CloudEvents plugin for Jenkins

Google Summer of Code Program 2021 Project Proposal

Nivedita Prasad
niveditaprasad81@gmail.com
Github handle

Project Abstract. This project idea proposes to implement a Jenkins plugin that extends the Jenkins to make cloudevents both discoverable and subscribable. Users should be able to set a global configuration to allow users to subscribe and discover cloudevents.

Project Description.

- ★ As the CI/CD world is moving more towards interoperability between multiple platforms like GitHub and GitLab, Jenkins should also be compatible with the same interoperability standards.
- ★ Some of these standards with respect to communication between different CI/CD platforms are put forth by the cloudevents specification. This spec outlines the structure of cloudevents, which are produced or consumed by entities which support it, hence making those entities compatible with other CI/CD platforms which also support them allowing them to work together.
- ★ Implementation
    - ○ **Global Plugin Configuration**

      In the Global Plugin Configuration add endpoints from which the events can be listened on. Other infrastructures which employ events can use this as the source for Jenkins events.

      Subscription points can be added by creating multiple sources on a single Jenkins instance.

      Ex- We can consider an asynch Pipeline step.

      Step-1: Enable Async Support

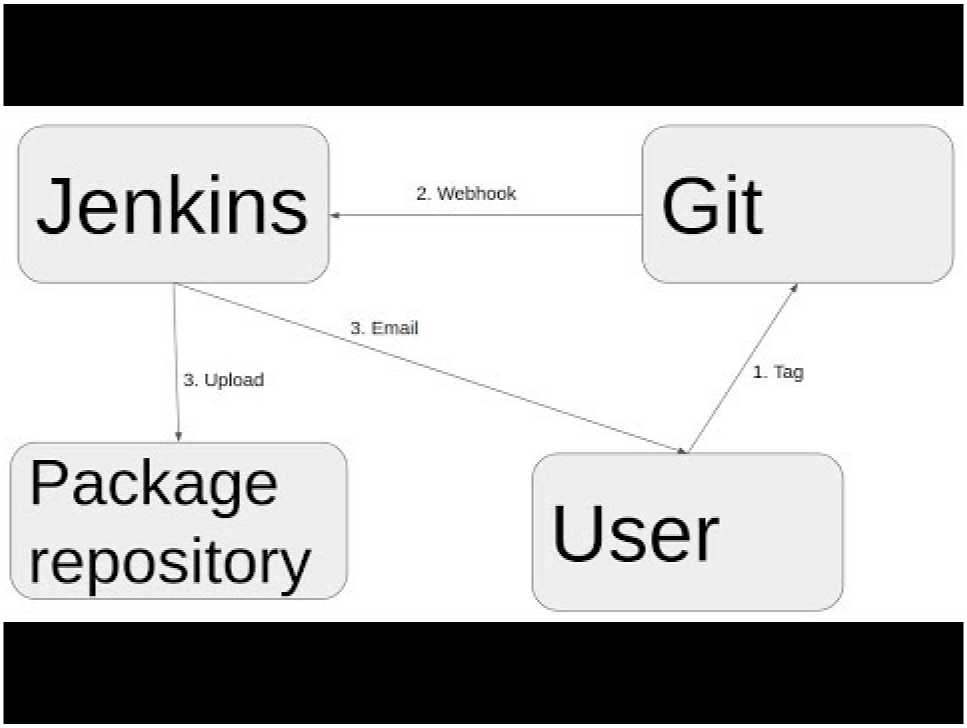
      Step-2: Add @Async Annotation to a method

      Step:3 Executor (Customize of Default)

    - ○ **Enabling CloudEvents for a Job**
        - ■ Job execution (when a build starts and finishes)
        - ■ Job step execution (when each step in the job starts and finishes)
        - ■ Job queue (when a job enters or changes state in the job queue)
        - ■ SCM Checkout (when the job checks out files from source control)

Resource	Event	Event Type
Job Execution	Started	`io.jenkins.event.job.started.v1`
	Running	`io.jenkins.event.job.running.v1`
	Succeeded	`io.jenkins.event.job.successful.v1`
	Failed	`io.jenkins.event.job.failed.v1`

○ **Trigger Job in CloudEvent**
Trigger running of a Job after listening on a specific cloudevent. Inspiration can be taken from the generic webhook-trigger plugin for implementation.



Example: It would work as Knative Eventing like Cloud Event receivers work - use an SDK which just listens on an HTTP port for the incoming events and decodes to an object, e.g. https://github.com/knative/eventing-sources/blob/master/cmd/event_display/main.go

Just import the CloudEvents JS SDK to handle the HTTP transport and proxy the events to the handler functions.
https://github.com/cloudevents/sdk-javascript

Project Deliverables

Note: The dates in the GSoC Timeline takes precedence over this document

- GSoC Timeline: https://developers.google.com/open-source/gsoc/timeline

- May 17 - June 7 (Community bonding) - To utilize the time of this phase I will ensure that I am connected with the mentors and community through means of mailing lists and IRC channel. I will discuss more about the project idea with a mentor.

- June 7 (Coding begins) - During this period I will start working on Jenkins plugin development and on Enabling CloudEvents for a Job.

- July 12 - July 16 (Phase 1 Evaluation) - During this phase, I will ensure that Enabling CloudEvents for a Job work properly.

- August 16 - August 23 (Final Evaluation) - I will work on how Jenkins should listen to the CloudEvents mainly on listening to the CloudEvents

- August 31 (Final Results Announced) - The time period between the final evaluation and result announcement will be utilized to make some changes in the user documentation for a better understanding, of how global configuration allows users to subscribe and discover cloudevents.

- Post GSoC - After the completion of my GSOC project I will ensure to remain connected with the community, work on the plugin to make it more interoperable, and also try to make it interoperable on other platforms, explore other areas of the Jenkins project.

Proposed Schedule

March 29 - April 13 (Application Period)

During this period I will familiarize myself with the project idea. I will read the plugin doc of Jenkins and also I read about CloudEvents that how to integrate GlobalPlugin into CloudEvents. I had watched the meeting of Cloud Native Sig Discussion on CloudEvents to get a more clear idea of workflow.

April 14 - May 17 (Acceptance Waiting Period)

During this period I will continue with the same step in the application period but in a broad manner, I will start making plugin development from scratch which creates CloudEvent. I also start studying other plugins like Prometheus and GitLab that how their codebase work.

May 17 - June 7 (Community Bonding Period)

During this period I will start discussing my idea regarding codebase. And also I discuss my idea regarding Global Plugin Configuration which I take from Prometheus or GitLab PLugin. So that my idea gets more clear regarding implementation and codebase.

June 7 - July 16 (Coding period 1)

I will start working on Enabling CloudEvents for a Job and on Global Plugin Configuration add endpoints from which the events can listen

July 16 - Aug 23 (Coding period 2)

I will continue working on the plugin to add support for trigger running of a Job after listening to a specific cloudevent on an endpoint that is being set by the user.

Future Improvements

CloudEvent platform is totally new for me, and after making the Jenkins Plugin, I will continue to work on this to improve features and to make it more user-friendly.

Continued Involvement

In the future, I will continue to work on Jenkins issues. Mainly, I would like to add more features to CloudEvent Jenkins Plugin.

Conflict of Interests or Commitment

State whether you may have any conflicts that may prevent you from working at full potential or may not allow you to contribute code you develop during GSoC. Possible reasons would include participating in a summer internship, or a scholarship/fellowship at your school that restricts your ability to claim a copyright on your work. Although working or being funded by your school can be OK with GSoC, we need to know about your other commitments, and also need to be certain that the code you write can be contributed by you to Jenkins without a claim by your employer or University.

If you are an accepted student in the United States of America on an F1 visa, you must obtain an authorization before you can participate. Discovering this after you've been accepted results in the slot being lost.

Major Challenges foreseen

- Listening to CloudEvents in Jenkins seems to be a major challenge because in JAVA SDK receiving and emitting events codebase is not available like JS

References

- Cloudevents
- CloudEvents SDK for Java
- Jenkins.io
- Cloud-Native Meeting
- Generic-Webhook Trigger doc

Relevant Background Experience

- I had watched the meeting recording and also CNCF talks on CloudEvent to understand it working and to get an idea. I also check the codebase of GitLab, Knative for an implementation idea. Currently, working on CloudEvent JAVA SDK codebase to implement.

Personal

I'm a sophomore student pursuing a BTech in Computer Science and Engineering at Siliguri Institute of Technology, WestBengal, India. I'm a frontend developer and DevOps /Kubernetes enthusiast, also love to contribute to the Documentation of Open-Source Organization. I am eager to explore different areas of technology. I started learning networking, DevOps, Kubernetes in mid-November in 2020. I'm interested in Java-client in K8's although my primary language is Java. On the way to learning K8's, I came across CI/CD, Jenkins.
I also participated in hackathons like MLH, DEVCOV and I also love to give talks on OpenSouce in the community.

Availability and commitments

I will be available for the entire GSoC Period. Although, my semester examination will commence in the week of July 2021 and will take utmost up to 5 days.

Experience

Free Software Experience/Contributions (optional):

- Contributed to Kubernetes/Community: https://github.com/kubernetes/community/pull/5399

Language Skill Set

- Java(Intermediate)
- Python
- Git

Reference Links and Web URLs (optional):

- Twitter
- Github
- LinkedIn