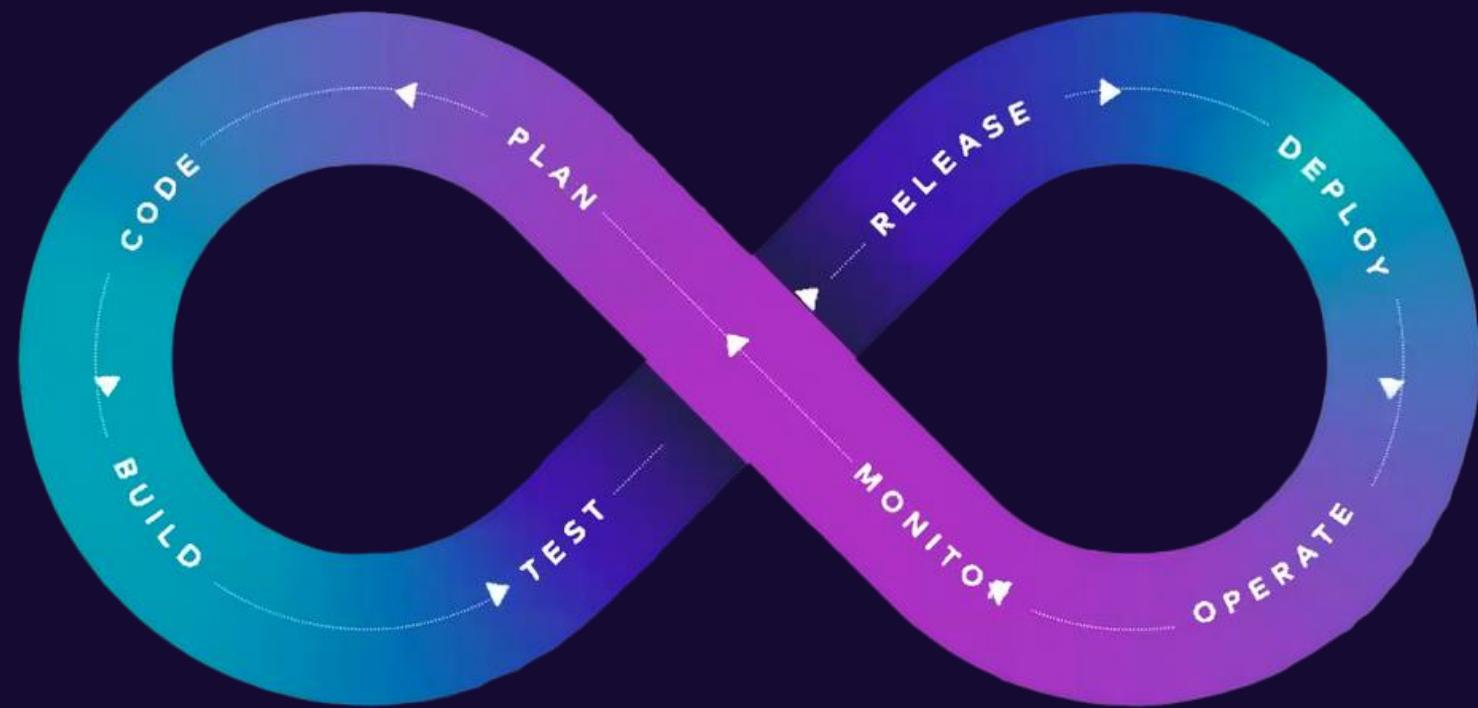


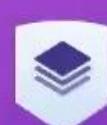


Solution de sécurité pour
développeurs



Maël Chevalier
Léo Charreau

IDE



Snyk Open Source



Snyk Code

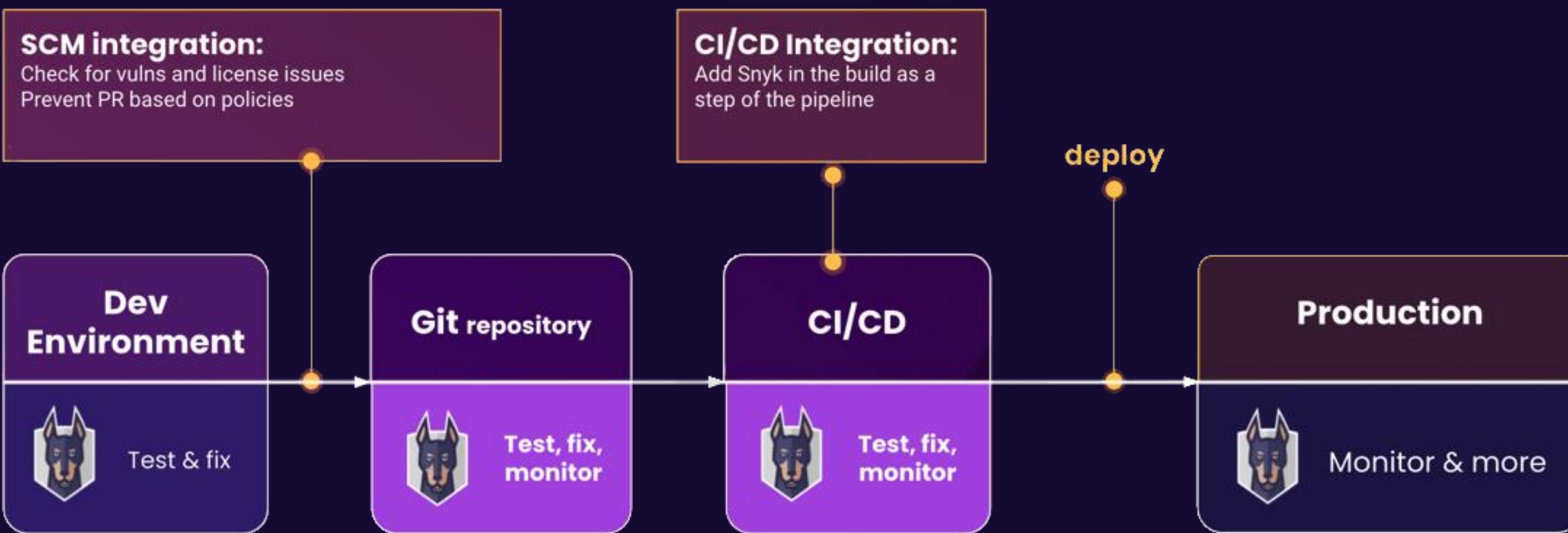


Snyk IaC



Snyk Container

Intégration



Correction auto.



Rapport

The screenshot shows a Snyk security report interface. At the top left is a user profile icon of a Doberman. Below it is the word "Rapport". The main area displays a summary of the report, including sections for "IMPORTED BY", "PROJECTS OWNER", "ENVIRONMENT", and "BUSINESS CRITICALITY". A large green button labeled "OPEN FIX REQUEST" with a gear icon is prominently displayed. On the left, there are filters for "ISSUE TYPE" (Reachable VULS, Exploit Maturity) and a search bar. The central part of the screen lists vulnerabilities with their scores: "MySQL.Data.EntityFrameworkCore" (Score 625) and "org.apache.struts:struts2-core" (Score 435). Each listing has a "FIX THIS VULNERABILITY" button.

Vulnerability	Score
MySQL.Data.EntityFrameworkCore	625
org.apache.struts:struts2-core	435

This screenshot shows a modal window titled "OPEN FIX PR" with the subtitle "ISSUES WITH A FIX". It contains a message: "AN UPGRADE OR PATCH IS AVAILABLE TO FIX THESE ISSUES". Below this, there are five small red icons with white letters (H, M, H, H, H) corresponding to the vulnerabilities listed in the main report. At the bottom of the modal is a large green button labeled "OPEN FIX REQUEST" with a gear icon.



Snyk Learn

snyk Learn

Lessons

Learning paths

Learning progress

JNDI injection

To exploit an example application vulnerable to JNDI injection we first create a malicious RMI server:

```
1 import java.rmi.registry.*;
2 import com.sun.jndi.rmi.registry.*;
3 import javax.naming.*;
4 import org.apache.naming.ResourceRef;
5
6 public class MaliciousRMIServer {
7     public static void main(String[] args) throws Exception {
8         //create our malicious RMI registry on port 1097 of our hosting server"
9         Registry registry = LocateRegistry.createRegistry(1097);
10
11         //this payload exploits unsafe reflection in org.apache.naming.factory.
12         BeanFactory
13         // NOTE: class namespace changed to jakarta.el.ELProcessor since Java 9
14         ResourceRef ref = new ResourceRef("javax.el.ELProcessor", null, "", "",
```

☕ ▾ **Code injection**

- Code injection: the basics**
- Code injection in action**
- Code injection under the hood**
- Code injection mitigation**



▼	2	comptegithubfac/term1	1 C 7 H 15 M 6 L	⊕
●	Code analysis	0 C 1 H 1 M 2 L	Tested 2 days ago	⚙️
M	pom.xml	1 C 6 H 14 M 4 L	Tested 2 hours ago	⚙️

25 of 25 issues

Sort by highest priority score ▾

C com.h2database:h2 - Remote Code Execution (RCE)

VULNERABILITY | CWE-94 ⓘ | CVE-2022-23221 ⓘ | CVSS 9.8 ⓘ | CRITICAL | SNYK-JAVA-COMH2DATABASE-2348247 ⓘ

SCORE 811

Introduced through com.h2database:h2@1.4.200
Fixed in com.h2database:h2@2.1.210

Show more detail ▾

NEW ⓘ Learn about this type of vulnerability ⓘ

Exploit maturity PROOF OF CONCEPT

Ignore Partially fix this vulnerability

H org.springframework.security:spring-security-web - Authorization Bypass

VULNERABILITY | CWE-285 ⓘ | CVE-2022-22978 ⓘ | CVSS 8.2 ⓘ | HIGH | SNYK-JAVA-ORGSPRINGFRAMEWORKSECURITY-2833359 ⓘ

SCORE 731

Introduced through org.springframework.boot:spring-boot-starter-security@2.6.7
Fixed in org.springframework.security:spring-security-web@5.5.7, @5.6.4

Show more detail ▾

Ignore Partially fix this vulnerability

H com.h2database:h2 - Remote Code Execution (RCE)

VULNERABILITY | CWE-94 ⓘ | CVE-2022-23221 ⓘ | CVSS 9.8 ⓘ | CRITICAL | SNYK-JAVA-COMH2DATABASE-2348247 ⓘ

SCORE 726

Intégration TER

Merci pour votre attention.



Maël Chevalier
Léo Charreau