

Understand modern deployment patterns

1 minute

End-users always use your application differently. Unexpected events will happen in a data center, multiple events from multiple users will cooccur, triggering some code that hasn't been tested in that way.

To overcome, we need to embrace that some features can only be tested in production.

Testing in production sounds a bit scary, but that shouldn't be the case.

When we talked about separating our functional and technical releases, we already saw that it's possible to deploy features without exposing them to all users.

When we take this concept of feature toggling and use it with our deployment patterns, we can test our software in production.

For example:

- Blue-green deployments.
- Canary releases.
- Dark launching.
- A/B testing.
- Progressive exposure or ring-based deployment.
- Feature toggles.

Take a critical look at your architecture

Are your architecture and the current state of your software ready for Continuous Delivery?

Topics you might want to consider are:

- Is your software built as one giant monolith, or is it divided into multiple components?
- Can you deliver parts of your application separately?
- Can you guarantee the quality of your software when deploying multiple times a week?
- How do you test your software?
- Do you run one or multiple versions of your software?
- Can you run multiple versions of your software side by side?
- What do you need to improve to implement Continuous Delivery?

Next unit: Knowledge check

[Continue >](#)