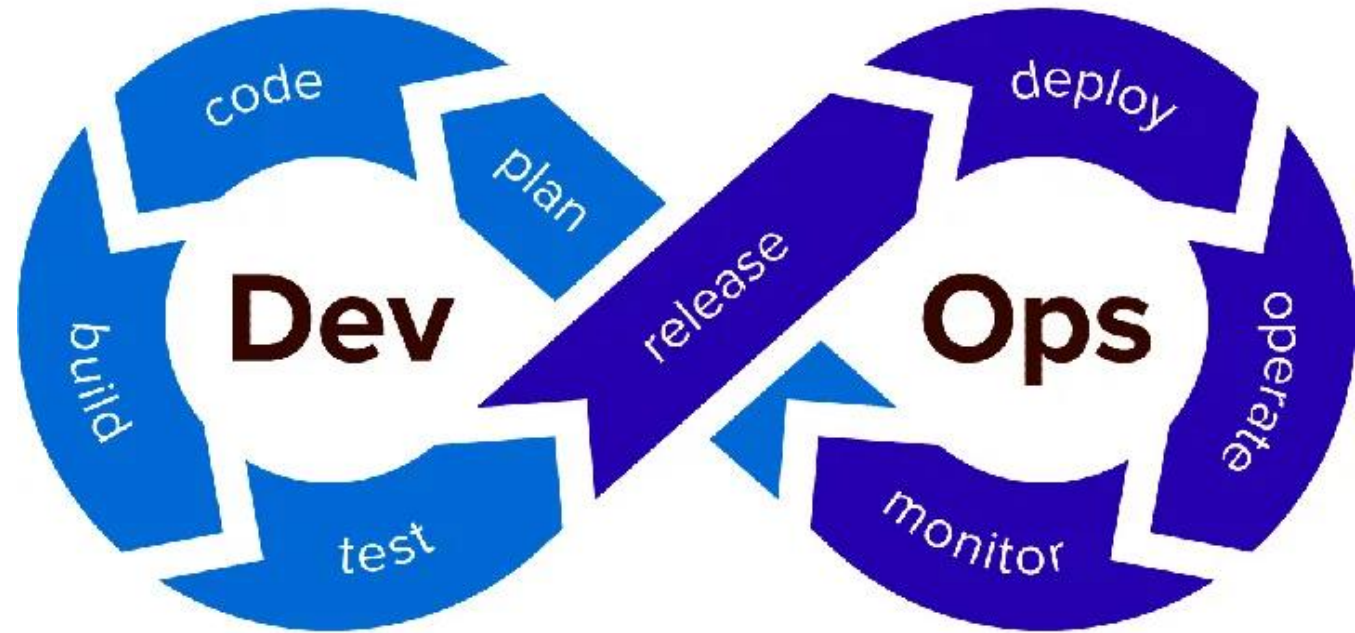


DevOps

From Dev to Ops to Dev

DevOps

Dev to Ops

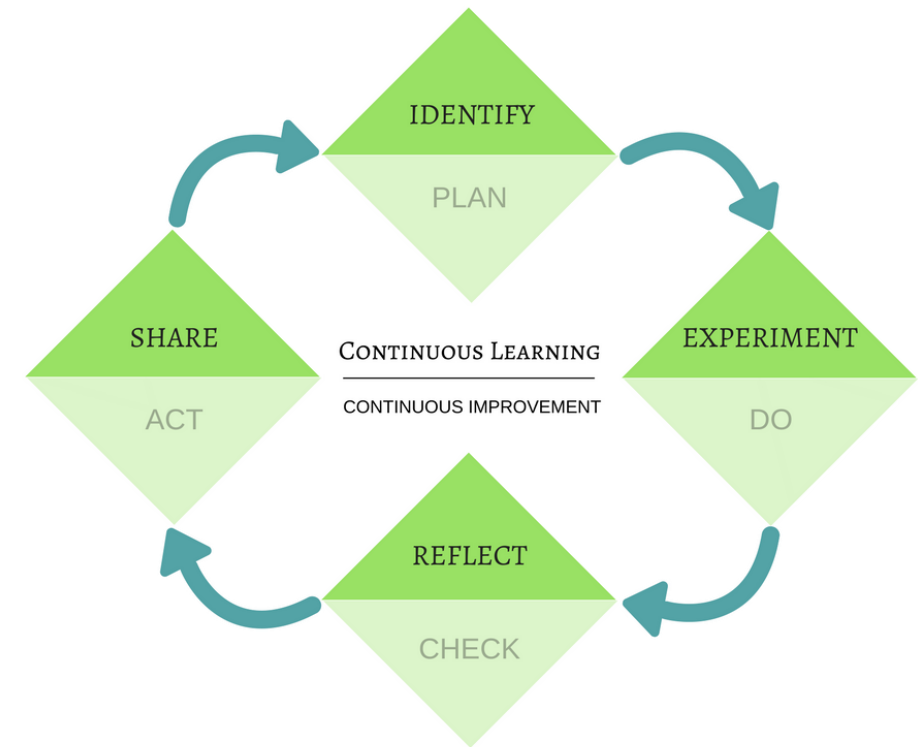


Outline (6 modules)

- *Industrialisation du déploiement logiciel (Pipelines de Continuous Delivery)*
- *Industrialisation de la construction et de la validation du logiciel*
- ***Gestion de configurations (construction et déploiement) pour différents environnements d'exécution***
- *Stratégies de surveillance d'exécution*
- *Automatisation de la surveillance d'exécution*
- *Gestion de l'évolution*

Outline (Methodology)

- Agile Learning
 - → Context
 - Me
 - Tooling (Introduction)
 - Roadmaps (DevOps/Application)
 - → Trying out / Applying / Testing
 - You
 - → Retrospective
 - Us
 - → Improve/Share
 - Us



DevOps

Gestion de configurations (construction et déploiement) pour différents environnements d'exécution

Outline

- Challenges
- Configurations
- Config Files
- Feature Flags

Introduction

Introduction

- Challenges
 - Configuration for Environments
vs
 - Builds for Environments
- Configuration Rollouts
- Handle multiple configurations

Configurations

Types and Locations

Configuration Types

- Log level
- Database
- URL
- Services
- ...

Configuration Locations

- 'Static'
 - Code
 - Config files
- 'Dynamic'
 - Databases
 - Service Calls
- ...

Config Files

Spring Boot Environment Configs

Properties

Spring Boot -> Profiles

- application.properties

server.port=

database.url=

...

- Properties files for environment
- Maven Profiles -> Spring Profiles

Properties (Spring)

- application.properties
server.port=8080
- application-dev.properties
server.port=8081
- application-prod.properties
server.port=8082
- ...

Properties (Maven)

- Pom.xml

```
<profiles>
  <profile>
    <id>dev</id>
    <build>
      <plugins>
        <plugin>
          <groupId>org.springframework.boot</groupId>
          <artifactId>spring-boot-maven-plugin</artifactId>
          <configuration>
            <profiles><profile>dev</profile></profiles>
          </configuration>
        </plugin>
      </plugins>
    </build>
  </profile>
  <profile>
    <id>prod</id>
    <build>
      <plugins>
        <plugin>
          <groupId>org.springframework.boot</groupId>
          <artifactId>spring-boot-maven-plugin</artifactId>
          <configuration>
            <profiles><profile>prod</profile></profiles>
          </configuration>
        </plugin>
      </plugins>
    </build>
  </profile>
</profiles>
```

- \$mvn spring-boot:run -Pprod

Feature Flags

Dynamic Configuration

Feature Flags

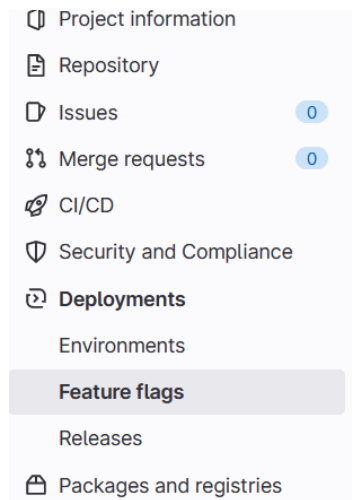
- Deployment Strategies
 - Blue Green
 - **Canary**
 - **A/B Testing**
 - ...

Feature Flags

- Feature in the Code
- API Calls to check execution
- Environment-based
- User-based

Gitlab -> Unleash

- Server
- User/Env based
- Integrated with Gitlab and environments



Feature flags

[View user lists](#)

[Configure](#)

[New feature flag](#)



Get started with feature flags

Feature flags allow you to configure your code into different flavors by dynamically toggling certain functionality. [More information](#)

Unleash

- Feature Flags Configuration

Configure feature flags

×

Install a [compatible client library](#) and specify the API URL, application name, and instance ID during the configuration setup. [More Information](#)

⚠

Set the Unleash client application name to the name of the environment your application runs in. This value is used to match environment scopes. See the [example client configuration](#).

API URL

https://mi-git.univ-tlse2.fr/api/v4/feature_flags/unleash/401

🔄

Instance ID

2FbySt_rppm

🔄

⚠

Regenerating the instance ID can break integration depending on the client you are using.

To prevent accidental actions we ask you to confirm your intention. Please type **Project Template** to proceed or close this modal to cancel.

Regenerate instance ID

Close

Unleash

- Environments

Project information

Repository

Issues0

Merge requests0

CI/CD

Security and Compliance

Deployments

Environments

Feature flags

Releases

Available2Stopped0

Clean up environmentsEnable review appNew environment

Q Search by environment name

> dev

Stop

> prod

Stop

Unleash

- Feature Flag

P Project Template

Project information

Repository

Issues0

Merge requests0

CI/CD

Security and Compliance

Deployments

Environments

Feature flags

Releases

Packages and registries

Infrastructure

Monitor

Analytics

Wiki

Snippets

Settings

New feature flag

Name *

feature_xyz

Description

My test feature

Strategies

Enable features for specific users and environments by configuring feature flag strategies.

Add strategy

Type

All users

Select strategy activation method?

Environments

+ prod X

Search

prod

dev

Select strategy activation method?

Unleash

- Libraries
 - Java -> https://docs.getunleash.io/sdks/java_sdk
- Add library to code
 - Maven:

```
<dependency>  
  <groupId>io.getunleash</groupId>  
  <artifactId>unleash-client-java</artifactId>  
  <version>9.1.0</version>  
</dependency>
```

Unleash

- Init Unleash

```
@Value("${spring.profiles.active}")
private String activeProfile;

//[...]

UnleashConfig config = UnleashConfig.builder()
    .appName(activeProfile) //Set tu the current environment of your application
    .instanceId("THE_ID_FROM_GITLAB")
    .unleashAPI("THE_URL_FROM_GITLAB")
    .build();

Unleash unleash = new DefaultUnleash(config);
```

- Use feature flag

```
if (unleash.isEnabled("my_feature")) {
    // do something
} else {
    // do something else
}
```