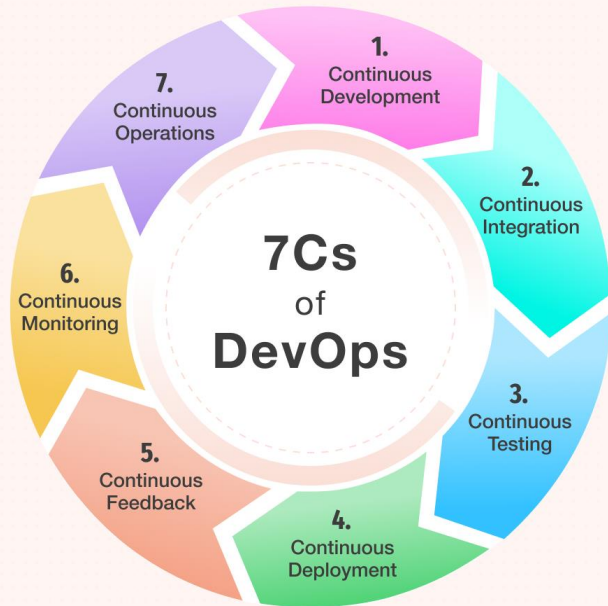


---

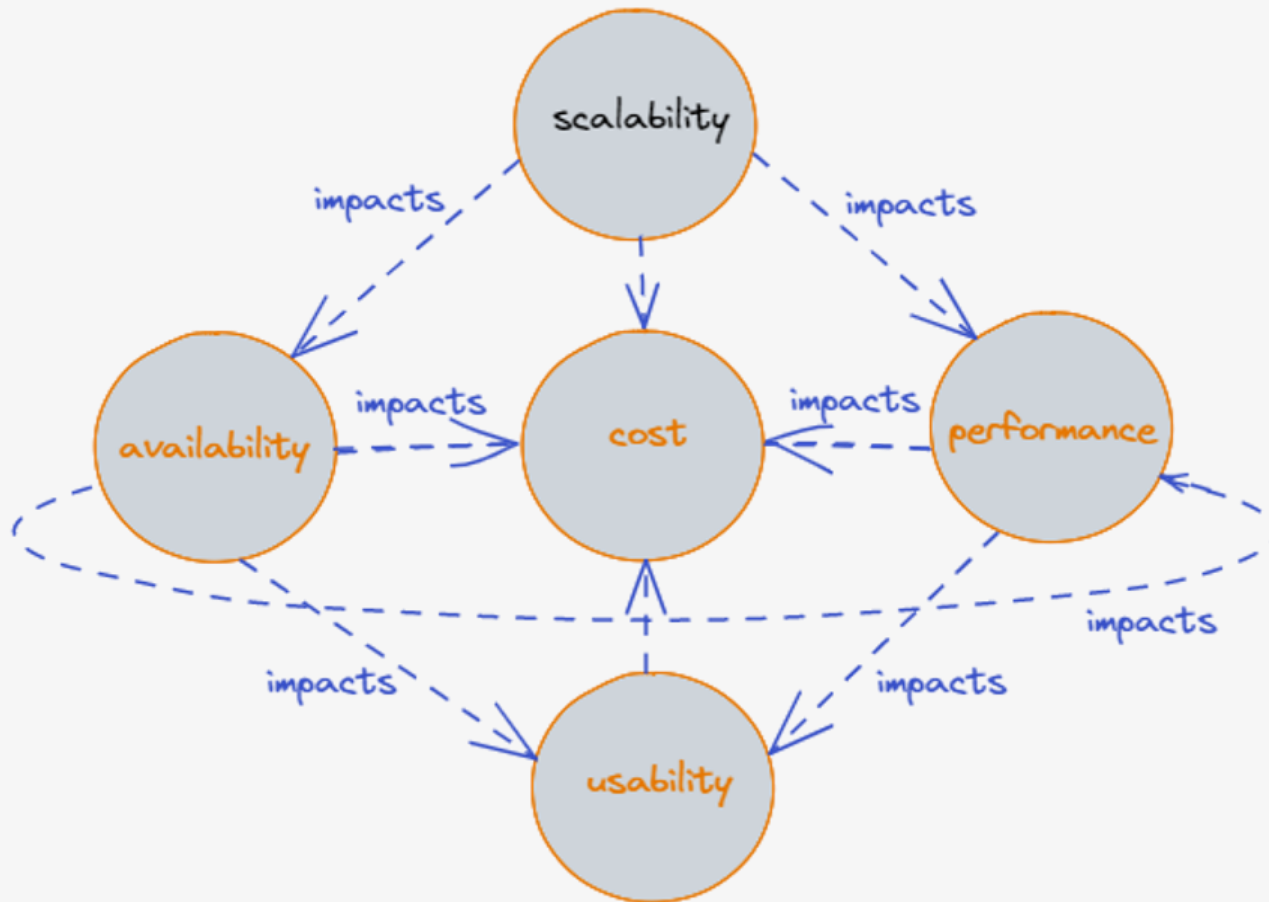
## Concepts

- Scalability
- Availability
- Continuous Integration
- Automation & Deployments

# 7c's of DevOps



1. Continuous Development
2. Continuous Integration
3. Continuous Testing
4. Continuous Deployment
5. Continuous Feedback
6. Continuous Monitoring
7. Continuous Operation



# Scalability

- **Definition:**

Scalability is the ability of a system to handle increasing workloads by adding resources.

- **Importance:**

Enables businesses to grow and meet user demand without performance degradation.

- **Real-World Examples:**

Online shopping platforms like Amazon handling traffic spikes during sales.

Video streaming services like Netflix scaling to serve millions of users.

# Types of Scalability

## Vertical Scaling (Scaling Up):

- Adding more power (CPU, RAM) to an existing machine.
- **Example:** Upgrading from a server with 8GB RAM to 32GB RAM.

## Horizontal Scaling (Scaling Out):

- Adding more machines to distribute the load.
- **Example:** Adding more servers to handle web requests.

## Elastic Scaling:

- Dynamically scaling resources up or down based on demand
- **Example:** Auto-scaling in cloud environments.

# Components of a Scalable System

## Architecture:

- Microservices vs. Monolithic Architecture.
- Stateless vs. Stateful services.

## Databases:

- Relational databases (e.g., MySQL, PostgreSQL) vs. NoSQL databases (e.g., MongoDB, Cassandra).
- Techniques like sharding and replication.

## Infrastructure:

- Cloud computing and containers (Docker, Kubernetes).

# Key Scalability Techniques

## • Load Balancing:

- Distributing traffic across multiple servers.

**Example Tools:** NGINX, HAProxy, AWS Elastic Load Balancer.

## Caching:

- Storing frequently accessed data to reduce load on databases.

**Example Tools:** Redis, Memcached.

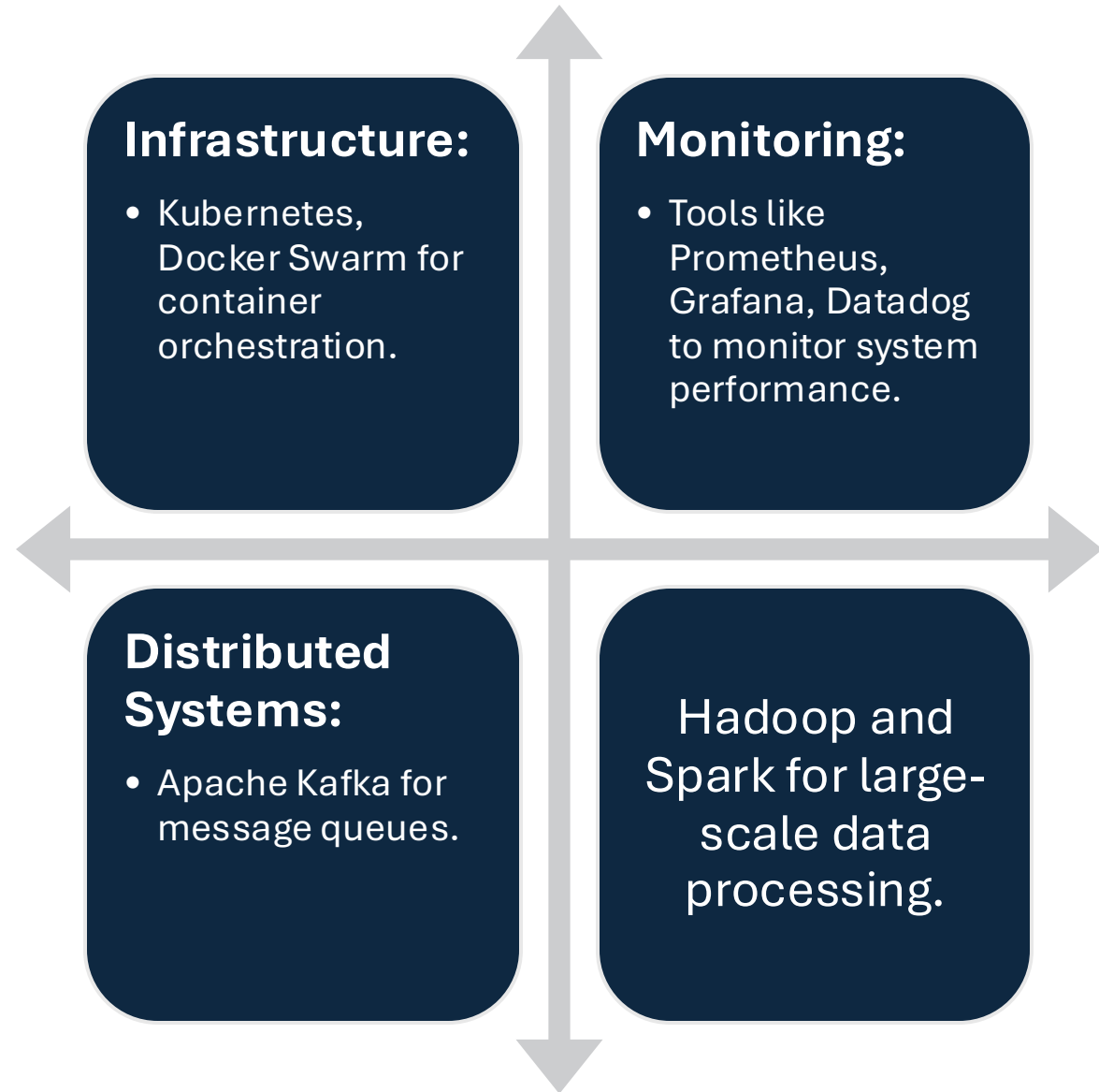
## • Database Optimization:

- Indexing, query optimization, and denormalization.

## Partitioning Data:

- Divide large datasets into smaller chunks for easier management.

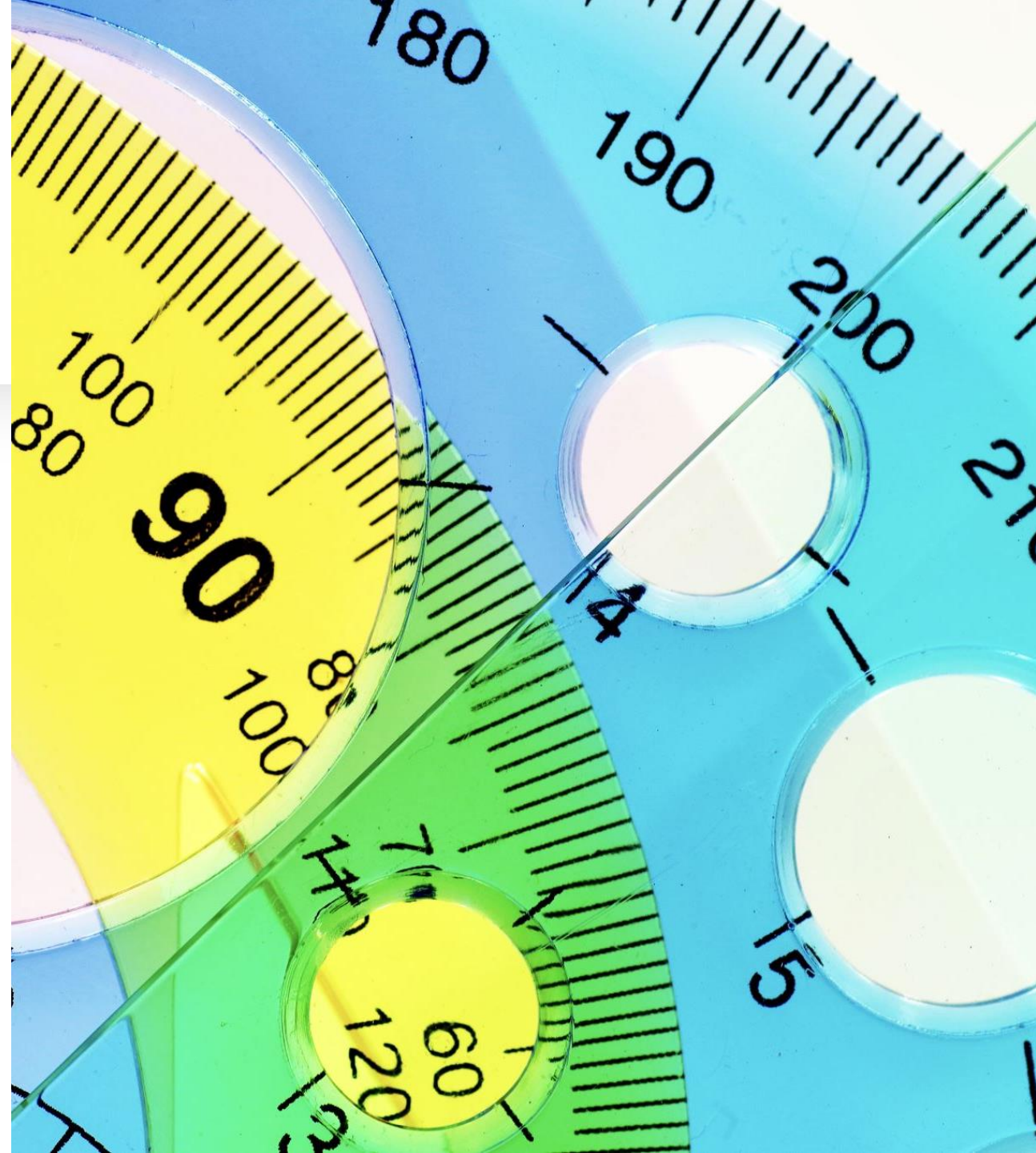
# Tools and Technologies for Scalability





# Measuring Scalability

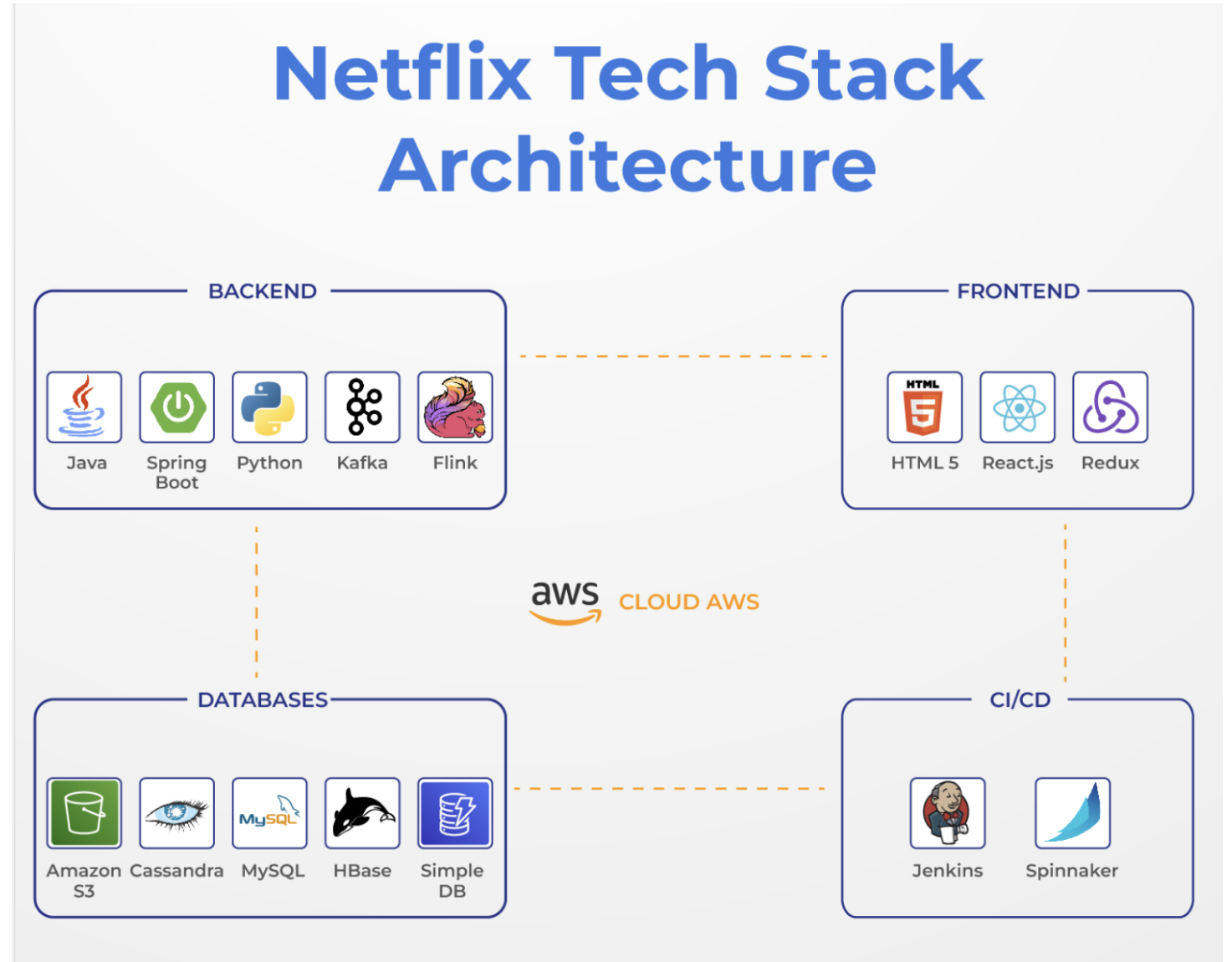
- **Key Metrics:**
  - Response Time, Throughput, Resource Utilization.
- **Load Testing:**
  - Tools like Jmeter to simulate traffic.





# Real-World Scenarios

Case Studies:  
-- Netflix



# Lab Exercises

- Vertical Scaling – Add more resources
- Horizontal Scaling with two servers
- Caching –
  - Simple caching
  - Redis