GOVTECH
SINGAPORE

SHIP-HATS 2.0 Docker Pipeline Webinar

Learning Events | Level 200 Tech

GOVTECH
SINGAPORE

ABOUT US

CLOUD | DEVOPS
MACHINE LEARNING

Implementation and training services

CLOUD ENABLED PTE LTD

cloud computing

Dev Ops

INDIA | SINGAPORE | DUBAI

SINCE 2013

PHONE +65 -81320344

EMAIL

reach@thecloudenabled.com

WEBSITE

www.thecloudenabled.com

ANIL BIDARI

CEO @ CLOUD ENABLED

Understanding Docker Pipeline

shiphats

Pipeline Templates

SHIP-HATS 2.0 offers a library of CI/CD pipeline templates that simplifies configuration efforts.

E2E templates

- Main ci file to build your pipeline
- Build→Test→Deploy .yml files with all variable keys defined (only values you need to add)

Modular templates

- Leverage these if Anything specific task in pipeline required
- Example – check webapp is ready or not

Review E2E Templates

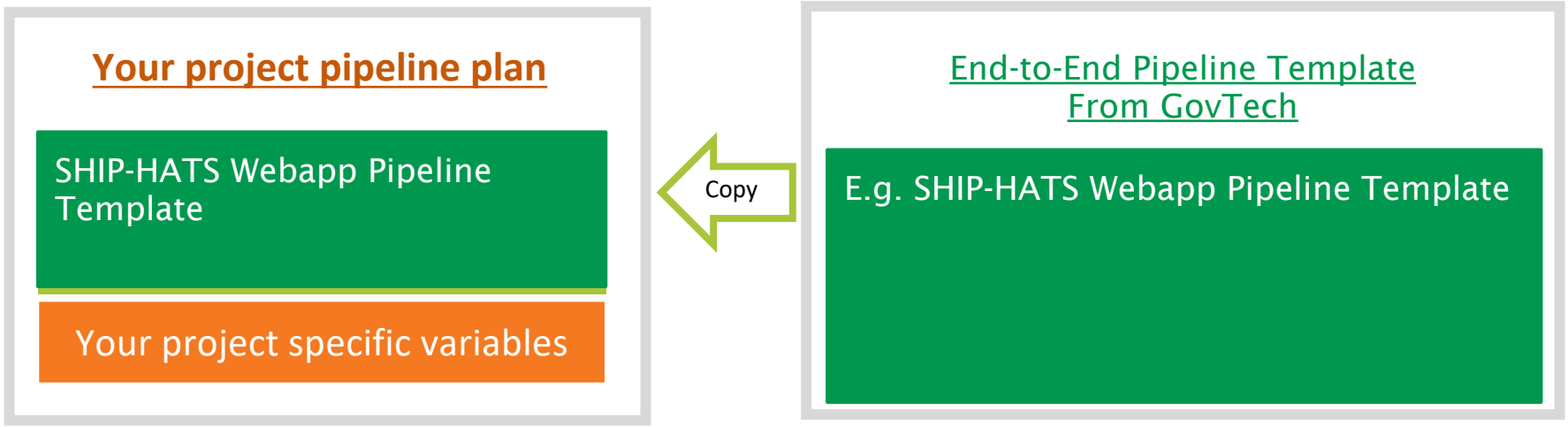
1. Go to Developer Portal: SHIP-HATS Pipeline Templates
2. Check for E2E Templates
3. Let's review SHIP-HATS Webapp E2E Template

E2E templates

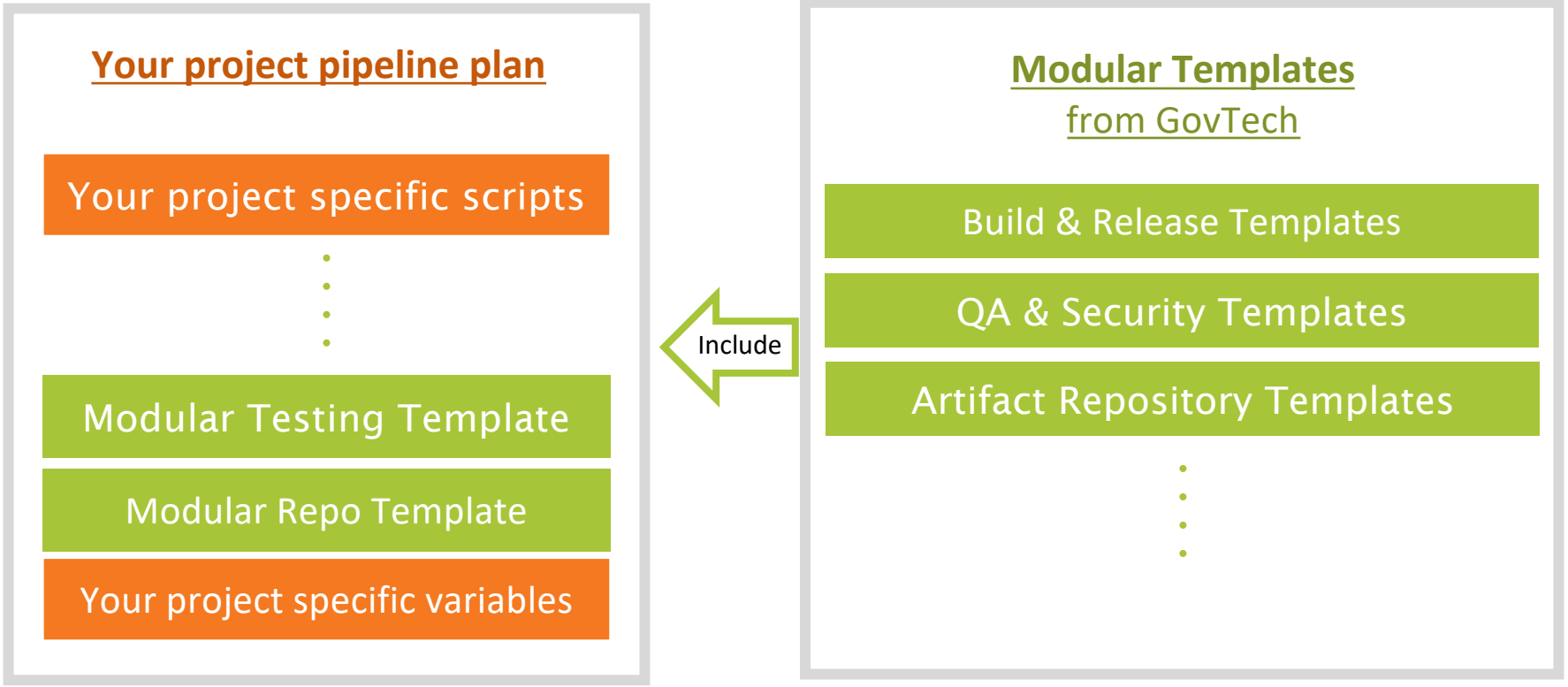
Template	Description	Template Type
SHIP-HATS Docker Image CI Pipeline Templates	This template allows you to deliver a compliant CI pipeline for a standard docker application (regardless of language) considering best practices as well as security aspects.	End to end Template
SHIP-HATS Docker Multi Services App E2E Templates	This template allows you to deliver a compliant CI/CD pipeline for a standard docker application (regardless of language) considering best practices as well as security aspects.	End to end Template
SHIP-HATS Docker Single Service App E2E Templates	This template allows you to deliver a compliant CI/CD pipeline for a standard docker application (regardless of language) considering best practices as well as security aspects.	End to end Template
SHIP-HATS Webapp E2E Templates	This template allows you to deliver a compliant CI/CD pipeline for a standard web application (regardless of language) considering best practices as well as security aspects.	End to end Template

Access E2E templates in GitLab here

End-to-end Pipeline Template

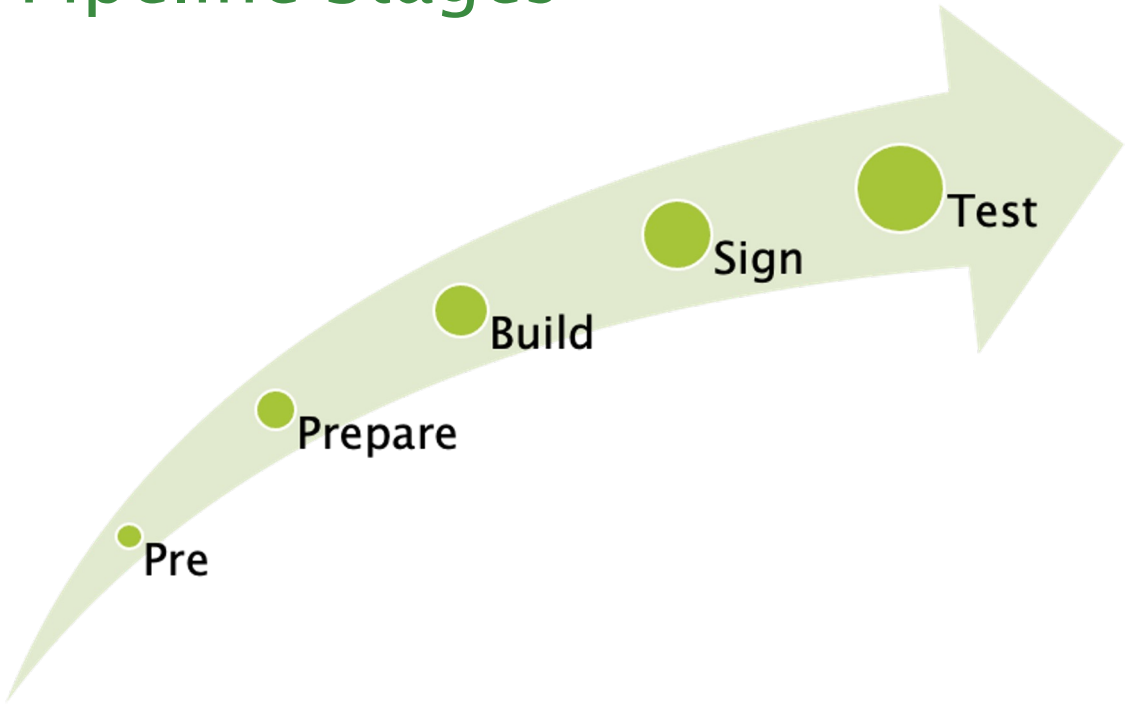


Modular Pipeline Templates



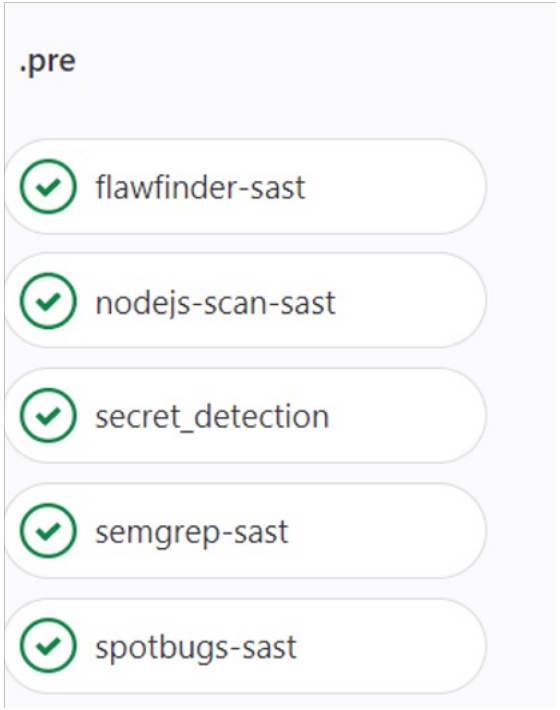
Sample Docker Pipeline walkthrough

Docker Pipeline Stages



We are using a nodejs app in this tutorial

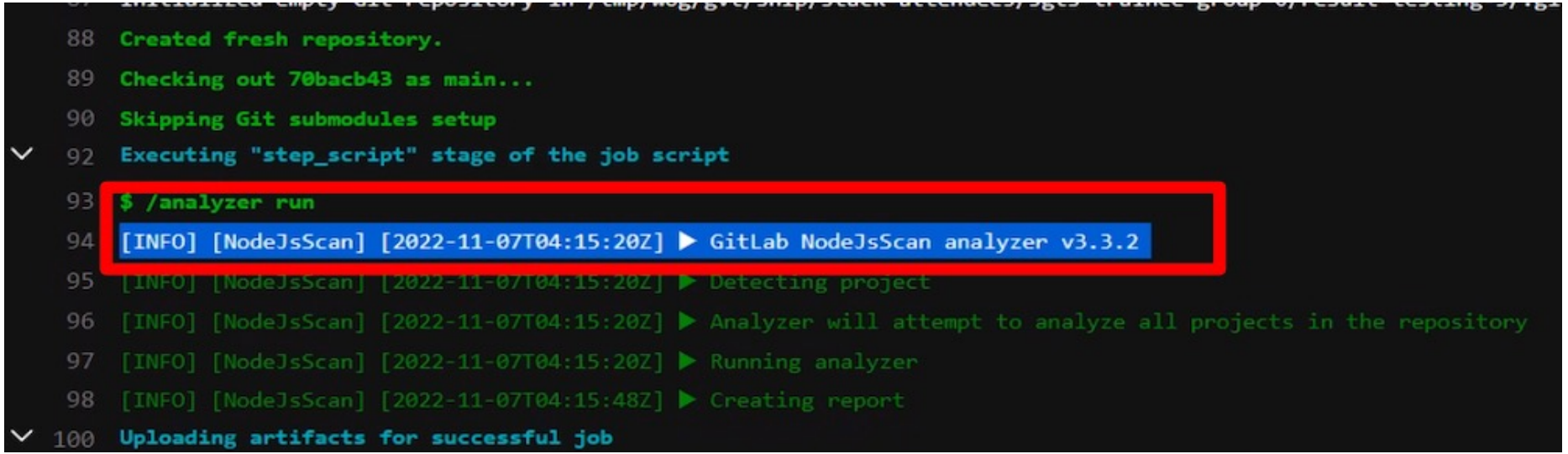
Pre-Stage



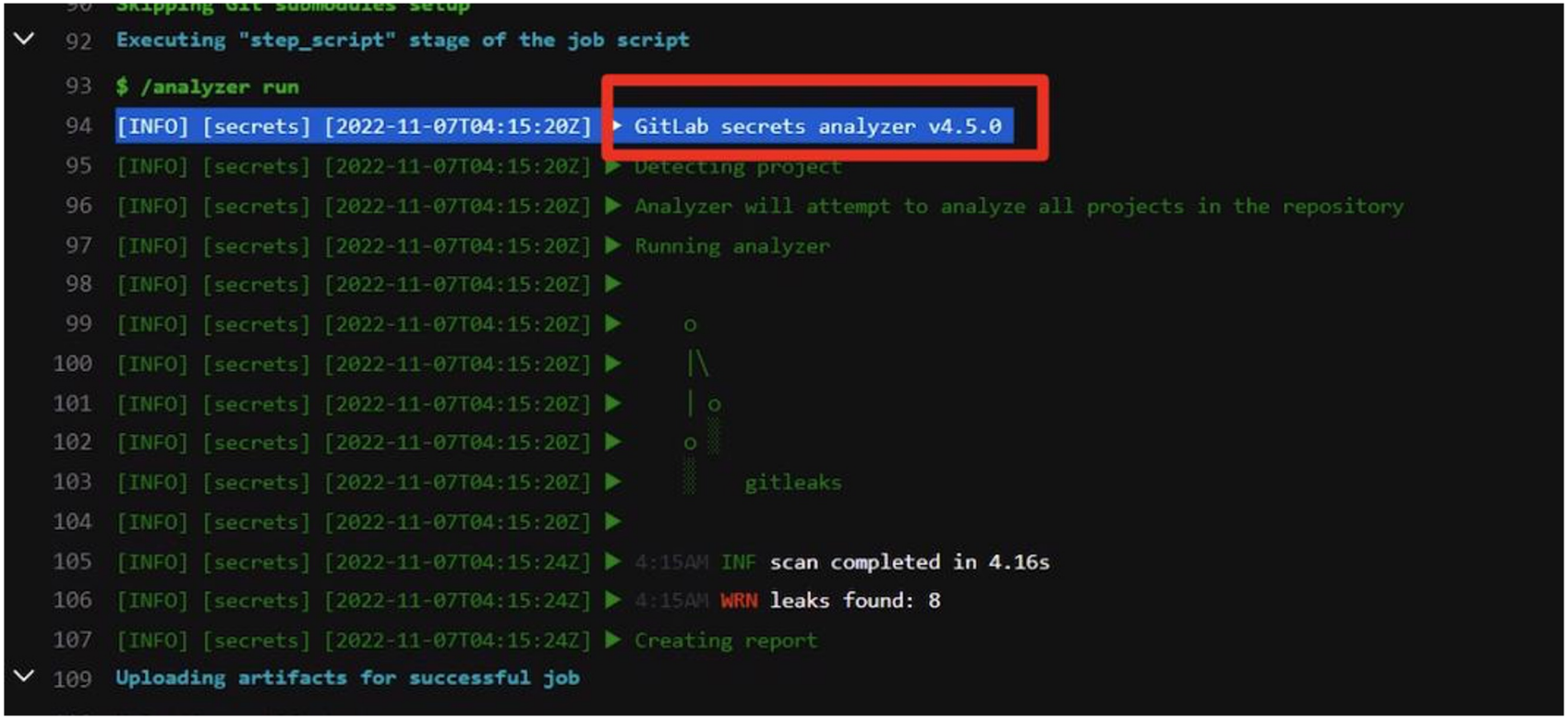
Pre-Stage

```
15 Initialized empty Git repository in /builds/wog/gvt/ship/stack-attendees/sgts-trainee-group-0/result-testing-3/.git/
16 Created fresh repository.
17 Checking out 70bacb43 as main...
18 Skipping Git submodules setup
20 Executing "step_script" stage of the job script
21 Using docker image sha256:f53dd3cfe140639d05f587913865f0e6cbab5f3441d7f73cf8752df516433fec for registry.gitlab.com/security-products/gemnasiu
   st registry.gitlab.com/security-products/gemnasium@sha256:d73db9517813851fd82758a23f59ba34cb5250d07a474ae8c1300eca300899d5 ...
22 $ /analyzer run
23 [INFO] [Gemnasium] [2022-11-07T04:14:34Z] ▶ GitLab Gemnasium analyzer v3.10.5
24 [INFO] [Gemnasium] [2022-11-07T04:14:37Z] ▶ Using commit cde471a8a4e9bcea8b0b7d8e1b3035bfea896cb6f
25  of vulnerability database
26 [INFO] [Gemnasium] [2022-11-07T04:14:39Z] ▶ using schema model 14
27 [INFO] [Gemnasium] [2022-11-07T04:14:39Z] ▶ Cannot auto-remediate dependency file, not supported: package-lock.json
28 [INFO] [Gemnasium] [2022-11-07T04:14:39Z] ▶ Cannot auto-remediate dependency file, not supported: views/angular.min.js
30 Uploading artifacts for successful job
31 Uploading artifacts...
32 **/gl-sbom-*.cdx.json: found 1 matching files and directories
33 Uploading artifacts as "archive" to coordinator... 201 Created  id=393142 responseStatus=201 Created token=noAmgtkH
```

Pre-Stage



Pre-Stage

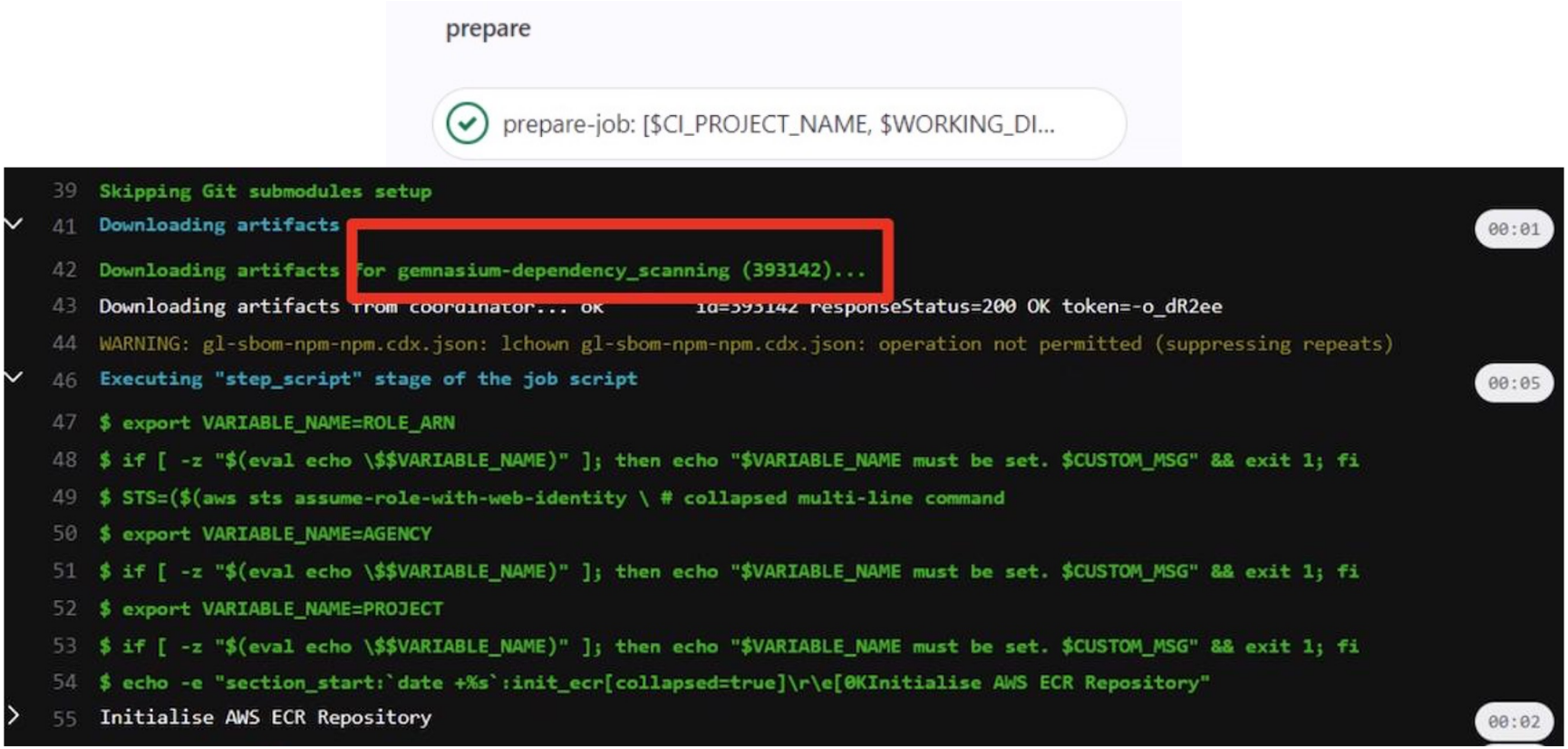


Pre-Stage

```
89 Executing "step_script" stage of the job script
90 $ /analyzer run
91 [INFO] [Semgrep] [2022-11-07T04:15:18Z] ▶ GitLab Semgrep analyzer v3.8.0
92 [INFO] [Semgrep] [2022-11-07T04:15:18Z] ▶ Detecting project
93 [INFO] [Semgrep] [2022-11-07T04:15:18Z] ▶ Analyzer will attempt to analyze all projects in the repository
94 [INFO] [Semgrep] [2022-11-07T04:15:18Z] ▶ Running analyzer
95 [INFO] [Semgrep] [2022-11-07T04:15:41Z] ▶ Creating report
96 [INFO] [2022-11-07T04:15:41Z] ▶ /tmp/wog/gvt/ship/stack-attendees/sgts-trainee-group-0/result-testing-3/gl-sast-report-post.json written
98 Uploading artifacts for successful job
```

```
95 [INFO] [Spotbugs] [2022-11-07T04:15:21Z] ▶ GitLab Spotbugs analyzer v3.3.0
96 [INFO] [Spotbugs] [2022-11-07T04:15:21Z] ▶ Detecting project
97 [WARN] [Spotbugs] [2022-11-07T04:15:21Z] ▶ No match in /tmp/wog/gvt/ship/stack-attendees/sgts-trainee-group-0/resul
```

Prepare-Stage

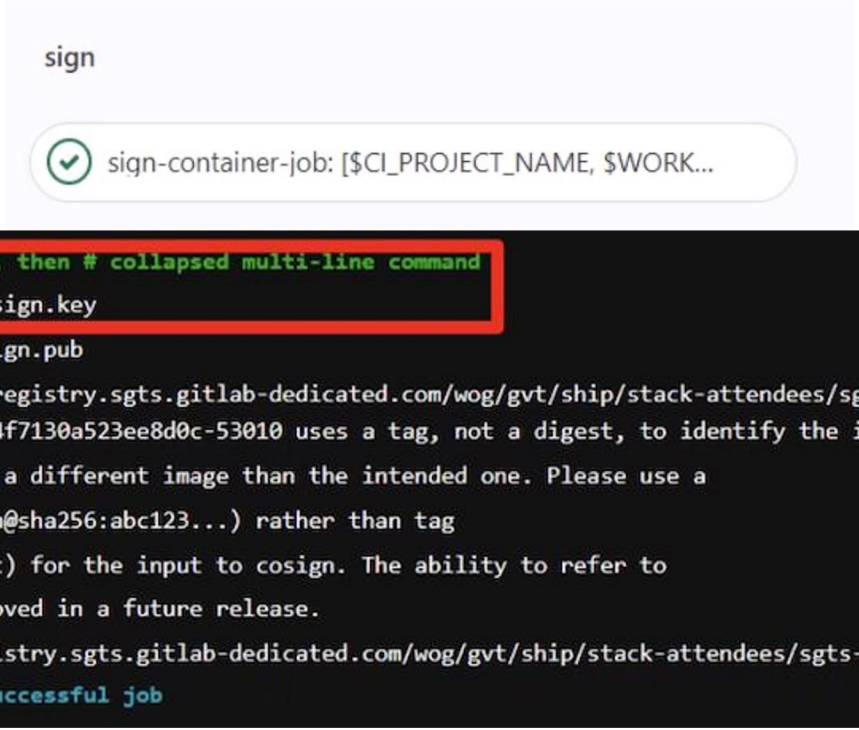


Build-Stage

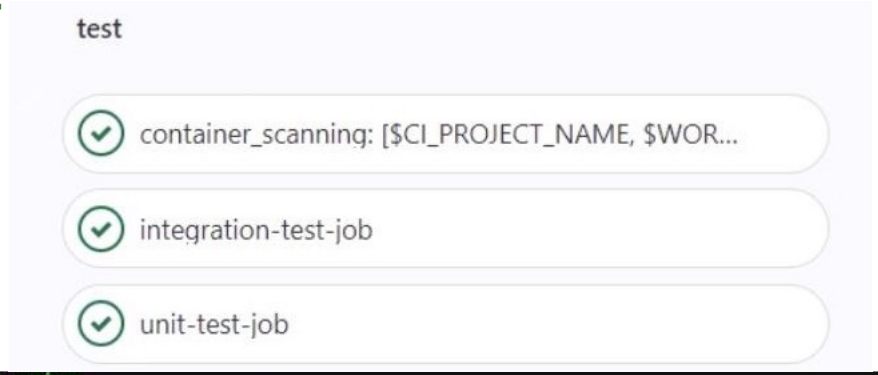

```
302 70bacb435c26ae3cebb902b34f7130a523ee8d0c-53010: digest: sha256:60ee601d47c8cee5070f8d34da92dc6ab687e71b79d6215eae60a9c40b4e7e6
    6 size: 3042
303 $ docker save $DOCKER_TARGET_REGISTRY/$DOCKER_TARGET_IMAGE > $OUTPUT_IMAGE_ARTEFACT
304 $ mkdir -p ./build-image-job/
305 $ echo "OUTPUT_IMAGE_ARTEFACT=$OUTPUT_IMAGE_ARTEFACT;" > ./build-image-job/env_$PRODUCT_IMAGE_NAME.sh
306 $ chmod +x ./build-image-job/env_$PRODUCT_IMAGE_NAME.sh
308 Uploading artifacts for successful job                                                          00:10
309 Uploading artifacts...
310 result-testing-3.tar: found 1 matching files and directories
311 ./build-image-job/env_result-testing-3.sh: found 1 matching files and directories
312 Uploading artifacts as "archive" to coordinator... 201 Created  id=393146 r          oken=K5CT_H-z
```

Using Custom Setup

Sign-Stage



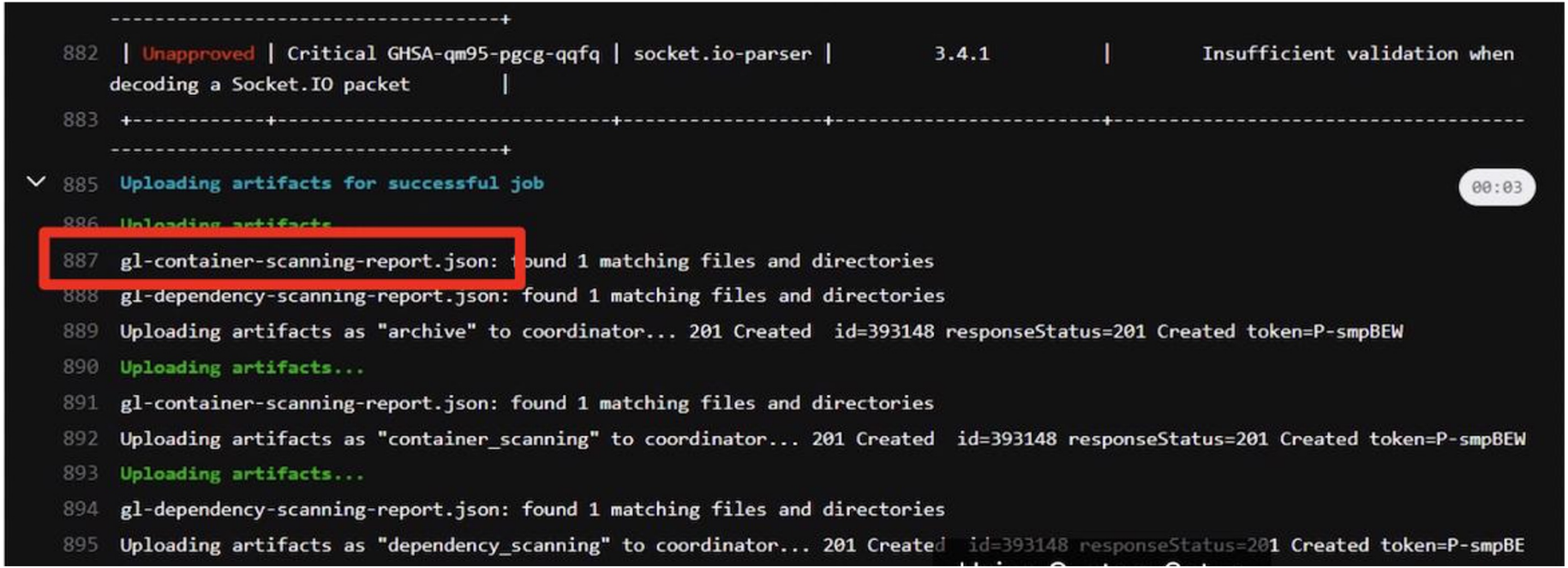
Test-Stage



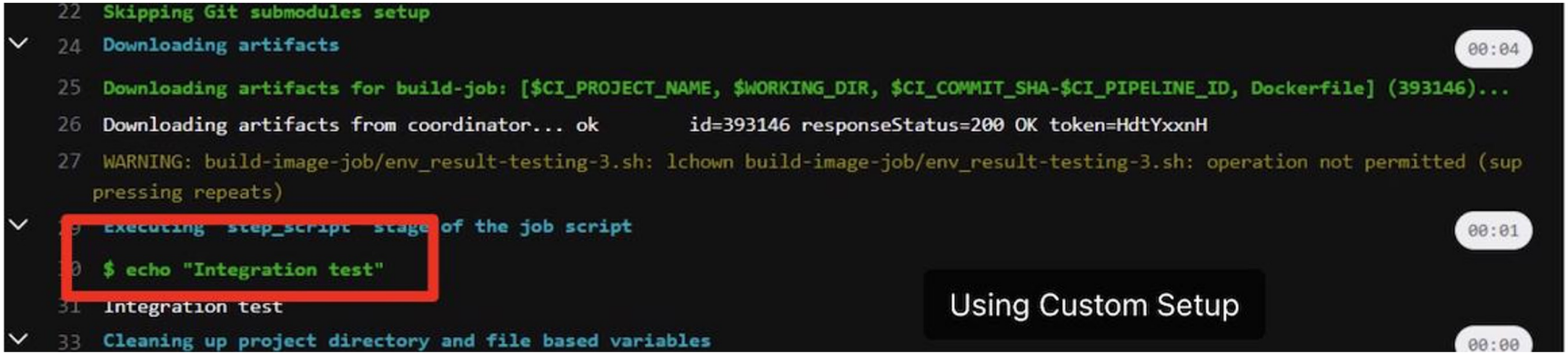

```
18 Skipping Git submodules setup
20 Executing "step_script" stage of the job script                    00:11
21 Using docker image sha256:f89e06552ceb36788979a8240f7ae34410cfcc5edd5e82b8a46e491cc84373b2 for registry.gitlab.com/security-products/container-scanning/grype:5 with digest registry.gitlab.com/security-products/container-scanning/grype@sha256:92b026c24b82abce1513a9d544358404ca86bd4ee4d4de708311051d1242c25b ...
22 $ gtcs scan
23 [INFO] [2022-11-07 04:20:28 +0000] [container-scanning] > Scanning container from registry registry.sgts.gitlab-dedicated.com/wog/gvt/ship/stack-attendees/sgts-trainee-group-0/result-testing-3:70bacb435c26ae3cebb902b34f7130a523ee8d0c-53010 for vulnerabilities with severity level UNKNOWN or higher, with gcs 5.2.4 and Grype Version: 0.50.2, advisories updated at 2022-11-06T08:18:58+00:00
```

Using Custom Setup

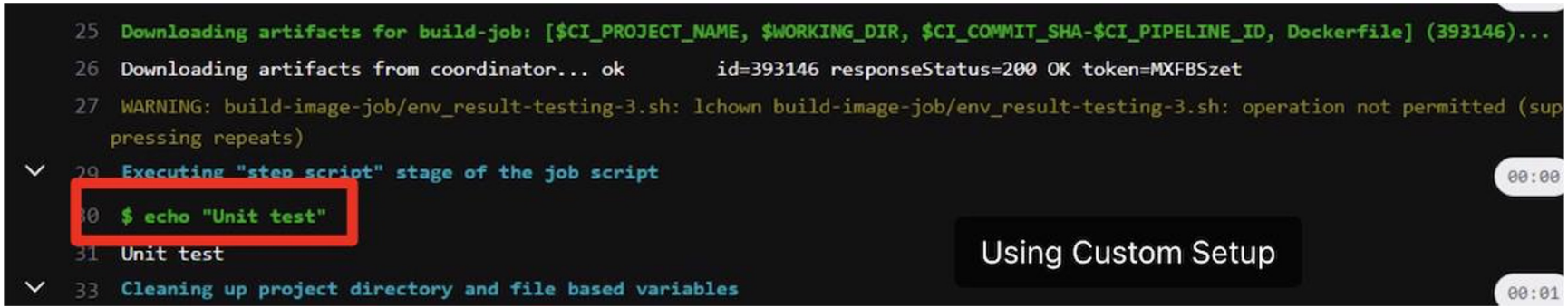
Test-Stage



Test-Stage | Integration-test



Test-Stage | Unit-test



Deploy-Stage

Deploy-Stage

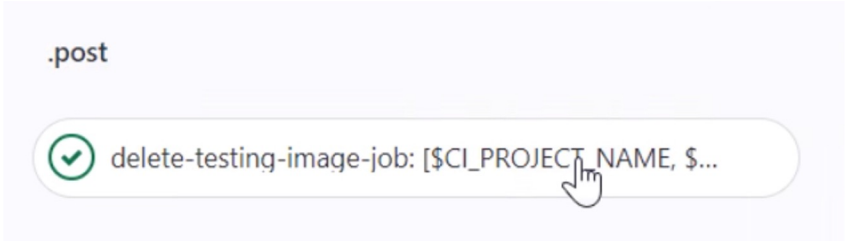
```
  DOCKER_PASSWORD: $AWS_ECR_TOKEN

build-and-push-docker-image:
 image:
   name: gcr.io/kaniko-project/executor:debug
   entrypoint: [""]
 tags:
   - ship_docker
 variables:
   DOCKERFILE_PATH: "./"
   DOCKERFILE_NAME: "Dockerfile"
   DIGESTFILE_NAME: "$CI_PROJECT_DIR/digest"
   OPTS: ""
 script:
   - export VARIABLE_NAME=DOCKER_TARGET_REGISTRY
   - !reference [.check-variable-sh, script]
   - export VARIABLE_NAME=DOCKER_TARGET_IMAGE
   - !reference [.check-variable-sh, script]
   - !reference [.check-docker-auth-config-sh, script]
   - export DOCKER_CONFIG=~/.docker/
   - /kaniko/executor --context "$DOCKERFILE_PATH" --dockerfile "$DOCKERFILE_PATH/$DOCKERFILE_NAME" --destination "$DOCKER_TARGET_REGISTRY/$DOC
 artifacts:
   paths:
     - $DIGESTFILE_NAME
   expire_in: 1 week
```

Deploy-Stage

```
35  $ if [ -f "$HOME/.docker/config.json" ]; then # collapsed multi-line command
7   cosign.pub exists
8   $ cosign verify --key cosign.pub $DOCKER_REGISTRY/$IMAGE_NAME:$IMAGE_VERSION
    35c26ae3cebb902b34f7130a523ee8d0c-53010 --
40  The following checks were performed on each of these signatures:
```

Post-Stage

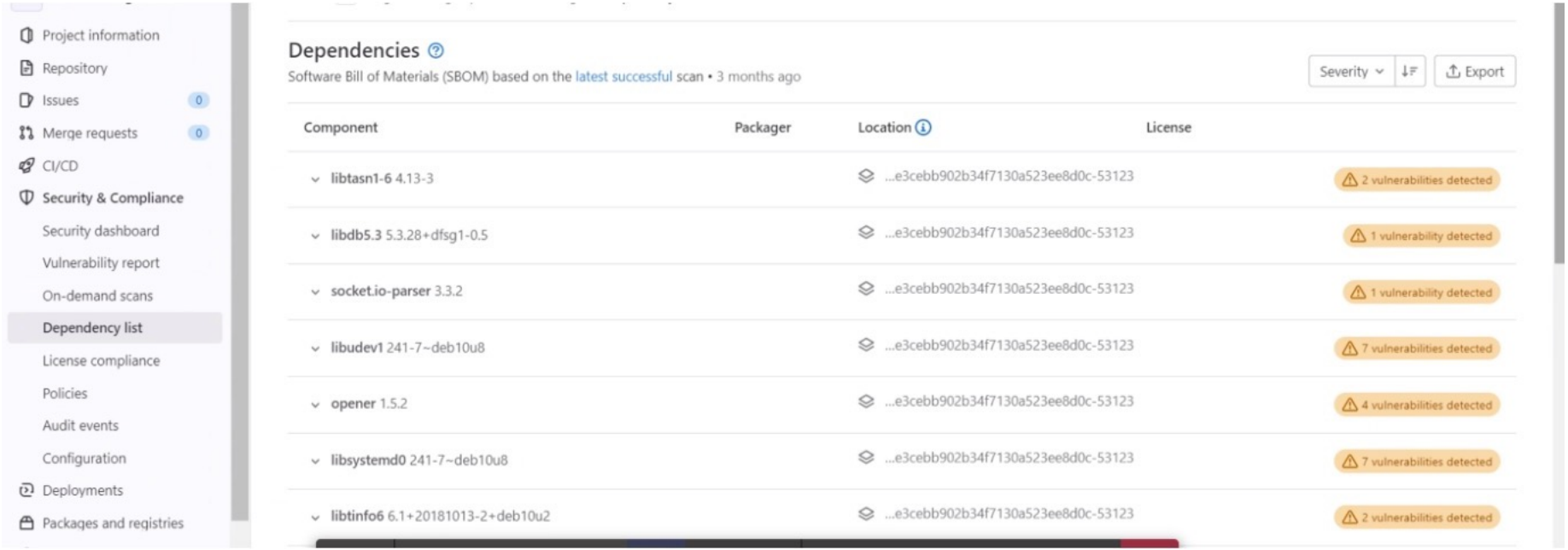



```yaml
delete-testing-image-job: # delete image for testing from TEST_REGISTRY
  stage: .post
  extends: .delete-gitlab-cr-image
  variables:
    IMAGE_NAME: ""
    IMAGE_VERSION: $VERSION
  when: always
  allow_failure: true
  parallel:
```

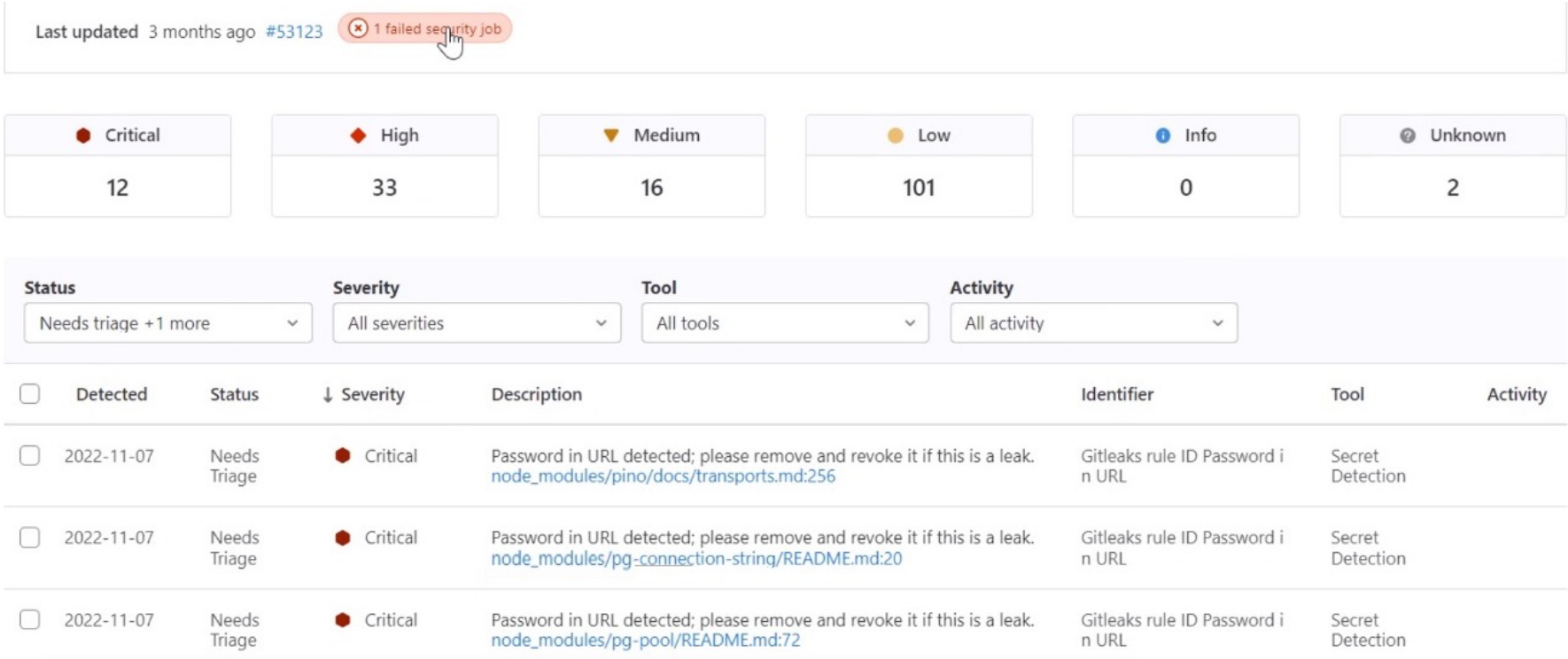
Post-Stage

```
48 $ if [ -z "$(eval echo \$$VARIABLE_NAME)" ]; then echo "Expected $VARIABLE_NAME but not found in json. Unable to proceed. $CUSTOM_MSG" && exit 1; fi
49 $ DIGEST=$( curl --header "PRIVATE-TOKEN:$API_TOKEN" "https://$CI_SERVER_HOST/api/v4/projects/$CI_PROJECT_ID/registry/repositories/$REPOSITORY_ID/tags/$IMAGE_VERSION" | jq -r --arg VALUE "$IMAGE_VERSION" '. | select(.name | contains($VALUE)) | .digest' )
50   % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
51                                  Dload  Upload   Total   Spent    Left  Speed
52   0     0    0     0    0     0      0      0 --:--:-- --:--:-- --:--:--     0
53 100   599  100   599    0     0   1035      0 --:--:-- --:--:-- --:--:--  1034
54 $ curl --request DELETE --header "PRIVATE-TOKEN:$API_TOKEN" "https://$CI_SERVER_HOST/api/v4/projects/$CI_PROJECT_ID/registry/repositories/$REPOSITORY_ID/tags/$IMAGE_VERSION"
55   % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
56                                  Dload  Upload   Total   Spent    Left  Speed
57 100     3  100     3    0     0     16      0 --:--:-- --:--:-- --:--:--    16
58 200$ CHECK_MESSAGE=$( curl --header "PRIVATE-TOKEN:$API_TOKEN" "https://$CI_SERVER_HOST/api/v4/projects/$CI_PROJECT_ID/registry/repositories/$REPOSITORY_ID/tags/$IMAGE_VERSION" | jq '. | select(.message != null) | .message' )
59   % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
60                                  Dload  Upload   Total   Spent    Left  Speed
61   0     0    0     0    0     0      0      0 --:--:-- --:--:-- --:--:--     0
62 100    31  100    31    0     0    180      0 --:--:-- --:--:-- --:--:--   181
63 $ if [ -z "$(eval echo \$CHECK_MESSAGE)" ]; then echo "Failed to delete image." && exit 1; fi
64 $ curl --request DELETE --header "PRIVATE-TOKEN:$API_TOKEN" "https://$CI_SERVER_HOST/api/v4/projects/$CI_PROJECT_ID/registry/repositories/$REPOSITORY_ID/tags/${DIGEST/:/-}.sig"
65   % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
66                                  Dload  Upload   Total   Spent    Left  Speed
```

Dependency Findings



Security Findings



Compliance Framework

Automate adopting DevSecOps best practices based on industry pipeline security & IM8

Examples:

- SCA including Dependency scanning
- SAST / DAST / Container scanning
- Gating before deployment to high stake environments
- Reports generation as part of provenance
- Signing and verification of signature on artefacts
- Checksum verification of artefacts
- Use of artifactory

We will iteratively improve these to help you meet DevSecOps policy.

Highly recommended

Compliance Framework

Benefits

Set up CI/CD pipeline faster
Leverage GitLab's OTS security tools and reporting tools
Flexibility to change to **non-GitLab Alternative Tools** in 2.0
Better quality by achieving compliance to industry standards.

Know your Resources: Activity

Bookmark Learning Events for past and upcoming events

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

Reach out to us at https://go.gov.sg/she for feedback, feature requests - we priorities based on demand!

Thank You

shiphats