SGTS Product Overview

shiphats

A Product of GDS, GovTech

Agenda

- Say Hello to THE Singapore Government Tech Stack (SGTS)
- Introducing **SHIP-HATS: AN SGTS Product**
  - Under-the-hood
  - Benefits
  - Technical Overview
- How to get started
  - Self Learn with Tech Docs
  - Workshops & Webinars
  - Subscription Model
- What's next?

GOVTECH SINGAPORE

Singapore Government Tech Stack (SGTS)

Say Hello to

SGTS

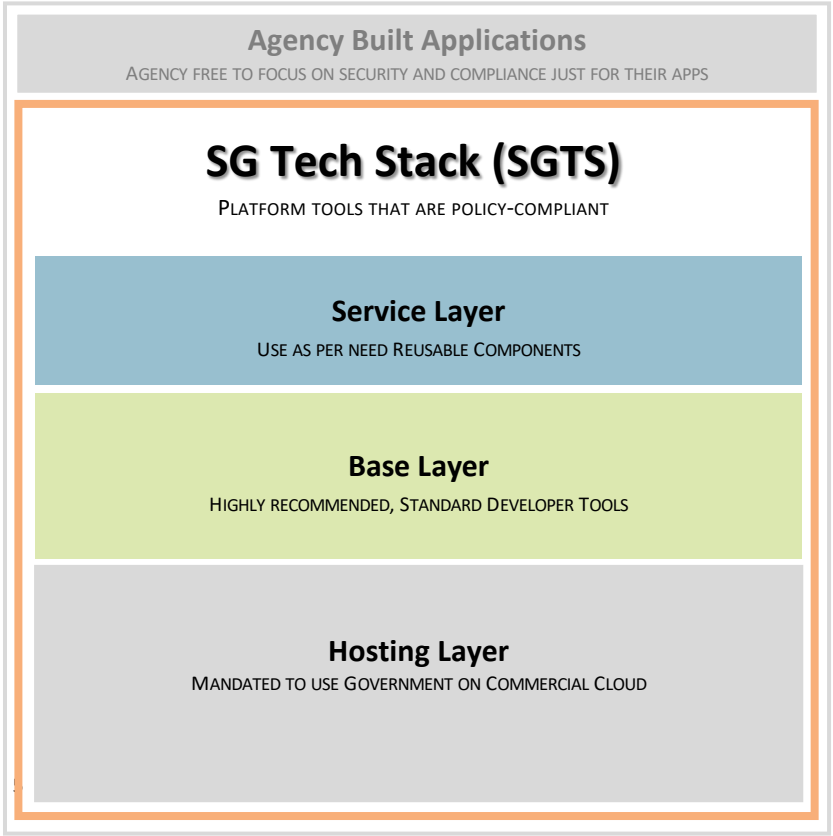
A set of **central platform tools** that streamlines and **simplifies the development and monitoring process** and enables code reuse across WOG to build and monitor secure, high quality applications.

#platformengineering

GOVTECH SINGAPORE

Say Hello to

SGTS



GOVTECH SINGAPORE

SGTS - Unpacked

Agency Built Applications

AGENCY FREE TO FOCUS ON SECURITY AND COMPLIANCE JUST FOR THEIR APPS

SG Tech Stack (SGTS)

PLATFORM TOOLS THAT ARE POLICY-COMPLIANT

Service Layer

USE AS PER NEED REUSABLE COMPONENTS

Base Layer

HIGHLY RECOMMENDED, STANDARD DEVELOPER TOOLS

Hosting Layer

MANDATED TO USE GOVERNMENT ON COMMERCIAL CLOUD

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
- ADEX (SDX)
- APEX Cloud
- APEX Marketplace

Toolchain
- SHIP-HATS

shiphats

Runtime
- Container Stack

container stack

Monitoring
- StackOps

stackops

Service Management
- Developer Portal
- SEED & TechPass
- TechBiz

Benefits of SGTS

Engineers

Access to templates, and self-service capabilities with **automated** infrastructure operations that are IM-compliant, to **reduce cognitive workload** on engineers.

PMs/CIOs

Increased transparency into the development process, and greater operational insights

Agencies

Levelling up of **DevSecOps maturity** within agencies and across the government

Vendor Partners

Enhanced capabilities ready to partner various government agencies deliver apps using SGTS

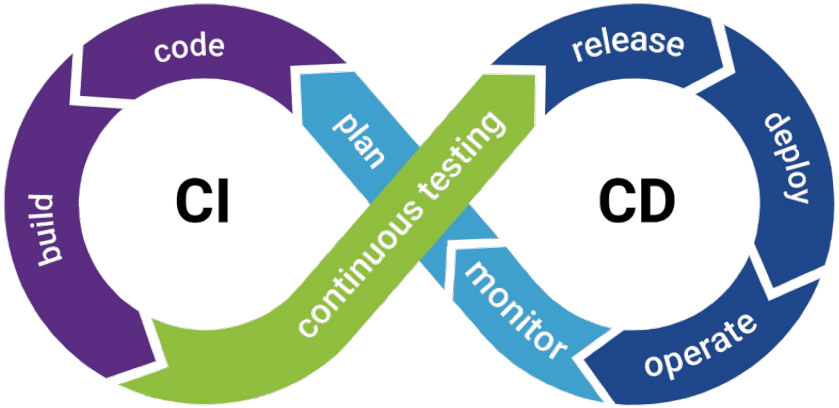
SHIP-HATS

An SGTS Product

What is

SHIP-HATS

An **end-to-end Source Code Management** and **Continuous Integration/Continuous Delivery** (CI/CD) toolchain with security and governance guardrails for developers to build, test, and deploy code to production.



CI/CD pipelines automate incremental code changes to be delivered quickly and reliably to production

GOVTECH SINGAPORE

Who should use

SHIP-HATS

- For all **Agency Systems that are Cloud Confidential (Eligible) and below**
- **Mandatory for GovTech-owned systems**
- Users can be **Public officers or Vendors**
- Subscription **by Agencies**

GOVTECH SINGAPORE

SHIP-HATS 2.0. Product Offering

Tools under-the-hood

Access (by default)

Subscription Mgmt

techbiz

Tools Provisioning

SHIP-HATS Portal

Sign in

techpass

SEED

Planning	Build	Build Testing SAST	Other Tests Code quality scanning, DAST, Container Scanning	Deploy & Release

GitLab-Native

GitLab
The One DevOps Platform

- **GitLab Ultimate Tier Licenses** which comes with features including value stream measures and dashboards
- Offers integrated native tools from planning till deployment
- Dedicated SaaS managed instance hosted in Singapore

Alternative Tools

Confluence

Jira

nexus repository pro

nexus lifecycle

Fortify on Demand

sonarqube

pCloudy.com

Tool Selection

Agencies must assess whether Gitlab native is sufficient (e.g. whether the gitlab-native tool supports the language/framework used etc).

[Tool comparison available on DevPortal]

1 GitLab Native only

- ✓ Lean teams
- ✓ Simple use cases
- ✓ New Projects
- ✓ Value for $$

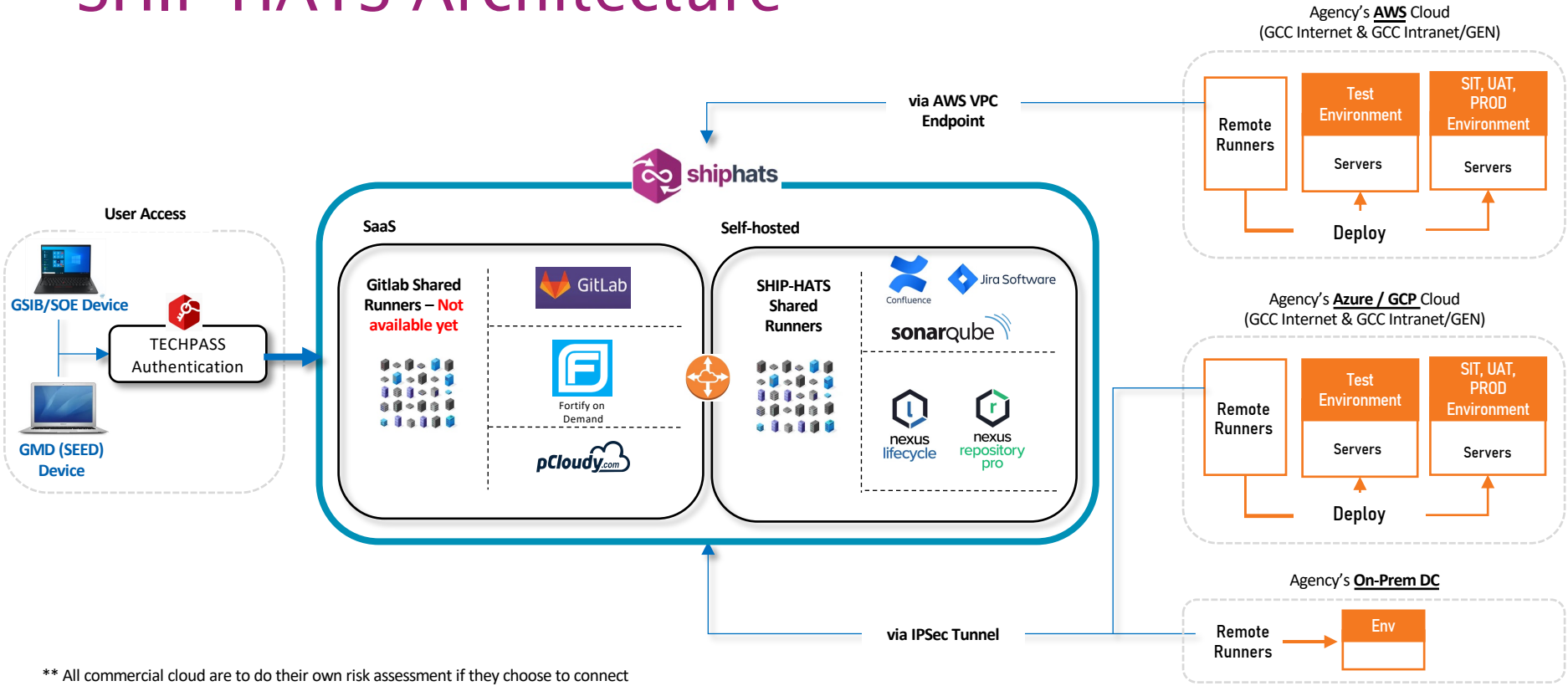
2 GitLab & Alternative tools

- ✓ Already on alternative tools
- ✓ Complex use cases

- ❖ **Both** options **meet Security** Policy needs
- ❖ Alternative tools are available as add-on anytime

GOVTECH SINGAPORE

SHIP-HATS Architecture



** All commercial cloud are to do their own risk assessment if they choose to connect remote runners to SHIP-HATS Gitlab over the internet.

GOVTECH SINGAPORE

Benefits of SHIP-HATS

#SHIPHATSforAgencies

50 Agencies use SHIP-HATS

300+ Agency Systems

Over 3K Active User

50+ Pipeline Templates offered

MINISTRY OF MANPOWER
2019 *4-year active user*

HTX
2019 *4-year active user*

Health Promotion Board
2020 *3-year active user*

Ministry of Education
SINGAPORE
2021 *2-year active user*

Why choose SHIP-HATS?

1

Faster time to market

- ✓ **In-built compliance** and best practices through centralized
- ✓ **Templates** and **tools** for all users
- ✓ **No procurement** overheads to get started

2

Security for Public Sector Needs

- ✓ CI/CD **platform complies to security** needs
- ✓ **SaaS benefits for Confi-Cloud (Eligible) Systems** as dedicated instance is hosted in Singapore

3

WoG Collaboration

- ✓ **Common repo** of images for reuse
- ✓ Contribute to templates
- ✓ Learn faster and replicate learnings across WOG

4

Metrics & Data for DevSecOps

- ✓ **GitLab Dashboards**: Security, DORA metrics, DevOps Adoption, Value Stream Analytics
- ✓ Greater transparency and reduces system reporting overheads

Value-Added Features

SHIP-HATS 2.0. Value-Added Features

As a subscriber, agencies will also have access to the following:

- Platform Enablement Features
- CI/CD Tools and Features
- Security Baseline & Testing Tools
- Metrics and Value Stream Measurements (VSM)

Platform-Enablement

e.g. Runners

Option 1: SHIP-HATS Shared Runners

- Hosted by SHIP-HATS team
- Created on-demand
- Available for all SHIP-HATS users at no additional costs
- No overheads for Agencies to maintain runners.
- 4 variants: **CStack, Docker, Windows** and GitLab Shared Runners

Option 2: Self-hosted Remote Runners

- Hosted by the agency
- Agencies to bear the costs of hosting their own runners
- Can be configured for Group or Project level access
- Full-control of the runners

GOVTECH SINGAPORE

CI/CD Tools & Features

e.g.
Pipeline Templates

1	**5 End-to-End Pipeline Templates** Pipeline skeletons
2	**40~Modular Templates** can be include as per need
3	**Reference Pipelines** fully implemented pipeline samples
4	**Pipeline COE** Reusable container images

GOVTECH SINGAPORE

1

5 End-to-End Pipeline Templates

Pipeline skeletons

Your project pipeline plan

End to End Pipeline Template

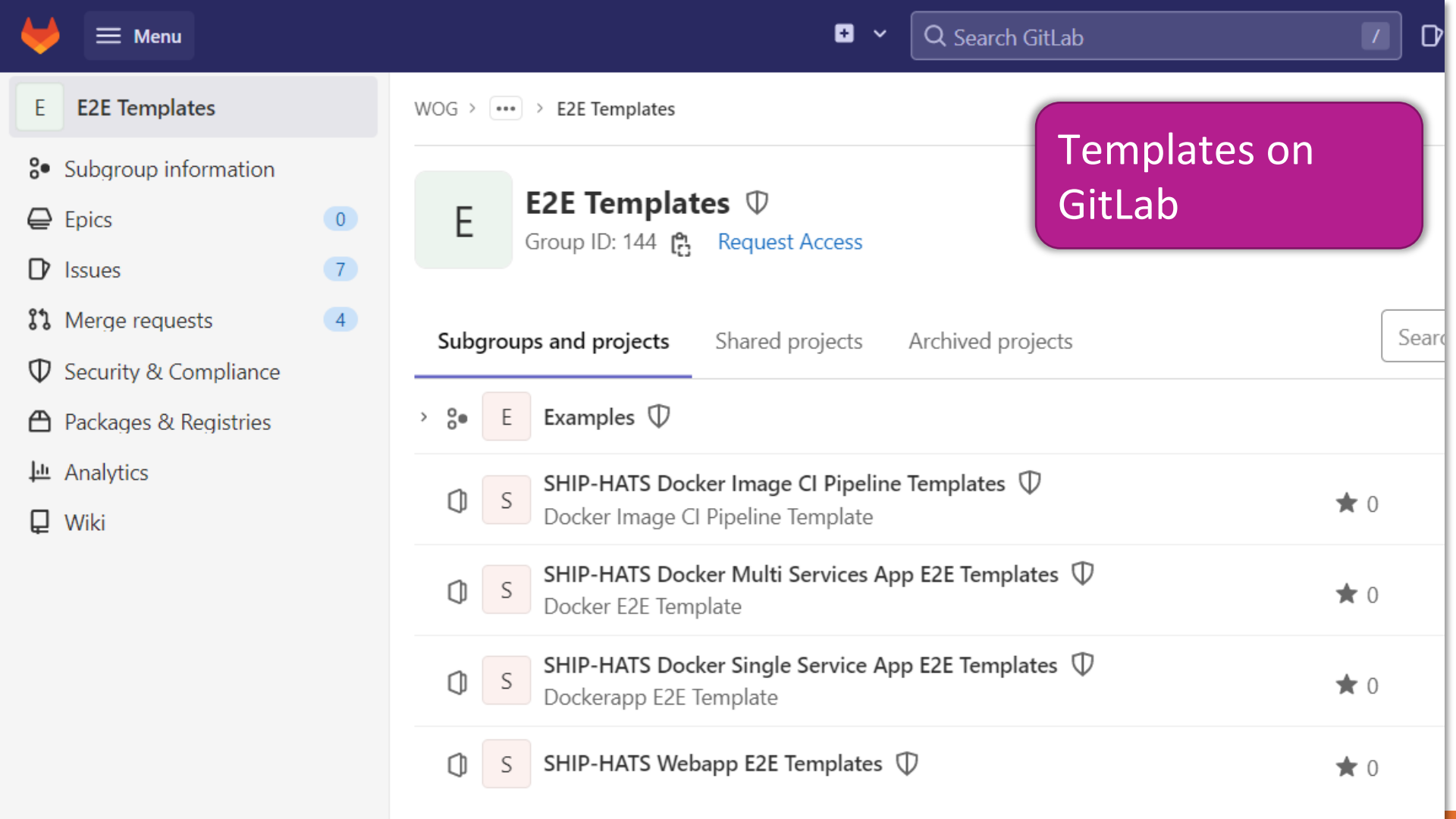
Your project specific variables

Copy

End to End Pipeline Templates
from GovTech

Example Use Case:
.NET applications hosted in Azure App Service

GOVTECH SINGAPORE


Menu
Search GitLab
E2E Templates
Subgroup information
Epics 0
Issues 7
Merge requests 4
Security & Compliance
Packages & Registries
Analytics
Wiki
WOG
E2E Templates
E
E2E Templates
Group ID: 144
Request Access
Templates on GitLab
Subgroups and projects
Shared projects
Archived projects
Examples
SHIP-HATS Docker Image CI Pipeline Templates
Docker Image CI Pipeline Template
SHIP-HATS Docker Multi Services App E2E Templates
Docker E2E Template
SHIP-HATS Docker Single Service App E2E Templates
Dockerapp E2E Template
SHIP-HATS Webapp E2E Templates

Yaml for E2E

.gitlab-ci.yml 6.05 KiB — Open in Web IDE | Lock | Replace | Delete

```yaml
include:
  - local: BUILD.gitlab-ci.yml
  - local: TEST.gitlab-ci.yml
  - local: DEPLOY.gitlab-ci.yml
  - project: "WOG/GVT/ship/ship-hats-templates"
    ref: "main"
    file:
      - /templates/vars/.gitlab-ci-sonatype-vars.yml
      - /templates/.gitlab-ci-publish-to-nexus.yml
      - /templates/.gitlab-ci-check-app-readiness.yml
      - /templates/.gitlab-ci-run-test.yml
      - /templates/.gitlab-ci-create-report.yml

stages:
  - build
  - static-test
  - deploy-to-testing-env
  - runtime-test
  - sign
  - publish
  - deploy-to-prod #to PROD

image: $NEXUSREPO_DOCKER_PROXY_HOSTNAME/alpine:latest

variables:
  # app-specific variables
  WORKING_DIR: "" # app for eg. relative path to application from project
  OUTPUT_ARTEFACT: "" # "$WORKING_DIR/app.zip" for eg. compiled app to be scanned and uploaded
  UNIT_TEST_REPORT: "" # "$WORKING_DIR" for eg. relative path to run unit testing
  WEBAPP_URL: "" # URL to reach project web application
  WEBAPP_NAME: "" # web application name that can be searched when service is up
  VERSION: "" # version of web application to release to artefact repository.

  # for e2e integration testing if robot framework is used
  RF_TESTSCRIPT_FOLDER: "" # "$WORKING_DIR/test-automation/E2eTest/robotframework_testscripts"

  # for dast (pen) testing in compliance framework
  DAST_WEBSITE: $WEBAPP_URL # target URL to perform dast for web app in testing env
  DAST_XML_REPORT: "dast-owasp-zap-report.xml" # DAST report that's viewable from your build's

  # for sonarqube scan in compliance framework
  # check out with SHIP-HATS team for account provisioning
  # see README.md#provide-sensitive-variables for list of variables required to include ship-sc

  # for nexus iq scan in compliance framework
  # check out with SHIP-HATS team for account provisioning
  # see README.md#provide-sensitive-variables for list of variables required to include ship-ne
  SCAN_TARGETS: $OUTPUT_ARTEFACT # target to perform scan on

  # for fod sast and fod dast in compliance framework
  # check out with SHIP-HATS team for account provisioning
  # see README.md#provide-sensitive-variables for list of variables required to include ship-fortify-sast-fod-with-report and ship-fortify-dast

  # for deploy-to-prod stage to know the source of artefact
  DEPLOY_ARTEFACT: prod/$ARTEFACT_ID.$ARTEFACT_PACKAGE

build-job: # build. define in BUILD.gitlab-ci.yml file
  stage: build

lint-job: # perform a lint check. define in TEST.gitlab-ci.yml file
  stage: static-test

unit-test-job: # perform a unit testing. define in TEST.gitlab-ci.yml file
  stage: static-test

depcheck-job: # perform a dependency check to check for unused dependencies. define in TEST.gitlab-ci.yml file
  stage: static-test

deploy-testing-job: # deploy to a testing env from DEPLOY.gitlab-ci.yml file
  stage: deploy-to-testing-env

wait-for-webapp-ready-job: # wait for the app service to be ready, using a template from WO
  extends: .wait-for-app-and-assert-text
  needs: ["deploy-testing-job"]
  stage: deploy-to-testing-env
  environment:
    name: testing
    action: prepare
  variables:
    URL: "$WEBAPP_URL"
    TEXT: "$WEBAPP_NAME"

e2e-test-job: # run e2e UI test using robot framework, you may edit to use any e2e testing
  stage: runtime-test
  variables:
    SERVICE: selenium/standalone-chrome:4
    ROBOT_IMAGE_VERSION: latest
    OPTS: -v BROWSER:headlesschrome
    OUTPUT_DIR: $RF_TESTSCRIPT_FOLDER/reports
    OUTPUT_XML: report.xml
    LOG_NAME: log
    REPORT_NAME: report
    TEST_TARGET: $RF_TESTSCRIPT_FOLDER
  extends: .run-robot-test
  allow_failure: true
```

BUILD.gitlab-ci.yml 459 bytes — Open in Web IDE | Lock

```yaml
include:
  - project: "WOG/GVT/ship/ship-hats-templates"
    ref: "main"
    file:
      - /templates/vars/.gitlab-ci-sonatype-vars.yml

image: $NEXUSREPO_DOCKER_PROXY_HOSTNAME/alpine:latest

.build-webapp:
  script:
    - echo "implement building webapp here."

build-job: # do not change the name of this job, required by compliance
  extends: .build-webapp
  # artifacts:
  #   when: always
  #   paths:
  #     - $OUTPUT_ARTEFACT # required by compliance
```


GOVTECH SINGAPORE

2

40~Modular Templates
can be include as per need

Your project pipeline plan

- Your project specific scripts
- Modular Testing Template
- Modular Repo Template
- Your project specific variables

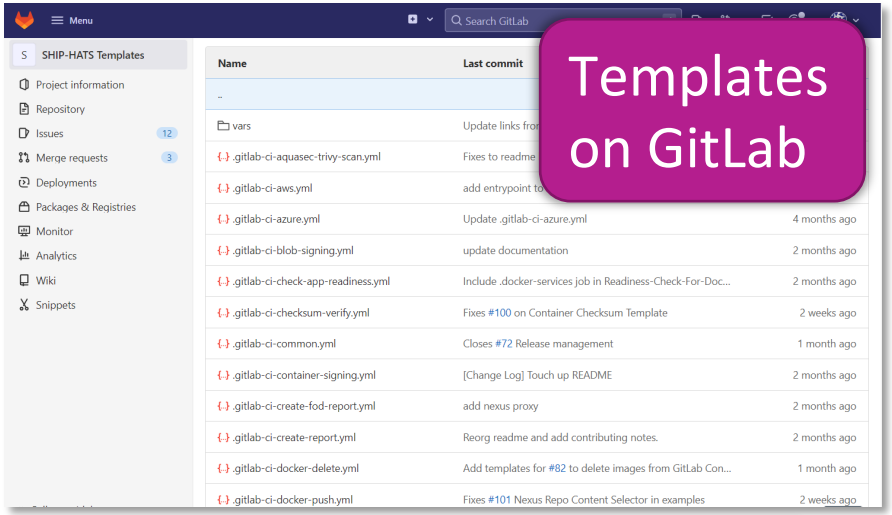
Include

Modular Templates from GovTech

- Build & Release Templates
- QA & Security Templates
- Artifact Repository Templates

GOVTECH SINGAPORE

Modular Pipeline Templates



Templates on GitLab

BUILD.gitlab-ci.yml 459 bytes — Open in Web IDE | Lock

```yaml
include:
  - project: "WOG/GVT/ship/ship-hats-templates"
    ref: "main"
    file:
      - /templates/vars/.gitlab-ci-sonatype-vars.yml

image: $NEXUSREPO_DOCKER_PROXY_HOSTNAME/alpine:latest

.build-webapp:
  script:
    - echo "implement building webapp here."
```

.gitlab-ci-publish-to-nexus.yml 3.31 KiB — Open in Web IDE | Lock | Replace | Delete

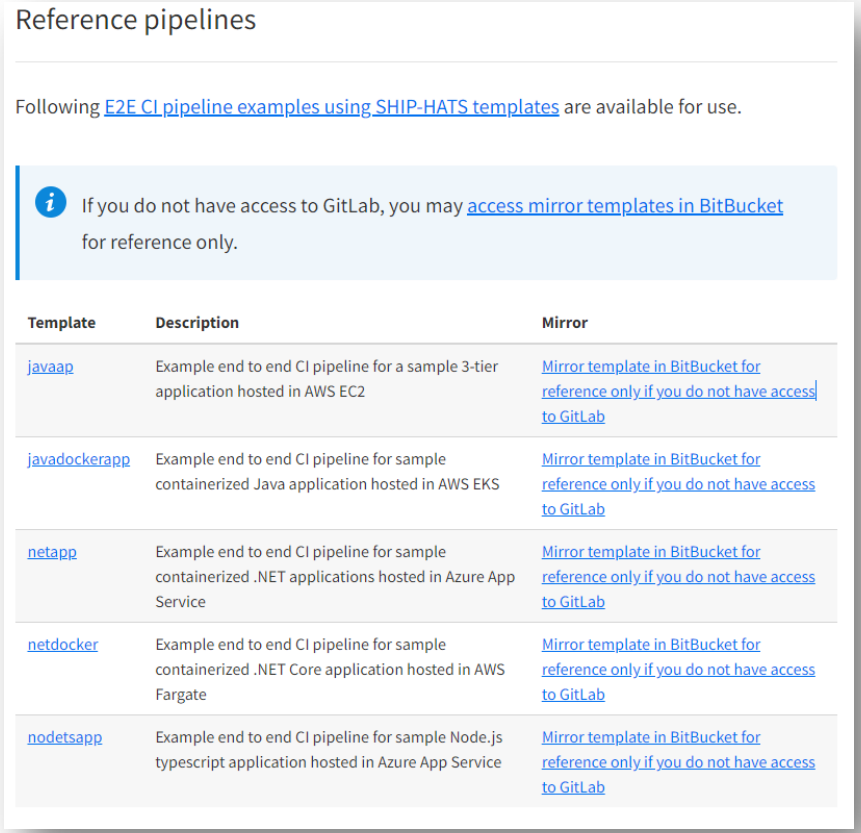
```yaml
include:
  - local: "/templates/vars/.gitlab-ci-sonatype-vars.yml"
  - local: "/templates/.gitlab-ci-nexus-configure.yml"
  - local: "/templates/.gitlab-ci-common.yml"

# Include these variables in your .gitlab-ci.yml
# i.e.
# variables:
#   MAVEN_SETTINGS_SERVER_ID: "test-demo-releases"
#   NEXUSREPO_REPO_ID: "test"
#   NEXUSREPO_REPO_GROUP_ID: "com.gt.shiphats.demo"
#   MVN_SETTINGS_FILE: "./settings.xml"
#   ARTEFACT: "./target/demo-0.0.1.jar"
#   ARTEFACT_ID: "demo"
#   ARTEFACT_VERSION: "0.0.1"
#   ARTEFACT_PACKAGE: "jar"

.publish-maven-artefact:
  image: $NEXUSREPO_DOCKER_PROXY_HOSTNAME/maven:latest
  tags:
    - non_privileged
    - no_root
    - cstack
  script:
    - export VARIABLE_NAME=MVN_SETTINGS_FILE
    - !reference [.check-variable-sh, script]
    - export VARIABLE_NAME=NEXUSREPO_REPO_GROUP_ID
    - !reference [.check-variable-sh, script]
    - export VARIABLE_NAME=ARTEFACT_ID
    - !reference [.check-variable-sh, script]
    - export VARIABLE_NAME=ARTEFACT_VERSION
    - !reference [.check-variable-sh, script]
    - export VARIABLE_NAME=ARTEFACT_PACKAGE
```

Yaml for modular

GOVTECH SINGAPORE

3

Reference Pipelines
fully implemented pipeline samples



Reference pipelines

Following E2E CI pipeline examples using SHIP-HATS templates are available for use.

If you do not have access to GitLab, you may access mirror templates in BitBucket for reference only.

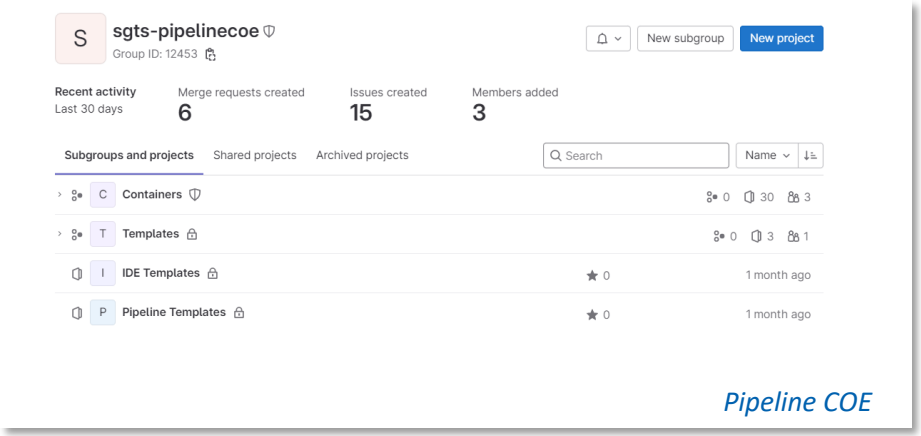
Template	Description	Mirror
javaap	Example end to end CI pipeline for a sample 3-tier application hosted in AWS EC2	Mirror template in BitBucket for reference only if you do not have access to GitLab
javadockerapp	Example end to end CI pipeline for sample containerized Java application hosted in AWS EKS	Mirror template in BitBucket for reference only if you do not have access to GitLab
netapp	Example end to end CI pipeline for sample containerized .NET applications hosted in Azure App Service	Mirror template in BitBucket for reference only if you do not have access to GitLab
netdocker	Example end to end CI pipeline for sample containerized .NET Core application hosted in AWS Fargate	Mirror template in BitBucket for reference only if you do not have access to GitLab
nodetsapp	Example end to end CI pipeline for sample Node.js typescript application hosted in Azure App Service	Mirror template in BitBucket for reference only if you do not have access to GitLab

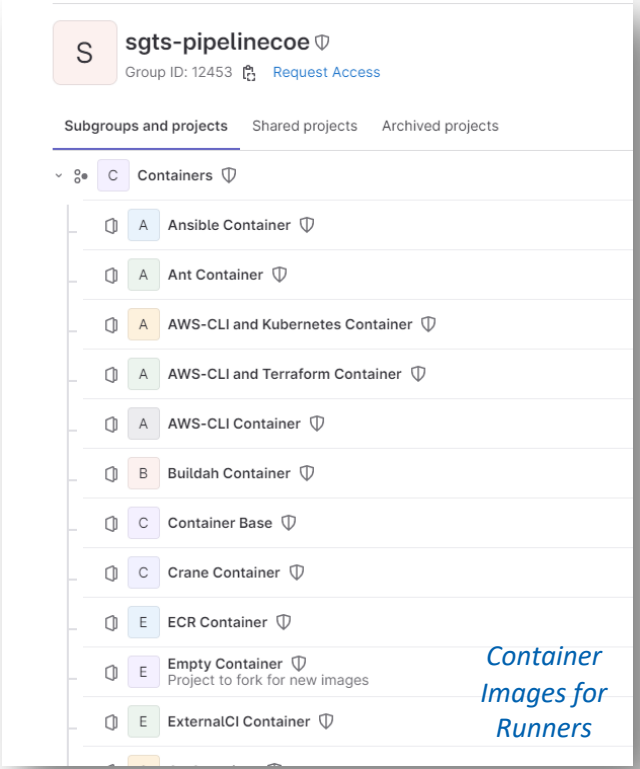
Developed by CTMO, SVC Group, GovTech

GOVTECH SINGAPORE

4 Pipeline COE
aka DevSecOps Governance Framework (DGF) (new branding)
Reusable code & images

- Feature provided by GitLab & maintained by the SHIP-HATS Team with a base set of **reusable code & images**
- Part of inner-sourcing efforts
- The SHIP-HATS team is **continually developing new reusable assets**
- Agencies can raise a Service Ticket to **suggest new assets**





GOVTECH SINGAPORE

Security Baseline & Testing Tools

Security Features

Benefits of Automated Security Testing:

- Tests are completed faster
- Consistency of running the same tests across incremental builds
- Repeatability allows the task to be build into a larger automated process
- Auditability of tests through logging and dashboards

To streamline and automate manual testing efforts, the following features were introduced

1	**Security with SHIP-HATS** Shifting left and codifying security
2	**Compliance Framework** Automates setting up controls
3	**Extended Code Analysis** Testing feature from CSG, available by default

GOVTECH SINGAPORE

Centralising automated security testing via SHIP-HATS

Impetus

- Lack of basic security capabilities across government
- Similar environmental set-up and needs
- Need for speedy remediation

Benefits

- Centralise security experts
- In-built compliance and best practices through centralized tools e.g. SHIP-HATS is preconfigured for OWASP Top 10 (an industry standard)
- Learn faster and replicate learnings across WOG*

What if we don't centralise?

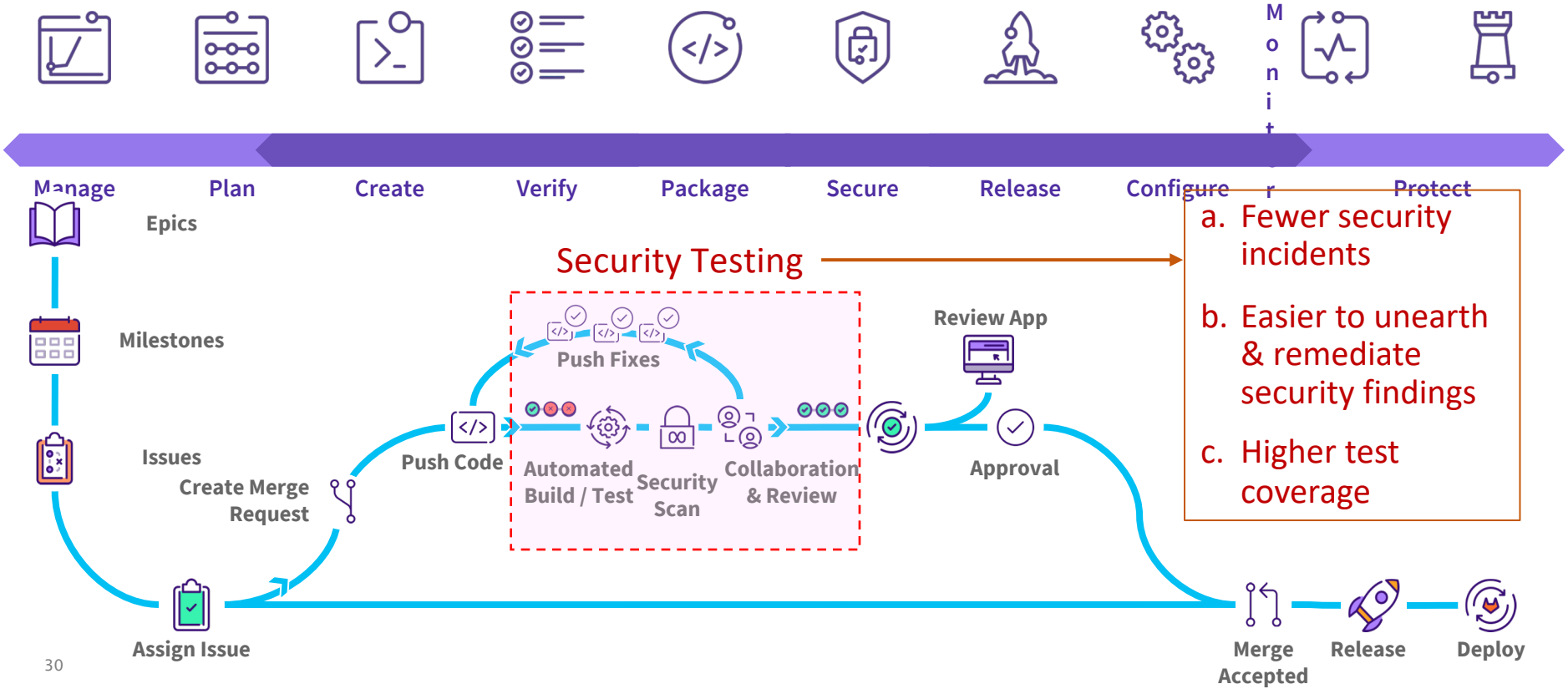
- Agencies will be left to configure tools on their own
- Longer onboarding/adoption times
- Higher manpower costs, esp. for agencies with lower DevSecOps maturity

*Apart from building-in the learnings via tools, best practices are also published in the DevOps Playbook (https://docs.developer.tech.gov.sg/docs/devsecops-playbook)

1

Security with SHIP-HATS
Shifting left and codifying security

Security testing is **incorporated in the CI/CD pipeline** and is **automated** (i.e. run every time the app is built)



1

Security with SHIP-HATS

Shifting left and codifying security

(in collaboration with CSG)

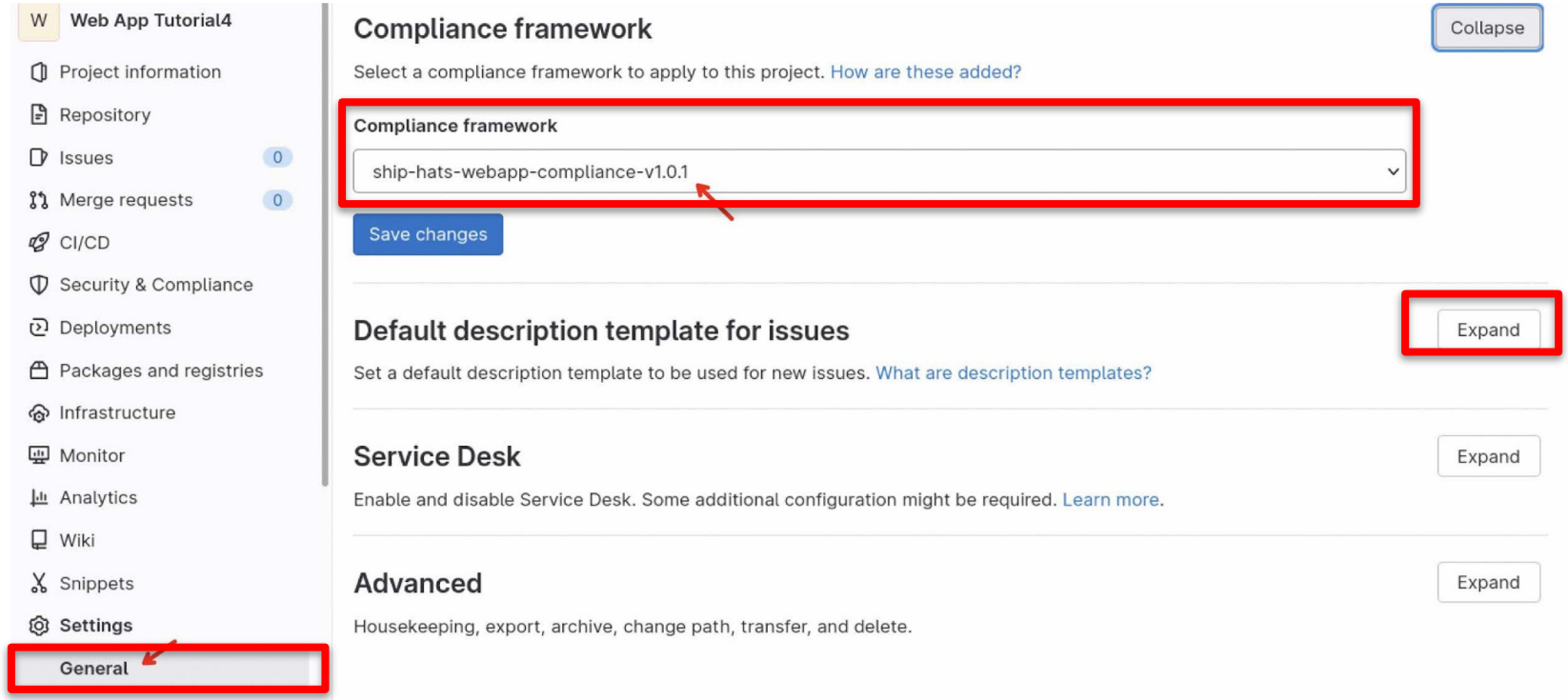
Design/Plan	Develop	Build	Test	Deploy	Operate & Monitor
Capabilities on SHIP-HATS					
Security Architecture (ISC)² Training via CAPE	Secure Development Secure Code Warrior (SCW)	Secure build management GitLab Build Runners	Performance test & monitoring pCloudy Test Farm	Approval for release to prod. GitLab Continuous Deployment	Continuous monitoring/logging AWS CloudWatch Logs
Threat modelling (ISC)² Training via CAPE	Code repo security controls GitLab Repo BitBucket (SHIP1.0)	Secrets Detection GitLab Security Analysers Security for Bitbucket (SHIP1.0)	Dynamic App Security Testing (DAST) GitLab DAST Fortify-on-Demand DAST	Artefact repository scanning GitLab Continuous Deployment	Continuous feedback & metrics AWS CloudWatch
	Code reviews GitLab Merge Requests	3rd Party component scanning GitLab Dependency Scanning Nexus Intelligence / IQ Server	Manual Pen Test CSAS Bulk Tender	Continuous vuln scan (Host) AWS Systems Patch Manager AWS Inspector	Web Application Firewall AWS Web Application Firewall
	Source code scanning (SAST) GitLab Security Analysers Fortify-on-Demand SAST	Container scanning GitLab Container Scanning Prisma Cloud (SHIP1.0)		Secrets management AWS Secrets Manager	
New Capabilities					
	Developer endpoint security controls Developers' Environment Endpoint Posture (DEEP)	Mobile app scanning Mobile App Security Hygiene (MASH)	Repeated Security Findings Extended Code Analysis (XCA)		Cloud security controls CloudSCAPE
	Secure-by-Default Designs Secure Infrastructure-as-Code (SIaC) Anyhow Code Also Secure ^				
Threat modelling (+) Integrative Threat Modelling Platform	Supply-chain Security Supply-chain Levels for Software Artifacts (SLSA)				Cloud architecture controls Cloud Architecture Compliance Tooling (CACTi)
					Container Threat Detection Container Anomaly Detection

^ Denotes Experimental

2

Compliance Framework
Automates setting up controls

- ✓ Use GitLab feature to automate adopting DevSecOps best practices and security needs
- ✓ Teams can apply the framework from UI and confident that their controls and requirements are set up correctly

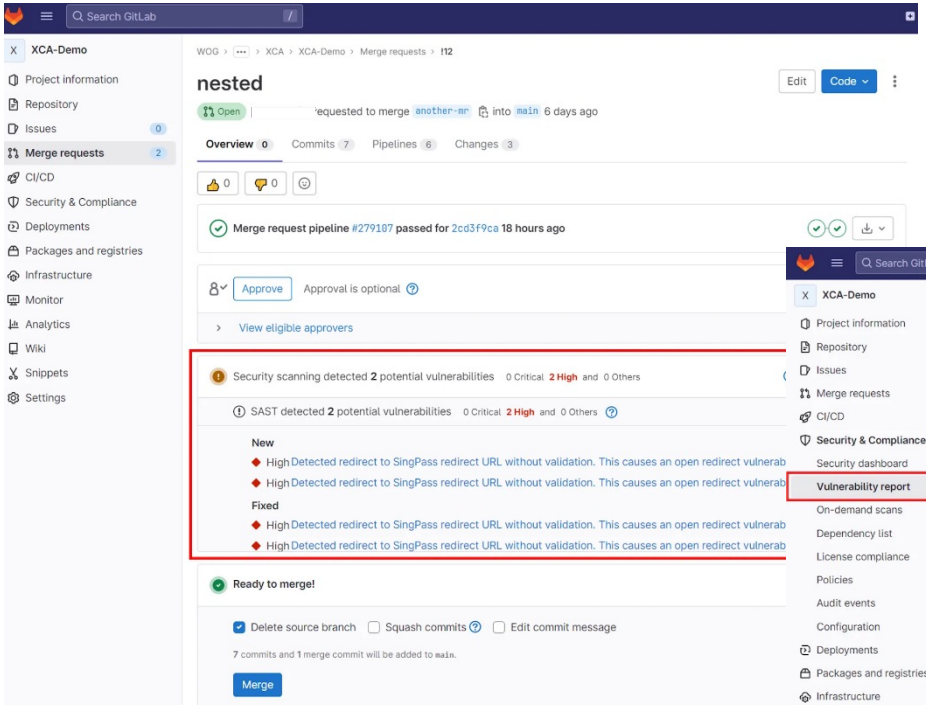


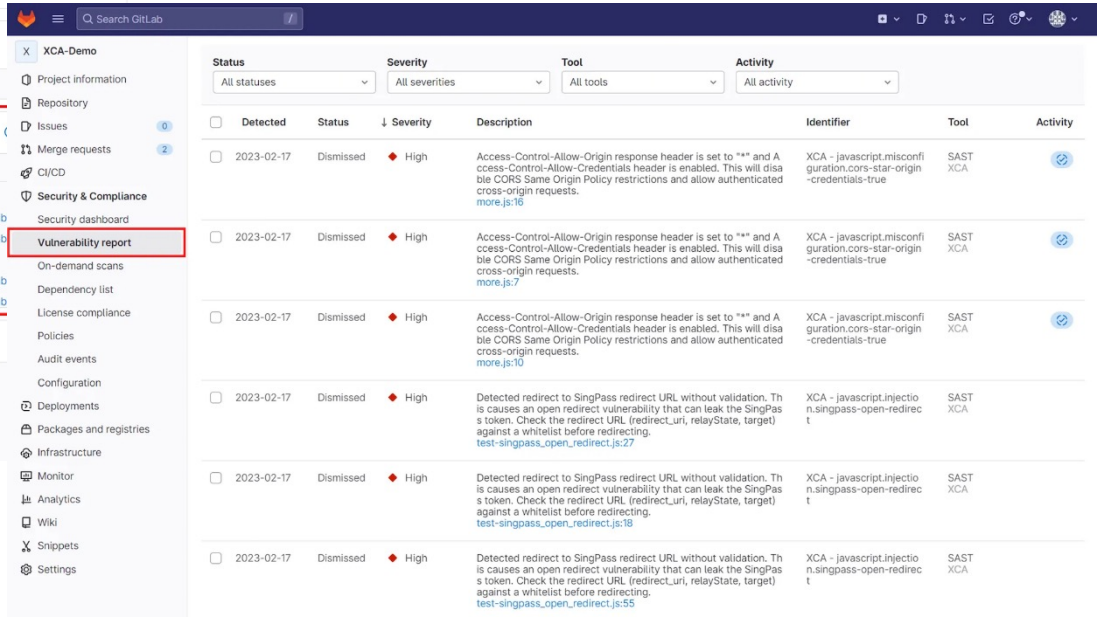
GOVTECH SINGAPORE

3

Extended Code Analysis

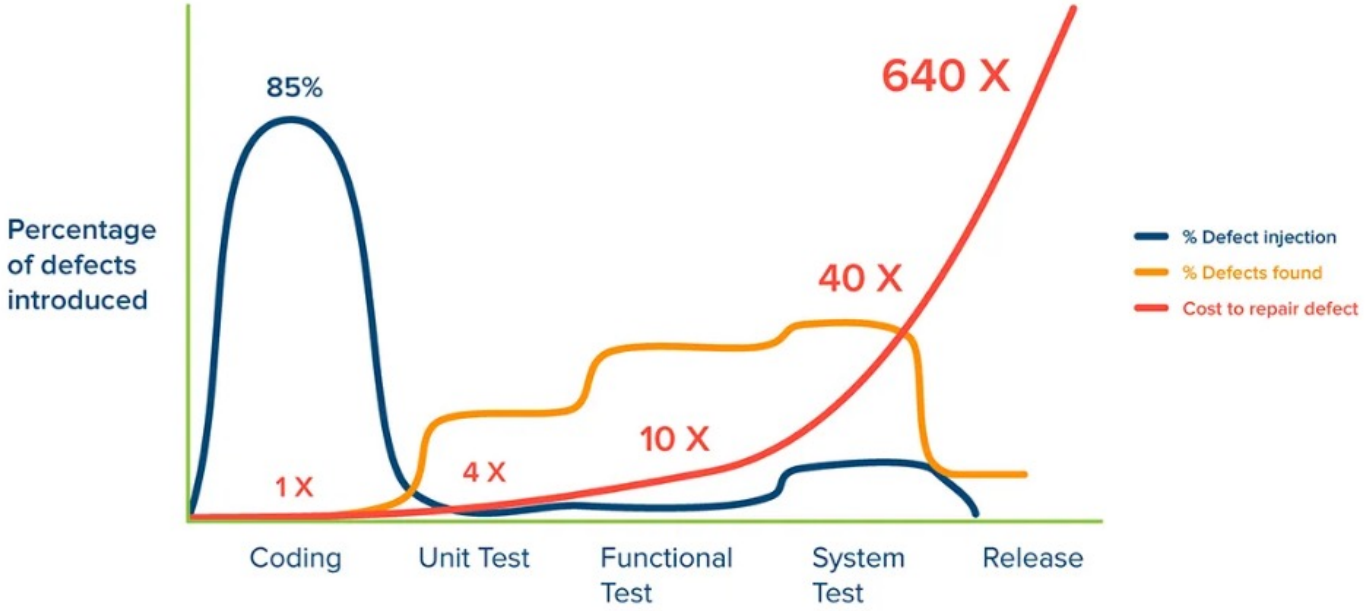
Testing feature from CSG, available by default





Apart from platforms, tools and features, we need to drive security best practices

Regular and more upstream security testing enables 'shift-left', where vulnerabilities are found and remediated earlier in the dev cycle resulting in fewer defects and reduced time/costs for rectification by 10x



Jones, Capers. *Applied Software Measurement: Global Analysis of Productivity and Quality.*

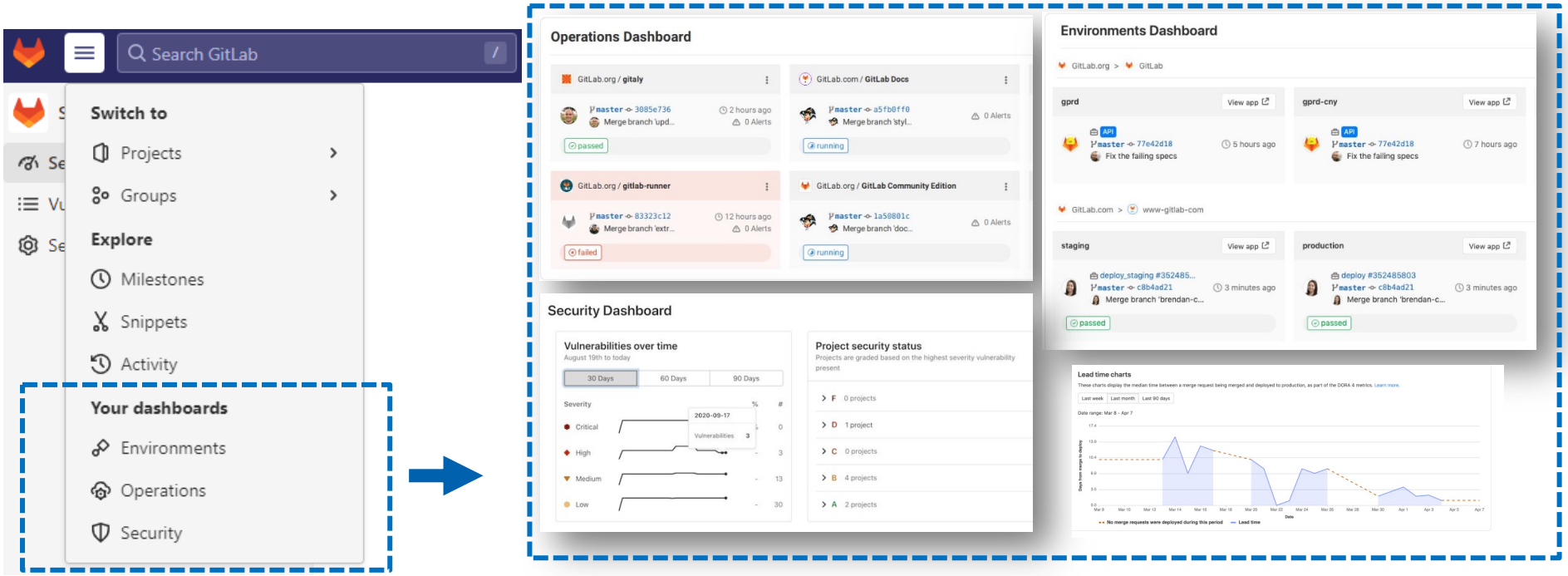
Metrics & Value Stream Metrics (VSM)

For Dev Productivity and DevSecOps Maturity

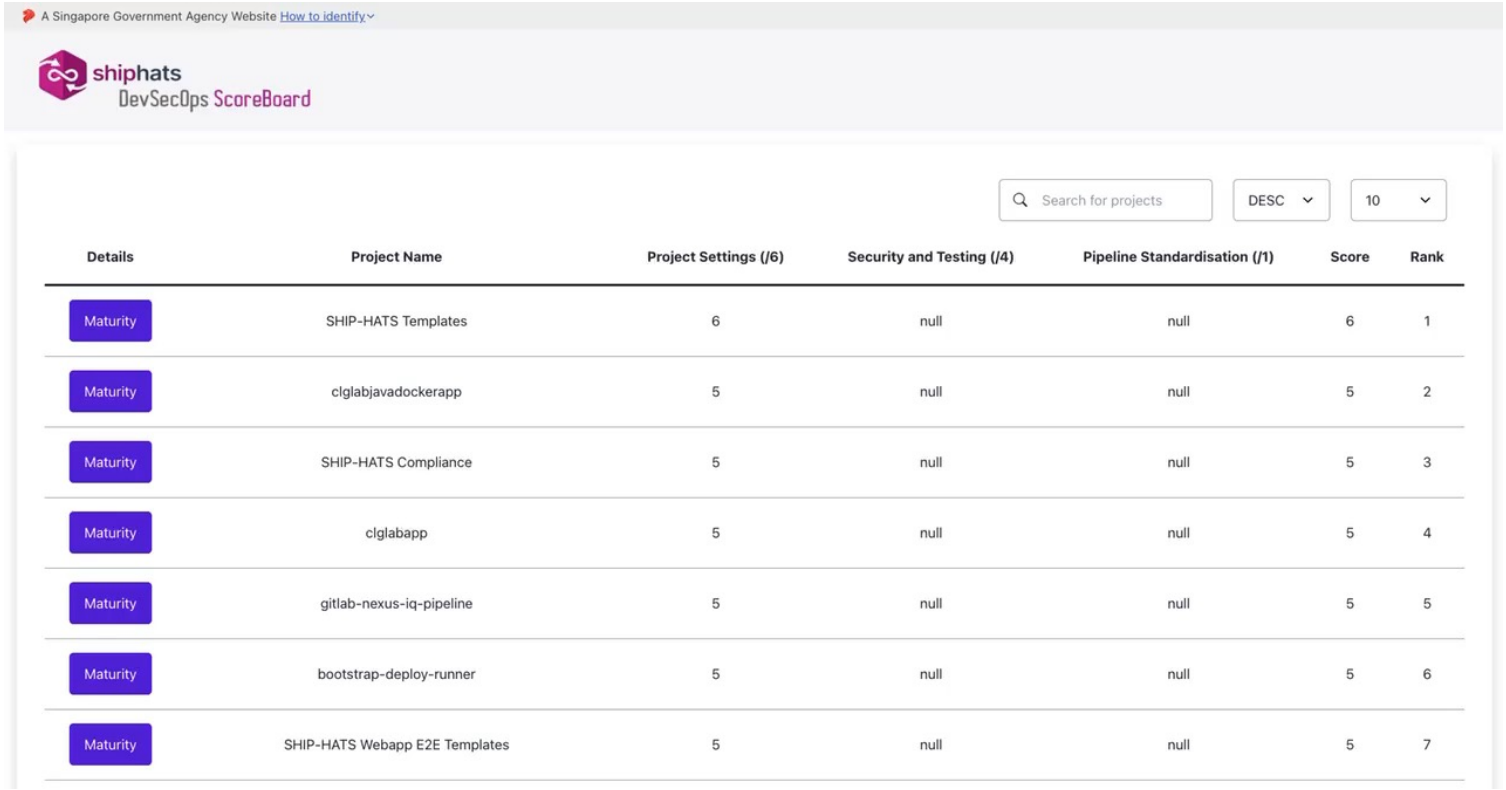
1	**GitLab-Native Dashboards** e.g. Security, Ops etc.
2	**DevSecOps Scoreboard (WIP!)**

GOVTECH SINGAPORE

1 **GitLab-Native Dashboards** e.g. Security, Ops etc.



2 DevSecOps Scoreboard (WIP!)



Details	Project Name	Project Settings (/6)	Security and Testing (/4)	Pipeline Standardisation (/1)	Score	Rank
Maturity	SHIP-HATS Templates	6	null	null	6	1
Maturity	clglabjavadockerapp	5	null	null	5	2
Maturity	SHIP-HATS Compliance	5	null	null	5	3
Maturity	clglabapp	5	null	null	5	4
Maturity	gitlab-nexus-iq-pipeline	5	null	null	5	5
Maturity	bootstrap-deploy-runner	5	null	null	5	6
Maturity	SHIP-HATS Webapp E2E Templates	5	null	null	5	7

Check DevPortal for Updates

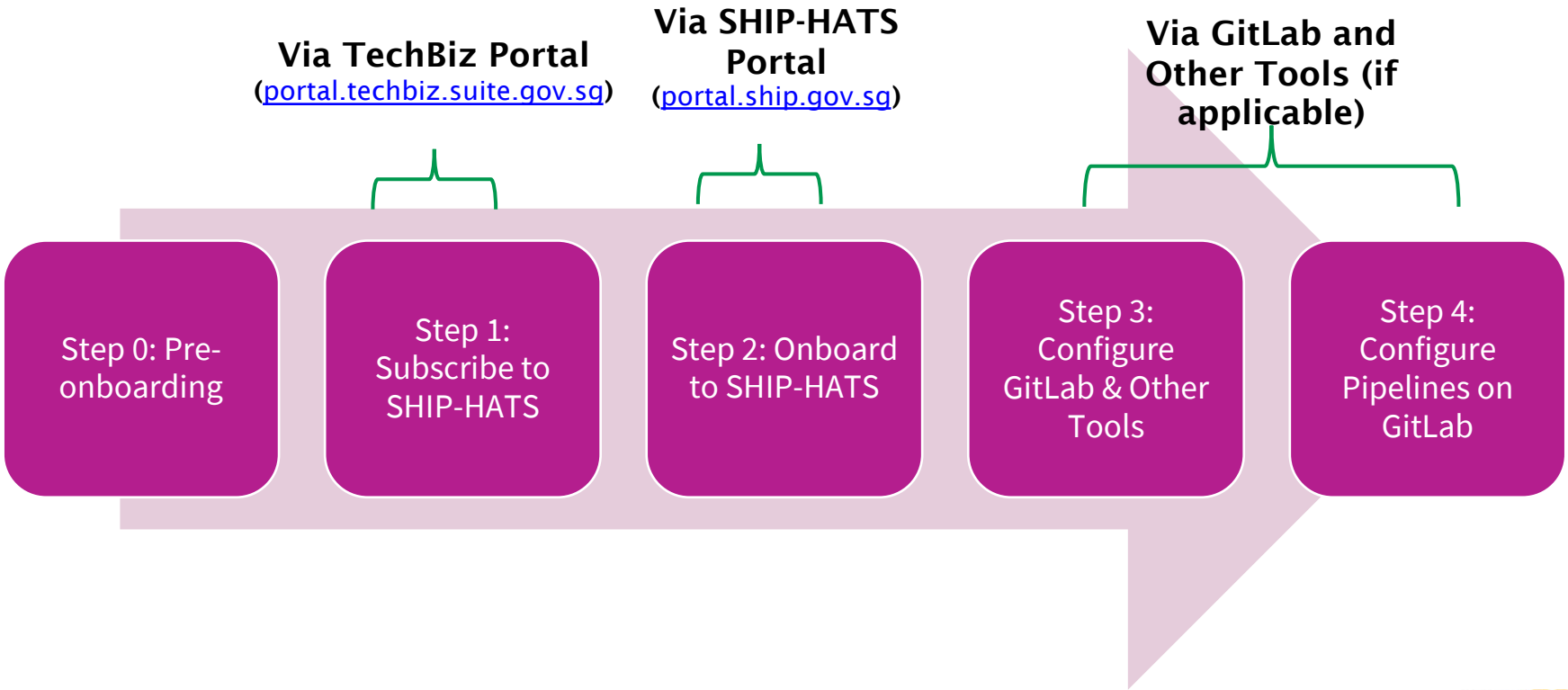
SHIP-HATS 2.0. Product Roadmap

Component	Q1	Q2	Q3	Q4
Platform Enablement	• Shared runners for Intranet deployment	• Runners to support GitLab-Native services • Jira/Confluence Cloud Pilot (JSM & Multi-tenancy)		• Jira/Confluence Cloud GA (JSM/Multi-tenancy)
CI/CD Tools and Features	• Low-Cost Flow (GitLab-Native) Pipeline Template • GitOps-based Templates	• GitLab Guest-Account Enablement with TechPass	• Limiting Feature for GitLab Guest Accounts	• Support for GCC + • BCP
Security Baseline & Testing Tools	• SemGrep (bg scanning) • DevSecOps Security Baseline – identification of baselines and and draft measures	• DevSecOps Security Baseline – roll-out to GDS	• **SLSA** • DevSecOps Security Baseline – roll-out to GovTech and WOG	• DevSecOps Security Baseline – intermediate baselines
Metrics and VSM	• DevSecOps Scoreboard POC – checking for compliance with IM8,	• DevSecOps Scoreboard Beta	• DevSecOps Scoreboard GA – Phase 1	• DevSecOps Scoreboard GA – Phase 2

Getting Started

***Onboarding Documentation** available on Developer Portal*

Subscribing and Onboarding to SHIP-HATS



GOVTECH SINGAPORE

Subscription Options

[Subscription documentation available on Developer Portal]

Pricing details on go.govsg/sh2indicative for Public Officers

Base Plan

- ✓ 3 GitLab Ultimate User License
- ✓ Shared pCloudy
- ✓ Standard Support 9.30am to 5:30pm
- ✓ up to 50 GitLab licenses per bundle

Notes

- **Start with Base Option**
- **Add-ons to maximise flexibility**

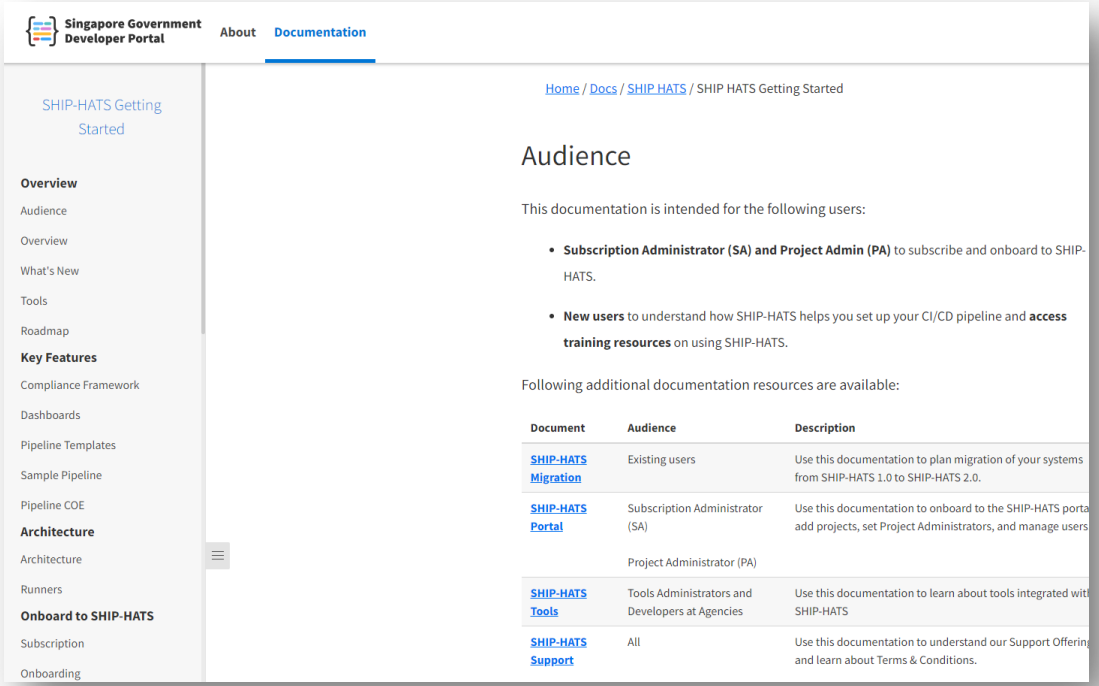
Flexible Add-ons

Component	What's Included	Notes
GitLab Ultimate	1 x GitLab Ultimate User License	Based on the number of users
Jira/Confluence	1 x Jira/Confluence License	Based on the number of users
Sonatype Nexus IQ & Nexus Repo	1 x Nexus IQ 1 x Nexus Repo User License	Based on the number of users
pCloudy Testing Farm	1 x pCloudy ios Device (dedicated) 1 x pCloudy Android Device (dedicated)	1 device can be used concurrently across multiple GitLab Projects, and tests can be run in parallel
SonarQube	1 x SonarQube Community Edition App 1 x SonarQube Dev Edition App	1 app can be used for 1 subscription (i.e. shared across GitLab projects) 1 app can be used for 1 subscription (i.e. shared across GitLab projects)
Fortify-on-Demand	1 x FOD App	1 app can be shared across multiple GitLab Projects, but tests cannot be run in parallel.
Additional support	Per hour (after 5:30 PM)	Outside office hours

Self-learn with

Technical Documentation

Everything on the Internet – Search on Google
Or
Visit https://go.gov.sg/ship-hats-docs

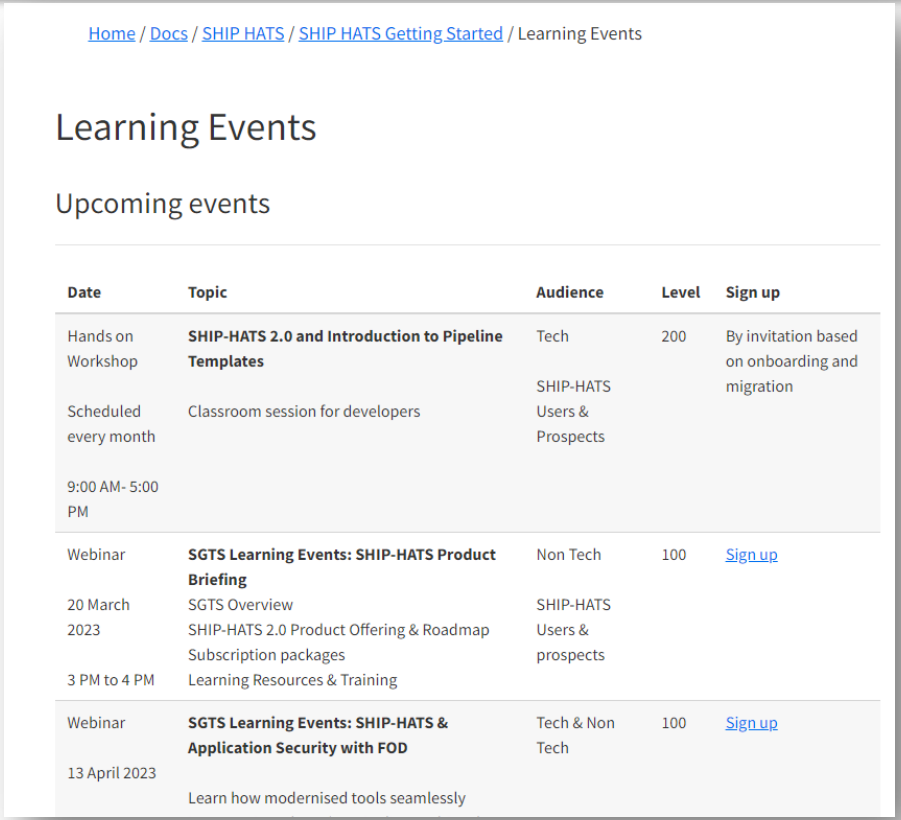


GOVTECH SINGAPORE

Attend with

Workshops & Webinars

https://go.gov.sg/ship-hats-learning-events



Home / Docs / SHIP HATS / SHIP HATS Getting Started / Learning Events

Learning Events

Upcoming events

Date	Topic	Audience	Level	Sign up
Hands on Workshop Scheduled every month 9:00 AM- 5:00 PM	**SHIP-HATS 2.0 and Introduction to Pipeline Templates** Classroom session for developers	Tech SHIP-HATS Users & Prospects	200	By invitation based on onboarding and migration
Webinar 20 March 2023 3 PM to 4 PM	**SGTS Learning Events: SHIP-HATS Product Briefing** SGTS Overview SHIP-HATS 2.0 Product Offering & Roadmap Subscription packages Learning Resources & Training	Non Tech SHIP-HATS Users & prospects	100	Sign up
Webinar 13 April 2023	**SGTS Learning Events: SHIP-HATS & Application Security with FOD** Learn how modernised tools seamlessly	Tech & Non Tech	100	Sign up

GOVTECH SINGAPORE

SHIP-HATS Subscription

https://go.gov.sg/sh2indicative

Accessible via GSIB

- Materials for Subscription
- Indicative Pricing, Pricing Calculator
- Pricing Simulations
- Subscription how-to videos

GOVTECH SINGAPORE

Next Steps

- Reach out to us at https://go.gov.sg/she

- Technical documentation https://go.gov.sg/ship-hats-docs

GOVTECH SINGAPORE

SGTS Products

- SGTS Products
  Explore the products
- SGTS Product Documentation
  Read the docs

GOVTECH SINGAPORE

SGTS Learning Series

--- Online Webinar ---

Learning Events available on Developer Portal

SHIP-HATS 2.0 Product Briefing Monday, 20 Mar 2023 3 to 4 PM

Get to know the **CI/CD tool** in Singapore Government Tech Stack including roadmap, features and resources.
Level 100 Tech & Non-Tech

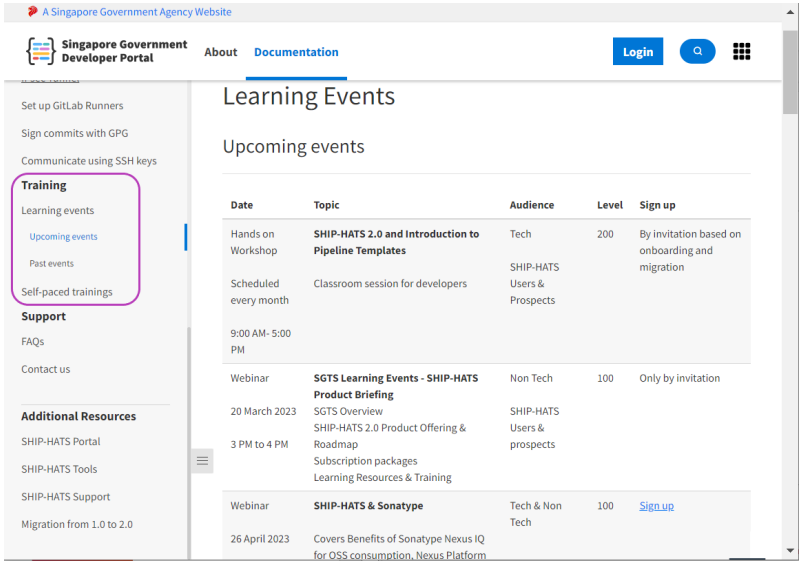
SHIP-HATS & Application Security with FOD

Thursday 13 Apr 2023 4 to 5 PM
Learn how modernised tools seamlessly integrate into SHIP-HATS.
Level 100 Tech & Non-Tech

SHIP-HATS & Sonatype Wednesday 26 Apr 2023 4 to 5 PM

Learn the best practices of using Sonatype in SHIP-HATS.
Level 100 Tech & Non-Tech

Join us and remember to bring your questions for live Q&A.



Thank You