

1. Create gitlab project

The first screenshot shows the GitLab user profile page for 'sunshine04178'. A red box highlights the 'New project/repository' button in the top-left navigation menu. The 'Activity' section shows a list of recent pushes and project creations.

The second screenshot shows the 'frogbot-scan' project page. A red box highlights the project name 'F frogbot-scan' in the breadcrumb navigation. The page displays project information, including 4 commits, 4 branches, and 0 tags. A table lists the project's files and their last commit details.

Name	Last commit	Last update
multi1	maven image	32 minutes ago
multi2	maven image	32 minutes ago
multi3	maven image	32 minutes ago
.gitlab-ci.yml	maven image	32 minutes ago
README.md	Initial commit	2 hours ago
mvnw	add project	2 hours ago
mvnw.cmd	add project	2 hours ago
pom.xml	add project	2 hours ago
rpm-script	add project	2 hours ago
settings.xml	add project	2 hours ago
training.spec	add project	2 hours ago

2. Download maven project and copy to your project of frogbot-scan

```
git clone https://gitlab.com/sunshine04178/frogbot-scan.git (your project address)
unzip project-examples-master.zip
mv project-examples-master/maven-examples/maven-example/*
/Users/hailangz/test/frogbot-scan
```

3. Add yml

```
cd /Users/hailangz/test/frogbot-scan  
  
vim .gitlab-ci.yml
```

```
frogbot-scan:  
image: maven:3.6.3  
rules:  
- if: $CI_PIPELINE_SOURCE == 'merge_request_event'  
when: manual  
variables:  
  FROGBOT_CMD: "scan-pull-request"  
  JF_GIT_BASE_BRANCH: $CI_MERGE_REQUEST_TARGET_BRANCH_NAME  
  # Repository scanning is triggered by any push to the default branch.  
  # If you'd like a different branch to be scanned, replace  
  # $CI_DEFAULT_BRANCH in the line below with the name of the branch, wrapped  
  # with quotes ("").  
- if: $CI_COMMIT_BRANCH == $CI_DEFAULT_BRANCH || $CI_PIPELINE_SOURCE ==  
  "schedule"  
variables:  
  FROGBOT_CMD: "scan-repository"  
  JF_GIT_BASE_BRANCH: $CI_COMMIT_BRANCH  
variables:  
# [Mandatory]  
# JFrog platform URL (This functionality requires version 3.29.0 or above  
# of Xray)  
JF_URL: $JF_URL  
# [Mandatory if JF_USER and JF_PASSWORD are not provided]  
# JFrog access token with 'read' permissions for Xray  
JF_ACCESS_TOKEN: $JF_ACCESS_TOKEN  
# [Mandatory if JF_ACCESS_TOKEN is not provided]  
# JFrog user and password with 'read' permissions for Xray  
# JF_USER: $JF_USER  
# JF_PASSWORD: $JF_PASSWORD  
# [Mandatory]  
# GitLab access token. Ensure the token has the following permissions,  
# depending on your GitLab deployment type:  
# Self hosted: api, read_api, read_user, read_repository.  
# Cloud: api, read_api, read_repository  
JF_GIT_TOKEN: $USER_TOKEN  
# Predefined GitLab variables. There's no need to set them.  
JF_GIT_PROVIDER: gitlab  
JF_GIT_OWNER: $CI_PROJECT_NAMESPACE  
JF_GIT_REPO: $CI_PROJECT_NAME  
JF_GIT_PULL_REQUEST_ID: $CI_MERGE_REQUEST_IID  
# [Optional, default: https://gitlab.com]  
# API endpoint to GitLab  
# JF_GIT_API_ENDPOINT: https://gitlab.example.com  
# [Optional]  
# By default, the Frogbot workflows download the Frogbot executable as  
# well as other tools
```

```
# needed from https://releases.jfrog.io
# If the machine that runs Frogbot has no access to the internet, follow
these steps to allow the
# executable to be downloaded from an Artifactory instance, which the
machine has access to:
#
# 1. Login to the Artifactory UI, with a user who has admin credentials.
# 2. Create a Remote Repository with the following properties set.
# Under the 'Basic' tab:
# Package Type: Generic
# URL: https://releases.jfrog.io
# Under the 'Advanced' tab:
# Uncheck the 'Store Artifacts Locally' option
# 3. Set the value of the 'JF_RELEASES_REPO' variable with the Repository
Key you created.
# JF_RELEASES_REPO: ""
# [Optional]
# Configure the SMTP server to enable Frogbot to send emails with detected
secrets in pull request scans.
# SMTP server URL including should the relevant port: (Example:
smtp.server.com:8080)
# JF_SMTP_SERVER: ""
# [Mandatory if JF_SMTP_SERVER is set]
# The username required for authenticating with the SMTP server.
# JF_SMTP_USER: ""
# [Mandatory if JF_SMTP_SERVER is set]
# The password associated with the username required for authentication
with the SMTP server.
# JF_SMTP_PASSWORD: ""
#####
#
## If your project uses a 'frogbot-config.yml' file, you should define ##
## the following variables inside the file, instead of here. ##
#####
#
# [Mandatory if the two conditions below are met]
# 1. The project uses yarn 2, NuGet, or .NET to download its dependencies
# 2. The `installCommand` variable isn't set in your frogbot-config.yml
file.
#
# The command that installs the project dependencies (e.g "nuget restore")
JF_INSTALL_DEPS_CMD: ""
# [Optional, default: "."]
# Relative path to the root of the project in the Git repository
# JF_WORKING_DIR: path/to/project/dir
# [Default: "*.git*;*node_modules*;*target*;*venv*;*test*"]
# List of exclusion patterns (utilizing wildcards) for excluding paths in
the source code of the Git repository during SCA scans.
# JF_PATH_EXCLUSIONS: "*.git*;*node_modules*;*target*;*venv*;*test*"
# [Optional]
# Xray Watches. Learn more about them here:
https://www.jfrog.com/confluence/display/JFROG/Configuring+Xray+Watches
# JF_WATCHES: <watch-1>,<watch-2>...<watch-n>
# [Optional]
```

```
# JFrog project. Learn more about it here:
https://www.jfrog.com/confluence/display/JFROG/Projects
# JF_PROJECT: <project-key>
# [Optional, default: "FALSE"]
# Displays all existing vulnerabilities, including the ones that were
added by the pull request.
# JF_INCLUDE_ALL_VULNERABILITIES: "TRUE"
# [Optional, default: "FALSE"]
# When adding new comments on pull requests, keep old comments that were
added by previous scans.
# JF_AVOID_PREVIOUS_PR_COMMENTS_DELETION: "TRUE"
# [Optional, default: "TRUE"]
# Fails the Frogbot task if any security issue is found.
# JF_FAIL: "FALSE"
# [Optional]
# Relative path to a Pip requirements.txt file. If not set, the Python
project's dependencies are determined and scanned using the project
setup.py file.
# JF_REQUIREMENTS_FILE: ""
# [Optional, Default: "TRUE"]
# Use Gradle wrapper.
# JF_USE_WRAPPER: "FALSE"
# [Optional]
# Frogbot will download the project dependencies if they're not cached
locally. To download the
# dependencies from a virtual repository in Artifactory, set the name of
the repository. There's no
# need to set this value, if it is set in the frogbot-config.yml file.
# JF_DEPS_REPO: ""
# [Optional]
# Template for the branch name generated by Frogbot when creating pull
requests with fixes.
# The template must include ${BRANCH_NAME_HASH}, to ensure that the
generated branch name is unique.
# The template can optionally include the ${IMPACTED_PACKAGE} and
${FIX_VERSION} variables.
# JF_BRANCH_NAME_TEMPLATE:
"frogbot-${IMPACTED_PACKAGE}-${BRANCH_NAME_HASH}"
# [Optional]
# Template for the commit message generated by Frogbot when creating pull
requests with fixes
# The template can optionally include the ${IMPACTED_PACKAGE} and
${FIX_VERSION} variables.
# JF_COMMIT_MESSAGE_TEMPLATE: "Upgrade ${IMPACTED_PACKAGE} to
${FIX_VERSION}"
# [Optional]
# Template for the pull request title generated by Frogbot when creating
pull requests with fixes.
# The template can optionally include the ${IMPACTED_PACKAGE} and
${FIX_VERSION} variables.
# JF_PULL_REQUEST_TITLE_TEMPLATE: "[🐸 Frogbot] Upgrade
${IMPACTED_PACKAGE} to ${FIX_VERSION}"
# [Optional, Default: "FALSE"]
# If TRUE, Frogbot creates a single pull request with all the fixes.
```

```

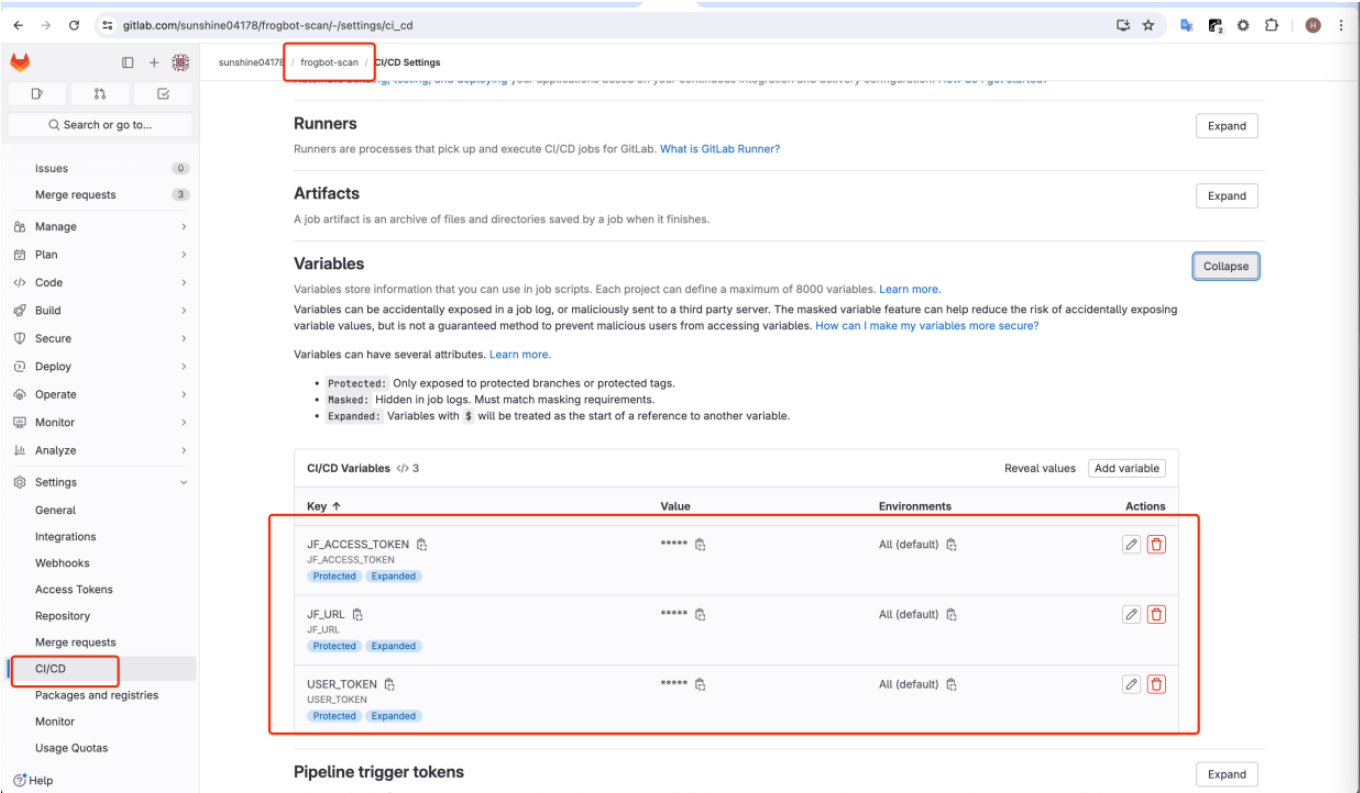
# If FALSE, Frogbot creates a separate pull request for each fix.
# JF_GIT_AGGREGATE_FIXES: "FALSE"
# [Optional, Default: "FALSE"]
# Handle vulnerabilities with fix versions only
# JF_FIXABLE_ONLY: "TRUE"
# [Optional]
# Set the minimum severity for vulnerabilities that should be fixed and
commented on in pull requests
# The following values are accepted: Low, Medium, High, or Critical
# JF_MIN_SEVERITY: ""
# [Optional, Default: eco-system+frogbot@jfrog.com]
# Set the email of the commit author
# JF_GIT_EMAIL_AUTHOR: ""
# [Optional]
# List of comma-separated(,) email addresses to receive email
notifications about secrets
# detected during pull request scanning. The notification is also sent to
the email set
# in the committer git profile regardless of whether this variable is set
or not.
# JF_EMAIL_RECEIVERS: ""
# [Optional]
# Set the list of allowed licenses
# The full list of licenses can be found in:
# https://github.com/jfrog/frogbot/blob/master/docs/licenses.md
# JF_ALLOWED_LICENSES: "MIT, Apache-2.0"
# [Optional]
# Avoid adding extra info to pull request comments. that isn't related to
the scan findings.
# JF_AVOID_EXTRA_MESSAGES: "TRUE"
# [Optional]
# Add a title to pull request comments generated by Frogbot.
# JF_PR_COMMENT_TITLE: ""
script:
# For Linux / MacOS runner:
- |
getFrogbotScriptPath=$(if [ -z "$JF_RELEASES_REPO" ]; then echo
"https://releases.jfrog.io"; else echo
"${JF_URL}/artifactory/${JF_RELEASES_REPO}"; fi)
curl -fLg
"$getFrogbotScriptPath/artifactory/frogbot/v2/[RELEASE]/getFrogbot.sh" |
sh
./frogbot ${FROGBOT_CMD}
# For Windows runner:
#
# - $getFrogbotScriptPath = $(if
([string]::IsNullOrEmpty($env:JF_RELEASES_REPO)) {
"https://releases.jfrog.io" } else {
"($env:JF_URL)/artifactory/($env:JF_RELEASES_REPO)" })
# - Invoke-WebRequest -Uri
"$getFrogbotScriptPath/artifactory/frogbot/v2/[RELEASE]/getFrogbot.sh" -
UseBasicParsing | ForEach-Object { & $_.Content }
# - .\frogbot ${FROGBOT_CMD}

```

4.commit code

```
git add .
git commit 'init'
git push
```

5.config Variables



6.Run CI

gitlab.com/sunshine04178/frogbot-scan/-/jobs

sunshine04178 / frogbot-scan / Jobs

Filter jobs

Status	Job	Pipeline	Coverage
Manual	#6803199386: frogbot-scan refs/merge-requests/3/h... manual	#1282615665 created by Stage: test	
Manual	#6803199068: frogbot-scan refs/merge-requests/2/h... manual	#1282615619 created by Stage: test	
Manual	#6803198186: frogbot-scan refs/merge-requests/1/h... manual	#1282615351 created by Stage: test	
Passed	#6803195693: frogbot-scan main 42 minutes ago	#1282614894 created by Stage: test	
Failed	#6802867917: frogbot-scan main 2 hours ago	#1282549953 created by Stage: test	
Passed	#6802860678: frogbot-scan main 2 hours ago	#1282546719 created by Stage: test	
Passed	#6802851229: frogbot-scan main 2 hours ago	#1282546719 created by Stage: test	
Failed	#6802844391: frogbot-scan	#1282546719 created	

gitlab.com/sunshine04178/frogbot-scan/-/jobs/6803195693

sunshine04178 / frogbot-scan / Jobs / #6803195693

Search job log

```
38 "Technology": "maven",
39 "WorkingDirectory": "/tmp/jfrog.cli.temp.-1715139256-4106193322",
40 "Descriptors": [
41   "/tmp/jfrog.cli.temp.-1715139256-4106193322/pom.xml",
42   "/tmp/jfrog.cli.temp.-1715139256-4106193322/multi1/pom.xml",
43   "/tmp/jfrog.cli.temp.-1715139256-4106193322/multi2/pom.xml",
44   "/tmp/jfrog.cli.temp.-1715139256-4106193322/multi3/pom.xml"
45 ]
46 }
47 ]
48 03:34:18 [Info] Running SCA scan for maven vulnerable dependencies in /tmp/jfrog.cli.temp.-1715139256-4106193322 directory...
49 03:34:18 [Info] Calculating Maven dependencies...
50 03:34:18 [Info] Downloading JFrog's Dependency from https://releases.jfrog.io/artifactory/xsc-gen-axe-analyzer-manager-local/v1/RELEASES
51 03:34:21 [Info] Extracting archive: /tmp/jfrog.cli.temp.-1715139256-2423421812/analyzerManager.zip to /tmp/jfrog.cli.temp.-1715139256-2
52 03:34:27 [Info] Scanning 19 maven dependencies...
53 03:34:27 [Info] Waiting for scan to complete on JFrog Xray...
54 03:34:39 [Info] Running applicability scanning...
55 03:34:44 [Info] Running secrets scanning...
56 03:34:48 [Info] Found 0 secrets
57 03:34:48 [Info] Running IaC scanning...
58 03:34:50 [Warn] got unsupported operating system error from analyzer manager
59 03:34:50 [Info] Running SAST scanning...
60 03:34:52 [Warn] got unsupported operating system error from analyzer manager
61 03:34:52 [Info] Xray scan completed
62 03:35:05 [Info] Creating Pull Request from: frogbot-commons-io_commons-io-9037b9584f8cdf16bb64ec557e1488c5 to: main
63 03:35:05 [Info] Created Pull Request updating dependency 'commons-io:commons-io' to version '2.7'
64 03:35:10 [Info] Creating Pull Request from: frogbot-org.codehaus.plexus-plexus-utils-9655d1ea6a64aae0c5f0e4d252f906b to: main
65 03:35:10 [Info] Created Pull Request updating dependency 'org.codehaus.plexus:plexus-utils' to version '3.0.24'
66 03:35:15 [Info] Creating Pull Request from: frogbot-org.apache.commons-commons-email-bd8cfa418f75f3f9dcb1e95d86584a29 to: main
67 03:35:16 [Info] Created Pull Request updating dependency 'org.apache.commons:commons-email' to version '1.5'
68 03:35:17 [Info] Frogbot "scan-repository" command finished successfully
69 Cleaning up project directory and file based variables
70 Job succeeded
```

Duration: 1 minute 29 seconds
Finished: 43 minutes ago
Queued: 0 seconds
Timeout: 1h (from project)
Runner: #12270807 (l1aDqxSn) 1-blue.saas-linux-small-amd64.runners-manager.gitlab.com/default

Commit d243b2f6
maven image

Pipeline #1282614894 Passed for main
test

Related jobs
frogbot-scan

7.Check result

7 / 8

