

# A Comparative Analysis of the Rise of Git-Based Distributed Collaborative Hosting Platforms: Survey, Performance Test, and Comparison

Md. Salah Uddin<sup>1</sup>, Kazi Jahid Hasan<sup>2</sup> and Rifat Tasnima<sup>3</sup>

<sup>1</sup>Institute of Software Technology, Graz University of Technology, Austria

<sup>1,2&3</sup>Department of Multimedia and Creative Technology, Daffodil International University, Dhaka, Bangladesh

E-mail: salah.mct@diu.edu.bd, jahid.mct@diu.edu.bd

(Received 7 September 2023; Revised 30 October 2023, Accepted 25 November 2023; Available online 30 November 2023)

**Abstract** - With the widespread use of the Internet, cloud services have steadily become more prevalent worldwide. The Internet is a typical pattern of distributed network structure. Initially, the term “distributed collaboration” [1] was mostly used in the computer-related industries to describe how individuals may work together remotely to accomplish a common goal at various times and locations in the network era with improved mobility. The community of software developers is constantly searching for an essential collaborative platform for organizing software projects and collaborative learning. This research study offers a thorough examination of cloud-based Git repository hosting services, examining both their present situation and potential for the future. The article meticulously examines year-by-year upgrades and follows the release dates of major cloud-based Git hosting platforms to offer insights into how these platforms have changed over time. Market share information and the importance of the main companies are evaluated, providing insightful information on platform preferences among enterprises and developers. The study explores the complex ecosystems of Bitbucket Server [7], GitLab [6], GitHub [5], and Bitbucket, as well as Azure DevOps [9] and AWS CodeCommit [8]. The report presents well-informed projections regarding the future of these services based on meticulous data analysis and professional observations. Deeper interaction with DevOps pipelines, improved security, and expanded support for containerization and micro-services architectures are some of the trends that are anticipated.

**Keywords:** Git, Distributed Collaboration, GitHub, GitLab, Bitbucket, Cloud Platform

## I. INTRODUCTION

The use of cloud-based Git repository hosting services has emerged as a fundamental transition in the quickly changing world of software development. These services have completely changed how software is created, maintained, and distributed since they offer a platform for collaborative development, version control, and seamless deployment. A crucial part of the contemporary software development toolset, cloud-based Git repository hosting services are explored in-depth both in their current condition and their potential future.

The on-premises version control systems that were formerly commonplace in the software development industry have

given way to cloud-based platforms that offer scalability, dependability, and a wide range of functionality. These services are crucial in enabling teams to work efficiently in the age of rapid application development and continuous integration.

Traditional software development methods have been impacted by the proliferation of cloud-based Git repository hosting services. However, despite their rising popularity, there is a dearth of thorough study that examines the characteristics, historical progression, and potential applications of these services. This study seeks to close this information gap and give interested parties a better understanding of the topic. This study aims to bridge this knowledge gap and provide stakeholders with a deeper understanding the features, historical evolution, year-wise updates, current status, and future trends of these services. Furthermore, the research may indicate areas where additional innovation and development are required to improve the capabilities and security of these services in accordance with growing software development requirements.

## II. BACKGROUND STUDY

Cloud-based Git repository hosting services have become an essential part of contemporary software development methods in recent years. The way programmers collaborate on code has completely changed thanks to Git, a distributed version control system developed by Linus Torvalds in 2005. Git repository hosting services in the cloud expanded Git’s functionalities by giving software development team’s easy-to-use, scalable, and collaborative platforms.

Software development workflows have been completely reimaged by the possibilities offered by cloud-based Git repository hosting providers. These features include:

*Collaboration Tools:* By supporting concurrent development, code reviews, and bug tracking, these services allow developers to work together on code invisibly.

*Version Control:* Git uses its strong version control capabilities to keep track of changes made to code repositories, providing options for rollback and traceability.

*Continuous Integration/Continuous Deployment (CI/CD):* Automation of code testing, building, and deployment through integration with CI/CD pipelines speeds up software delivery and accuracy.

*Security:* Code repositories are protected by strong security methods, such as access restriction, authentication, and encryption.

*Scalability:* Platforms built on the cloud offer the scalability to manage projects of various sizes, from little startups to enormous corporations.

*Reliability:* Low downtime and low data loss are ensured by high availability and disaster recovery features.

*Third-party Integrations:* Productivity is increased through integration with a variety of development and project management technologies.

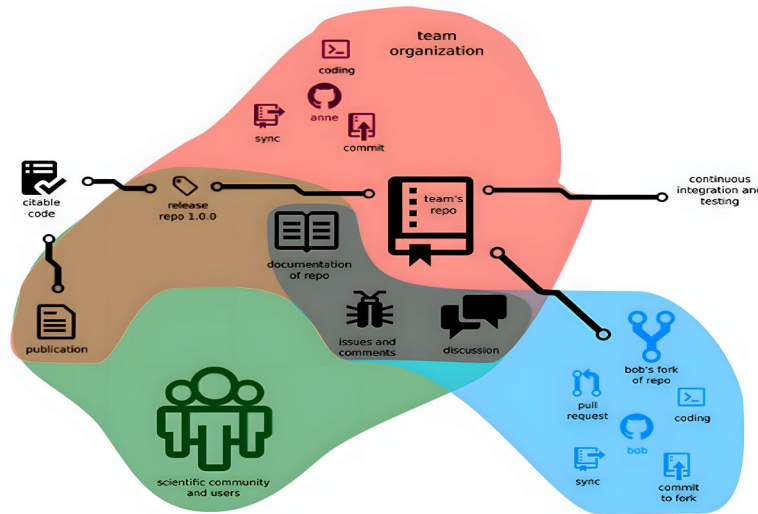


Fig. 1 The Structure of a Git based Project

### III. METHODOLOGY

This research employs a multifaceted methodology designed to thoroughly analyze cloud-based Git repository hosting services, encompassing their features, historical development, year by year updates, current status, and future trends. The subsequent research techniques has been used:

*A. Data Collection:* It has been verified to gather information on annual updates and significant feature upgrades using official documentation and release notes. We sent emails invit-ing developers, students, and the organization to the interviews. The interviews took place in person and over Skype and lasted between 10 and 20 minutes. The interviewer recorded the audio and took notes.

*B. Market Analysis:* Surveying businesses and developers to learn about their preferences and usage of various platforms.

*C. Data Analysis:* Comparative and statistical evaluation of cloud-based and on-premises features Git repository hosting options have been studied to look for trends, correlations, and patterns.

*D. Expert Interviews:* Interviewing experts and industry leaders in the fields of version control systems and software

development to gain insight about the existing environment and future developments for cloud-based Git hosting.

RQ1: How does Git hosting support software development?

RQ2: What are the motivations for and benefits of using Git hosting?

RQ3: What challenges are related to the use of Git hosting?

*E. Predictive Modeling:* Utilizing predictive modeling methods to project future trends and advances in cloud-based Git repository hosting services while taking into account elements like technology progress, market demands, and user preferences.

*F. Surveys and Questionnaires:* In order to collect qualitative and quantitative information on user satisfaction, is-sues, and expectations, survey and questionnaire distribution to developers, DevOps experts, and enterprises using cloud-based Git hosting is being undertaken.

RQ1: Which Git platform do you use to track your projects?

RQ2: What is the purpose to use Git hosting?

RQ3: Do you use new collaborating features: Branching and Forking?

RQ4: How do you handle naming branches and commits?

RQ5: How do you think Git hosting Automate your tasks?

RQ6: How do you think Git hosting make your code easily citable and cite source code?

RQ7: Do you use GitHub to Be Social: Follow and Watch?

*G. Ethical Considerations:* Ensuring ethical data gathering procedures, such as informed permission for surveys and interviews to preserve respondents’ identity and privacy.

*H. Validation:* Confirming the accuracy and dependability of the analysis by having the conclusions of the research validated by peer review and professional opinion.

**IV. EXPERIMENTAL RESULTS**

This study uses a thorough research methodology to provide a holistic overview of cloud-based Git repository hosting services, providing insightful analysis and predictions for the benefit of academics, industry professionals, and other stakeholders in the software development industry.

*A. Github:* The well-known social code sharing website GitHub uses the Git distributed version management system. GitHub was founded in 2008 by Tom Preston Werner, Chris Wanstrath, and PJ Hyett [5]. With over 83 million active users, GitHub is the world’s most popular Git repository hosting platform. It is wellknown for having a large development community and for providing a wide range of services such as code review, issue tracking, project management, CI/CD, security scanning, and more.

TABLE I YEAR-WISE UPDATE OF GITHUB

Year-Wise Updates		
	Year	Features
GitHub	2009	Mercurial repositories
	2010	Pull requests
	2011	Private repositories
	2012	GitHub Pages
	2013	GitHub Enterprise
	2014	GitHub Flow
	2015	GitHub Actions
	2016	GitHub Packages
	2017	GitHub Code spaces
	2018	GitHub Topics
	2019	GitHub Sponsors
	2020	GitHub Discussions
	2021	GitHub Copilot
	2022	GitHub Profiles
2023	GitHub Code spaces Beta	

*B. GitLab:* In 2011, Dmytro Zaporozhets launched The GitLab, an open source code sharing platform [6]. GitLab is a popular Git repository hosting platform recognized for its feature rich capabilities such as self-hosting and Code review, issue tracking, project management, CI/CD, and

more. It is also well known for its compatibility with other Atlassian products.

TABLE II YEAR-WISE UPDATE OF GITLAB

Year-Wise Updates		
	Year	Features
GitHub	2012	Private repositories
	2013	Self-hosting
	2014	Git Lab Pages
	2015	Git Lab Enterprise
	2016	GitLab Flow
	2017	GitLab CI/CD
	2018	GitLab Security scanning
	2019	GitLab Packages
	2020	GitLab Code spaces
	2021	GitLab Topics
	2022	GitLab Sponsors
	2023	GitLab Discussions

*C. Bitbucket:* Bitbucket, the renowned Git repository hosting service, integrates seamlessly with other Atlassian products. It is well known for its significant emphasis on security. Jesper Nhr founded GitHub in 2008 [7]. Bitbucket’s primary functions are code generation and code review. Bitbucket also provides a variety of features, such as merge checks, Git Large File Storage (LFS), two-step verification, bitbucket pipelines, IP whitelisting, and pull requests with code review and comments.

TABLE III YEAR-WISE UPDATE OF BITBUCKET

Year-Wise Updates		
	Year	Features
Bitbucket	2009	Private repositories
	2010	Mercurial repositories
	2011	Pull request
	2012	Bitbucket pipelines
	2013	Bitbucket pages
	2014	Bitbucket enterprise
	2015	Bitbucket flow
	2016	Bitbucket Codespaces
	2017	Bitbucket Topics
	2018	Bitbucket Sponsors
	2019	Bitbucket Discussions
	2020	Bitbucket pipelines Beta
	2021	Bitbucket Codespaces Beta
	2023	Bitbucket pipelines GA

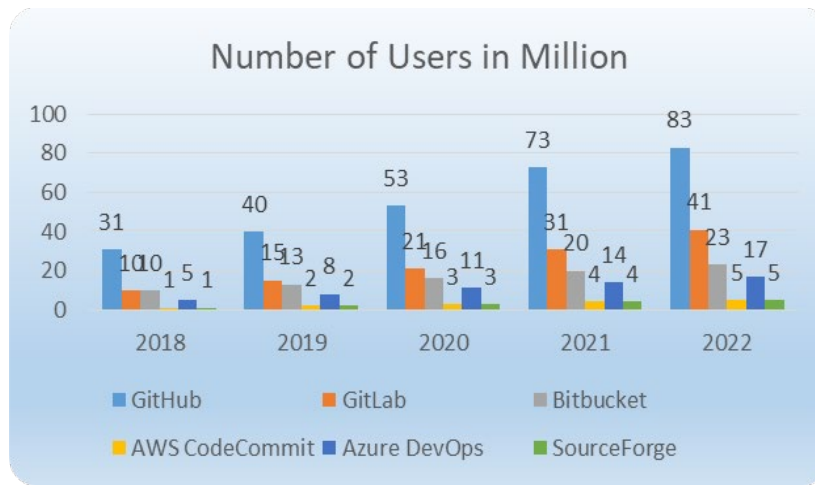


Fig. 2 User Analysis of Git hosting platforms

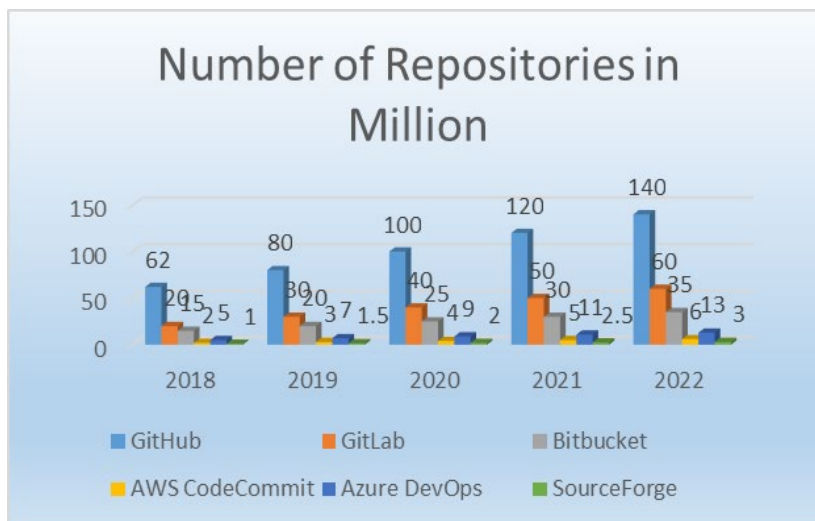


Fig. 3 User Repositories Analysis of Git hosting platforms

*D. User Retention Analysis:* 80% of Git repositories are idle, according to a Gitlab analysis from 2023. This means they have not been updated in at least six months. The numbers for the aforementioned systems (GitHub, GitLab, Bitbucket, AWS CodeCommit, Azure DevOps, and SourceForge) are likely to be similar. However, keep in mind that the number of inactive repositories on each platform will most likely vary depending on the site’s unique user base and the types of projects hosted on it. Platforms popular with open source projects, for example, may have a higher percentage of dormant repositories than platforms popular with commercial organizations.

Furthermore, from one platform to the next, the term “inactive repository” may be defined differently. For instance, some systems would only classify a repository as inactive if it hasn’t received an update in at least a year, while other platforms might classify it as inactive if it hasn’t received an update in at least two years. It is estimated that 60-70% of the repositories on the platforms you named are

not currently active after taking all of these considerations into account. This indicates that there are a huge number of dormant repositories on these networks.

For several reasons, inactive repositories can be problematic. They can first occupy priceless platform storage space. Second, they might make it challenging to locate live repositories. Third, they could make the platform appear less well-known than it actually is. There are several solutions available to the issue of dormant repositories.

Automatically deleting inactive repositories after a predetermined amount of time is one approach. Allowing users to designate repositories as inactive is an additional choice. This would enable the platform to keep track of the number of inactive repositories and take action to lower that number.

*E. Comparison Based on Features:* After analyzing survey results, interview we have found following data:

TABLE IV COMPARISON BETWEEN DIFFERENT HOSTING PLATFORMS BASED ON FEATURES

Features	GitHub	GitLab	Bitbucket
Assign Requests to Teammates	Yes	Yes	Yes
Attach Milestones, Projects, and Labels to Provide Context	Yes	Yes	No
Subscribe to be Notified When the Pull Request Changes	Yes	Yes	Yes
Diff of Changes Between Source and Base Branch	Yes	Yes	Yes
One-Click Merge and Delete Source Branch	Yes	No	No
Integration with External Continuous Integration Tools	Yes	Yes	Yes
Pull Request Templates to Ensure Contributing Guidelines are Being Followed	Yes	No	No
Conversations Around Parts of the Code that Require Resolution	Yes	Yes	No
Required Reviews to Ensure that Every Pull Request is Signed Off by Someone Before the Merge	Yes	Yes	Yes
Option to Require Reviewer Approval Before Merge	Yes	Yes	Yes
WIP (Work In Progress) Indicator to Open Merge Requests Before They're Ready to be Merged	Yes	Yes	No

## V. CONCLUSION

We examined market share statistics and identified the major participants in the cloud-based Git repository hosting industry in order to assess the state of the market. With a significant portion of users and repositories, GitHub continues to hold a dominant position. Additionally, GitLab and Bitbucket have sizable user communities, particularly among businesses. Additionally, we looked at the creation of specialized systems that each catered to certain user preferences and needs, such as Gogs, Git-bucket, AWS CodeCommit, Azure DevOps, and Bitbucket Server. This study provides a comprehensive overview of cloud-based Git repository hosting options for academics, professionals, and students. The information presented in this paper should help stakeholders make wise choices, make the most of various platforms, and move ahead with agility and foresight in the ever changing world of contemporary software development.

## REFERENCES

- [1] Z. Yang *et al.*, "Git-Based Distributed Collaborative Learning: Theories, Tools and Features," in *Proceedings of the 2nd International Conference on Education: Current Issues and Digital Technologies (ICECIDT 2022)*, 2022, doi: 10.2991/978-2-494069-02-2\_15.
- [2] L. Dabbish, C. Stuart, J. Tsay, and J. Herbsleb, "Social coding in GitHub: Transparency and collaboration in an open software repository," in *International Conference on Computer Supported Cooperative Work (CSCW)*. ACM, pp. 1277-1286, 2012.
- [3] Y. Perez-Riverol *et al.*, "Ten Simple Rules for Taking Advantage of Git and GitHub," *PLoS Comput Biol*, vol. 12, no. 7, pp. e1004947, 2016, doi: 10.1371/journal.pcbi.1004947.
- [4] S. Tiwari, S. Rathore, S. Farooq, and P. Patani, "How Students Are Using GitHub? A Survey," in *15th Innovations in Software Engineering Conference*, 1-5, 2023, doi: 10.1145/3511430.3511454.
- [5] GitHub website: <https://github.com/>
- [6] GitLab website: <https://gitlab.com/>
- [7] Bitbucket website: <https://bitbucket.org/>
- [8] Amazon website: <https://aws.amazon.com/codecommit/>
- [9] Azure website: <https://azure.microsoft.com/en-us/products/devops>
- [10] Gitlab 2023 study on inactive projects: <https://news.itsfoss.com/gitlab-inactive-projects-policy/>
- [11] A. Zagalsky *et al.*, "The Emergence of GitHub as a Collaborative Platform for Education," in *Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work & Social Computing*, 2015.
- [12] C. Chandrasekara and P. Herath, *Hands-on GitHub Actions: Implement CI/CD with GitHub Action Workflows for Your Applications*. Apress, 2021.
- [13] GitHub, "The 2020 state of the octoverse - community report," 2020. [Online]. Available: [octoverse.github.com](https://octoverse.github.com)
- [14] T. Kinsman, M. Wessel, M. A. Gerosa, and C. Treude, "How do software developers use GitHub actions to automate their workflows?" in *International Conference on Mining Software Repositories (MSR)*, 2021. [Online]. Available: [octoverse.github.com](https://octoverse.github.com).