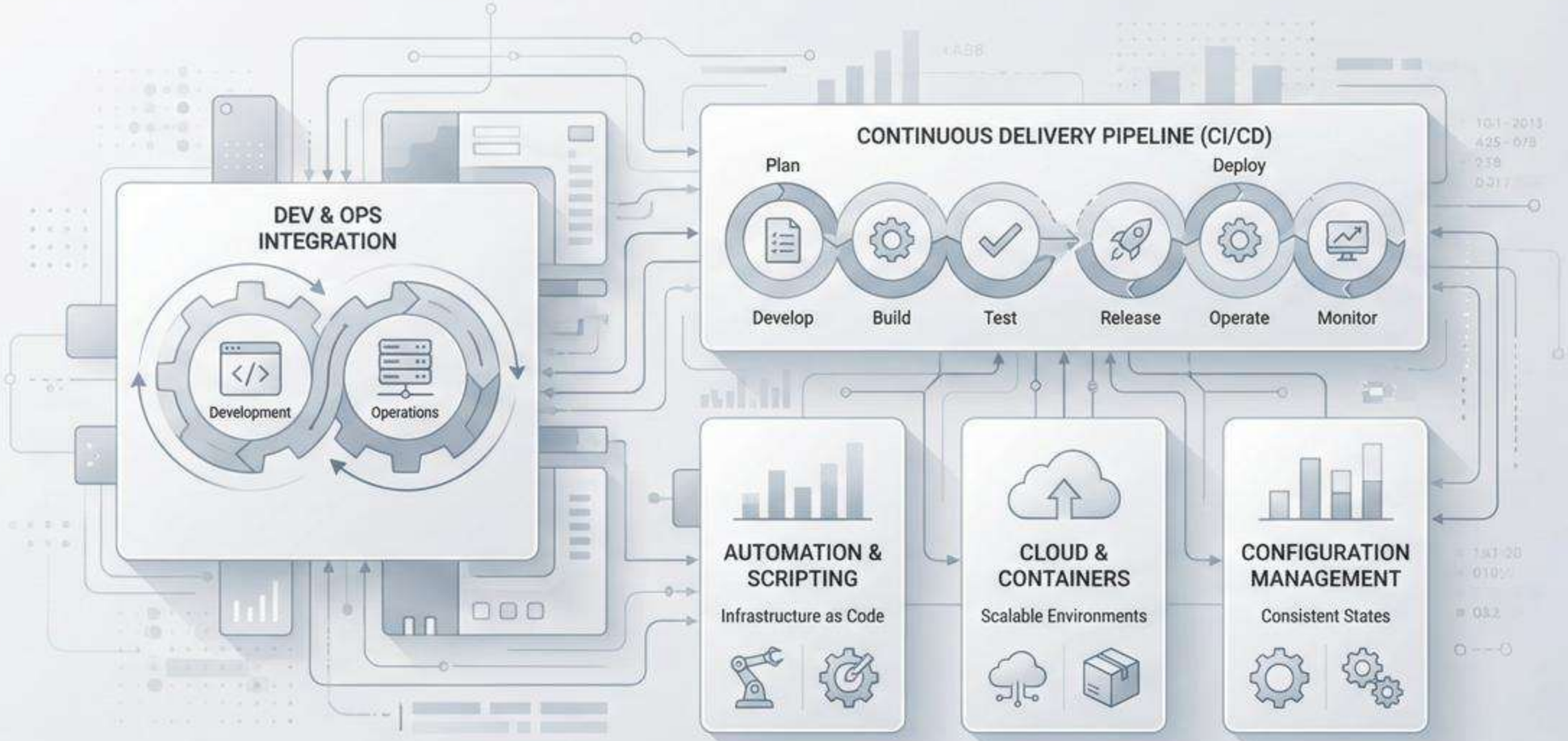
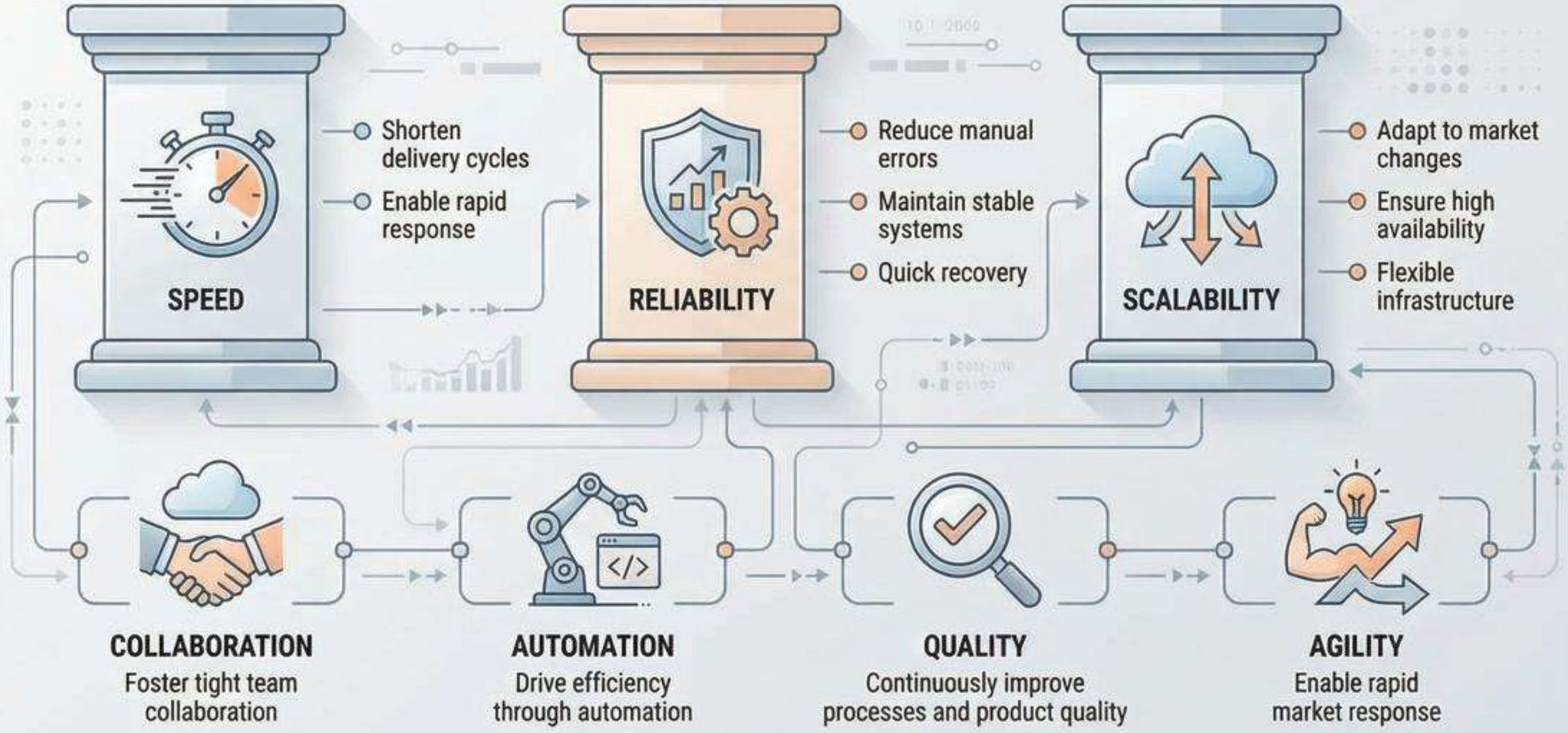
DEVOPS UNLOCKED: CODE TO CLOUD

A Strategic Introduction to Collaboration, Automation, and Continuous Delivery

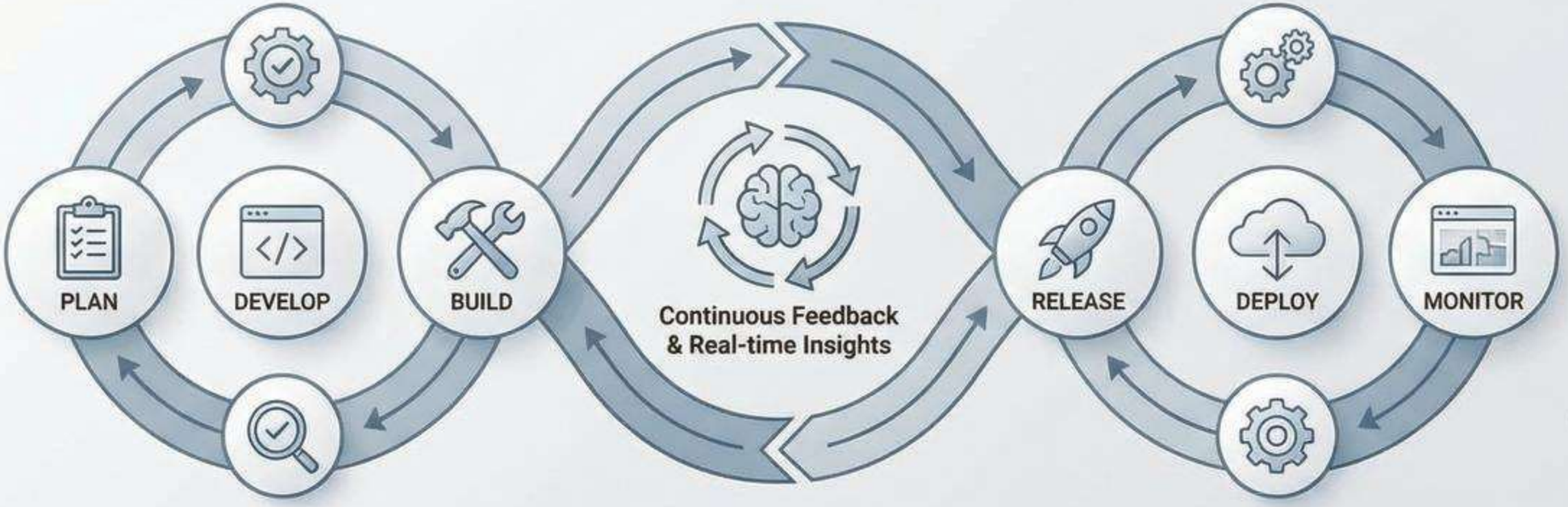


DEVOPS PRIMARY GOALS: SPEED, RELIABILITY, SCALABILITY

A Foundation for High-Performance Software Delivery and Continuous Improvement



DEVOPS AND SOFTWARE DEVELOPMENT LIFE CYCLE (SDLC)

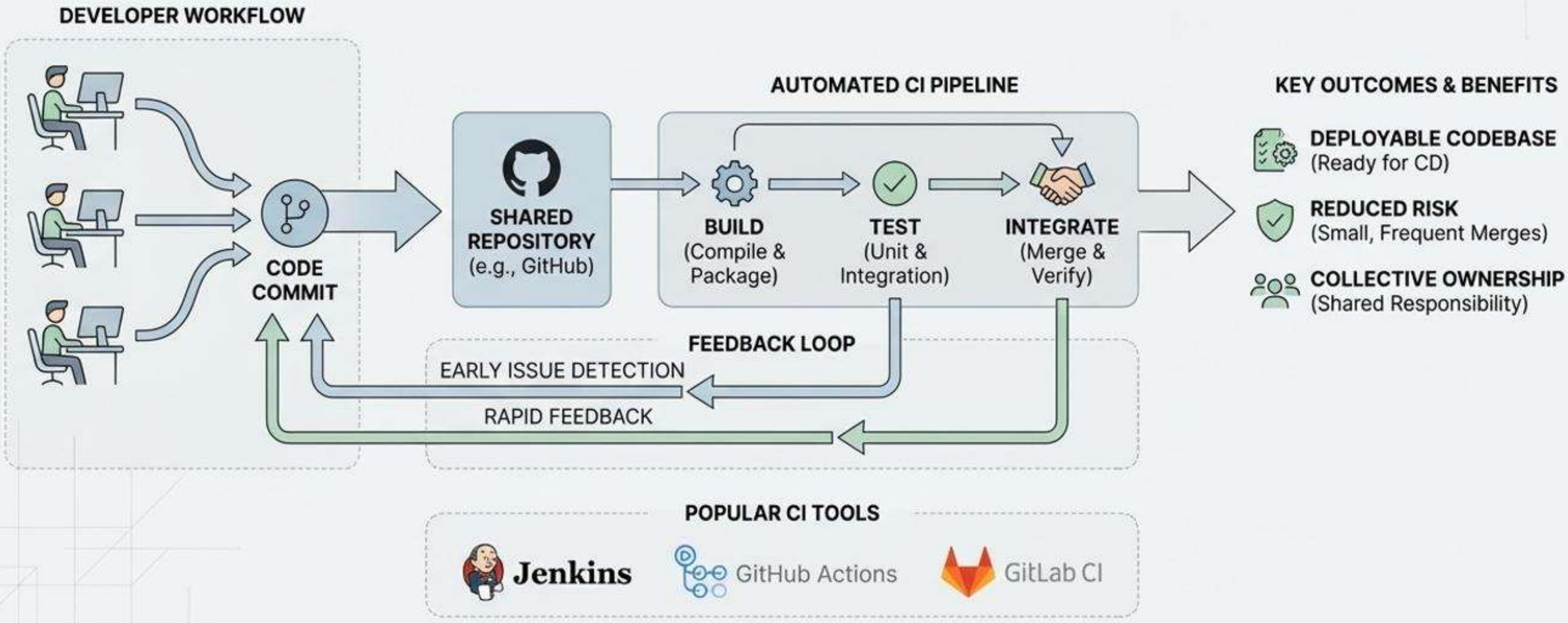


NO HAND-OFF DELAYS

EARLY DEFECT DETECTION

RAPID & SAFE DELIVERY

Continuous Integration (CI): Frequent Merges & Automated Feedback



CONTINUOUS DEPLOYMENT (CD): AUTOMATING RELEASE TO PRODUCTION

Extending Continuous Integration for Faster, Reliable Software Delivery

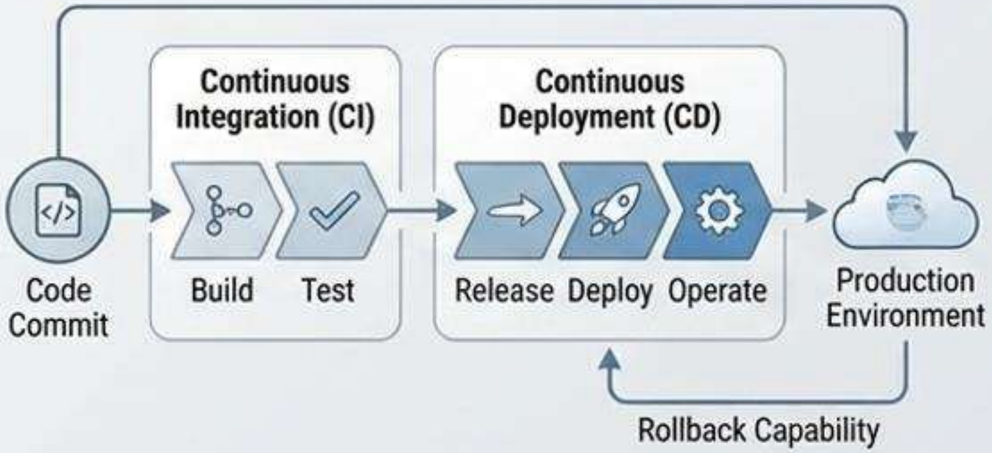
THEORY: WHAT IS CONTINUOUS DEPLOYMENT?

Automated Push to Production:
Every validated build is deployed without manual intervention.

Eliminates Manual Bottlenecks:
Removes delays in the release process.

Automated Pipelines:
Handles environment provisioning, testing, and rollout.

Faster Time-to-Market:
Users get new features and fixes within minutes.



EASY LEARNING POINTS: BENEFITS

Faster Releases:
Frequent, smaller updates.

Reduced Human Error:
Consistent, automated process.

Instant Recovery:
Quick rollback from failures.

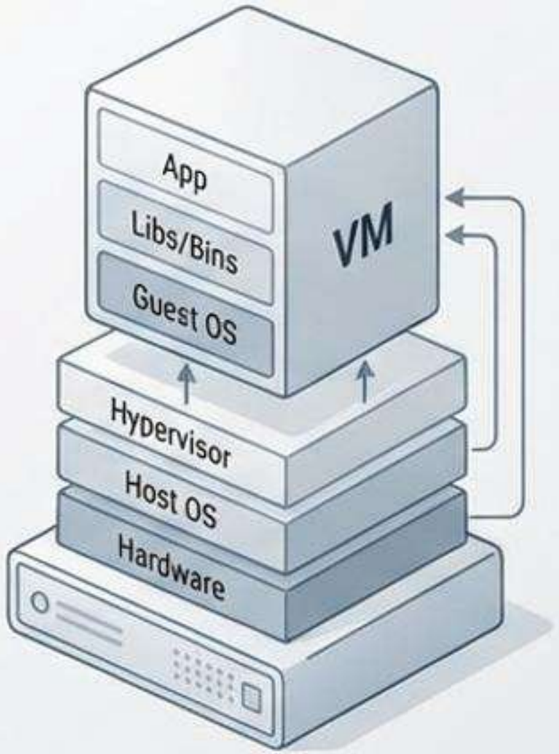
Improved Collaboration:
Seamless Dev & Ops workflow.

POPULAR CD TOOLS

Jenkins | GitHub Actions | GitLab CI | CircleCI | Argo CD

CONTAINERS & VIRTUAL DEVELOPMENT

Package Applications, Dependencies, and Libraries for Portability and Speed

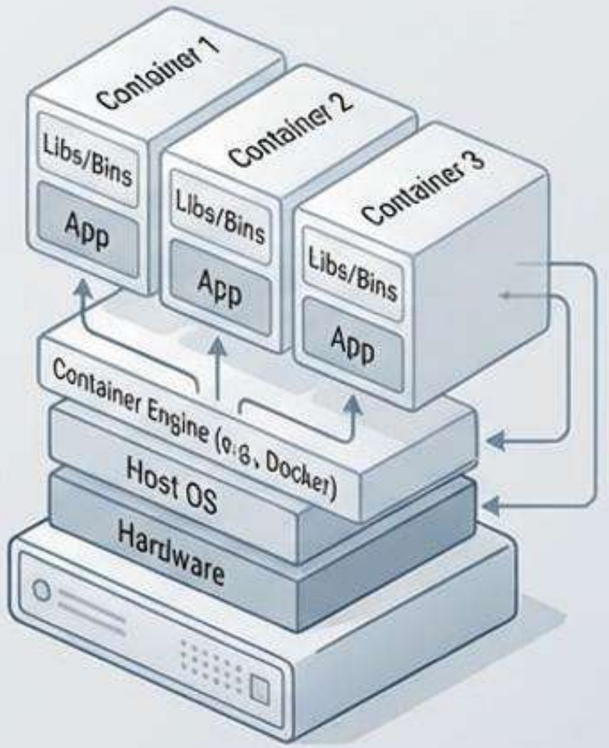


VIRTUAL MACHINE (VM)
Heavy, slow startup, own OS.
High Overhead. Isolated.

EASY LEARNING POINTS

- Containers are fast
- Easy to move between systems
- Same app runs everywhere
- Docker is a popular container tool

Key Benefit: Simplify dependency management & scaling across laptops, data centers, and clouds.

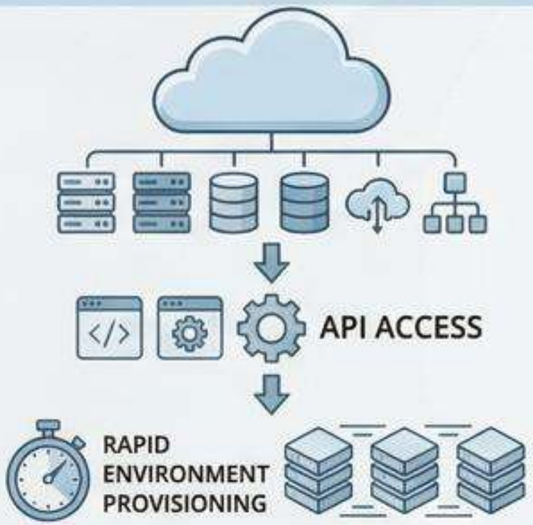


CONTAINER
Lightweight, fast startup, shares OS.
Low Overhead, Dense Utilization. Portable.

PUBLIC CLOUD PROVIDERS & DEVOPS

On-Demand Resources, Scalability, and Global Reach for Modern Software Delivery

ON-DEMAND & AUTOMATED RESOURCES



Spin up identical environments in minutes via APIs. Align perfectly with DevOps automation for faster delivery.

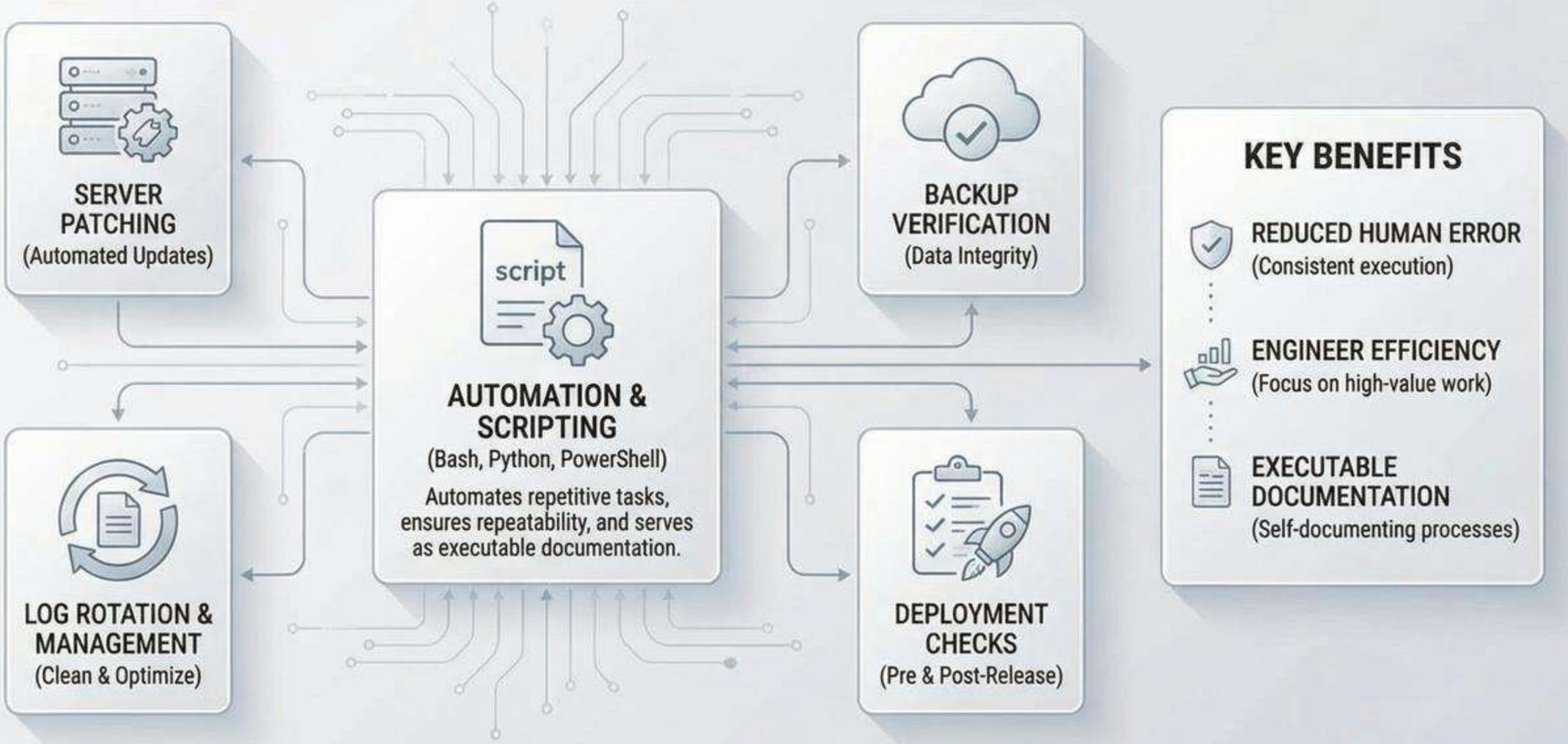
ELASTIC SCALABILITY & COST EFFICIENCY

Scale elastically with fluctuating user demand. Pay only for resources used, optimizing infrastructure costs.

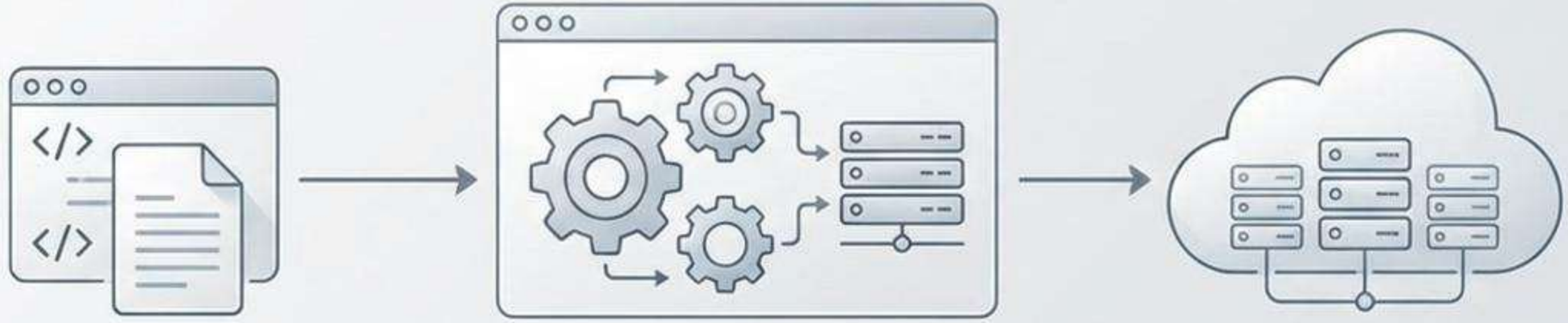
GLOBAL AVAILABILITY & RESILIENCE

Leverage global zones for resilient, low-latency deployments. Ensure high availability and business continuity.

DEVOPS AUTOMATION: SCRIPTING POWER



CONFIGURATION MANAGEMENT TOOLS



CODE
(Infrastructure as Code)

CONFIGURATION MANAGEMENT ENGINE
(e.g., Ansible, Terraform, Chef, Puppet)

SCALABLE INFRASTRUCTURE
(Desired State)

IDEMPOTENT PROVISIONING

Ensure consistent state regardless of previous configuration. Repeatable & reliable.

PREVENT CONFIGURATION DRIFT

Automated enforcement maintains desired state, avoiding unintended changes.

IMMUTABLE INFRASTRUCTURE

Replace, don't patch. Deploy new, pre-configured components.

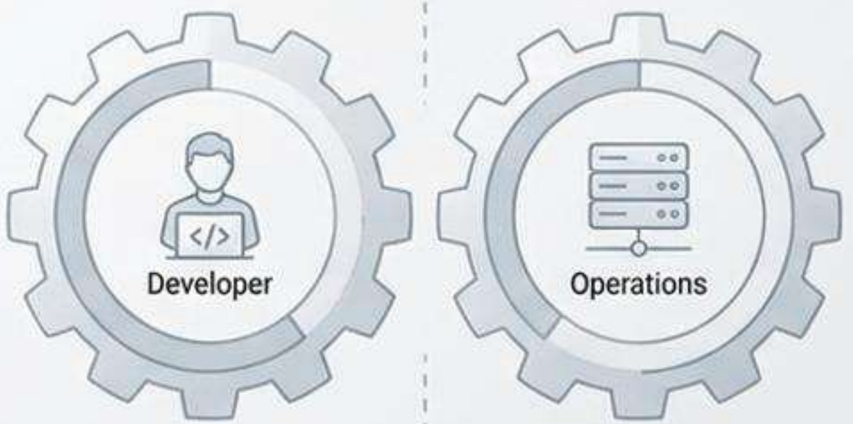
REVIEW & COLLABORATION

Manage infrastructure changes via pull requests, like application code.

DEVOPS ROLES: CROSS-FUNCTIONAL COLLABORATION

From Siloed Teams to Collective Accountability & Shared Responsibility

TRADITIONAL SILOS



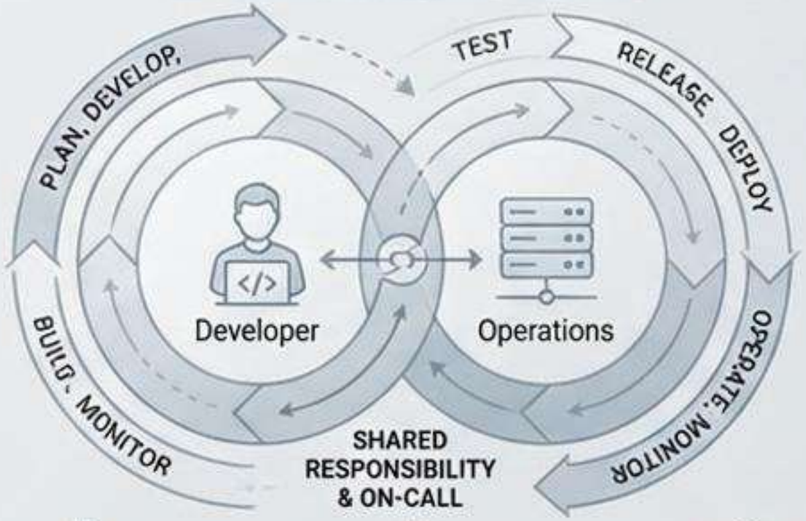
DEVELOPMENT (DEV)

OPERATIONS (OPS)

Separate, disconnected functions.
Delays & Errors.

DEVOPS TRANSFORMATION

CROSS-FUNCTIONAL TEAMS



FASTER RESOLUTION

Rapid issue detection & fixing through shared knowledge.

HIGHER QUALITY RELEASES

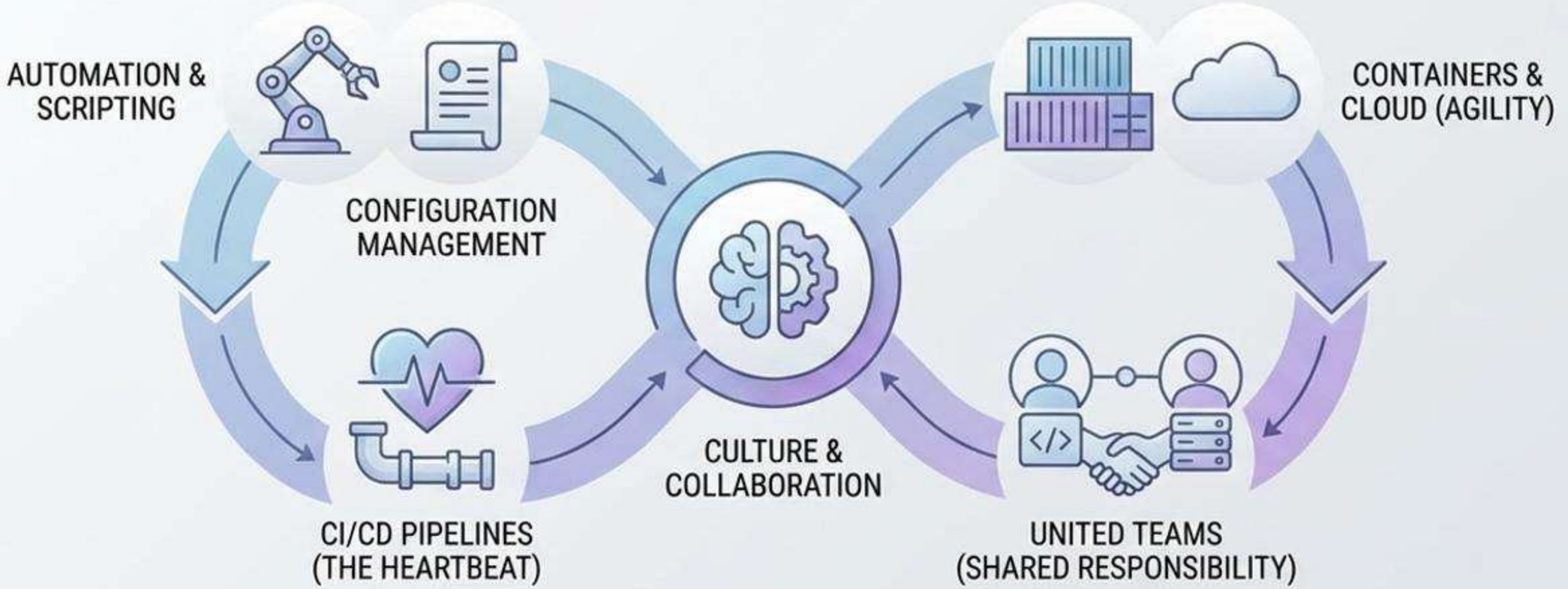
Automated testing & continuous feedback.

CULTURE OF ACCOUNTABILITY

Collective ownership for customer value.

DevOps dissolves the divide, creating one unified, efficient team.

DEVOPS CORE PRINCIPLES



DevOps is culture first, tools second. Automation and collaboration are engines; CI/CD provides the heartbeat; containers and cloud supply agility; united teams deliver fast, reliable, and continuously improving software.