GOVTECH
SINGAPORE

# SHIP-HATS 2.0. Configuring SHIP-HATS 101 (Part 1) - GitLab


GovTech Services Group CTMO

17th May 2023



GOVTECH
SINGAPORE

# Agenda

- Quick Recap of SHIP-HATS
- Platform Checklist
- Quick Recap of Gitlab
- Gitlab Checklist
- Pipeline Checklist
- What's next?
- Survey





GOVTECH SINGAPORE

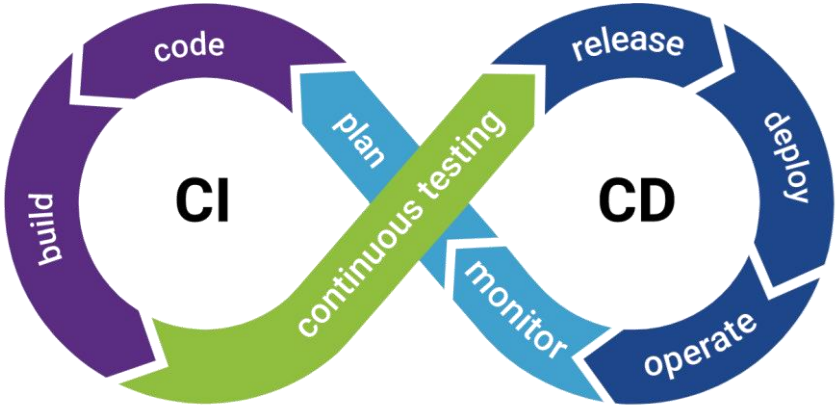
# SHIP-HATS

## A Quick Recap

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) | Planning | Build | Build Testing<br>SAST | Other Tests<br>Code quality scanning, DAST, Container Scanning | Deploy & Release |
|---|---|---|---|---|---|

**Subscription Mgmt**

techbiz

**Tools Provisioning**

SHIP-HATS Portal

**Sign in**

techpass

SEED

## GitLab-Native

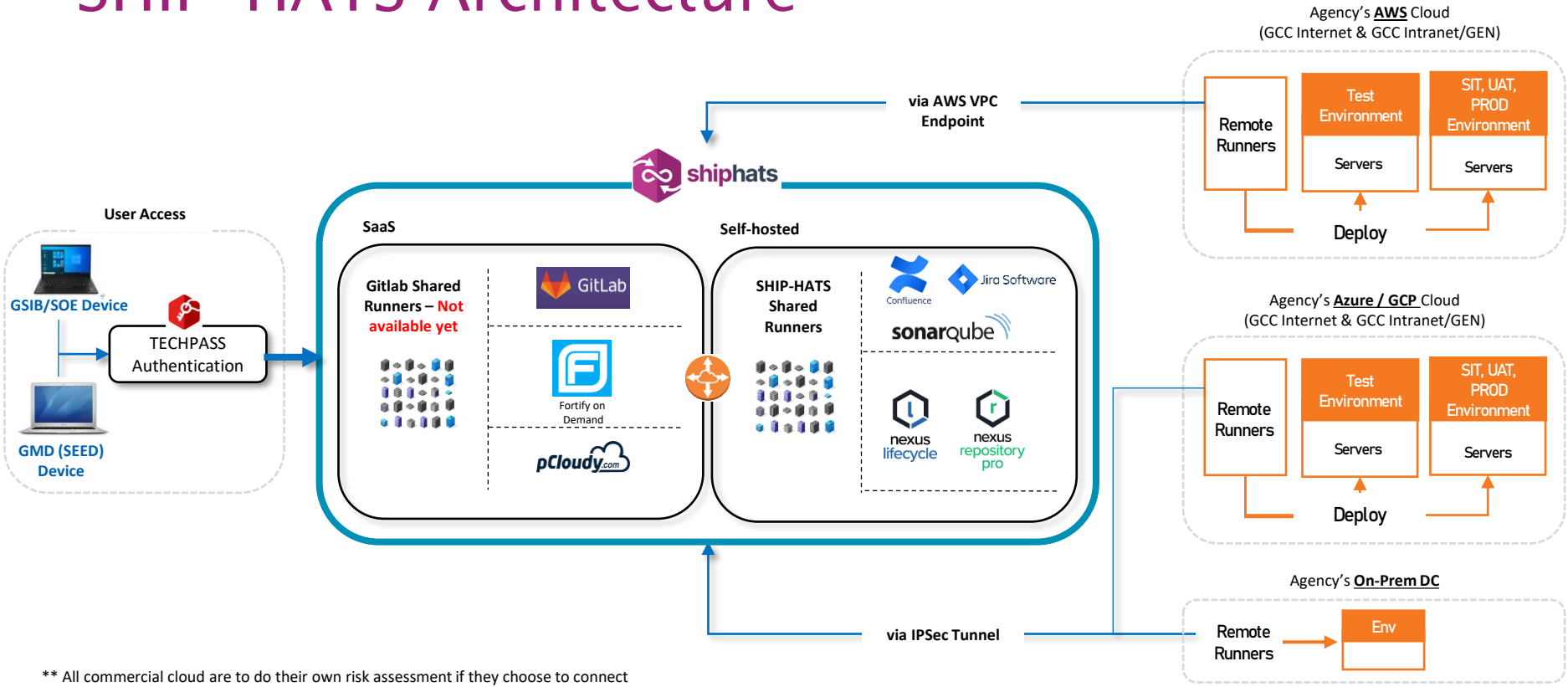
GitLab
The One DevOps Platform

- **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

| Doc/ Proj Mgmt | Artefact Repository | SAST/ DAST | Dependency Scanning | Code Quality Scanning | Device Farm |
|---|---|---|---|---|---|
| Confluence<br>Jira Software | nexus repository pro | Fortify on Demand | nexus lifecycle | sonarqube | pCloudy.com |

# SHIP-HATS Architecture

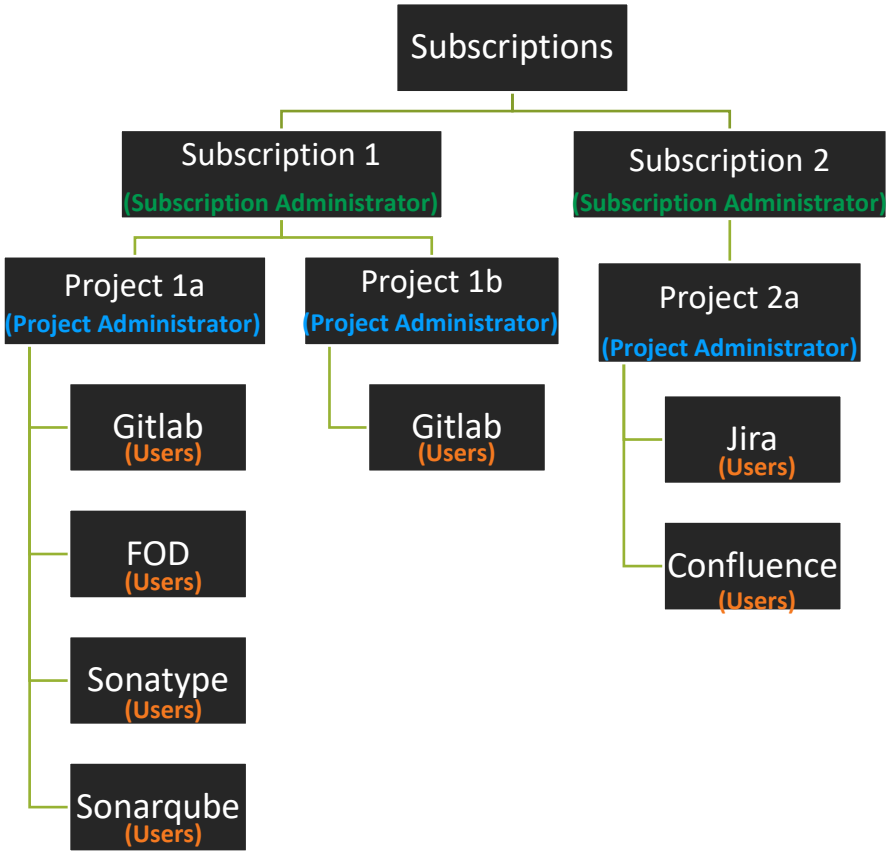



** All commercial cloud are to do their own risk assessment if they choose to connect remote runners to SHIP-HATS Gitlab over the internet.




GOVTECH SINGAPORE

# Subscription Roles and Responsibilities




## Subscription Administrator

- Manage the subscription via TechBiz
- Create projects and assign project administrators via SHIP-HATS Portal

## Project Administrator (via SHIP-HATS Portal)

- Provision tools* for the project
- Manage users and their roles for each provisioned tool
- Manage **tokens**** for each provisioned tool

## Users

- Use the tool(s) per the assigned role within the project

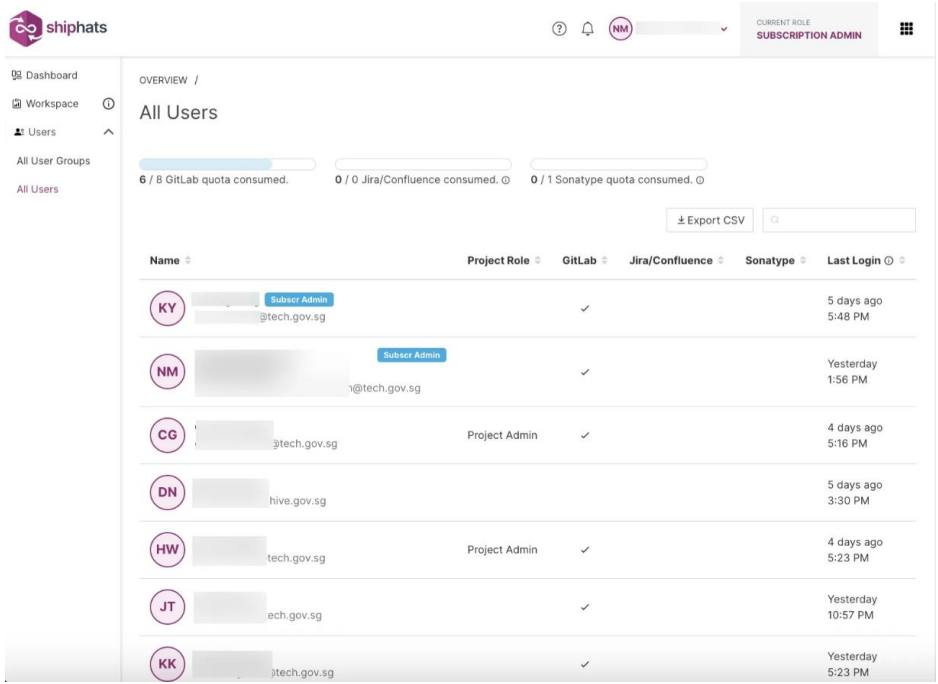
*Same instances of tools used for all subscriptions on SHIP-HATS. Provisioning here involve creating resources e.g. projects in the given tool

**Tokens are also assigned minimum privileges to support pipeline integration

# Platform Checklist

# Platform Access – Users

Validate that the platform is accessible to the intended users by checking the users included in the subscription.



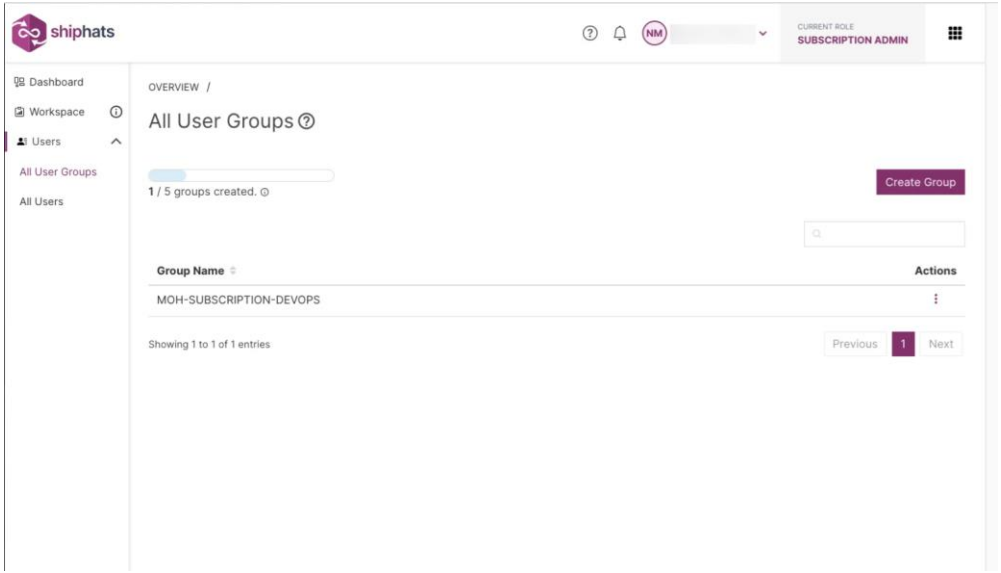

**Assign project admin wisely**

They automatically have **privileged roles** in the following tools

- Owner in Gitlab
- Administrator in Jira/Confluence
- GT-Manager in FOD
- Owner with claim components in Nexus IQ
- Administrator in Sonarqube

# Platform Access – User Group

Validate that the user groups include the intended users




Do you know

1. User groups can be assigned privileges in the various tools except Gitlab and FOD. You may have to validate the privileges these groups have in each tool to review this.

2. User groups and users are for the entire subscription. You may have to work with subscription admin and multiple project admins to review this for a subscription shared by multiple projects.




GOVTECH SINGAPORE

# Tools Access Control and Configuration

- Validate access control (users/groups) and configuration of the resources in each tool:
  - Gitlab groups/projects (more to follow in this webinar)
  - Jira projects
  - Confluence spaces
  - FOD applications
  - Nexus IQ organizations/applications
  - Sonarqube projects
  - Nexus Repository repositories

- You can use both SHIP-HATS Portal and the individual tool's UI to update role/permission for each tool except FOD (SHIP-HATS Portal only) and Nexus Repositories (Service Desk only).

- Do you know that it is possible to assign a user/group outside of your subscription to resources in a tool in your subscription?





GOVTECH SINGAPORE

# GitLab

## A Quick Recap


shiphats

## FREE

- Community Support
- Basic DevOps

## PREMIUM

**IMPROVE EFFICIENCIES and CONTROL**

- Self-managed reliability
- Release Controls
- Enterprise Agile Planning
- Advanced CI/CD
- Faster code reviews
- Priority Support
- Advanced DevOps

## ULTIMATE

**IMPROVE SECURITY, COMPLIANCE and PLANNING**

- Value Stream Management
- Portfolio Management
- Compliance
- Security risk mitigation
- Advanced security testing
- Self-managed reliability
- Release Controls
- Enterprise Agile Planning
- Advanced CI/CD
- Faster code reviews
- Priority Support
- Secure DevOps

# 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
- ✓ Migrating agencies

**Migrating agencies are advised to migrate as-is**

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





GOVTECH SINGAPORE

# GitLab Native Tools

**Review tooling assessment** in Developer Portal
**Review GitLab documentation** on what is supported/not supported

| Tool/Feature | Example Limitations/Issues |
|---|---|
| Security Dashboard | • Shows only results of scans from most recent completed pipelines on default branch<br>• Suppression of false positives apply to entire project<br>• Does not include findings for Nexus IQ |
| SAST | Limited support for Java 8 and .NET Framework |
| SCA | • No support for .NET using PackageReference<br>• Scan only first Maven POM file for project with multiple POM files<br>• Fail with error 128 for projects using Go, npm, yarn, bundler |
| Code Quality | Only on self-hosted runners as it requires DinD |

Note: The intent is not to discourage you from using Gitlab-native tools but to be clearly aware of any gap/issue so that you can plan and take step accordingly.

# 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 (not available yet)

## Option 2: Self-hosted Remote Runners

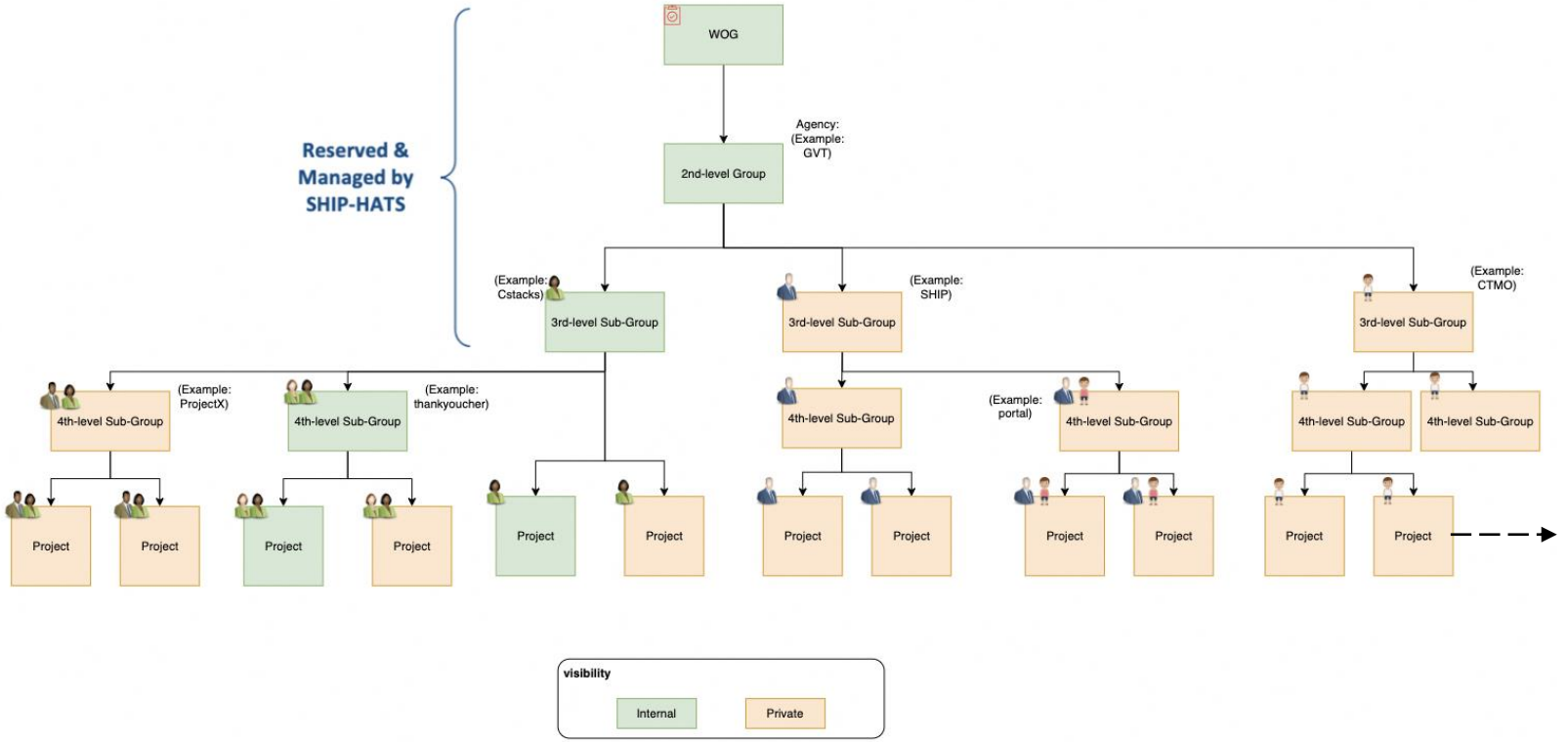
- Hosted by the agency
- Agencies to bear the costs of hosting their own runners
- Can be configured for Group or Project level access
- Full-control of the runners

**Use shared runners** where possible to minimize exposure for remote runners.

**Use CStack over Docker runners** where possible to minimize use of root. Consider Pipeline COE for runner images.

**Secure remote runners** not just on their hosts but also from the pipeline. You can potentially run a malicious job on the dedicated runner, which may have access to sensitive resources.

# Hierarchy and Structure




**Avoid deeply nested hierarchy** as it can become challenging to manage and maintain
e.g. it will not be trivial to check the membership and configurations of every subgroup/project

# Roles and Responsibilities

An example for illustration:

| Who | GitLab Role | Responsibilities |
|---|---|---|
| SHIP-HATS Project Admin | Owner | Manage groups and projects members and settings |
| Engineer | Developer | Source/test code and pipeline development |
| Project Manager | Reporter | Track project/issues |
| Manual Tester | Reporter | Test and open issues |

**Assign roles to Gitlab groups** instead of users to simplify permission management

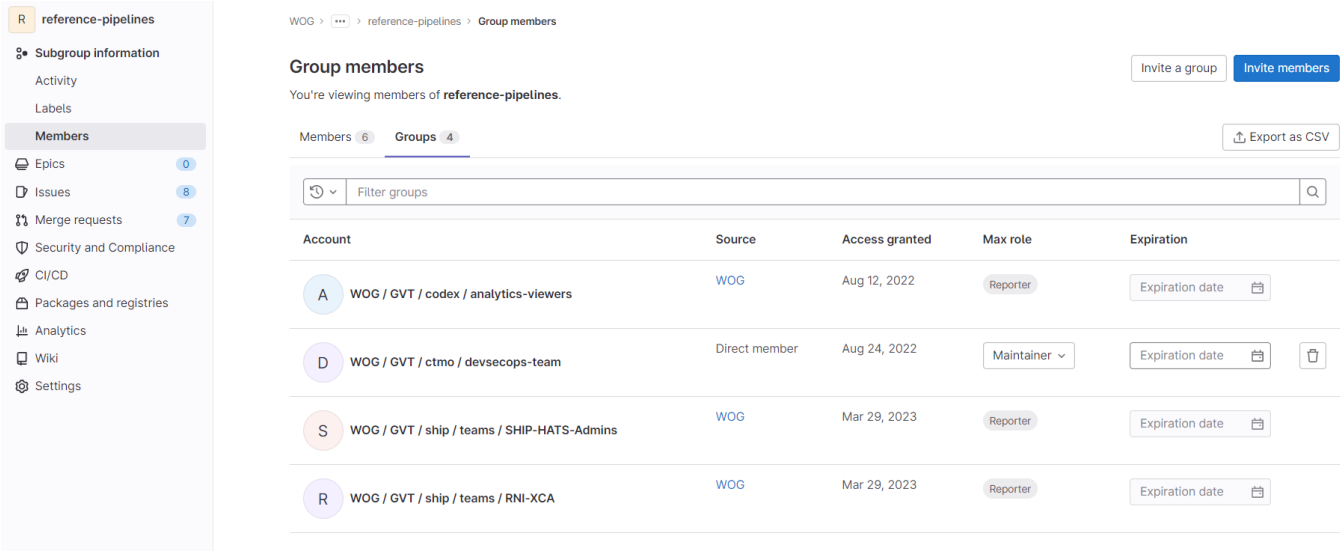
**Assign Owner/Maintainer wisely**.
They can
- View all group/project variables (including secrets) in plain text
- Remove technical controls enforced via group/project settings e.g. approvals
- Manage project membership

**Consider restricting what a developer can do.**
By default, they can
- Edit the pipeline
- Dismiss a vulnerability finding

# GitLab Checklist

# Access Control

## Validate the membership of each GitLab group/project



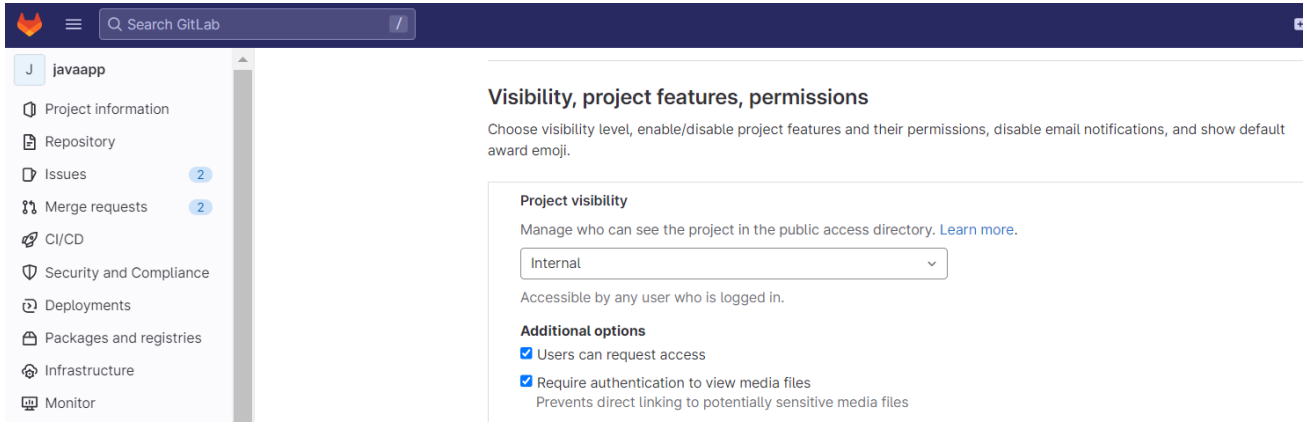




GOVTECH SINGAPORE

# Visibility Level

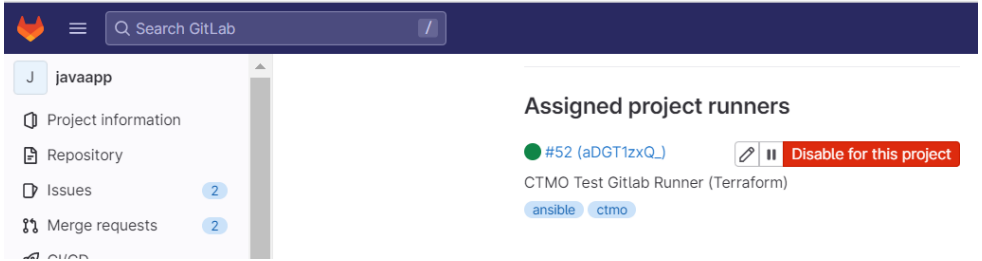
Validate visibility level is "Private“ for each GitLab group/project




SHIP-HATS Gitlab is shared by **ALL** tenants.
Setting a visibility other than “Private” mean that every SHIP-HATS users can access the group/project.

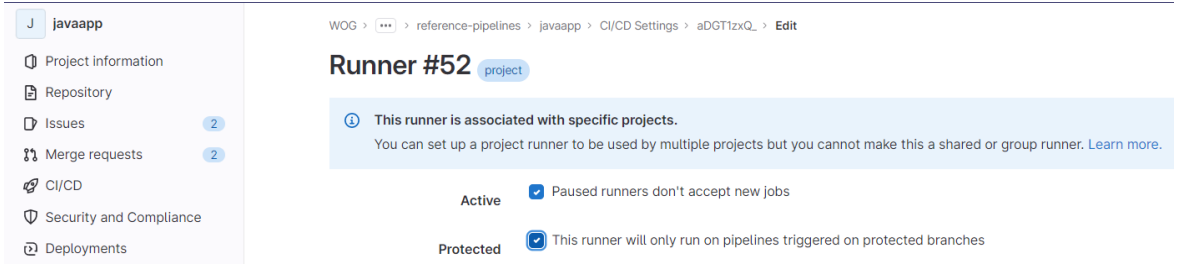
# Registered Runners

- Validate remote group and project runners include only those intentionally registered.




Do you know that remote runners can be registered by Owner/Maintainer without additional approval?

- Validate "sensitive" remote group and project runners are protected



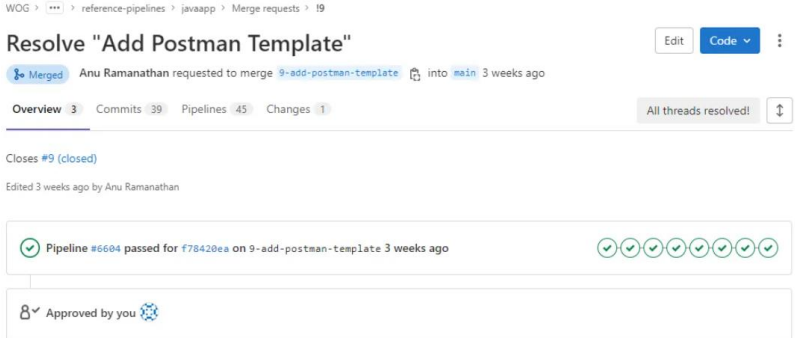

Do you know that by default, a runner can pick up jobs from any branch, including unprotected branches for which a developer can directly push changes?




GOVTECH SINGAPORE

# Peer Review

- Validate peer review is configured for each project
  1. Protect the target branch so that a merge request is required to make a change
  2. Require approval for merge requests for protected branches
  3. Prevent approval by author and users who add commits for merge request
  4. Prevent editing approval rules in merge request
  5. Require owner to approve changes to the pipeline YAML




**Enforce peer review**
While this might appear to slow down development, the intent is to avoid deploying unreviewed changes to prod. Reviewer can leverage pipeline to assess quality of change.

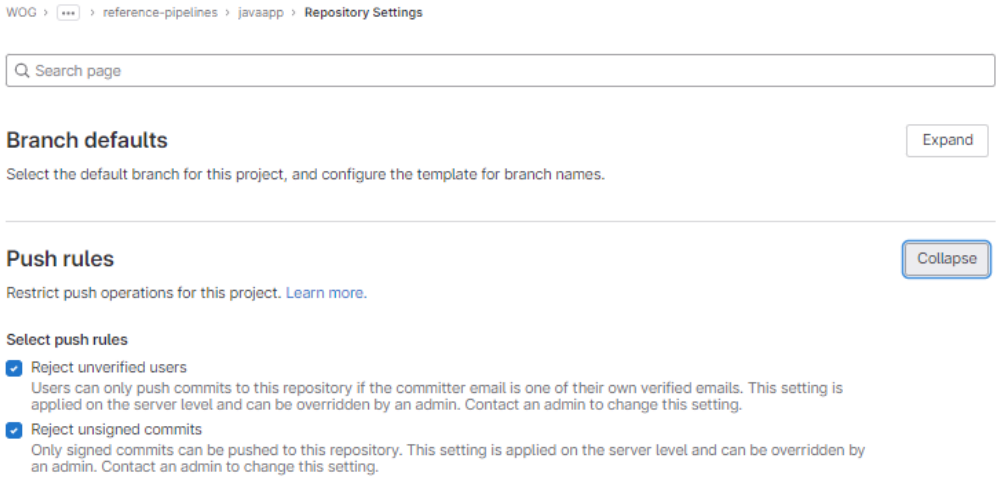
**[Reminder] Choose Owner/Maintainer wisely**
Owner/maintainer can unconfigure the settings here.




GOVTECH SINGAPORE

# Signed Commits

- Validate unverified users and unsigned commits are rejected at each group/project (enabled by default)




- Do you know that using SSH for authentication does not prevent impersonation for commits?

- **[Reminder] Choose Owner/Maintainer wisely**
  Owner/maintainer can unconfigure the settings here.




GOVTECH SINGAPORE

# Pipeline Checklist

# Reports

Validate the pipeline automatically generates the following reports

- Unit Test Report
- UI Test Report
- SAST Report
- Dependency Scanning Report
- Code Quality Scan Report
- Unit Test Code Coverage Report
- API Test Report
- DAST Report
- Container Scanning Report
- Secret Detection Report

and

- Saved as job artifacts (default expiry 14 days) and/or


- Published to artifact repo (default expiry 6mths for Nexus Repo)

You can also use the dashboards in the individual tools but they usually reflect the latest scan.

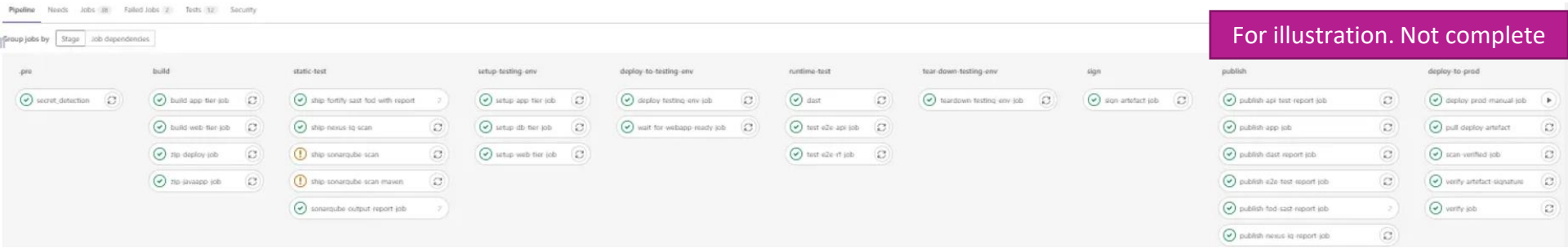



GOVTECH SINGAPORE

# Quality Gates

## Validate that for each required test/scan

1. The pipeline invokes the test/scan
   - Unit Test
   - SAST
   - Code Quality Scan
   - API Test
   - DAST
   - Container Scanning
   - UI Test
   - Dependency Scanning
   - Secret Detection
2. The test/scan completes successfully
3. The test/scan passes the success criteria




Do you know that a “green tick” can means that the job completes successfully but test/scan may **NOT** have passed? It is possible to script each test/scan job so that it fails if the success criteria is not met but you will have to review the script to ensure that this hard gating has been implemented.



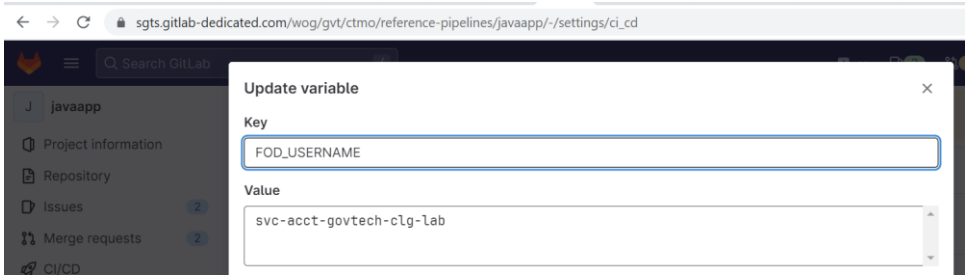

GOVTECH SINGAPORE

# Tokens

## Validate that the project token* is used for integration with

- Nexus IQ
- Nexus Repository
- FOD
- Sonarqube

*Not supported for Jira/Confluence as yet




Do you know that using a personal token imply

1. The corresponding user may have additional privileges that he/she should not have (e.g. publish to Nexus Repo)
2. It may not be possible to differentiate between the pipeline and the user in the tools' audit logs

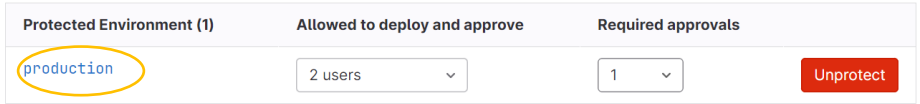
**Save the project token as a masked group/project variable** (if not using a secret manager)
If you cannot find the project token in the list of group/project variable, it may have been **hardcoded** in the pipeline YAML.

# Deployment Approvals

## Validate that deployment to production requires approval

1. Tag deployment job in pipeline as deployment to PROD
2. Configure PROD as protected environment
3. Configure who are allowed to deploy to PROD and #approvals required

```
68  deploy-testing-env-job:
69    extends: .prep-for-ansible
70    stage: deploy-to-prod-env
71    environment:
72      name: production
73    script:
```




**Restrict who can edit the pipeline YAML**

By default, a developer can edit the pipeline YAML and can therefore remove the environment tag from the deployment job (along with the need for approvals). Consider

1. Require owner to approve changes to pipeline YAML
2. Separate project for deployment




GOVTECH SINGAPORE

# What’s Next?

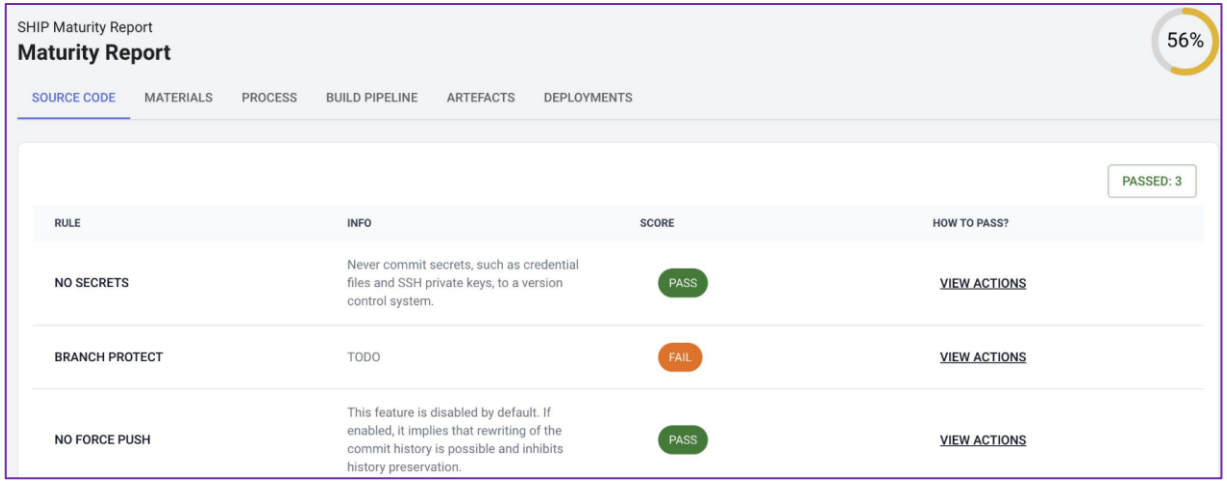
*Check DevPortal for Updates*

# SHIP-HATS 2.0. Product Roadmap

| Component | Q1 | Q2 | Q3 | Q4 |
|---|---|---|---|---|
| **Platform Enablement** | • Shared runners for Intranet deployment | • Runners to support GitLab-Native services<br>• 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<br>• GitOps-based Templates | • GitLab Guest-Account Enablement with TechPass | • Limiting Feature for GitLab Guest Accounts | • Support for GCC +<br>• BCP |
| **Security Baseline & Testing Tools** | • SemGrep (bg scanning)<br>• DevSecOps Security Baseline – identification of baselines and and draft measures | • DevSecOps Security Baseline – roll-out to GDS | • **SLSA**<br>• 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 |

Covers some of the checks in this webinar

# DevSecOps Scoreboard




**Collaboration between GDS/CSG/OSG/CTMO**

Capture DevSecOps metrics at the various granularity levels for better oversight on maturity and process improvement.

**Looking for pilot users!**

Reach out to Liyana_MUHAMMAD_FAUZI@tech.gov.sg to participate in the pilot

# Documentation (Requires SHIP-HATS Access)

- **PM/BA Checklist:** https://sgts.gitlab-dedicated.com/groups/wog/gvt/ctmo/reference-pipelines/-/wikis/PM-BA-Checklist-(SHIP-HATS-2.0)
- **Considerations for using GitLab:** https://sgts.gitlab-dedicated.com/groups/wog/gvt/ctmo/reference-pipelines/-/wikis/Considerations-for-Using-GitLab
- **CD Approaches:** https://sgts.gitlab-dedicated.com/groups/wog/gvt/ctmo/reference-pipelines/-/wikis/Gitlab-CD-Approaches





GOVTECH SINGAPORE

# Upcoming Webinars

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

- GitLab as a PM tool
- Configuring SHIP-HATS 101 (Part 2) – Alternative Scanning Tools

Sign up here



GOVTECH SINGAPORE

# Key Timelines

| Tool | Decommission Date | Replacement Tool |
|---|---|---|
| Fortify WebInspect (OnPrem) | 31 July 2023 | GitLab SAST/DAST or Fortify On Demand |
| Fortify SCA (OnPrem) | 31 July 2023 | GitLab SAST/DAST or Fortify On Demand |
| Digital.ai | 26 May 2023<br><br>**Note:** The date has been corrected to reflect the actual date of decommissioning. | GitLab |
| Bitbucket | January 2024 | GitLab |
| Bamboo | January 2024 | GitLab |
| Prisma Cloud | January 2024 | GitLab |



GOVTECH SINGAPORE

# Survey

Please do fill in the below survey to share

1. your feedback on this webinar and
2. if you want to attend a hands-on workshop version of this webinar!

https://form.gov.sg/645dda8053ed3e0012c04112



GOVTECH SINGAPORE

# Thank You