



DevOps  
INSTITUTE

# Continuous Delivery Ecosystem Foundation<sup>SM</sup>

## BLUEPRINT

Continuous Delivery (CD) is a software engineering approach in which teams produce incremental software changes in short cycles ensuring that the software can be released safely at any time. A DevOps toolchain automates a continuous delivery pipeline to deliver software changes faster, more frequently, securely, with reduced cost and risk.

### Continuous Delivery Concepts

Collaborative management, design practices, continuous integration, continuous testing, infrastructures, toolchains, security, monitoring and delivery / deployment practices, work on incremental changes frequently using automation and fast feedback loops.

### Continuous Integration and Testing

Code changes are committed frequently to a version managed trunk branch. Images built from merged code are saved in an artifact repository. Tests conducted throughout the pipeline catch risky failures before release while completing tests quickly to avoid bottlenecks.

### Security Assurance (DevSecOps)

Vulnerabilities are identified and fixed as a part of the pipeline by integrating security practices into team activities, processes and tool chains, including automated security scans and monitoring of code, images and deployments.

### Collaborative Culture

Align cross-functional lean, agile teams around business goals. Embrace "The Three Ways of DevOps". Master collaboration, affinity, effective tooling and organization changes that support increasing scale with quality built-in.

### Monitoring & Improvement

Real-time active monitoring and analytics make tests, processes, and application performance measures visible for real-time decision analytics at each stage of the pipeline to prevent bottlenecks and to identify improvements.

### Design Practices for CD

High performance CD ecosystems use loosely coupled API-based modular service-oriented architectures (e.g. microservices) and 12-Factor Apps design practices, enable apps to be separately packaged, processed, tested and delivered in separate images or containers.

### Infrastructure & Toolchains

Resilient, elastic infrastructures, such as virtual and cloud-based systems defined "as-code" and "as-a-service" support on-demand, auto-scalable, immutable deployment environments. Tests such as Chaos Monkey identify failure points for improving reliability of infrastructure and toolchains.

### Continuous Delivery & Deployment

Automated configuration management, release automation, modular packages deployed using orchestrated virtualized, containerized, applications enable deployable production-ready artifacts and deployed safely using strategies such as Blue/Green, Feature Flag and Canary.

