

DevSecOps Metrics

SGTS Learning Series

25th October 2023



Agenda



01

DevSecOps Metrics

Why, what and how

02

GitLab Dashboards

Pre-defined dashboards

03

DevSecOps Maturity Report and Dashboard MVP

Demo of the upcoming dashboard for SHIP-HATS Customers to monitor compliance with DevSecOps policies and best practices

Stack – Government-in-a-Box

www.developer.tech.gov.sg



Agency Specific IT System

SG Tech Stack (SGTS)

Service Layer

REUSABLE COMPONENTS

Base Layer

STANDARDISED DEV TOOLS

Network and Infra Services

Government on Commercial Cloud (GCC)

Legend: ● Live ● Planned ● Work in progress

Analytics

- Personalise
- WOGAA

Design

- SG Design System

Digital Identity

- Corppass
- Verify
- Identiface
- Authorise
- Singpass Login
- Notify
- MyInfo
- Sign

IoT

- DECADA IoT Stack

Data Science

- Video Analytics
- Transcribe
- GovText
- Fraud Detect

Data Transfer

- Cloud File Transfer
- Video Exchange

Communications

- APEX on Prem
- APEX (SDX)
- APEX Cloud
- APEX Marketplace

Toolchain

- SHIP-HATS



Runtime

- Container Stack
- IaC



Monitoring

- StackOps
- Endpoint Det & Res



Service Management

- Developer Portal
- SEED & TechPass
- TechBiz

SHIP-HATS has gained traction in the past two years



1

Reusable components sets us on a path to accelerate adoption

- ✓ **Easier for lean teams** to adopt DevSecOps with integrated GitLab
- ✓ **WOG collaboration, innersourcing & code contribution** is now possible
- ✓ 50+ Templates – to set pipelines, image repo etc.
- ✓ Closer integration with CSG e.g. XCA available as default
- ✓ Compliance pipeline feature to drive CICD best practices across WOG
- ✓ GovTech Source Code Mandate

2

Enhanced support & enablement to meet the users where they are

- ✓ **Handhold technical onboarding** by Solution Consultant team
- ✓ **Training Dev Sat +7% points** – ~180 Public Officers & vendors **monthly workshop** (95% NPS) and **bi-weekly webinar** 200-300 attendees
- ✓ **Enabling self help with open tech documentation.** Dev Sat improved **+7% points**

3

Automated DevSecOps measurements

- ✓ **Productivity Metrics** can be now measured with value stream data
- ✓ Upcoming DevSecOps Maturity Report and Dashboard

Dev Sat 75% even during migration

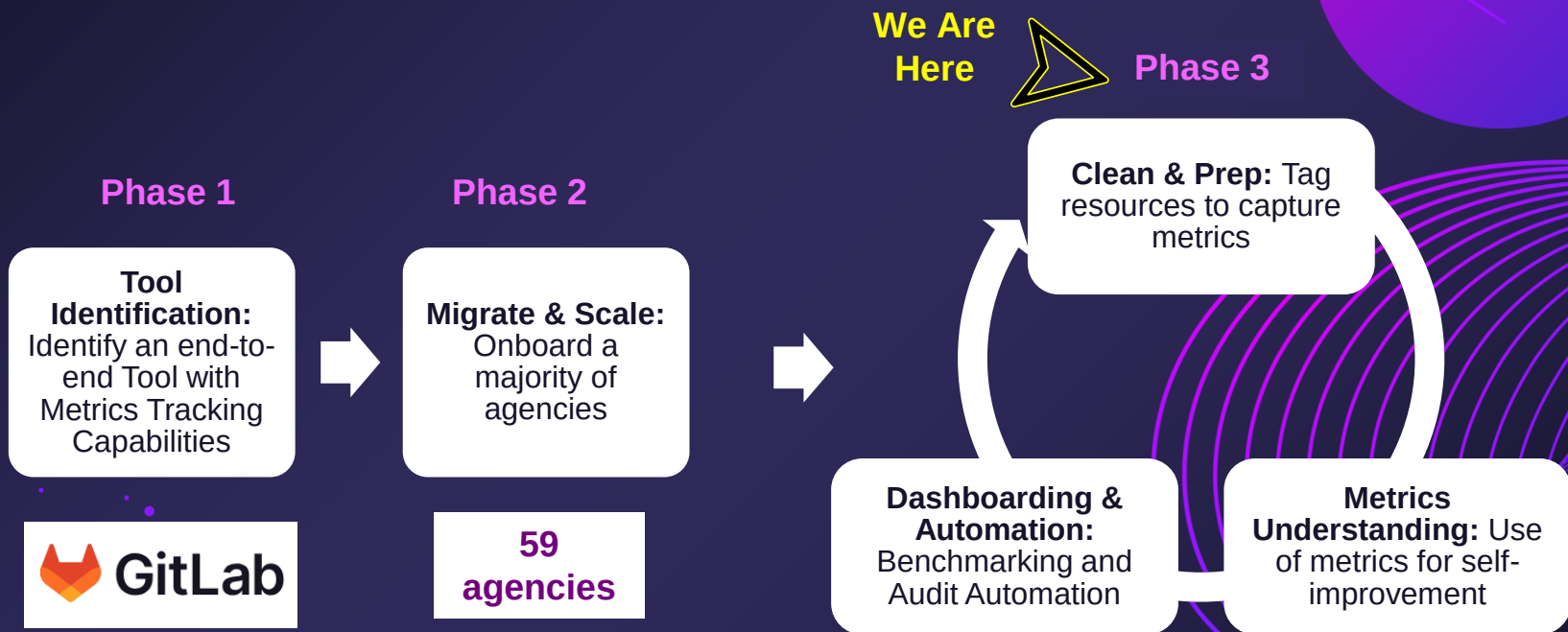
58 Agencies use SHIP-HATS incl. HTX, ICA, URA, MOM, MOE etc.

159 systems onboarded in 12 months to 552 Systems

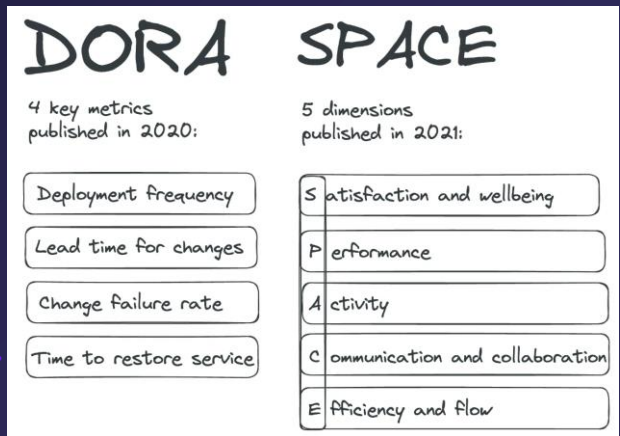
4,978 active users

Cumulative cost avoidance estimated at \$216m over 5 years*

Capturing DevSecOps Metrics

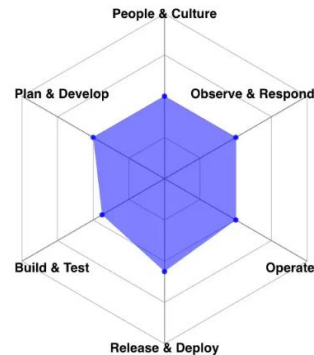


DevSecOps Measurements: Many Definitions



Results by Competency

Here is a breakdown of your results by competency area:



OWASP Devsecops Maturity Model



DSOMM

Introducing DORA

1. DORA - "DevOps Research and Assessment"
2. A research program that was founded in 2014 by Dr. Nicole Forsgren, Jez Humble, and Gene Kim
3. **State of DevOps report** provides insights on high-performing organizations.
4. **DevOps Assessment tool** identifies areas for improvement.
5. **High-performing IT organizations achieve business goals**

Priority

Deployment frequency

Lead time for changes

Change failure rate

Time to restore service

Introducing DORA

Priority

Deployment frequency

Lead time for changes

Change failure rate

Time to restore service

Deployment Frequency

High deployment frequency is generally associated with better software delivery performance

Measures how often code changes are deployed to production

Introducing DORA

Deployment frequency

Lead time for changes

Change failure rate

Time to restore service

Lead Time for Changes

Measures the time it takes for code changes to go from development to production

Short lead times are generally associated with better software delivery performance

Introducing DORA

Deployment frequency

Lead time for changes

Change failure rate

Time to restore service

Change Failure Rate

Measures the percentage of code changes that result in a service incident or require remediation

Low change failure rates are generally associated with better operational performance

Introducing DORA

Deployment frequency

Lead time for changes

Change failure rate

Time to restore service

Mean Time to Recover (MTTR)

Short MTTR is generally associated with better operational performance

Measures how long it takes to restore service when a service incident occurs

Industry Benchmarks vs SG Gov Realities

Software delivery performance metric	Elite	High	Medium	Low
 Deployment frequency For the primary application or service you work on, how often does your organization deploy code to production or release it to end users?	On-demand (multiple deploys per day)	Between once per week and once per month	Between once per month and once every 6 months	Fewer than once per six months
 Lead time for changes For the primary application or service you work on, what is your lead time for changes (i.e., how long does it take to go from code committed to code successfully running in production)?	Less than one hour	Between one day and one week	Between one month and six months	More than six months
Time to restore service For the primary application or service you work on, how long does it generally take to restore service when a service incident or a defect that impacts users occurs (e.g., unplanned outage or service impairment)?	Less than one hour	Less than one day	Between one day and one week	More than six months
Change failure rate For the primary application or service you work on, what percentage of changes to production or released to users result in degraded service (e.g., lead to service impairment or service outage) and subsequently require remediation (e.g., require a hotfix, rollback, fix forward, patch)?	0%-15%	16%-30%	16%-30%	16%-30%

- Do government practices enable us to benchmark against industry norms?

What are some peculiarities in the gov construct that might limit us?

How to use Deployment Frequency as a metric?



For Dev Teams: Deployment Frequency as a metric

Dev Teams

Continuous
Improvement

Risk Management

Agility

1. Continuous Improvement

- Identify trends and make improvements in development and release processes
- Measure how often new code or updates are deployed

2. Risk Management

- Insights into the stability of the deployment process

3. Agility

- Higher deployment frequency reflects an agile development approach
- Allows teams to respond quickly to market demands and customer feedback

For Product Owners: Deployment Frequency as a metric

1. Feedback Loops

- Crucial for product iterations and improvements
- Frequent deployments enable faster feedback loops with users

2. Alignment with Business Goals

- Align efforts with business objectives
- Ensure timely release of features and fixes

3. Competitive Advantage

- Ability to deploy frequently can be a competitive advantage
- Stay ahead in the fastpaced tech industry

Product Owners

Feedback Loops

Alignment with
Business Goals

Competitive
Advantage

Strategies to avoid “Gaming” or Manipulations

1. **Define Clear Criteria** tied to meaningful changes, such as new features, bug fixes, or improvements, rather than trivial or inconsequential updates.
2. **Automate Data Collection** to collect deployment data that reduces the potential for manual manipulation or errors in reporting.
3. **Code Reviews and Approval Processes** so that changes are thoroughly reviewed and meet the criteria for deployment.
4. **Regular Checks** to look for unusual patterns or anomalies that could indicate gaming. Be vigilant in monitoring trends and anomalies.
5. **Focus on Outcomes**; Shift the focus from deployment frequency to outcomes that matter, such as customer satisfaction, system reliability, and business impact. This can reduce the temptation to game the metric.

Deployment Frequency in SHIP-HATS Today

- 500+ systems, across 58 agencies
- Majority are not tagged to production resulting in 0 deployments

	Elite	High	Med	Low	0
Systems (over the past 90 days)	6	8	57	86	396

Elite	High	Medium	Low
On-demand (multiple deploys per day)	Between once per week and once per month	Between once per month and once every 6 months	Fewer than once per six months

What does it mean for us?

DevSecOps Culture

- Collaboration
- Communication
- Continuous improvement



Good Practices

- Shift-Left: Integrate security testing and test early
- Implement CI/CD to automate the build, test and deploy processes
- Role segregation
- Etc.



Outcomes

- Apps for citizens and businesses that are secure and of high quality
- Customer satisfaction
- Agility to respond to policy and changing needs

Measured by DORA,
Insights derived by tools like the DevSecOps Dashboard



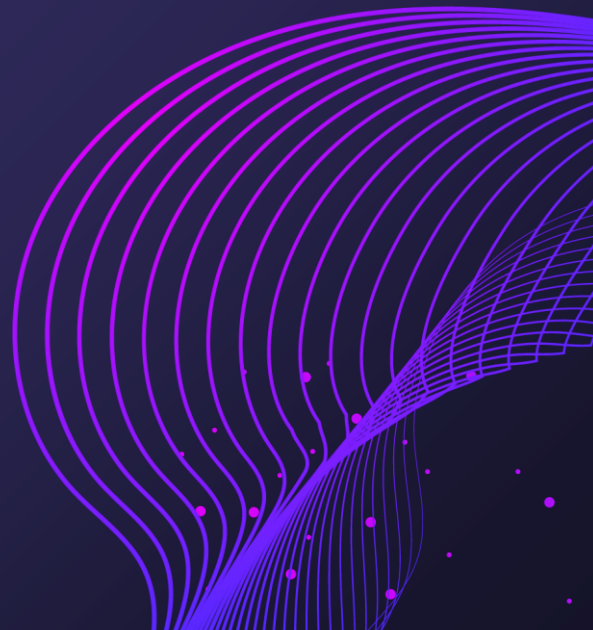
DevSecOps Maturity Report & Dashboard

Case study: Example of Insecure DevSecOps Practice

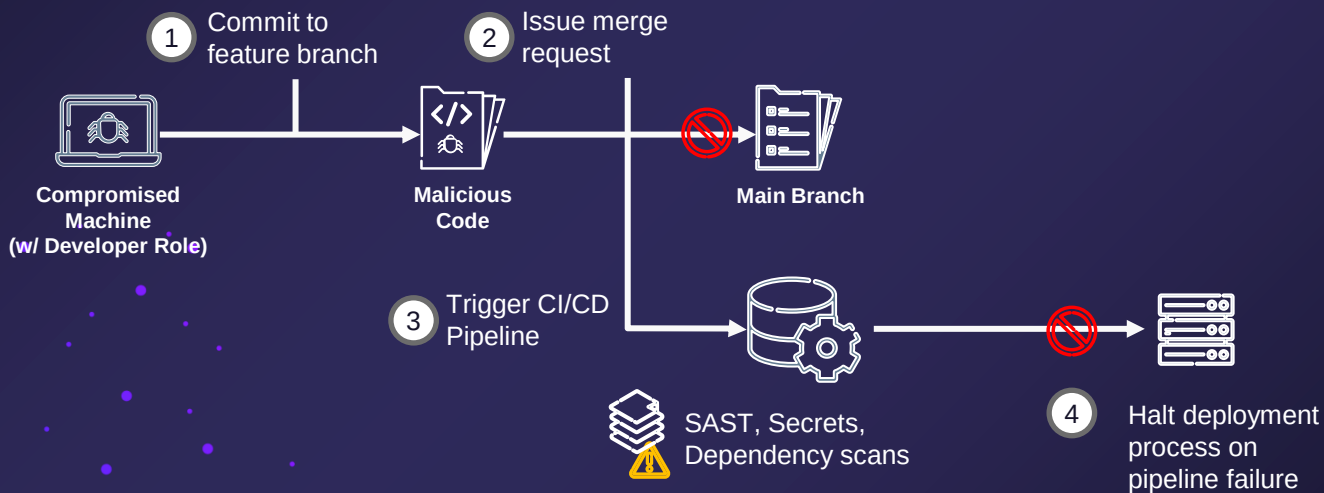
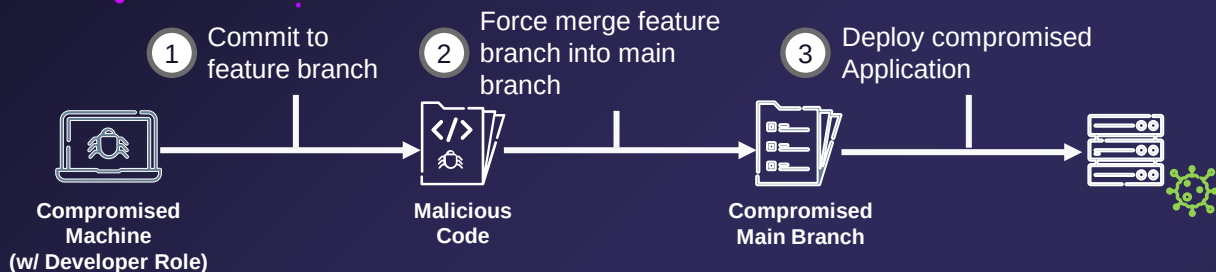


Sushi

- In Sept 2021, SushiSwap's token launchpad (MISO) got hacked for \$3 million
- The attacker injected a malicious code into the front-end private repository (miso-studio) on GitHub, substituting the legitimate wallet address with their own
- 864.8 Eth tokens collected for an automobile auction were diverted to the attacker's wallet
- The PRs review process wasn't enforced in git branch protection, enabling the attacker to independently deploy the malicious code



Bad vs Good DevSecOps Practice



3 key challenges for good DevSecOps practices



Not knowing what good looks like

Secure configurations and best practices



Lack of visibility on current security posture

Overall view and statistics of issues



Dealing with manual governance toil

Collation of information, manual periodic scans

And we can solve these challenges by...



Not knowing what good looks like

Codifying security controls and best practices, as well as providing prescriptive guidance for teams to follow



Lack of visibility on current security posture

Providing consolidated reports/dashboard for shared visibility and insights on DevSecOps practice maturity across projects



Dealing with manual governance toil

Providing automated scans on development pipelines configurations, practices and tools used, and consolidating that data for easy reference

Introducing: DevSecOps Maturity Report and Dashboard

	Report	Dashboard
Target users	Devs	Devs, PMs, CIOs/CISOs
Branch	Any branch	Main branch
Frequency of Scans	Every commit	Every 24hr
Onboarding	Include CI/CD template provided by SHIP-HATS	By-default for all SHIP-HATS Users
Use-Case	Observe real-time updates for a single repository whenever a commit is made	View results collectively
Availability	Available Today	MVP Mode – Coming Soon!
Data source	SHIP-HATS GitLab	

The DevSecOps Maturity Report



- The DevSecOps Maturity Report provides developers immediate feedback on their DevSecOps maturity score for any branch in their code repository
- Developers would have to include a template (provided by SHIP-HATS) into their CI/CD pipelines
- The report can be configured to be updated each time there is a new commit, and can be viewed on GitLab pages

ENVIRONMENT VARIABLES

PROJECT_ID

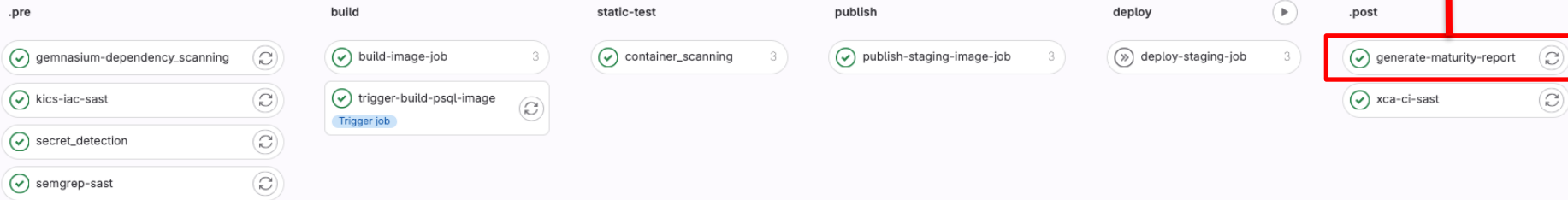
PROJECT_TOKEN



```
include:
  - project: "WOG/GVT/ship/ship-hats-templates"
    ref: "main"
    file:
      - "/templates/.gitlab-ci-generate-maturity-report.yml"

stages:
  - generate-report

gen-report-job:
  stage: generate-report
  extends: .generate-maturity-report
```





i This is a non exhaustive list of checks to ensure full compliance to DevSecOps maturity as per various standards. We are constantly working to add more checks and improve computation of checks to help uplift WOG DevSecOps maturity. It is in our plans to enable such that users are able to check off items that do not apply to them. This [project](#) is open for innersourcing contributions and constructive ideas are welcome. **X**

SHIP HATS Maturity Report

QD Controller

Standard

IM8



Passed:

9

Failed:

7

[SOURCE CODE](#)

[PROCESS](#)

[MATERIALS](#)

[PIPELINE](#)

[ARTEFACTS](#)

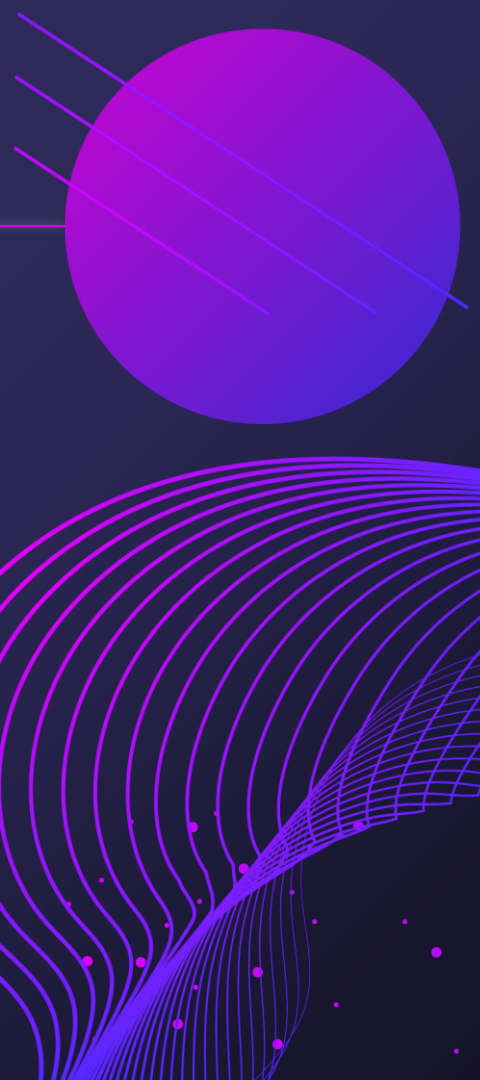
[DEPLOYMENTS](#)

☐ SHOW FAILED ONLY

RULE	INFO	SCORE	HOW TO PASS?
PUSH RULES	-	FAIL	^
PUSH RULES UNSIGNED COMMITS ARE REJECTED	Rejecting unsigned commits is essential for maintaining the integrity and security of a software dev... more	FAIL	VIEW ACTIONS
PUSH RULES SECRET FILES ARE NOT ALLOWED TO BE PUSHED	Prevention of pushing secret files is of paramount importance to uphold data security and maintain t... more	PASS	VIEW ACTIONS
PUSH RULES MERGE REQUEST CANNOT BE APPROVED BY THE COMMITTER	Enforcing the rule that merge requests cannot be approved by the committer is essential to maintaini... more	PASS	VIEW ACTIONS
PUSH RULES MERGE REQUEST CANNOT BE APPROVED BY THE AUTHOR	Enforcing the rule that merge requests cannot be approved by the author is essential for maintaining... more	PASS	VIEW ACTIONS
PUSH RULES UNVERIFIED USERS ARE NOT ALLOWED TO COMMIT	Disallowing unverified users is paramount for maintaining the robustness and security of the softwar... more	PASS	VIEW ACTIONS
MERGE REQUEST	-	FAIL	^
MERGE REQUEST RULES ONLY MAINTAINERS ARE ALLOWED TO APPROVE MERGE REQUEST	Enforcing the rule that only maintainers can approve merge requests is vital to maintaining a contro... more	FAIL	VIEW ACTIONS
MERGE REQUEST RULES MERGE REQUEST APPROVAL RULES HAVE AT LEAST ONE APPROVER	Enforcing the rule that merge request approval rules must have at least one approver is essential fo... more	PASS	VIEW ACTIONS

The DevSecOps Maturity Dashboard (Upcoming)

- The DevSecOps Maturity Dashboard will start off by providing users (Devs, PMs, CISOs, CIOs, etc) an overview of their DevSecOps maturity score for a project's main branch
- Projects that are on SHIP-HATS will be scanned automatically on a daily basis, and the dashboard will reflect the results accordingly
- In the future, the dashboard will be able to provide statistics and trends over time





Projects

🔍 Search

Filters

Maturity Level ▾



MASH Backend

wog/gvt/csg/acpd

DORA Metrics ⓘ DF: - LTC: - MTTR: - CFR: -

Members

C K X +10

Fail

17

Pass

12

Maturity Level ⓘ

Low

Last Scan

6 hours ago

MASH Frontend

wog/gvt/csg/acpd

DORA Metrics ⓘ DF: - LTC: - MTTR: - CFR: -

Members

C K X +10

Fail

15

Pass

14

Maturity Level ⓘ

Low

Last Scan

6 hours ago

CloudSCAPE Backend

wog/gvt/csg/acpd

DORA Metrics ⓘ DF: 14 days LTC: 5 days MTTR: 3 days CFR: 6.0%

Members

C K X +10

Fail

13

Pass

16

Maturity Level ⓘ

Medium

Last Scan

6 hours ago

CloudSCAPE Frontend

wog/gvt/csg/acpd

DORA Metrics ⓘ DF: 12 days LTC: 7 days MTTR: 2 days CFR: 5.0%

Members

C K X +10

Fail

14

Pass

15

Maturity Level ⓘ

Medium

Last Scan

6 hours ago

CodeSCAPE Frontend



Group: wog/gvt/csg/acpd Language: Javascript 78%, Java 20%, C++ 1.5%, Python 0.5%

Improve your maturity level by
remediating the failed checks

5

Failed

24

Passed



82.8%

Your Maturity level is High ⓘ

DORA Metrics ⓘ

9 per day

Deployment
Frequency

3 days

Lead time for
changes

1 day

Mean time to
recover

4.0 %

Change failure
rate

Scan Results

Group By: Category



👁 Passed Checks

SHIP-HATS GitLab Project Settings

0



15



Project Rules - Project Has Multiple Permission Levels

IM8, OWASP, SLSA



Push Rules - Unsigned Commits Are Rejected

IM8, CNCF, SLSA



Push Rules - Secret Files Are Not Allowed To Be Pushed

IM8, OWASP, CNCF, SLSA



Protected Branches - Default Branch Is Protected

IM8, OWASP, CNCF, SLSA



Protected Branches - Protected Branches Are Safeguarded Against Force Push

IM8, CNCF, SLSA

Push rules - Unsigned commits are rejected Pass

IM8, CNCF, SLSA

Description

Rejecting unsigned commits is essential for maintaining the integrity and security of a software development workflow. Unsigned commits lack a digital signature that is used to verify the authenticity of the commit author's identity and the integrity of the code changes. This opens the door to potential unauthorised and malicious code alterations, making it difficult to trace the origin of changes and leading to an increased risk of code injection, unauthorised access, and code tampering.

By enforcing signed commits through GPG/SSH keys, you reduce the likelihood of unauthorised changes in your project and foster a more secure and transparent collaboration environment among developers.

Remediation Steps:

1. Install [GPG](#) for your Operating System

For MacOS ensure that [homebrew](#) is installed

```
brew install gnupg
```

For Windows ensure that [chocolatey](#) is installed

```
chocolatey install gnupg
```

2. Run the following command to generate key pair

```
gpg --gen-key
```

Recap: Report & Dashboard

	Report	Dashboard
Target users	Devs	Devs, PMs, CIOs/CISOs
Branch	Any branch	Main branch
Frequency of Scans	Every commit	Every 24hr
Onboarding	Include CI/CD template provided by SHIP-HATS	By-default for all SHIP-HATS Users
Use-Case	Observe real-time updates for a single repository whenever a commit is made	View results collectively
Availability	Available Today	MVP Mode – Coming Soon!
Data source	SHIP-HATS GitLab	

Future Roadmap for dashboard

- Introduce the MVP to a select group of pilot users to collect their feedback and insights.
- For the dashboard to expand scope of coverage from main branch to all branches
- On demand updates instead of daily updates for the dashboard
- Register your interest today!



In Summary



- DevSecOps Metrics should be used meaningfully to better understand their dev activities and productivity
- Start by monitoring Deployment Frequency as a tool to identify gaps in product delivery
- Tag GitLab pipelines that are deploying to production in order to capture Deployment Frequency data
- Utilise tools like the DevSecOps Report/Dashboard to track compliance to IM and best practices



Thank you!
