2020. First coding phase demos. Presentation abstracts and links

Part 1, Aug 27, 2PM UTC

Git Plugin Performance Improvements

Presenter: Rishabh Budhouliya

Project abstract: The Git Plugin Performance Improvement Project aims to improve the performance of the git plugin. To do so, we've added a JMH benchmarks module inside the git client plugin which would enable a developer to benchmark any git operation. For the users, we've added a new functionality called the GitToolChooser, which will recommend an optimal git implementation on the basis of the size of the remote repository. Furthermore, we will present some data on the difference in performance due to the improvements throughout the project.

Talk abstract: The talk will be a demonstration of the newly added performance improvement features in the git plugin with visualisation of key metrics (for example: execution time) to show the gain in performance.

References:

- Project page: Git Plugin Performance Improvements
- Benchmarks and results: https://github.com/jenkinsci/git-client-plugin/pull/521
- Github Repository: https://github.com/jenkinsci/git-plugin and https://github.com/jenkinsci/git-client-plugin

GitHub Checks API for Jenkins Plugins

Presenter: Kezhi Xiong

Project abstract: The GitHub Checks API allows developers to report the CI integrations' detailed information rather than the binary pass/fail build status on GitHub pages. This project is about implementing this API as a new Jenkins plugin. By consuming this API, other plugins can easily create GitHub checks. Thus, any information during the Jenkins process like warnings, summaries, and durations can be directly shown on GitHub pages.

Talk abstract: In this talk, I'll go through what GitHub check run is, how it could benefit the developers. Then, I'll present how Jenkins plugins could integrate with that, and how Warnings NG and Code Coverage API plugins integrate with GitHub Checks.

References:

- Project page: https://www.jenkins.io/projects/gsoc/2020/projects/github-checks
- Checks API Plugin: https://plugins.jenkins.io/checks-api/
- GitHub Checks API Plugin: https://plugins.jenkins.io/github-checks/

Custom Jenkins distribution build service

Presenter: Sladyn Nunes

Project abstract: The main idea behind the project is to build a customizable jenkins distribution service that could be used to build tailor-made jenkins distributions. The service would provide users with a simple interface to customize the configuration, they want to build the instance with eg: plugins,jenkins version, docker image etc. Furthermore it would include a section for sharing community created distros so that users can find and download already built jenkins war/configuration files to use out of the box.

Talk abstract: We would be going through how to add plugins to the configuration and the various package configuration details that need to be entered so that the service can easily generate a packager-config.yml, essentially the first step of generating a fully usable jenkins.war, we would then go through the process of downloading a war file as well as taking a look at how one would go about accessing the community configurations.

References:

- Project page: https://www.jenkins.io/projects/gsoc/2020/projects/custom-jenkins-distribution-build-service
- Main repository: https://github.com/jenkinsci/custom-distribution-service

Part 2, Aug 28, 2PM UTC

Machine Learning Plugins for Data Science

Presenter: Loghi Perinpanayagam

Project abstract: The main goal of this project is integrating Machine Learning workflow including Data preprocessing, Model Training, Evaluation and Prediction with Jenkins build tasks. This plugin will be capable of executing code fragments via IPython kernel as currently supported by Jupyter Notebook. Kernels which are already installed can be configured for each build step and dumping visuals is an added feature in the plugin.

Talk abstract: Machine Learning has evolved rapidly in the software industry for recent years. Jenkins CD/CI can be a good practice to deliver a high reliable product in the end. Machine Learning plugin can be used to build Jupyter Notebooks and script files with proper kernel configurations.. In addition, the build wrappers could be used to convert Jupyter Notebooks to python/JSON and/or copy the files to the workspace for more actions.

This Machine Learning plugin will endeavour to satisfy the data science community together with the help of other plugins. Success of this plugin will definitely serve much benefits to the community and Jenkins.

References:

- Project page: https://www.jenkins.io/projects/gsoc/2020/projects/machine-learning
- Plugin page: https://plugins.jenkins.io/machine-learning/
- Repository: https://github.com/jenkinsci/machine-learning-plugin

Jenkins Windows Services: YAML Configuration Support

Presenter: Buddhika Chathuranga

Project abstract: On Windows machines, Jenkins server and client can be installed as Windows Services in order to get better robustness and manageability within the system. This is a functionality bundled into the Jenkins core directly. When installed as a service, Jenkins uses the Windows Service Wrapper executable (.NET, written in C#) which is being configured by XML config files. Currently, there are only a few configuration checks there (no XML Schema, limited validation, etc.), and it's often that the service wrapper is misconfigured by Jenkins users. In this project I update Windows Service Wrapper to support YAML files as configuration inputs and to introduce better configuration validation during the service installation and startup. Usage

of YAML should simplify configuration management in Jenkins, especially when automated tools are used.

Talk abstract: In the presentation I will talk about a brief description about Windows service wrapper. Then I will talk about project tasks. Under that first I will talk about YAML configuration support. Then about the new CLI and XML schema file. End of the talk I will demo on YAML configuration support and YAML verification with the JSON Schema.

References:

- Project page: https://www.jenkins.io/projects/gsoc/2020/projects/winsw-yaml-configs
- Repository: https://github.com/winsw/winsw
- YAML Support Documentation: https://github.com/winsw/winsw/blob/master/doc/yamlConfigFile.md

External Fingerprint Storage

Presenter: Sumit Sarin
Slides: link

Project abstract: File fingerprinting is a way to track which version of a file is being used by a job/build, making dependency tracking easy. The fingerprint engine of Jenkins can track usages of artifacts, credentials, files, etc. within the system. It does this by maintaining a local XML-based database. This leads to dependence on the physical disk of the Jenkins master. This project involved extending Jenkins core to support storing of fingerprints in an external storage, along with two reference implementations, backed by Redis and PostgreSQL respectively. Various functionalities like migration and cleanup were developed and released. Users can now use these plugins to externalize the storage of their fingerprints. This project was one step forward in developing a cloud native Jenkins.

Talk abstract: This presentation will start with a brief introduction about the fingerprinting engine in Jenkins and its use case. Then we will discuss the motivation behind externalizing these fingerprints. We will discuss the external fingerprint storage API built during the course of the project, and what features it allows the plugin developers to use. Then we will talk about the two reference implementations built by us during the project, backed by Redis and PostgreSQL. We will discuss fingerprint cleanup and migration strategies. A demo will be presented which will show the working of the plugins and their functionality in action. We will conclude the presentation with what potential future areas of improvement can be for this project where the community is more than welcome to contribute. And lastly a short Q&A with the developers behind this project.

References:

- Project page: External Fingerprint Storage
- Redis Fingerprint Storage Plugin: https://github.com/jenkinsci/redis-fingerprint-storage-plugin/
- PostgreSQL Fingerprint Storage Plugin: https://github.com/jenkinsci/postgresql-fingerprint-storage-plugin
- JEP: https://github.com/jenkinsci/jep/tree/master/jep/226
- Phase 1 Blog Post: https://www.jenkins.io/blog/2020/06/27/external-fingerprint-storage/
- Phase 2 Blog Post: https://www.jenkins.io/blog/2020/07/24/external-fingerprint-storage-phase-2/

Jenkins X: Consolidate Apps and Addons

Project abstract: The main aim of the project is to consolidate Apps and Addons inside Jenkins X to avoid confusion. We will do everything as an App and deprecate the use of the word Addon then migrate Addons to Apps via Helm chart, the Apps will includes system charts like Knative, Gloo, Istio, flagger, Prometheus and more, so only using the Apps framework to extend Jenkins X platform is the best solution at present.

Talk abstract: In this presentation, I will introduce the background of the project, as well as the solutions, and demonstrate the Apps I made.