

Carnegie Mellon University

Software Engineering Institute

Copyright 2023 Carnegie Mellon University.

This material is based upon work funded and supported by the Department of Defense under Contract No. FA8702-15-D-0002 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center.

The view, opinions, and/or findings contained in this material are those of the author(s) and should not be construed as an official Government position, policy, or decision, unless designated by other documentation.

References herein to any specific commercial product, process, or service by trade name, trade mark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by Carnegie Mellon University or its Software Engineering Institute.

NO WARRANTY. THIS CARNEGIE MELLON UNIVERSITY AND SOFTWARE ENGINEERING INSTITUTE MATERIAL IS FURNISHED ON AN "AS-IS" BASIS. CARNEGIE MELLON UNIVERSITY MAKES NO WARRANTIES OF ANY KIND, EITHER EXPRESSED OR IMPLIED, AS TO ANY MATTER INCLUDING, BUT NOT LIMITED TO, WARRANTY OF FITNESS FOR PURPOSE OR MERCHANTABILITY, EXCLUSIVITY, OR RESULTS OBTAINED FROM USE OF THE MATERIAL. CARNEGIE MELLON UNIVERSITY DOES NOT MAKE ANY WARRANTY OF ANY KIND WITH RESPECT TO FREEDOM FROM PATENT, TRADEMARK, OR COPYRIGHT INFRINGEMENT.

[DISTRIBUTION STATEMENT A] This material has been approved for public release and unlimited distribution. Please see Copyright notice for non-US Government use and distribution.

Internal use:* Permission to reproduce this material and to prepare derivative works from this material for internal use is granted, provided the copyright and "No Warranty" statements are included with all reproductions and derivative works.

External use:* This material may be reproduced in its entirety, without modification, and freely distributed in written or electronic form without requesting formal permission. Permission is required for any other external and/or commercial use. Requests for permission should be directed to the Software Engineering Institute at permission@sei.cmu.edu.

* These restrictions do not apply to U.S. government entities.

DM23-0952

Smoke Tests

Admin Test

Using environment-dependent configuration, establish that Observer(s) are sending messages to the broker. You may view the same output as below from the container's STDOUT logs. We show the same output using an interactive session. You may do the same or by using a scheduled task:

```
[I] vcaaron@mac-bigredapple ~/P/d/gitlab_agent (master)> cargo run --bin gitlab_projects
Finished dev [unoptimized + debuginfo] target(s) in 0.27s
Running `target/debug/gitlab_projects`
[*] lock file created. Proceed.
connecting to: amqps://ec2-52-204-150-74.compute-1.amazonaws.com:5671/%2f?auth_mechanism=external
Retrieving projects
https://gitlab.sandbox.labz.s-box.org/api/v4/projects
```

Figure 1- Run `gitlab_projects` binary

```
Running `target/debug/gitlab_projects`
[*] lock file created. Proceed.
connecting to: amqps://ec2-52-204-150-74.compute-1.amazonaws.com:5671/%2f?auth_mechanism=external
Retrieving projects
https://gitlab.sandbox.labz.s-box.org/api/v4/projects
https://gitlab.sandbox.labz.s-box.org/api/v4/projects/300/users
https://gitlab.sandbox.labz.s-box.org/api/v4/projects/300/runners
https://gitlab.sandbox.labz.s-box.org/api/v4/projects/300/pipelines
https://gitlab.sandbox.labz.s-box.org/api/v4/projects/299/users
https://gitlab.sandbox.labz.s-box.org/api/v4/projects/299/runners
https://gitlab.sandbox.labz.s-box.org/api/v4/projects/299/pipelines
https://gitlab.sandbox.labz.s-box.org/api/v4/projects/298/users
https://gitlab.sandbox.labz.s-box.org/api/v4/projects/298/runners
https://gitlab.sandbox.labz.s-box.org/api/v4/projects/298/pipelines
```

Figure 2- Output

Next, use `kubectl` to exec into the RabbitMQ container (or use Docker if you're running a local instance). Use "`rabbitmqctl`" from the interactive shell to observe messages being queued:

```
ubuntu@ec2-52-204-150-74:~$ sudo rabbitmqctl list_queues
Timeout: 60.0 seconds ...
Listing queues for vhost / ...
name      messages
gitlab_projects 2664
```

Figure 3- RabbitMQ's `rabbitmqctl` tool output

Start the GitLab Consumer. Establish that Consumer(s) are receiving messages from the broker and the queues are depleting:

```
[I] vcaaron@mac-bigredapple ~/P/d/gitlab_agent (master) [SIGINT]> cargo run --bin projects_consumer
Finished dev [unoptimized + debuginfo] target(s) in 0.61s
Running `target/debug/projects_consumer`
connecting to: amqps://ec2-52-204-150-74.compute-1.amazonaws.com:5671/%2f?auth_mechanism=external
[*] waiting to consume
[*] Connected to neo4j
[*] Transaction Committed
[*] Transaction Committed
```

Figure 4 - Running the projects_consumer binary

```
ubuntu@ec2-52-204-150-74:~$ sudo rabbitmqctl list_queues
Timeout: 60.0 seconds ...
Listing queues for vhost / ...
name      messages
gitlab_projects 0
```

Figure 5 - Observing the empty queues

Observe nodes being created in Neo4J Browser. Enter your Cypher query into the query editor/input box. The simple Cypher query we are using here is:

```
MATCH (n:GitlabProject) RETURN n LIMIT 25
```

```
MATCH (n:GitlabProject) RETURN n LIMIT 25
```

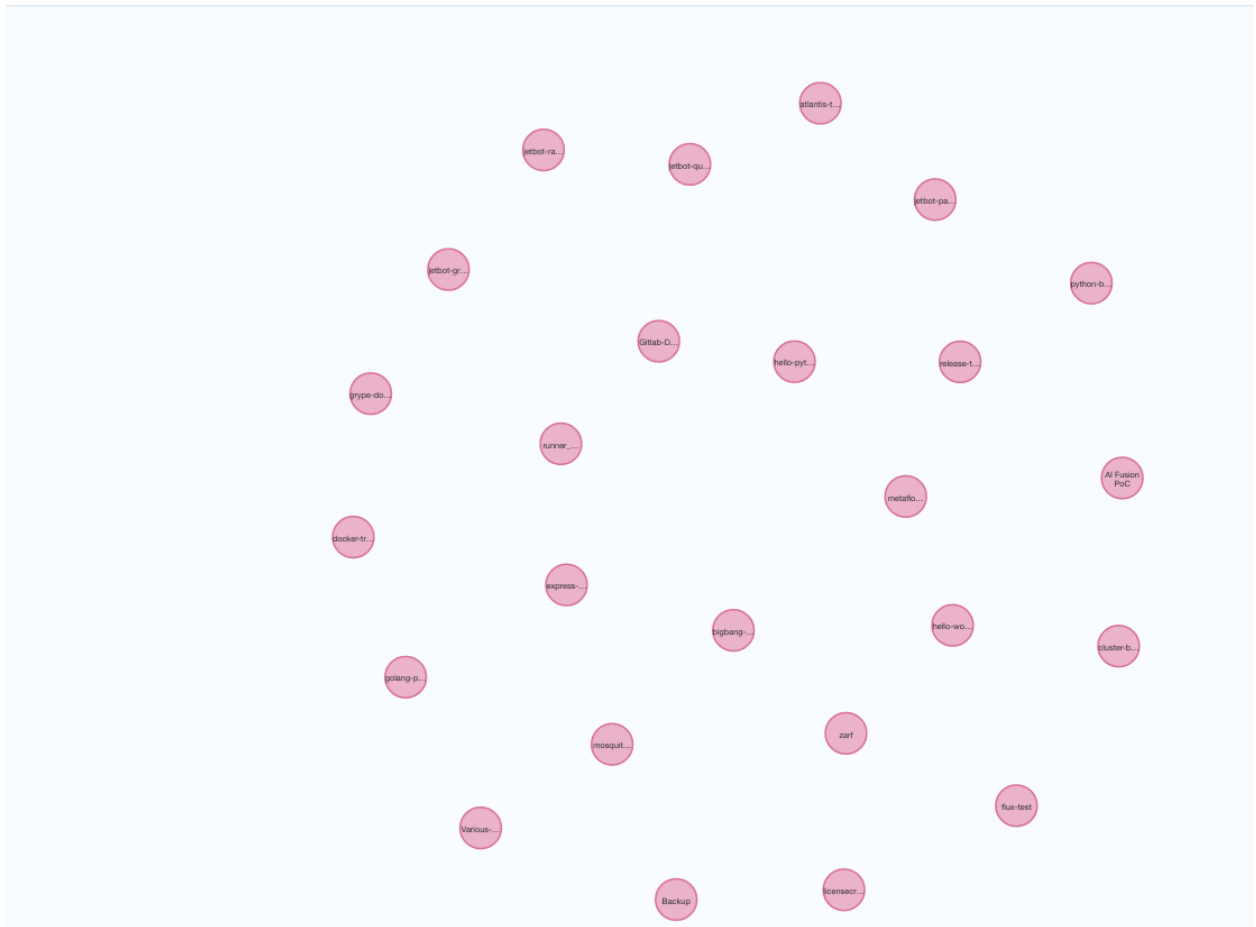


Figure 6 - Match created GitLab Project nodes

Note: Some queries will be logged to standard out by the consumer. An absence of output does not necessarily mean that queries are not being executed, however. It is best to confirm queries are being executed by manually querying neo4j for a desired node. You can do this by constraining your query, for example, by the name of the node you are creating. The query above might then be:

```
MATCH (n:GitlabProject {Name: "MyNewProject"}) RETURN n
```

The new nodes will be created, based on the observation interval chosen in the scheduler. For example, for GitLab Projects, the scheduler default we set it is every 30 seconds. This interval may differ, depending upon your Admin's configuration.

User Test

Establish connection to GitLab

Log into your Gitlab instance as a user with privileges to create objects such as a new project or user OR confirm your Gitlab instance's API is reachable by interacting with it's REST API and providing the same token you plan to use for Polar via the PRIVATE-TOKEN header. ([See Gitlab's documentation for more details on how to do this](#))

Below is an example of trying the API with Postman:

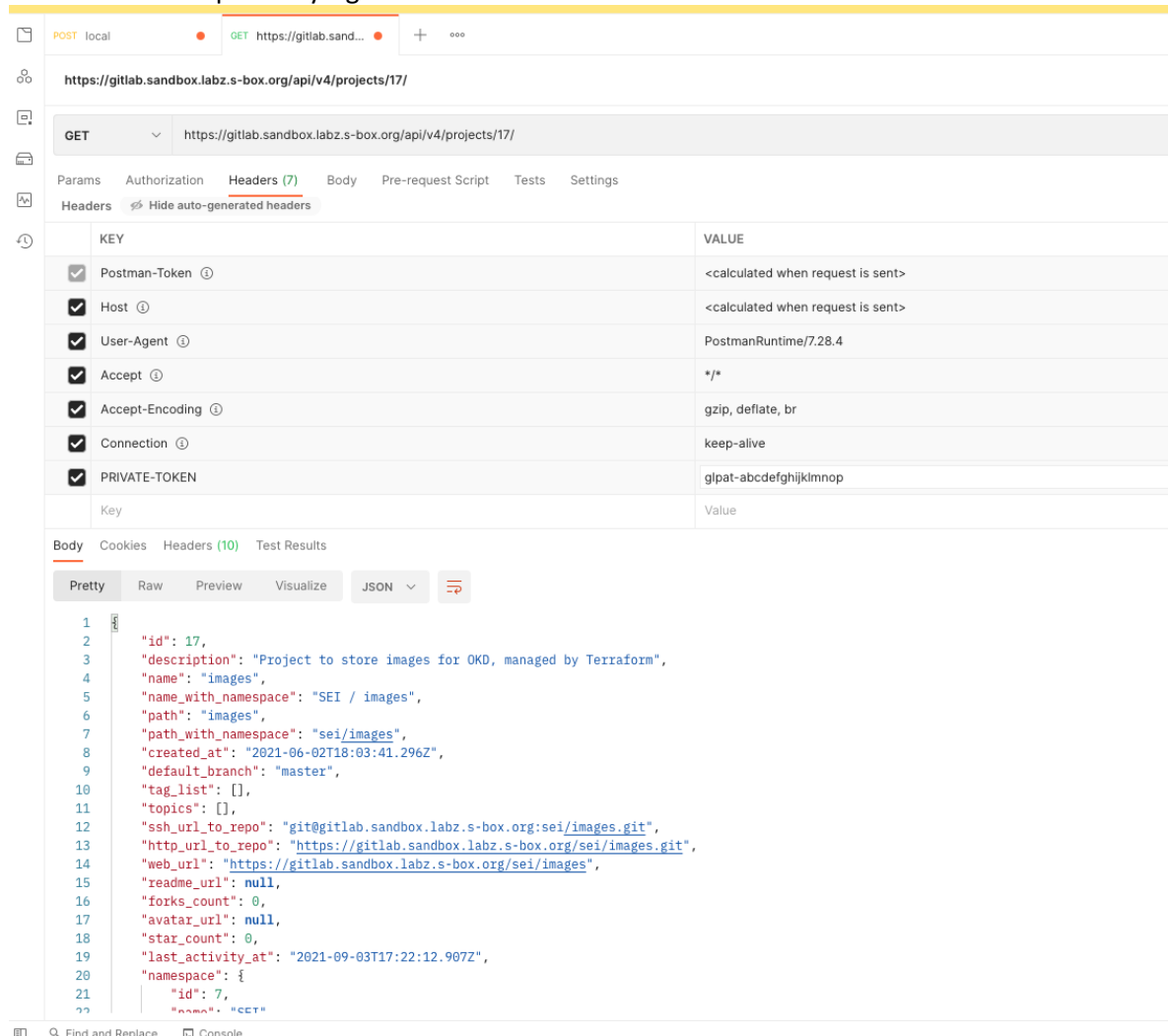


Figure 7 - Example of setting a GET request to GitLab API for a specific project

Create some items. Using either the Gitlab REST API or the GUI, create some Gitlab resource such as a new project:

Your work / Projects / New project / Create blank project

Create blank project

Create a blank project to store your files, plan your work, and collaborate on code, among other things.

Project name

Must start with a lowercase or uppercase letter, digit, emoji, or underscore. Can also contain dots, pluses, dashes, or spaces.

Project URL

 Pick a group or namespace / **Project slug**

Visibility Level ?

☒ Private
Project access must be granted explicitly to each user. If this project is part of a group, access is granted to members of the group.

☐ Internal
The project can be accessed by any logged in user except external users.

☐ Public
The project can be accessed without any authentication.

Project Configuration

☒ Initialize repository with a README
Allows you to immediately clone this project's repository. Skip this if you plan to push up an existing repository.

☐ Enable Static Application Security Testing (SAST)
Analyze your source code for known security vulnerabilities. [Learn more.](#)

UNCLASSIFIED - NO CUI

Figure 8 - Creating a new project in GitLab

Open Neo4J Browser and observe that the new items appear in the graph. After triggering the Gitlab observer for that resource (i.e., a project or user), and consuming the messages from the rabbitmq instance, you can query neo4j for that resource, reliably by that resource's id. A query for example could be:

```
MATCH (n:GitlabProject) where n.project_id = '17' return n
```

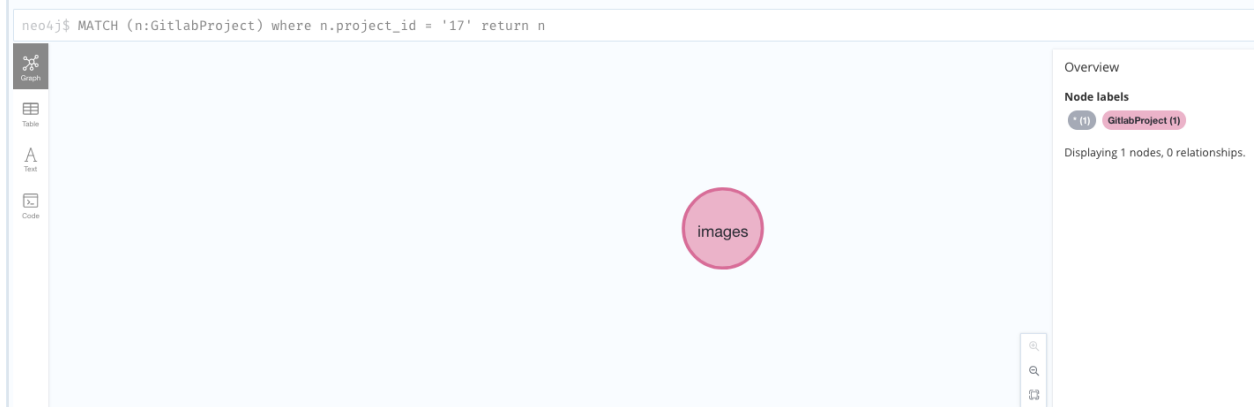


Figure 9 - Neo4J Browser - filtering query results for a specific project ID