

CI/CD

A DevOps Practice

Samuele Cozzi

Agenda

1. Continuous Integration
2. Continuous Delivery
3. Continuous Deployment
4. Benefits
5. Deployment Strategies
6. References

“

Continuous integration (CI) is the DevOps **practice** of automating the integration of code changes from multiple contributors into a single software project. It's a primary DevOps best practice, allowing developers to frequently merge code changes into a central repository where **builds and tests** .

”

[Learn more](#)

“

Continuous delivery is a DevOps **practice** where code changes are automatically **prepared for a release** to production. A pillar of modern application development, continuous delivery expands upon continuous integration by deploying all code changes to a testing environment and/or a production environment after the build stage.

”

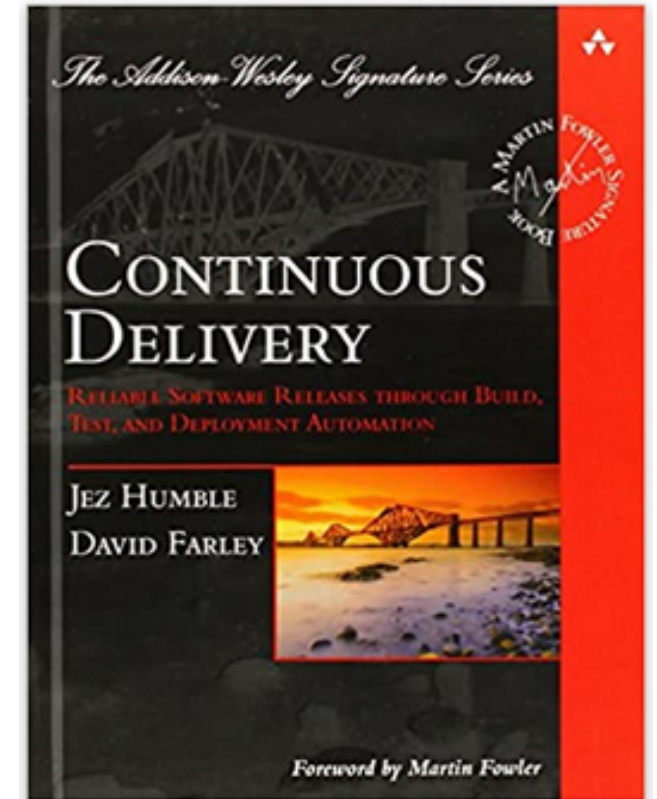
[Learn more](#)

“

Continuous Deploy is the ability to get changes - features, configuration changes, bug fixes, experiments - into production or into the hands of users safely and quickly in a sustainable way

”

[Learn more](#)



CI/CD

Continuous Integration

- Build: **A**
 - Unit Test: **A**
-
- Artifacts: M
 - Deploy to Test: M
 - Run Tests: M
 - Deploy to Prod: M

Continuous Delivery

- Build: **A**
 - Unit Test: **A**
 - Artifacts: **A**
 - Deploy to Test: **A**
-
- Run Tests: M
 - Deploy to Prod: M

Continuous Deploy

- Build: **A**
- Unit Test: **A**
- Artifacts: **A**
- Deploy to Test: **A**
- Run Tests: **A**
- Deploy to Prod: **A**

Key Ingredients

INTEGRATE

CD uses an integrated infrastructure

ORCHESTRATE

CD emphasizes orchestration of the environment

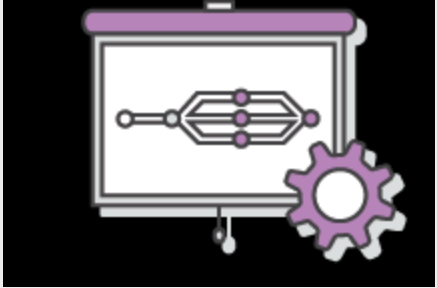
AUTOMATE

CD tasks are automated as much as possible

ACCELERATE

CD goal is to accelerate activities as early in the pipeline as possible

Benefits



Automate the
Software Release
Process



Improve Developer
Productivity



Find and Address
Bugs Quicker



Deliver Updates
Faster

Deployment Strategies

Rolling Deployment

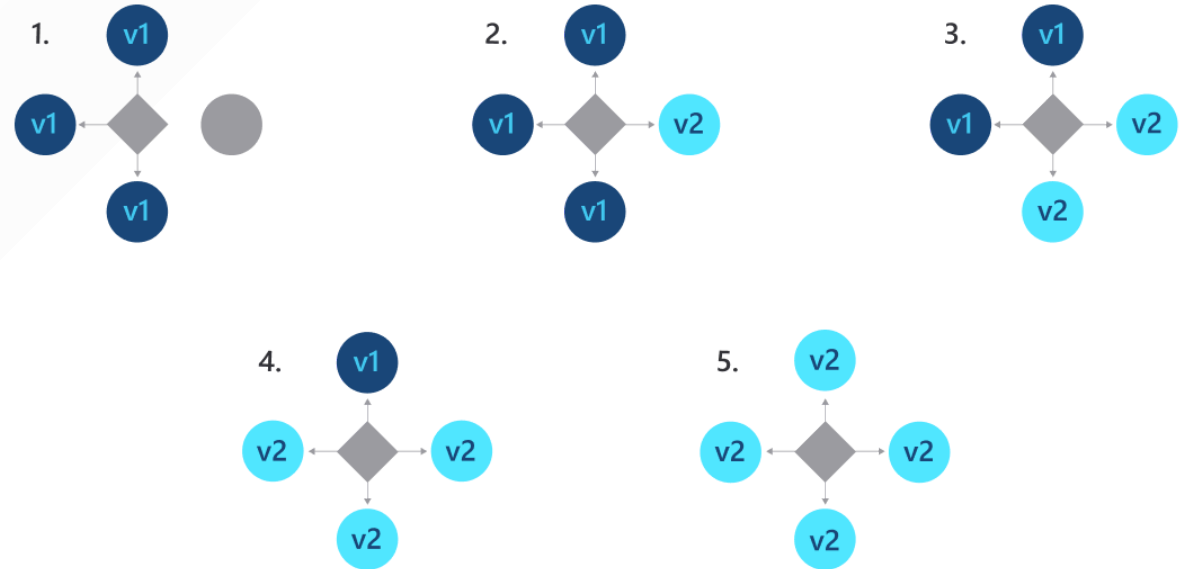
Blue/Green
Deployment

Canary Deployment

A/B Testing

Rolling Deployment

Kubernetes's default rollout method is a ramped or rolling deployment. This deployment slowly replaces pods one at a time to avoid downtime. Old pods are scaled down only after new pods are ready. If your deployment encounters problems, you can pause or cancel the Kubernetes deployment without taking the entire cluster offline.



Rolling Deployment

PRO

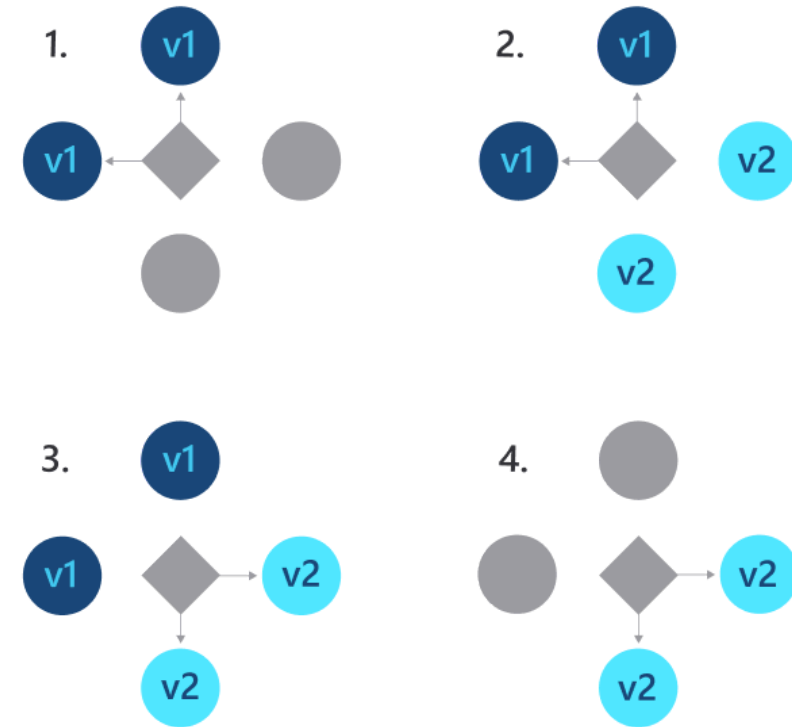
- Easy to set up
- Convenient for stateful applications that can handle rebalancing of the data

CON

- Rollout/rollback can take time.
- No control over traffic.

Blue/Green Deployment

A Blue-Green deployment strategy is one where the old and the new instances of the application or microservice run in parallel at the same time in production, and a load balancer switches traffic from the older version to the newer version instantly



Blue/Green Deployment

PRO

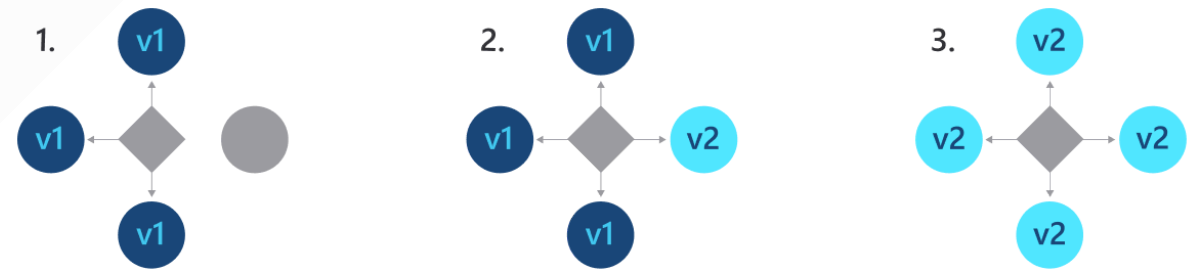
- Instant rollout/rollback.

CON

- Expensive as it requires double the resources.
- Proper test of the entire platform should be done before releasing to production.

Canary Deployment

Allow your customers to test your Kubernetes deployment by releasing the new version to a small group of them. You'll run one ReplicaSet of the new version along with the current version and then, after a specified period of time without errors, scale up the new version as you remove the old version.



Canary Deployment

PRO

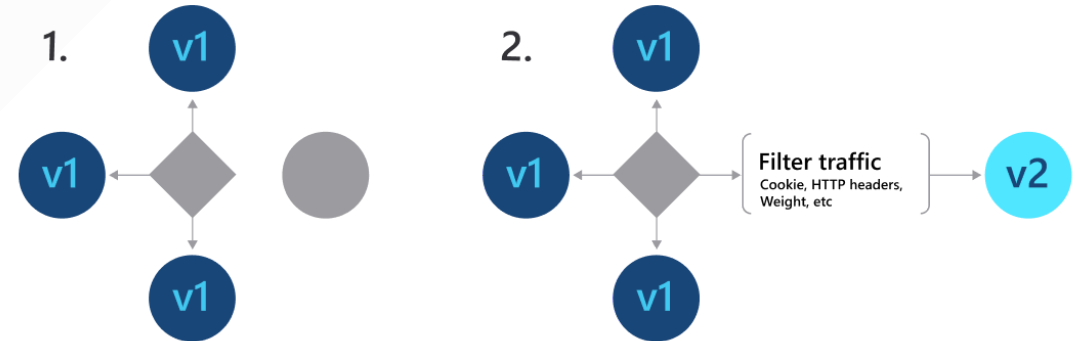
- Convenient for error rate and performance monitoring
- Fast rollback

CON

- Slow rollout

A/B Testing

A/B testing deployments consists of routing a subset of users to a new functionality under specific conditions. It is usually a technique for making business decisions based on statistics, rather than a deployment strategy.



A/B Testing

PRO

- Full control over the traffic distribution
- Several versions run in parallel

CON

- Requires intelligent load balancer
- Hard to troubleshoot errors for a given session, distributed tracing becomes mandatory.

Refernces

- [Atlassian Docs](#)
- [Continuous Integration Book](#)
- [Continuous Delivery Book](#)
- [Deployment strategy on Microsoft](#)
- [Deployment strategy on AWS](#)
- [Deployment strategies](#)