



Meetup: DevSecOps

In drei einfachen Schritten die eigene WebApp sicher entwickeln

Michael Fuchs

\$whoami



 @TheEXiiLE

 theexiile1305

- › Studium B. Sc. Informatik @MUAS
Evaluation und exemplarische Implementierung eines sicheren Cloud-Speichers mit der Web Cryptography API
→ DOI: [10.13140/RG.2.2.11844.14724](https://doi.org/10.13140/RG.2.2.11844.14724)
- › Student M. Sc. IT-Sicherheit @MUAS
IDM, Zero Trust Networks & Secure Software Development
- › wissenschaftlicher Mitarbeiter @SecLab MUAS
starke IDMs & Zero Trust Networks
- › Software Developer & Security Engineer @inovex

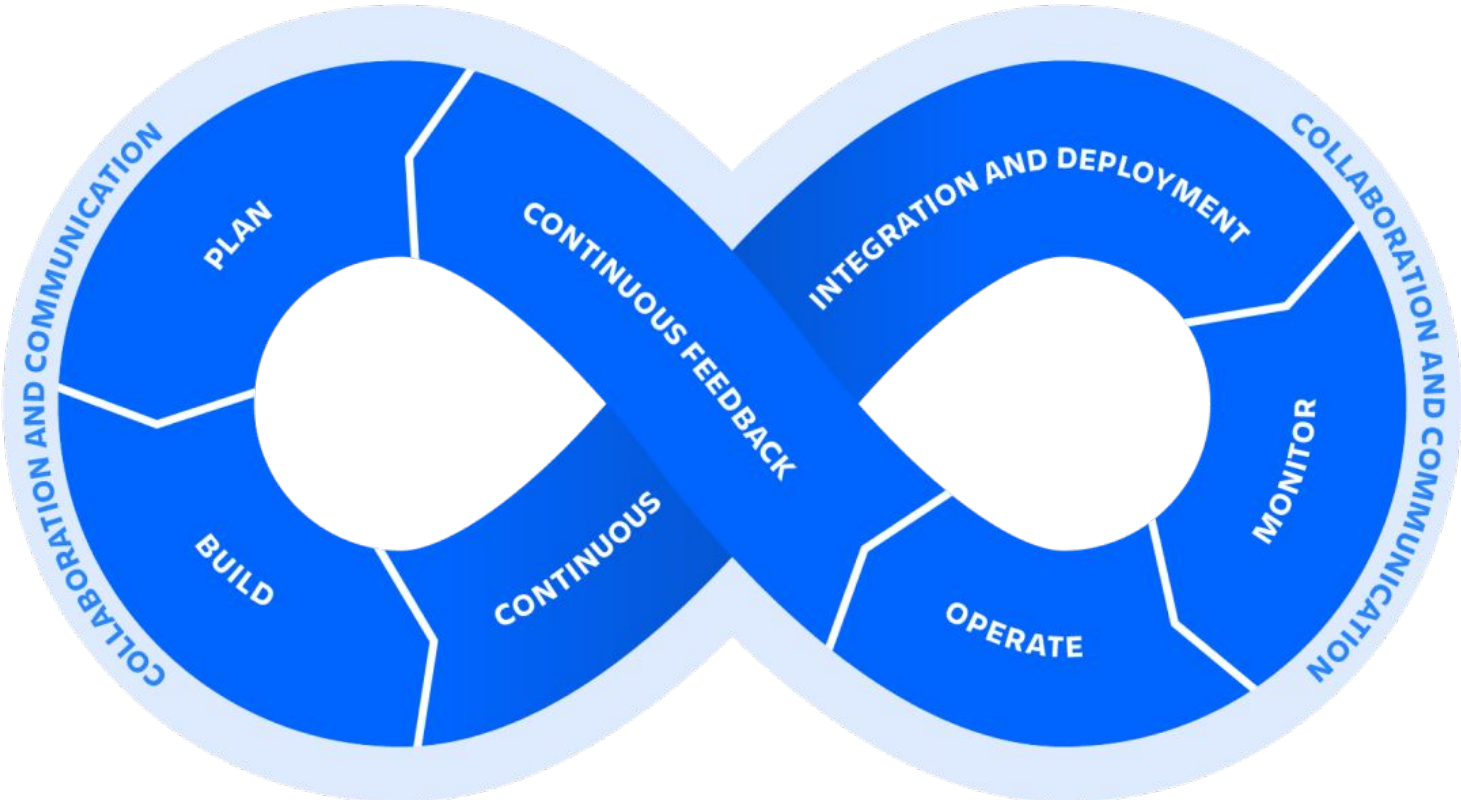
Motivation

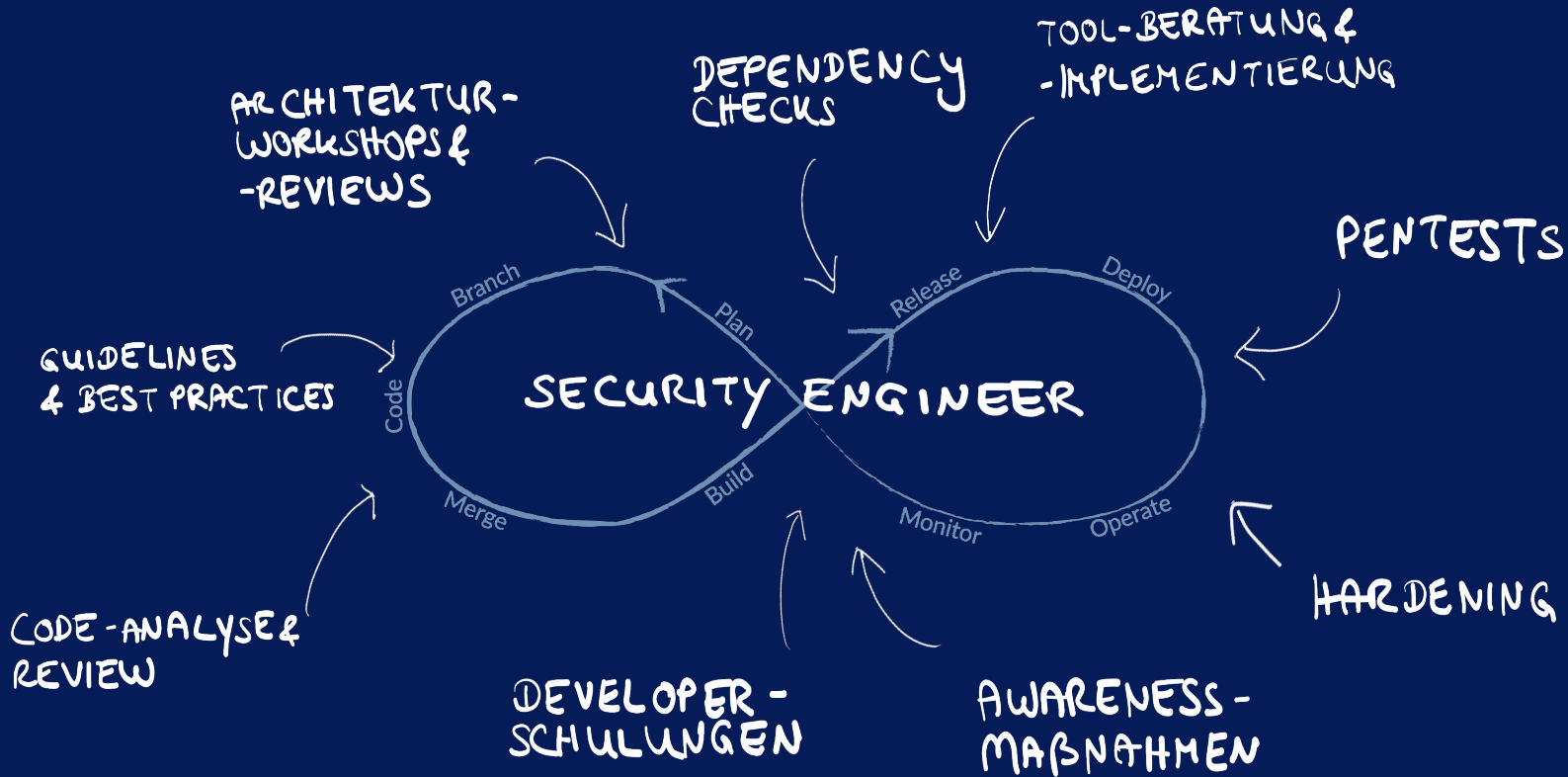


What are current challenges in Dev(Sec)Ops?

- › process/culture is based on the development team
- › rapid, practical programming conveyance
- › improved, also proactive security is needed
- › supply chain-attacks with increasing complexity

What actually constitutes DevOps?





How can we develop DevOps further?

DevOps moving at rapid pace

→ traditional security just can't keep up

DevSecOps

- › easier to manage rapid pace of development
- › management of large scale secure deployments
 - allows much smoother scaling

security as part of process is the only way to ensure safety

DevSecOps



What is DevSecOps

striving “Secure by Default”

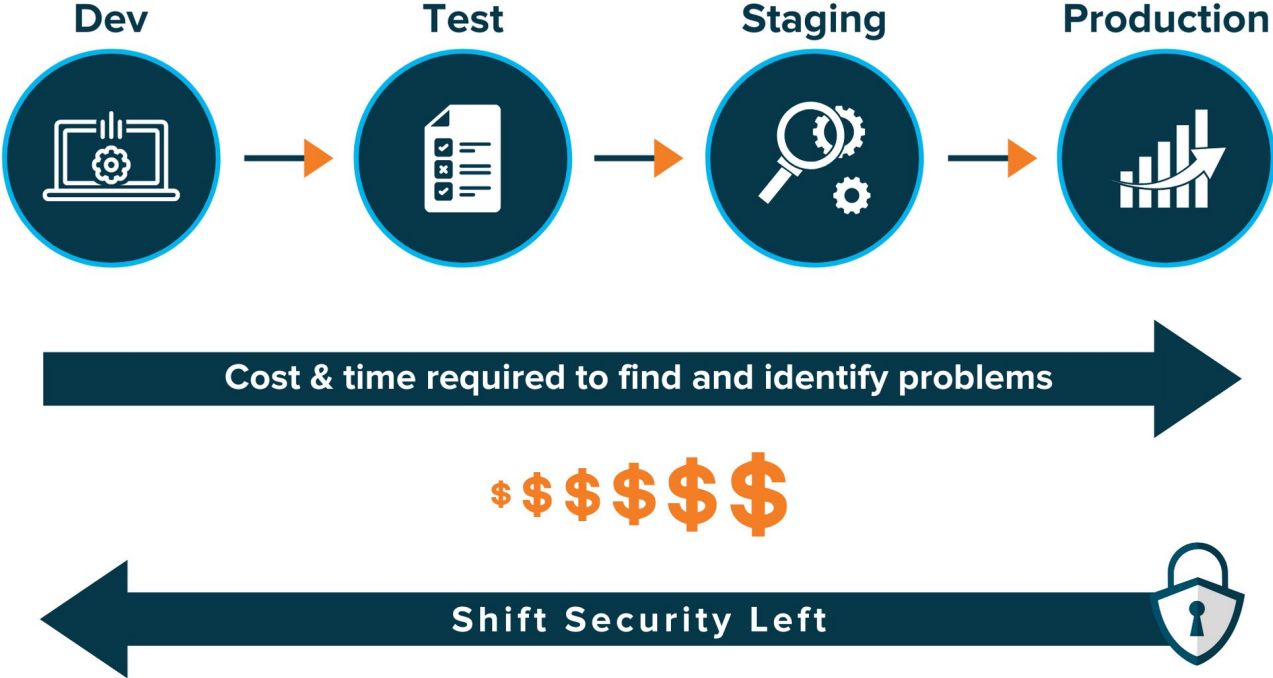
- › integrated security via **tools**
- › security as code **culture**
- › promote cross **skilling**



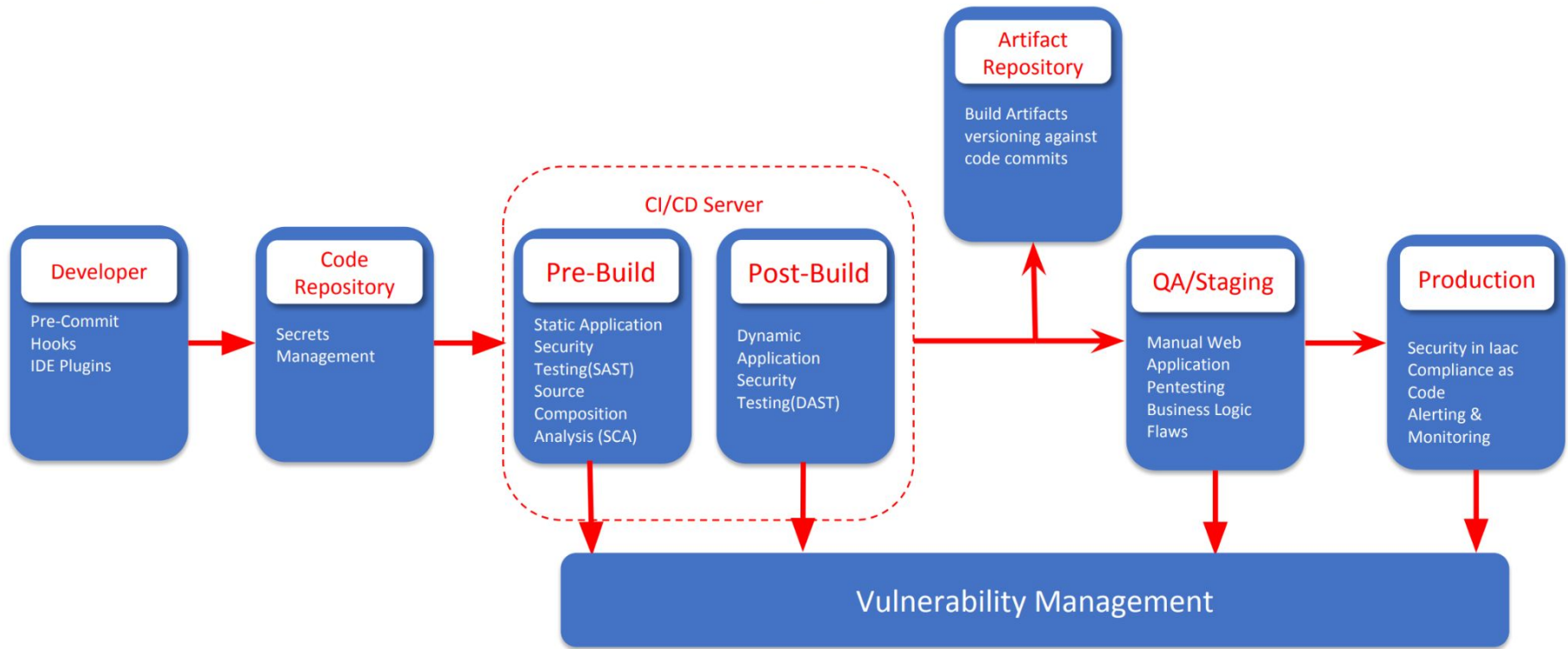
Key elements of DevSecOps

- › application/API Inventory
- › custom & Open-Source Code Security
- › runtime Protection
- › compliance monitoring
- › cultural factors

Shifting Left saves cost & time





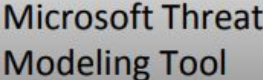



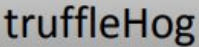
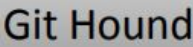


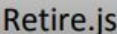












Injecting Sec in DevOps



Tools



What tools are available?

Threat Modelling Tools	  ThreatSpec	 Microsoft Threat Modeling Tool			
Pre-Commit Hooks	   git-secret	 truffleHog  Git Hound			
Software Composition Analysis	  Requires.io	 Retire.js			
Static Analysis Security Testing (SAST)	 Bandit	 RIPS	 sonarqube	 Pmd	
IDE Plugins	 skim	 CAT.net			
Secret Management	 HashiCorp Vault	 Keywhiz		 Confidant	

What tools are available?

Vulnerability Management	  
Dynamic Analysis Security Testing (DAST)	    
Security in Infrastructure as Code	 OpenVAS <small>Open Vulnerability Assessment System</small>   clair  
Compliance as Code	   DevSec Hardening Framework Docker Bench for Security
WAF	  

What are the issues with these tools?

- › Which one should I use for my problem?
- › every environment is different
- › advanced tooling requires
 - a lot of knowledge
 - complicated configuration
 - differentiation of false/positive rate
 - large investment costs

Which tools do I use to achieve quick results?

- › using beginner tools first **Why?**
 - quick win with a lot of benefits
 - no complex configuration is needed
 - can be used in roughly every environment

focus on:

- › secret management
- › dependency updates
- › error tracking

Secret Management



Secrets Management: Problems

- › often credentials are stored in config files
- › leakage can result in abuse scenario
- › secrets management allows you to tokenize the information

Secret Management: Solution

- › prevent leaking of sensitive information
 - before commit/push to repository
 - never share secrets via instant messaging apps, etc.

How to handle secrets than?

- › using dedicated software to share secrets with ease
 - based on gpg
 - easy to use and to manage receptions of secrets

Secret Management: pre-commit hooks

- › sensitive information often erroneously leaked due to accidental git commits
 - access keys, access tokens, SSH keys, etc.
 - pre-commit hooks on developer's side avoid that

Important:

- › if developers want they can circumvent this step hence use it like a defense in depth but don't fully rely on it

Practical Example:
pre-commit hooks with gitleaks

Practical Example: *detect-secrets*

usage with pre-commit framework

→ <https://github.com/pre-commit/pre-commit>

and adding gitleaks

```
repos:  
  - repo: https://github.com/zricethezav/gitleaks  
    rev: v8.2.0  
    hooks:  
      - id: gitleaks
```



Practical Example: *gitleaks* as pre-commit hook

via <https://github.com/zricethezav/gitleaks>

→ scanning with gitleaks detect

→ uncommitted changes in repository: gitleaks protect

→ Github Actions also possible

Practical Example: *gitleaks* with Github Actions

```
"github.com/jackc/pgconn"  
"github.com/jackc/pgx/v4"  
"github.com/jackc/pgx/v4/pgxpool"  
  
"github.com/gilcrest/diy-go-api/domain/errs"  
)  
  
const (  
    // DBHostEnv is the database host environment variable name  
    DBHostEnv string = "DB_HOST"  
    // DBPortEnv is the database port environment variable name  
    DBPortEnv string = "DB_PORT"  
    // DBNameEnv is the database name environment variable name  
    DBNameEnv string = "DB_NAME"  
    // DBUserEnv is the database user environment variable name  
    DBUserEnv string = "DB_USER"  
    // DBPasswordEnv is the database user password environment variable name  
    DBPasswordEnv string = "DB_PASSWORD"  
    // DBSearchPathEnv is the database search path environment variable name  
    DBSearchPathEnv string = "DB_SEARCH_PATH"  
  
    DB|  
)  
  
// PostgreSQLDSN is a PostgreSQL datasource name  
type PostgreSQLDSN struct {  
    Host      string  
    Port      int  
    DBName    string  
    SearchPath string  
  
datastore/datastore.go [+]  
-- INSERT --
```

Practical Example:

Secret Management with gopass

Practical Example: *gopass*

via <https://github.com/gopasspw/gopass>

- › git by default
- › easier management of recipients with gpg
- › binary data can also be stored
- › working on all OS

example:

<https://github.com/theexiile1305/meetup-devsecops-gopass>



Dependency Management



Dependency Management: Problems

- › product can be malfunction
- › able to use new features added in the latest versions
- › security issue fixes can be missed or delayed
- › reduce maintenance overheads of old versions
- › bug fixes are often contained in the new versions

Dependency Management: Solution

Requirements:

- › automated pull request
- › overview of release notes
- › maybe adoption, age and passing criteria

Basic prerequisite:

- › basic CI
- › tests should consists on unit/integration/security level
- › test sufficiently qualitatively as well as quantitatively

Practical Example:
*Dependency Updates with
renovate/depfu/dependabot*

Practical Example: *Dependency Updates*

via <https://github.com/theexiile1305/meetup-devsecops-mw>

- › several tools are available depending on tech-stack
 - *renovate, depfu, dependabot*
 - renovate enables self-hosted version
- › automated creation of pull requests with
 - release notes
 - age, adoption, passing and confidence measurements

Practical Example: *Dependency Updates*

Configure Renovate #3

Merged theexille1305 merged 1 commit into main from renovate/configure 6 hours ago

Conversation 0 Commits 1 Checks 0 Files changed 1 +6 -0

renovate bot commented 6 hours ago



Welcome to Renovate! This is an onboarding PR to help you understand and configure settings before regular Pull Requests begin.

To activate Renovate, merge this Pull Request. To disable Renovate, simply close this Pull Request unmerged.

Detected Package Files

- `gradle.properties` (gradle)
- `settings.gradle.kts` (gradle)
- `build.gradle.kts` (gradle)
- `gradle/wrapper/gradle-wrapper.properties` (gradle-wrapper)

Configuration Summary

Based on the default config's presets, Renovate will:

- Start dependency updates only once this onboarding PR is merged
- Enable Renovate Dependency Dashboard creation.
- If Renovate detects semantic commits, it will use semantic commit type `fix` for dependencies and `chore` for all others.
- Ignore `node_modules`, `bower_components`, `vendor` and various `test/test` directories.
- Autodetect whether to pin dependencies or maintain ranges.
- Rate limit PR creation to a maximum of two per hour.
- Limit to maximum 10 open PRs at any time.
- Group known monorepo packages together.
- Use curated list of recommended non-monorepo package groupings.
- A collection of workarounds for known problems with packages.

Would you like to change the way Renovate is upgrading your dependencies? Simply edit the `renovate.json` in this branch with your custom config and the list of Pull Requests in the "What to Expect" section below will be updated the next time Renovate runs.

What to Expect

With your current configuration, Renovate will create 3 Pull Requests:

- Update dependency `ch.qos.logback:logback-classic` to `v1.2.11`
- Update dependency `gradle` to `v7.5`
- Update dependency `io.sentry:sentry` to `v6.3.0`

Edit Code

Reviewers

No reviews

Assignees

No one—assign yourself

Labels

None yet

Projects

None yet

Milestone

No milestone

Development

Successfully merging this pull request may close these issues.

None yet

Notifications

Customize

Unsubscribe

You're receiving notifications because you're watching this repository.

1 participant

Lock conversation

Practical Example: *Dependency Updates*

Update dependency io.sentry:sentry to v6.3.0 #7

Edit <> Code ▾

Open renovate wants to merge 1 commit into `main` from `renovate/io.sentry-sentry-6.x`

Conversation 0

Commits 1

Checks 0

Files changed 1

+1 -1



renovate bot commented 2 hours ago

Contributor



This PR contains the following updates:

Package	Change	Age	Adoption	Passing	Confidence
io.sentry:sentry	6.0.0 -> 6.3.0	9d	20%	89%	neutral

Release Notes

▶ getsentry/sentry-java

Configuration

Schedule: Branch creation - At any time (no schedule defined), Automerge - At any time (no schedule defined).

Automerge: Disabled by config. Please merge this manually once you are satisfied.

Rebasing: Whenever PR becomes conflicted, or you tick the rebase/retry checkbox.

Ignore: Close this PR and you won't be reminded about this update again.

If you want to rebase/retry this PR, click this checkbox.

This PR has been generated by [Mend Renovate](#). View repository job log [here](#).

Reviewers

No reviews

Still in progress? Convert to draft

Assignees

No one—assign yourself

Labels

None yet

Projects

None yet

Milestone

No milestone

Development

Successfully merging this pull request may close these issues.

None yet

Notifications

Customize

Unsubscribe

You're receiving notifications because you're watching this repository.

0 participants

Lock conversation

Update dependency io.sentry:sentry to v6.3.0

Verified

d67e483

Practical Example: *Dependency Updates*

seclab > devsecops > Merge requests > !3

Update dependency org.springframework.boot:spring-boot-starter-parent to v2.7.2

Edit

Code

Open Michael Fuchs requested to merge [renovate/spring-boot](#) into [master](#) 2 minutes ago

Overview 0 Commits 1 Pipelines 1 Changes 1

This MR contains the following updates:

Package	Change	Age	Adoption	Passing	Confidence
org.springframework.boot:spring-boot-starter-parent (source)	2.0.1.RELEASE -> 2.7.2	7d	11%	?	neutral

Release Notes

▼ [spring-projects/spring-boot](#)

[v2.7.2](#)

[Compare Source](#)

:lady_beetle: Bug Fixes

- Publishing a docker image to a private registry fails without authentication [#31824](#)
- In a non-reactive application, health indicators in a parent context are not found [#31818](#)
- Dependency management for Derby is incomplete [#31814](#)
- ApplicationPid doesn't log a warning if it takes a long time to return [#31810](#)
- A router function with attributes causes /actuator/mappings to return a 500 response due to an UnsupportedOperationException [#31806](#)
- InstanceAlreadyExistsException when using Actuator with multiple context and JMX enabled [#31804](#)
- `UnionManagementConfigurationContext` in the test class loaded by system class loader throws `ClassNotFoundException` [#31803](#)

Error Tracking



Error Tracking: Problems

- › trail of events that lead to errors
- › releases provide visibility to which errors were addressed and which were introduced for the first time
- › quickly identify performance issues before they become downtime
- › enhancement of application monitoring with context

Error Tracking: Solution

- › central management of errors with
 - enrichment with click path and/or user behavior
 - especially from client side perspective
- › integration to other tools
 - e.g.: repository, management of tasks, CI/CD, communication, etc.
- › alerting whenever peak of unknown errors happen

Practical Example: *Error Tracking with Sentry*

Practical Example: *Error Tracking with Sentry*

The screenshot displays the Sentry web interface. On the left is a dark sidebar with navigation items: Projects, Issues, Performance, Releases, User Feedback, Alerts, Discover, Dashboards, Activity, Stats, and Settings. The main content area is titled 'Issues' and shows a list of unresolved issues. The first issue is an `IllegalStateException` with the message 'This is a test exception', reported 13 minutes ago. The second issue is a `NullPointerException`, reported 7 hours ago. Both issues are associated with the 'ktor' environment and the 'All Env' filter. The interface includes filters for environment, time range (14D), and search (is:unresolved). A table at the bottom shows the number of events and users for each issue.

	GRAPH:	24h 14d	EVENTS	USERS	ASSIGNEE
<input type="checkbox"/> <code>IllegalStateException</code> com.example.RoutingKt\$configureRo... This is a test exception New Issue KTOR-2 13min ago 7hr old			5	0	
<input type="checkbox"/> <code>NullPointerException</code> com.example.RoutingKt\$configureRo... New Issue KTOR-1 7hr ago 7hr old			1	0	

Practical Example: *Error Tracking with Sentry*

The screenshot displays the Sentry web interface for an issue titled "IllegalStateException com.example.RoutingKt...". The issue is marked as a "New Issue" and has 5 events, 0 users, and 1 assignee. The environment is "production" and the level is "error". The exception occurred in the "runtime" environment of "Oracle Corporation 1.8.0_332-heroku".

Environment Details:

- environment: production
- level: error
- runtime: Oracle Corporation 1.8.0_332-heroku
- runtime.name: Oracle Corporation
- server_name: 96ad1bf1-55fa-4143-b186-8aaefc8524a2.prvt.dyno.rt.heroku.com

Exception Details:

IllegalStateException
This is a test exception

Stack Trace (most recent call first):

- com.example.RoutingKt\$configureRouting\$1\$1 in invokeSuspend at line 18
- com.example.RoutingKt\$configureRouting\$1\$1 in invoke (5)
- io.ktor.server.routing.Route\$buildPipeline\$1\$1 in invokeSuspend at line 116
- io.ktor.server.routing.Route\$buildPipeline\$1\$1 in invoke (5)
- io.ktor.util.pipeline.SuspendFunctionGun in loop at line 123
- io.ktor.util.pipeline.SuspendFunctionGun in proceed at line 81
- io.ktor.util.pipeline.SuspendFunctionGun in executeSktor_utils at line 101
- io.ktor.util.pipeline.Pipeline in execute at line 77
- io.ktor.server.routing.Routing\$executeResult\$S\$inlined\$execute\$1\$1 in invokeSuspend at line 478
- io.ktor.server.routing.Routing\$executeResult\$S\$inlined\$execute\$1\$1 in invoke (5)
- io.ktor.util.debug.ContextUtilsKt in initContextInDebugMode at line 17
- io.ktor.server.routing.Routing in executeResult at line 174

Ownership Rules:

- Unknown User
- Unknown Bro...
- Oracle Corpor...
Version: 1.8.0_33...

Timeline:

- LAST 24 HOURS: 5 events
- LAST 30 DAYS: 5 events
- LAST SEEN: 13 minutes ago
- FIRST SEEN: 7 hours ago

Releases:

- See which release caused this issue

Issue Tracking:

- Link GitHub Issue

Tags:

- environment: production 100%
- level: error 100%
- runtime: Oracle Corporation 1.8.0_332-heroku 100%
- runtime.name: Oracle Corporation 100%
- server_name: 74087abd-9b6d-40f9-9f... 60%

Practical Example: *Error Tracking with Sentry*

The screenshot displays the Sentry web interface. A modal window titled "GitHub Issue" is open, allowing the user to create a GitHub issue from a selected Sentry error. The modal contains the following fields:

- Create** (selected) / **Link**
- GitHub Repository**: A dropdown menu showing "meetup-devsecops-mw".
- Title**: A text input field containing "IllegalStateException: This is a test exception".
- Description**: A text area containing the Sentry issue link: "Sentry issue: [KTOR-2](https://sentry.io/organizations/meetup-devsecops-inovex/issues/3460684660/?referrer=github_integration)" followed by the error details: "IllegalStateException: This is a test exception at com.example.RoutingKt\$configureRouting\$1\$1.invokeSuspend(Routing.kt:18) at com.example.RoutingKt\$configureRouting\$1\$1.invoke(Routing.kt) at com.example.RoutingKt\$configureRouting\$1\$1.invoke(Routing.kt)".
- Assignee**: A dropdown menu showing "Unassigned".
- Create Issue** button.

The background shows the Sentry error details for "IllegalStateException: This is a test exception" in the "production" environment. The error stack trace includes the following frames:

- io.ktor.server.routing.Routing\$executeResult\$inlined\$execute\$1 in invokeSuspend at line 478
- io.ktor.util.pipeline.Pipeline in execute at line 77
- io.ktor.util.pipeline.SuspendFunctionGun in execute\$ktor_utils at line 101
- io.ktor.util.pipeline.SuspendFunctionGun in proceed at line 81
- io.ktor.util.pipeline.SuspendFunctionGun in loop at line 123
- io.ktor.server.routing.Route\$buildPipeline\$1 in invoke at line 123

Outlook

- › **Static Application Security Testing (SAST)**
 - static code analysis that checks for pattern
 - point of source code and often give feedback

- › **Dynamic Application Security Testing (DAST)**
 - block bock testing
 - requires functioning product/application
 - based on libraries of automated attacks, fuzzing, etc.

Key Takeaways

- › Security is everyone responsibility
- › Embrace security as an integral part of the process
→ use feedback to refine the process
- › DevSecOps is not a one size fit all
→ your mileage will vary

Vielen Dank!

Michael Fuchs
Software Developer at
Application Development

inovex GmbH
Ludwig-Erhard-Allee 6
76131 Karlsruhe

michael.fuchs@inovex.de

