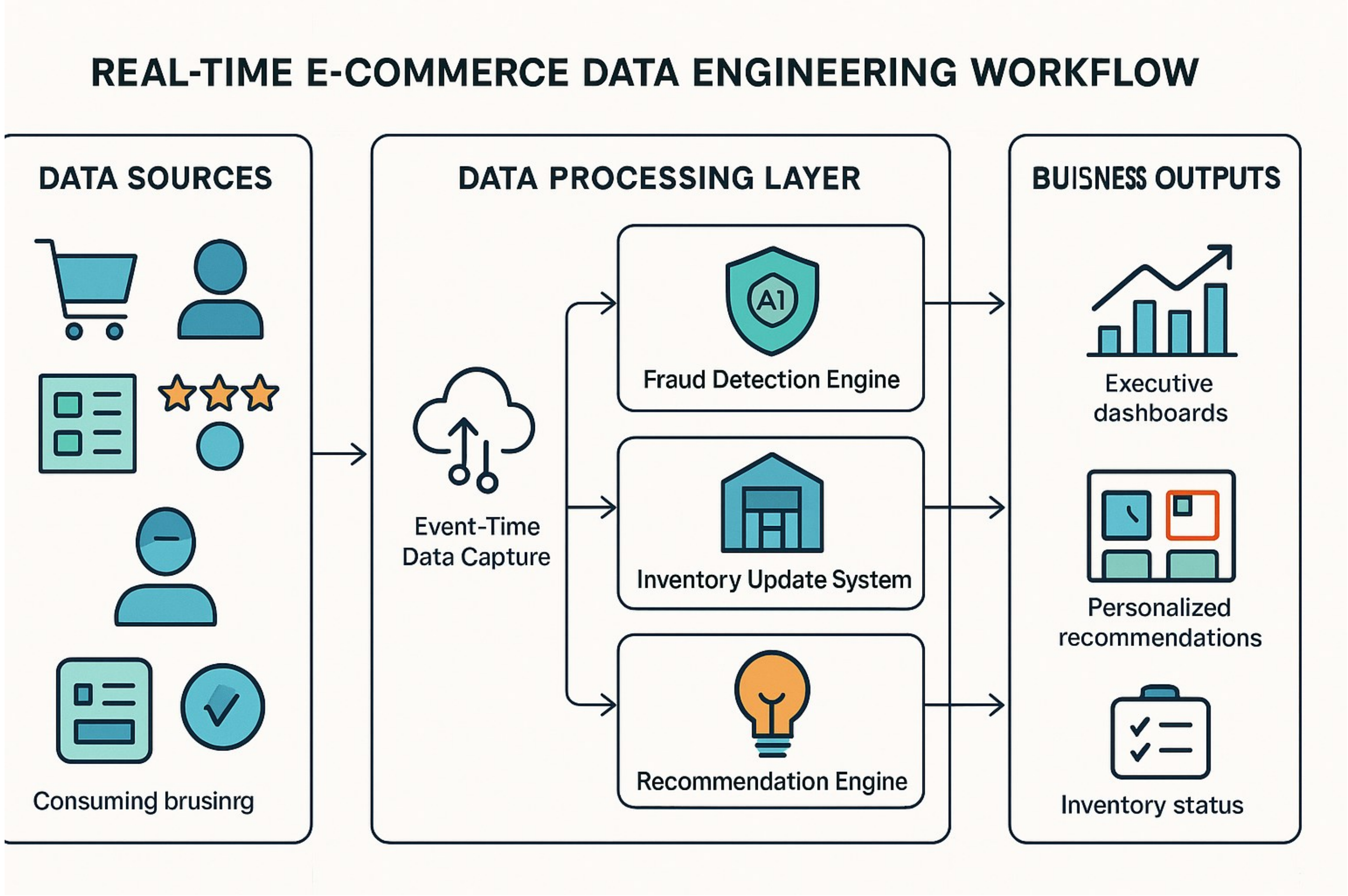
Data Engineering vs Other Roles

Role	Primary Focus	Key Skills	Deliverables
Data Engineer	Infrastructure & Pipelines	Python, SQL, Cloud, ETL tools	Data pipelines, Architecture
Data Analyst	Insights & Reporting	SQL, BI tools, Statistic	Reports, Dashboards
Data Scientist	Predictive Modeling	Python/R, ML, Statistics	Models, Predictions
ML Engineer	Production ML Systems	ML Ops, Deployment, Monitoring	ML Services, APIs

Introduction to Data Engineering

What is Data Engineering?

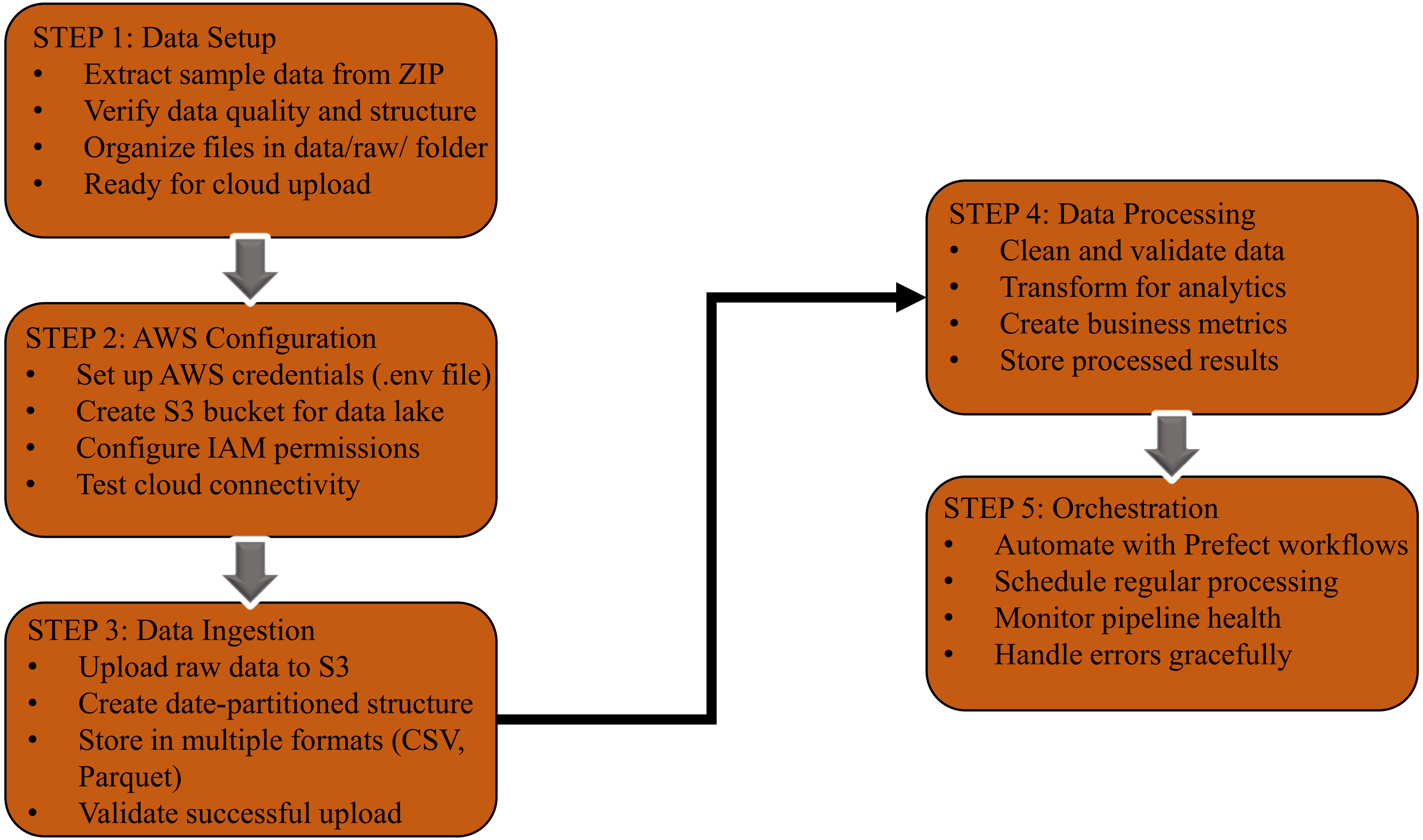
Importance of Data Engineering - Real-World Example



Generated by AI

Instructor: Navid Shirzadi

Hands on Project



STEP 1: Data Setup

- Extract sample data from ZIP
- Verify data quality and structure
- Organize files in data/raw/ folder
- Ready for cloud upload

STEP 2: AWS Configuration

- Set up AWS credentials (.env file)
- Create S3 bucket for data lake
- Configure IAM permissions
- Test cloud connectivity

STEP 3: Data Ingestion

- Upload raw data to S3
- Create date-partitioned structure
- Store in multiple formats (CSV, Parquet)
- Validate successful upload

STEP 4: Data Processing

- Clean and validate data
- Transform for analytics
- Create business metrics
- Store processed results

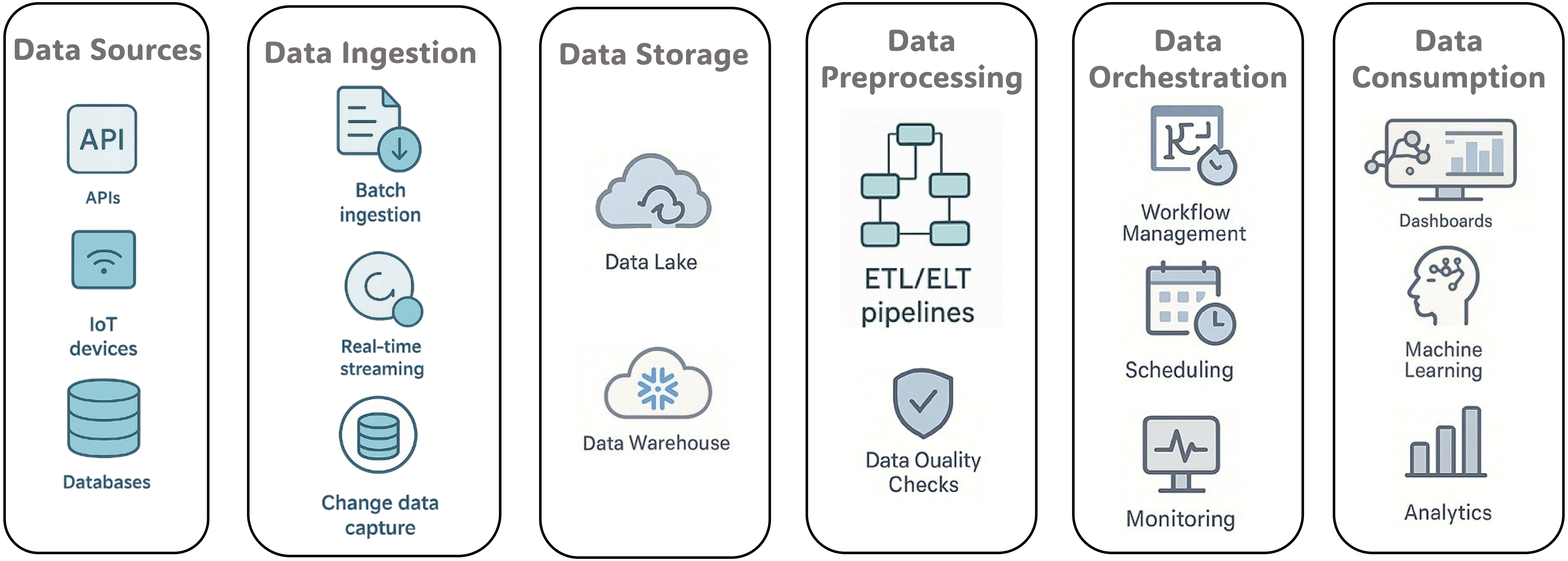
STEP 5: Orchestration

- Automate with Prefect workflows
- Schedule regular processing
- Monitor pipeline health
- Handle errors gracefully

Introduction to Data Engineering

What is Data Engineering?

The Data Engineering Ecosystem



Data Sources

APIs

IoT devices

Databases

Data Ingestion

Batch ingestion

Real-time streaming

Change data capture

Data Storage

Data Lake

Data Warehouse

Data Preprocessing

ETL/ELT pipelines

Data Ouality Checks

Data Orchestration

Workflow Management

Scheduling

Monitoring

Data Consumption

Dashboards

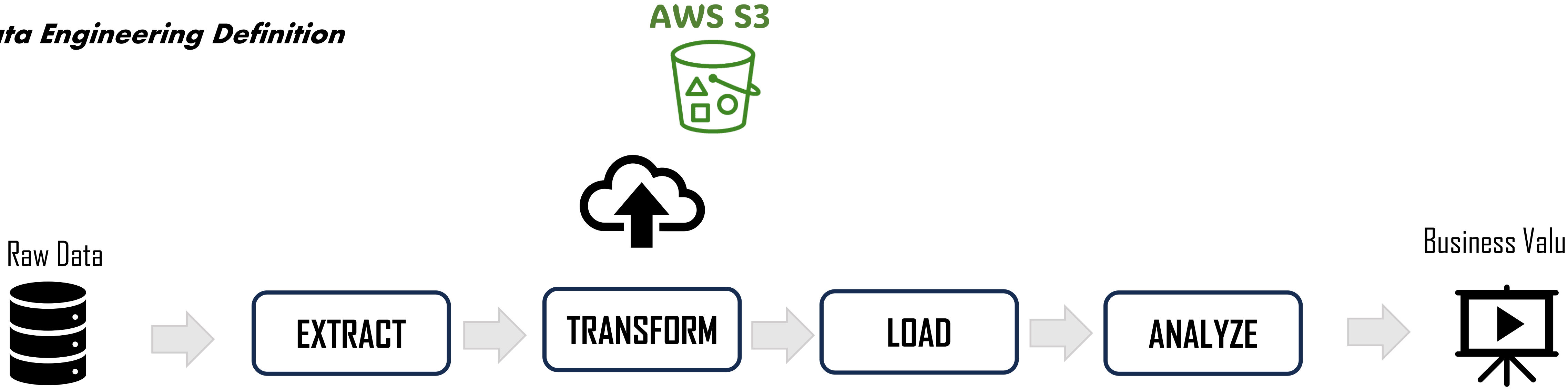
Machine Learning

Analytics

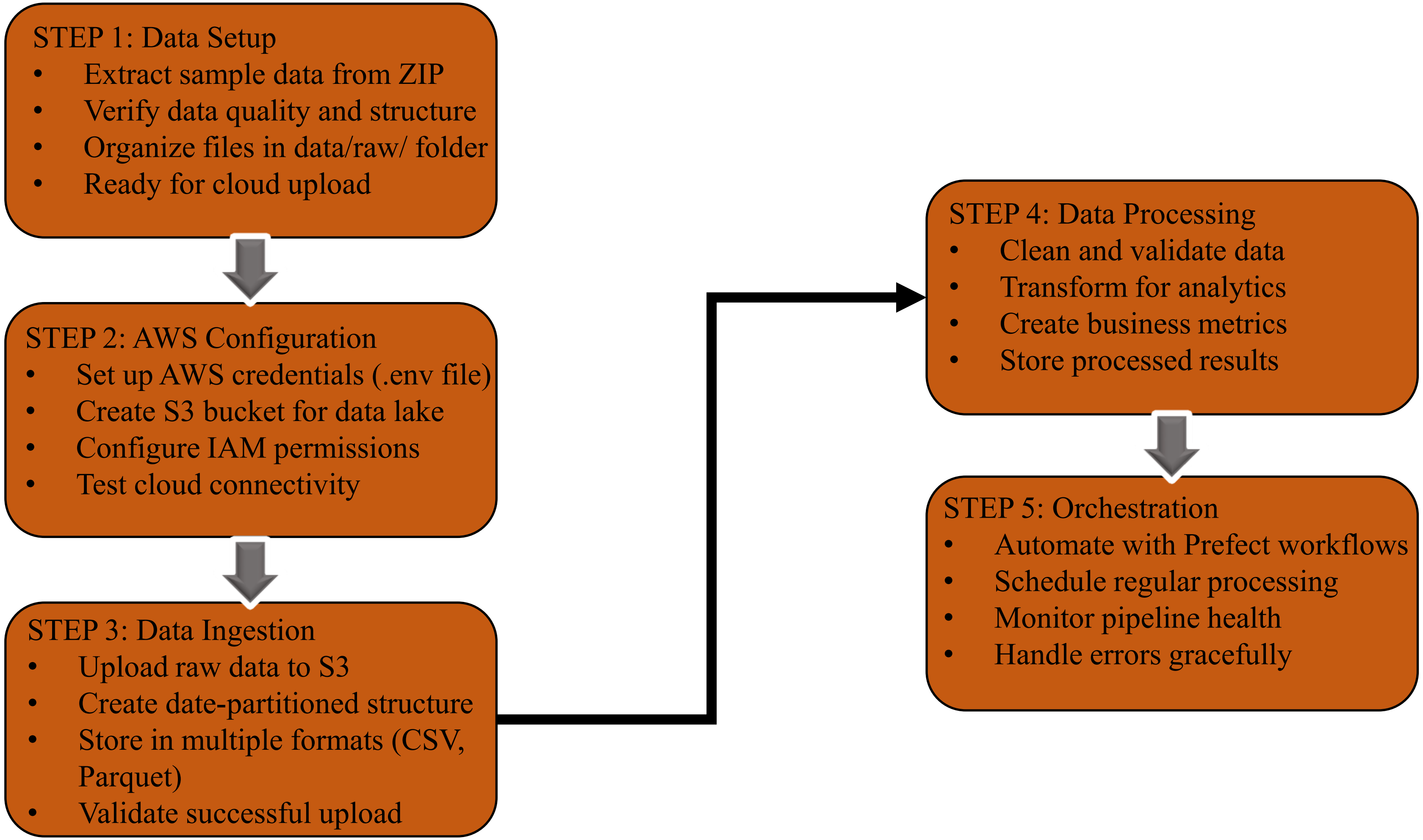
Introduction to Data Engineering

What is Data Engineering?

Data Engineering Definition



Hands on Project



STEP 1: Data Setup

- Extract sample data from ZIP
- Verify data quality and structure
- Organize files in data/raw/ folder
- Ready for cloud upload

STEP 2: AWS Configuration

- Set up AWS credentials (.env file)
- Create S3 bucket for data lake
- Configure IAM permissions
- Test cloud connectivity

STEP 3: Data Ingestion

- Upload raw data to S3
- Create date-partitioned structure
- Store in multiple formats (CSV, Parquet)
- Validate successful upload

STEP 4: Data Processing

- Clean and validate data
- Transform for analytics
- Create business metrics
- Store processed results

STEP 5: Orchestration

- Automate with Prefect workflows
- Schedule regular processing
- Monitor pipeline health
- Handle errors gracefully

Introduction to Data Engineering

What is Data Orchestration?

Coordination and management of data workflows and pipelines.

Scheduling

Managing dependencies

Monitoring

Scaling

❌ Without Orchestration

- Manual execution of data scripts
- No dependency management
- Difficult error handling and retries
- No visibility into pipeline status
- Hard to scale and maintain

✅ With Orchestration

- Automated scheduling and execution
- Clear dependency graphs
- Automatic retries and error handling
- Real-time monitoring and alerting
- Scalable and maintainable pipelines

Introduction to Data Engineering

Famous Orchestration Tools

Apache Airflow

PREFECT

dagster

AWS Step Functions

Azure Data Factory

Google Cloud Composer

Core Concept

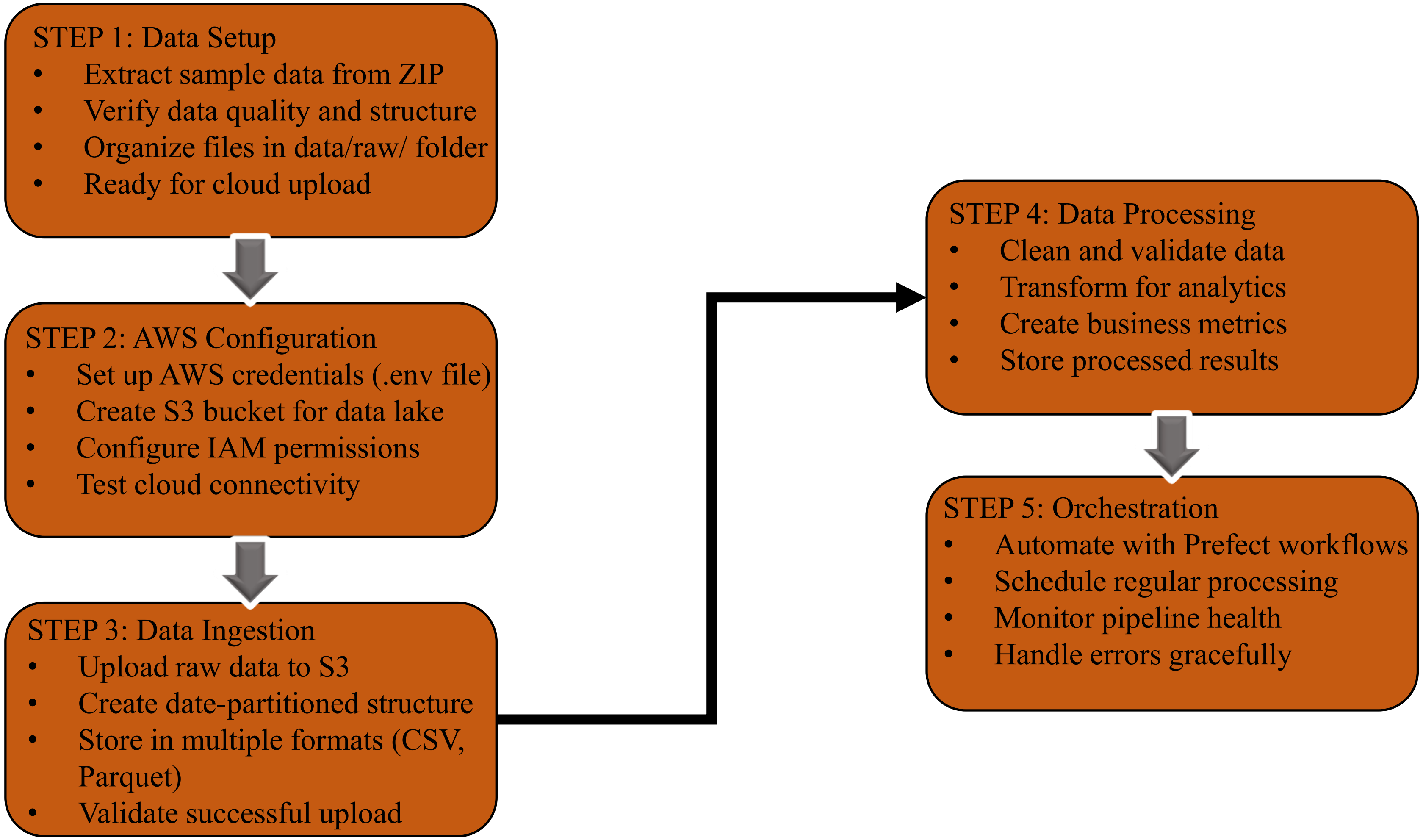
@flow The main workflow container that orchestrates tasks

```
@flow(name="data-pipeline)
def data_pipeline():
    raw_data = extract_data()
    clean_data = transform_data(raw_data)
    load_data(clean_data)
```

@Task Individual units of work within a flow

```
@task(name="extract", retries=3)
def extract_data():
    # Task implementation
    return data
```

Hands on Project



STEP 1: Data Setup

- Extract sample data from ZIP
- Verify data quality and structure
- Organize files in data/raw/ folder
- Ready for cloud upload

STEP 2: AWS Configuration

- Set up AWS credentials (.env file)
- Create S3 bucket for data lake
- Configure IAM permissions
- Test cloud connectivity

STEP 3: Data Ingestion

- Upload raw data to S3
- Create date-partitioned structure
- Store in multiple formats (CSV, Parquet)
- Validate successful upload

STEP 4: Data Processing

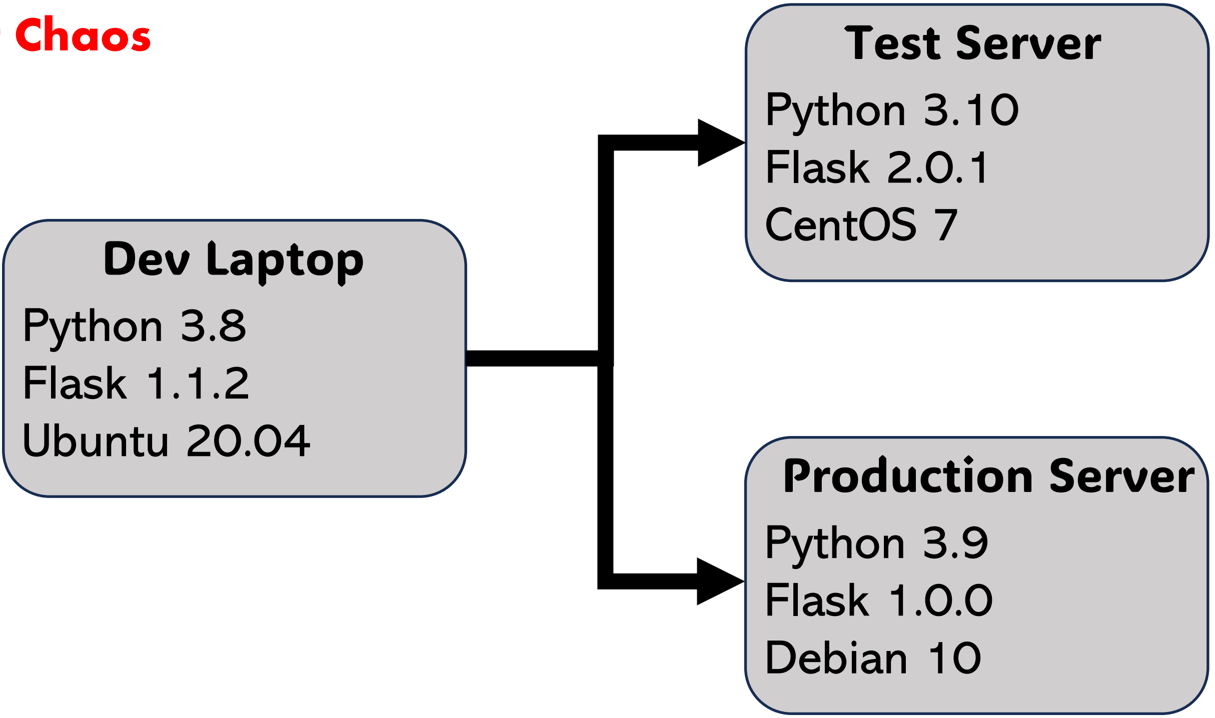
- Clean and validate data
- Transform for analytics
- Create business metrics
- Store processed results

STEP 5: Orchestration

- Automate with Prefect workflows
- Schedule regular processing
- Monitor pipeline health
- Handle errors gracefully

Containerization

Before: Environment Chaos



Dev Laptop
Python 3.8
Flask 1.1.2
Ubuntu 20.04

Test Server
Python 3.10
Flask 2.0.1
CentOS 7

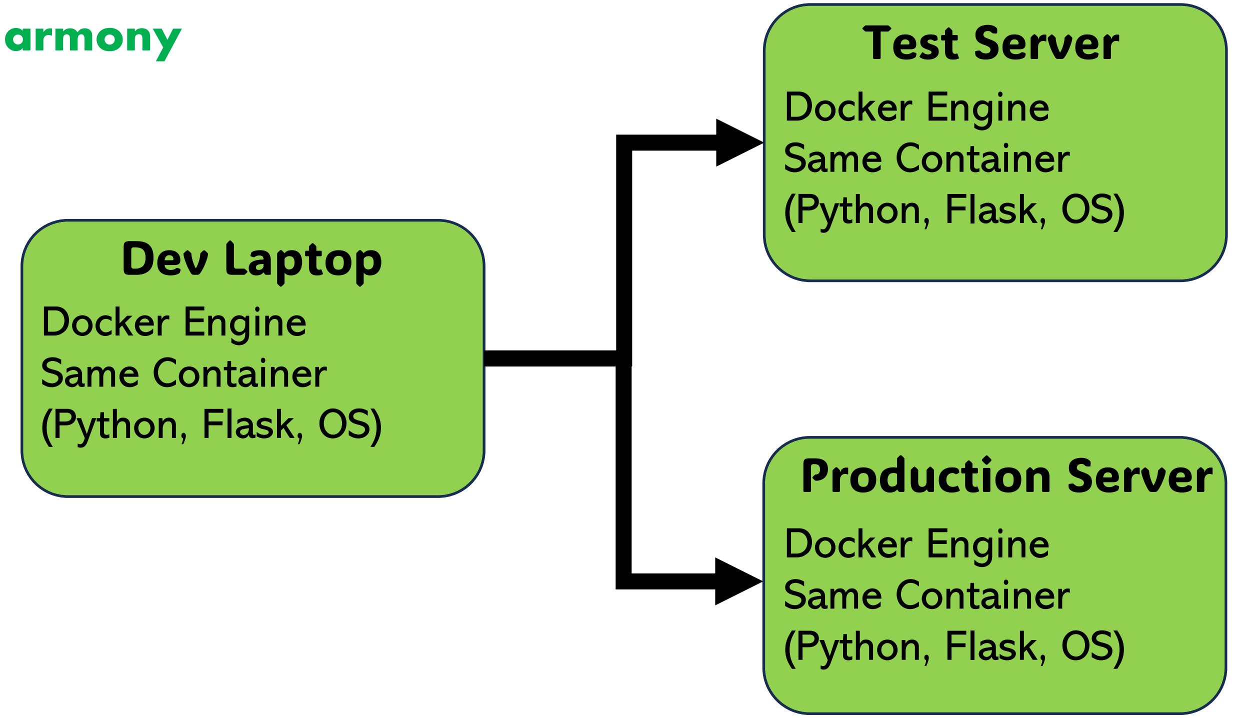
Production Server
Python 3.9
Flask 1.0.0
Debian 10

- ❌ Inconsistent behavior
- ❌ Hard-to-reproduce bugs
- ❌ Works on my machine" syndrome

Introduction to Data Engineering

Containerization

After: Dockerized Harmony



Dev Laptop

Docker Engine
Same Container
(Python, Flask, OS)

Test Server

Docker Engine
Same Container
(Python, Flask, OS)

Production Server

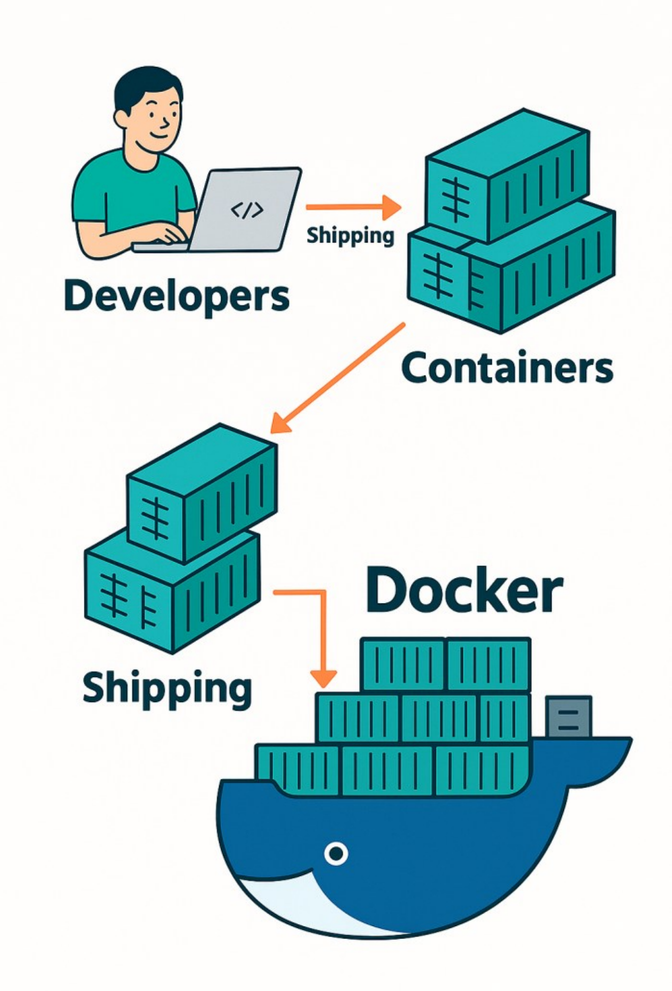
Docker Engine
Same Container
(Python, Flask, OS)

- ✓ Consistent behavior
- ✓ Easy debugging and testing
- ✓ Reliable, predictable deployments

Introduction to Data Engineering

Containerization

Docker



Generated by AI