GitProtect
by Xopero ONE

Git Backup Guide

How to protect GitHub,
Bitbucket and GitLab
data

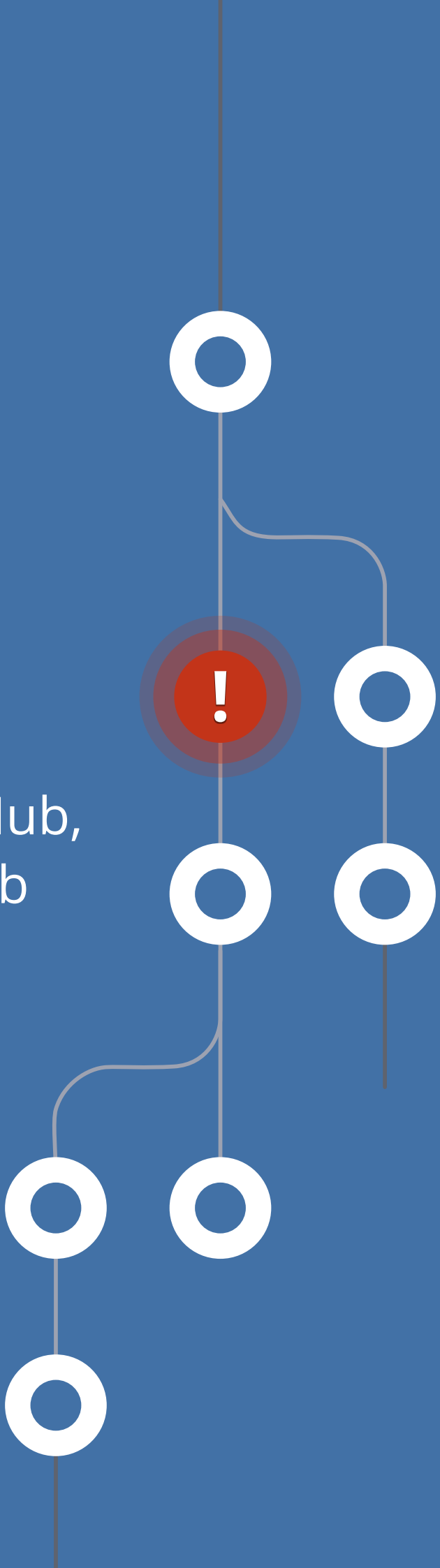


Table of contents

Introduction

If your organization uses version control systems like GitHub, GitLab and Bitbucket, you probably are aware that code as intellectual property is the most valuable asset inside your company - you and your team spent thousands of hours (and money) to write, support and improve projects. As CTO, IT manager, software-house owner, or team leader – you probably can imagine how much it would cost you to lose the code your team has been working on for months...

Let's not forget that while hosting your projects on GitHub, GitLab, or Bitbucket, it doesn't depend on your organization in 100% - now it's exposed to the risks of an external service provider as well.

Data breaches, systems downtime, policy changes, and more - all of those factors can limit access to your repositories on GitHub, GitLab, Bitbucket, and in conclusion, put your intellectual property at risk. And without proper protection of your IP, your business might not be able to harness the full potential of code created by your employees.

One of the hardest parts of being a leader is convincing your team and superiors that even if your code is hosted within such reliable companies like GitHub, GitLab, or Atlassian, it might get lost.

It is even harder to communicate with non-technical members of your team who still believe that Git itself or a cloud GitHub/GitLab/Bitbucket service is a backup. Well, it stores your code and files so after accidental deletion or modification, it should be able to bring your data back. But we can never treat production data we work on as a backup. That's the main lesson they should learn.

Let's go find you some arguments that will back you up during discussions with your superiors, team members, and even developers and convince them that GitHub, GitLab, or Bitbucket backup solution is something crucial for your organization.

Or if you are simply curious about GitHub, GitLab, or Bitbucket data protection - please feel welcome.

Chapter 1

Why backup GitHub, GitLab or Bitbucket - the risk of data loss

Reason #1

Shared Responsibility Model

As most SaaS providers, also GitHub, GitLab, and Atlassian rely on shared responsibility models that define which security duties are handled by the service provider and which belong to your organization as a user/customer. There are many ways the version control systems keep your code secure. They repair errors, handle hardware, server-side software failures, and data center outages. But regardless, your data is always your concern - you need to make sure it's properly protected. Even if considering legal and compliance issues - it is better to keep copies of the files for extended periods of time which some backup solutions provide.

For example, at Atlassian, the service provider handles the security of the applications themselves, the systems they run on, and the environments those systems are hosted within. They ensure compliance with standards such as PCI DSS and SOC2. You, as a customer, manage the information within your accounts, the users, access to your data, and control what apps you install and trust. Finally, you are responsible for ensuring your company is meeting compliance requirements. Just like in below image.

Probably that is why cloud service providers recommend its users to have a reliable third-party backup software, such as GitProtect powered by Xopero ONE.

Reason #2

Outages

Believe us or check it out, but there were many times that GitHub, GitLab,or Bitbucket went down, leaving many companies without access to their code and the possibility to work. Going further, with many financial losses.

Image: Atlassian Cloud Security Shared Responsibilities (Atlassian)

One of the biggest outages of GitLab happened in 2017. It was caused by a that they were eventually unable to recover. Specifically, they lost modifications to database and data such as projects, comments, user accounts, issues and snippets.

In June 2020, there was a major outage of the Github service that lasted for hours and impacted millions of developers.

That kind of outages can impact developers' productivity, especially if they occur during crucial launch windows. Think about your company - how long will you be able to work without access to your GitHub data? How much such an outage will cost your company? Are you able to afford it? Or you better prevent such situations and invest in reliable third-party backup software like GitProtect powered by Xopero ONE to quickly recover data and get back to code and work?

And GitHub downtime is only the tip of an iceberg...

Reason #3

Human errors

One of the most common issues when it comes to cybersecurity incidents is human error. It's the problem we can neverunderestimate. Especially, as developers tend to have one GitHub account that they use both for personal and professional purposes, sometimes mixing the repositories. Accidental deletion of branches, overwriting them, or even intentional deletion made by the frustrated employee (or even ex-worker, who still has access to the repository) - are some of the most common reasons for data loss. If you don't protect your data, that kind of problem can make your work completely useless.

Reason #4

Ransomware

Ransomware remains one of the most expensive threats for businesses of all time. It happens every 11 seconds and is projected that by the end of 2021 it will generate global losses of...20 billion dollars (compared to 325 million in 2015).

In 2019 BleepingComputer reported that attackers were targeting GitHub, GitLab, and Bitbucket users, wiping code and commits from multiple repositories and leaving behind only a ransom note and a lot of questions.

Business downtime caused by a ransomware attack usually lasts days. Then a company needs weeks to restore all systems, and without reliable backup software those attempts usually fall down. You can not believe that paying a ransom will give you a 100% guarantee of recovering your data. When it comes to the version control system, losing access to the data that stays encrypted, can cause downtime as well. Unless you have your Git backup and you can recover the data anywhere, from any point-in-time, and get back to work immediately. And most of all, not lose your data.

And we must remember the threat that lurks around every corner when it comes to data - malware attacks.

Reason #5

Hardware and Software Errors

Not only the human errors or hacker attacks can lead to losing access to your data, but it can be also influenced by many sorts of hardware and software failures.

Hardware and software failures can happen on the client-side, and if the data is not protected well enough, it all can be lost - especially if you work with Git only on your own local server.

Adding problems with synchronization, saving repositories, downloading it, you can see a full range of issues that can see a full range of issues that can slow down, postpone or disable the development process and expose your company to financial loss.

Chapter 2

Third-party repository backup vs. your own script

When it comes to files, endpoints, servers, or VMs - a third-party backup software is something obvious that nearly every business needs and should have. Unlike git repository backup which is not so obvious, but of equal importance.

Managing your own scripts - pros and cons

Managing backups in-house obligates you to manage all the infrastructure, processes, ongoing maintenance, and repair costs to make your internal backups. While in the beginning, it might seem cost-effective, in a long-term perspective maintenance cost and working hours of the employees managing backups can cost you a fortune.

PRO: Customization

Managing your own script lets you decide how it should work to meet your company's requirements and specifications. You know how it should integrate with other elements of your organization. Finally - you know what kind of data you want to protect, how often this backup should perform, and how - you can customize it.

CON: High long-term costs

If you want to make your own backups you have to delegate internal employees to work on it, test it on a regular basis, maintain it and enable some form of data retention - unless you have to keep in mind to manually remove older backup copies to make room for new ones... Even if it's just a part-time job of your employee, it distracts him from his core duties. And now - let's assume that you sacrificed your employee time and you finally have your own backup script. Now somebody has to test it and maintain it as a part of his routine. As in most software, not only in the backup case, most costs occur during use.

CON: Responsibility

Moreover, if the event of failure happens and your backup script fails so you won't be able to restore the data, the only person you can blame is yourself. Or at least your management will do that. Are you sure you need this additional responsibility on your shoulders?

Third-party repository backup - pros and cons

When you are buying a third-party repository backup you know you pay for a piece of mind, saving your employees time so they can focus on core duties, reducing maintenance, and administration costs, and data protection guarantee. Initial higher-cost seems now pretty slight when you consider it in the long-term. It turns out

that it's a pretty small investment for all of the security it gives...

PRO: All the best of backup solution

The third-party repository backup solution such as for instance GitProtect powered by Xopero ONE enables you to protect all GitHub and Bitbucket data - no matter what hosting service you use. You can backup all GitHub and Bitbucket: servers, repos, and metadata - both local and cloud. Including comments, requests, milestones, issues, wikis and much more.

Such dedicated git repository backup makes you sure you use years of experience of a backup service provider on the backup market that protects all mission-critical data - including files, endpoints, servers, virtual machines, SaaS, etc. (and on which btw. you can take advantage of).

So, except for some dedicated features, you have access to the best features such as:

- any storage compatibility (you can store your copies on SMB network shares, local disc resources, public clouds)
- full automation ("set-and-forget") and central management
- predefined backup plans or advanced plan customization (so you can adjust backup performance to your company requirements and specification)
- wide range of recovery options (including granular, point-in-time recovery, cross-user recovery)

Even if you delegate your best developers to write you a backup script, they probably won't be able to deliver you such advanced and secure features as a professional backup provider and won't ensure you with the same guarantee of data accessibility and recoverability.

PRO: Security and recovery guarantee

Speaking about best practices - for all third-party professional backup service providers security is an integral part of their DNA. They need to make sure that the data is protected, recoverable, and accessible anytime and as fast as you need it. As your business probably relies on software and digital assets more than ever before, make sure the repository backup software you use provides you with encryption (AES is desired), zero-knowledge encryption, and no-single-point-of-failure, web-based architecture. Additionally, if something is wrong with your copy, you should be informed about it by daily reports, logs and special notifications.

PRO: Lower long-term costs

You might think a third-party repository backup solution is an expensive option. But try to calculate how much you are going to pay for preparing internal repository protection procedures and backup scripts. Then, to this sum add hours spent on maintenance, tests, and administration of your employee and alternate cost - how much money would this employee bring you while he would do his normal work instead. And of course, take his word that your data is secured. We will make a bet, that initial higher costs seem pretty slight now

- long-term costs of a third-party backup solution now seem more attractive, and your employees can focus on what they are best at - their work. And bring you money.

CON: Limited control

Like with every kind of third-party software you don't have control over each aspect of its pricing, terms of services and eventual changes in the future. So you should consider what is more important to you - choosing a third-party software with limited control and team's focus on solving core business problems or preparing and maintaining your own backup procedures over which you have full control with devoting priceless time of your own internal developers.

PRO: Meeting the shared responsibility model

Whether you use GitHub or Bitbucket, like most SaaS providers, those two also rely on shared responsibility models. In short: service providers are responsible for the accessibility and availability of their platforms while you, as a data owner, are responsible for data protection. Are you sure that your own, internal solution is safe enough? Have you considered all possible scenarios of losing your data? Finally, do you have it tested? With a third-party backup solution you share this concern - now also an external company is responsible for keeping your data safe.

Chapter 3

GitHub, Bitbucket, and GitLab backup options - what to choose?

In this chapter, we show you the different types of tools that you can use to back up your version control system data. Please bear in mind that GitHub. GitLab or Atlassian, as most cloud services providers, rely on shared responsibility models in which you are responsible for data protection. They even encourage you to use third-party backup software to make sure your data is accessible and recoverable. So now, let's take a look at what external options you have.

Option one: Scripts

GitHub, Bitbucket, and GitLab are the platforms used by developers who see the need to back up their work. Thus on developers forums, inside Git communities you can find a lot of self-written scripts that provide you with some basic features and a bit of peace of mind.

Among some backup scripts are:

- backup-github.sh
- Python GitHub Backup
- based on PowerShellForGitHub
- koddr/github-backup-automation
- clockfort/GitHub-Backup
- atlassian-bitbucket-diy-backup
- ghe-backup by Zalando Tech

and more.

PRO: Using an open-source script is free and usually the code is pretty stable. You can check the entire script on GitHub or Bitbucket and decide whether it's sufficient for all your or your organization's requirements.

CON: Sometimes scripts do not provide you with a restore option, give you rather archiving features or require a specific data storage. So you can perform backups but if the event of failure occurs, you will need to write your own script to recover the data. Having backups without a reliable recovery method provides you with very limited possibilities. In the event of failure every minute matters, so besides backups it might be even more relevant to have a recovery option that makes your data accessible and recoverable from any point-in-time exactly when you need it.

Scripts - for who? Individual developers or small project owners. It's a solution for those who are in need of very rare backups, made rather sporadically. As it doesn't include a recovery option - it would be efficient only for people whose business does not rely on version control system data and might survive with no access to information for some while. Finally, it's efficient only assuming that afterward you or your team would be able to write a recovery script to not lose data.

You can even solve this by writing your own scripts if you have enough skills, time and resources. But please bear in mind that it can be tricky to get this right. Moreover maintaining such a tool might generate long-term costs and administrative time (as mentioned in the previous chapter).

Option 2: S3 Backup

Amazon S3 Backup is an open-source script made with the use of GitHub Action. GitHub Action is an internal GitHub tool that helps to build, test and deploy applications. In short - it automates every step of the development workflow.

S3 Backup app allows you to backup git repositories into popular Amazon S3 compatible object storage.

PRO: S3 Backup is an open-source script meaning it's for free and quite simple to use. You don't need to install any additional software. All changes are captured as it works with every push so you don't need to worry about missing some data in your backups.

CON: S3 Backup relates only to repositories so all metadata - including pull requests, wikis, projects, issues, etc. are not protected and can be lost. The biggest disadvantage is that it does not provide you with any retention settings and options - old backups are immediately replaced with new ones here. It gives a very poor recovery option - all you can recover is only the last full version and there is no possibility of point-in-time recovery. Just imagine that malicious actors can compromise backups by infecting a new copy that will replace an old reliable one. It opens them a gate to perform further attacks. Yes, it is possible with only one backup version stored.

Moreover - it does apply only to cloud storage so if you want to use your own, local infrastructure, it's not for you.

S3 Backup - for who? As S3 Backup does not differ a lot from the scripts mentioned above we can also recommend it to individual developers or small project owners who need to perform backups rather sporadically. As there are no retention settings and it simply replaces old copies with new ones it may be useful only when you don't need to track changes and recover historical data. In short - all you need is only the last state of data. Moreover, if you consider using S3 Backup, please make sure you have all the most important security measures in place to prevent any malicious actions.

Option 3 - GIT and GitHub API

Finally, you can use the official GitHub API to backup your GIT repository. First you need to clone and download a repository or wiki to your local machine. Once you have it done, you need to use API to export elements of your GitHub Enterprise Server repository (like issues, pull requests, forks, comments, etc.) to your local machine, create a zip archive and save it in some secure place - external hard drive or cloud service.

PRO: Using GIT and GitHub API allows you to create copies of the entire set of data - repositories together with all metadata.

CON: We would say that this option is rather a replica of your repositories and all metadata but not a backup itself - it's not encrypted, can be compromised, and lost. It's like an external copy of your repository saved in separate localization. It doesn't run automatically so it requires repeating this action over and over again.

GIT and GitHub API - for who? As this method doesn't work automatically, it's rather an option for archiving a GitHub repository and old projects. It doesn't provide any security measures for such copies so it shouldn't be treated as a backup. It might become useful for small project owners who simply want to keep access to older projects for any future use.

Option 4 - Git Backup powered by Xopero ONE

If your organization expect more from a backup than simply one copy of your data set (and mostly only repository copy) better consider a professional backup and recovery software for GitHub, Bitbucket and GitLab such as GitProtect powered by Xopero ONE which automatically protects all your version control systems data - including servers, repositories and metadata (pull requests, wikis, issues, branches, projects, etc.).

PRO: GitProtect powered by Xopero ONE is the software dedicated to GitHub, Bitbucket, and GitLab which includes the best features of enterprise backup software created by the vendor with more than 10 years of experience on the backup market. Trusted by thousands of customers and partners worldwide (including: T-Mobile, Orange, ESET, Subway, AVIS).

Whether you use GitHub, GitLab, or Bitbucket, you can **protect all your data** - including servers, repositories, and all metadata. And it does not matter if you use your version control system as a SaaS application or locally on your developers' devices.

When it comes to **storage**, you don't need to invest in an additional IT infrastructure - you can store backups locally (your local machine or any NAS, SAN devices) as well as any private or public cloud compatible with Amazon S3 (AWS, Azure, Wasabi, Xopero, etc.). It can even be a hybrid or multi-cloud environment as within one license you can have multiple storages.

All you need to do is to install the software and then you can only use a **central web-based management console** to set backup plans, recover the data, manage users, devices and storages. Thanks to this cloud-based architecture you have access anytime and anywhere - every time you need.

Once you add your GitHub, Bitbucket or GitLab account to Git Backup by Xopero ONE you can set automatic **backup plans** which include data to be protected, backup type (file backup or image backup), storage where the copies should be stored, schedule so

the time when the automatic backup should be performed and backup execution manner. New repo? It can be automatically added to your scheduled backup plan. Moreover, you can set a push as a trigger so the backup will perform automatically with every push you make.

To make it even more easy for you - you can choose a **predefined backup plan** from the list.

You have a **full control over retention** due to the Grandfather-Father-Son scheme - probably the most efficient way of rotation that allows you to manage the copies in the long-term perspective while requiring minimal space in data storage and enables fast recovery.

Moreover, having a full control over retention gives you a possibility to **archive unused repositories** and save your version control system's free space and save money.

During backup plan setting, you can even choose **encryption level** (all copies are encrypted with AES encrypted key considered as impossible to break but additionally you can change a force of this encryption) and **compression level** to control your storage capacity.

To make it even more safe for you - we have implemented a **Secure Password Manager** that enables you to create strong passwords that you don't need to memorize.

Git Backup also provides **audit logs and notifications**, so you can stay up to date and keep track of your copies for security and compliance purposes.

And finally, you have a wide range of **data recovery options**. Flexible, point-in-time recovery to a repository or local device makes Git Backup a very reliable and complete backup and recovery solution for your version control system.

CON: GitProtect powered by Xopero ONE is a paid solution but the price depends on the number of repositories to protect. Unlike in your own script case, you are not an architect here so you have less control over how it works and what features it has. But the list of features is quite long (and based on years of market experience) so probably it can be even wider than you expect. Considering you can use your own infrastructure and nearly every storage compatibility as well as repository archiving possibilities (and saving your version control system space) this price seems relatively low and reasonable. Adding any possible attacks and events of failures may even become an investment with a pretty high return. It is said that in the event of failure you can save 4$ on every 1$ spent on backup and disaster recovery solution.

Git Backup powered by Xopero ONE - for who? For every organization that treats its code as an Intellectual Property and relies on version control systems as GitHub, Bitbucket, or GitLab - regardless of its size, revenues, and even industry. It can be an enterprise, a small or medium-sized business that has an IT department as well as a software house and even individual

developers. It's for all organizations that are aware of data breach costs and legal penalties so want to prevent data loss and ensure business continuity.

Chapter 4

Git backup strategy - what should you consider?

Have you decided which backup solution is the best for your organization? Do you have enough arguments to convince your team or superiors that data protection of your GitHub, Bitbucket or GitLab account is an absolute must-have? If the answer is 2x YES - let's take a look at what you should consider in your backup strategy.

Choose data to be protected

Together with your organization's growth, grows the number of projects, repositories; and other data generated within your GitHub, Bitbucket, or GitLab account.

You should determine what kind of data is the most crucial for your organization and need to be protected and recoverable in the event of failure to keep your business working. Then, create a backup plan that will include all critical servers, repositories, metadata (including comments, requests, issues, milestones, releases, wikis & more) - both local and cloud.

Storage - where do you want to store your copies

You should be able to choose from a wide list of storage options – local, cloud (public or private), hybrid, or multi-cloud. Then You need to consider which option is the most accurate and efficient for your organization and choose the one that doesn't require any additional IT infrastructure investment and you can use your current resources.

Schedule and backup performance manner

Finally, define a schedule when your automatic backups should perform - it can be a specific time-range (days and hours) or backups can be triggered by every push.

You should have full control over retention and versioning that will eliminate the risk of compromising backups by attackers and let you archive unused repositories. The GFS (Grandfather-Father-Son) scheme seems like the most suitable option that defines the efficient way of rotation full, differential, and incremental backups that save your storage and enable fast recovery.

It would be great to have a wide range of options when it comes to additional backup plan settings like encryption or compression to control storage capacity and backup performance speed.

Recovery Time Objective and Recovery Point Objective

A good backup strategy has to have a well-thought disaster recovery plan that will eliminate downtime and guarantee

business continuity in case of any event of failure. What you need to do first is to define the two most important parameters – Recovery Point Objective (RPO) and Recovery Time Objective (RTO).

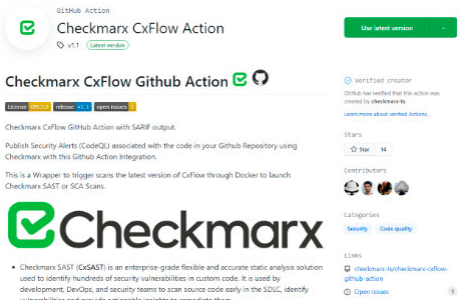
Then look for a GitHub, Bitbucket and GitLab backup solution that gives you a wide range of **data recovery options**. Flexible, point-in-time recovery to any place - local device or another repository is desirable.

Chapter 5

Git protection: consider these solutions in your security plan

GitHub ECOSYSTEM

We have selected a list of tools and solutions from the GitHub marketplace that will strengthen your data protection, facilitate management and unleash productivity.

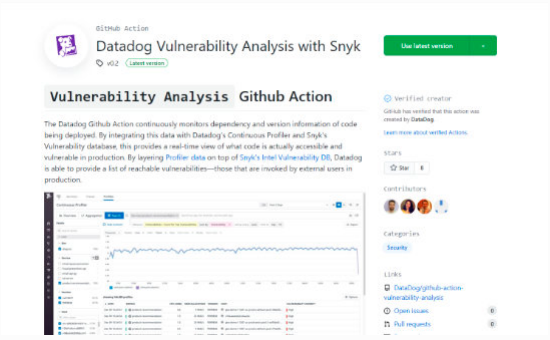


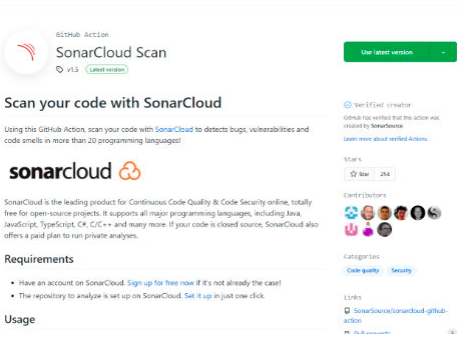
Checkmarx CxFlow Action
by Checkmarx

Publish Security Alerts (CodeQL) associated with the code in your Github Repository using Checkmarx with this Github Action Integration.

Datadog Vulnerability Analysis with Snyk by Datadog

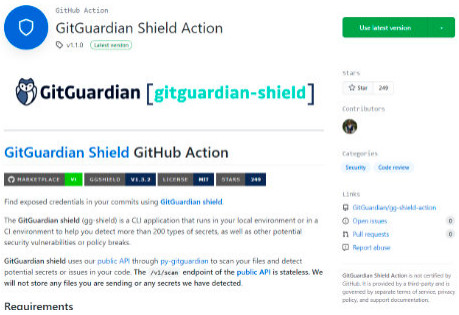
Continuously monitor dependency and version information of code being deployed. Get a real-time view of what code is actually accessible and vulnerable in production.





SonarCloud Scan
by SonarSource

Scan your code to detects bugs, vulnerabilities and code smells. It supports all major programming languages, including Java, JavaScript, TypeScript, C#, C/C++ and more.

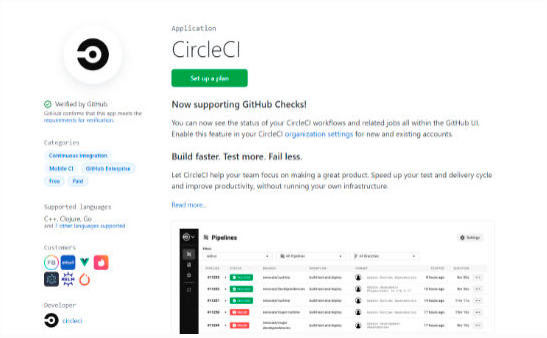


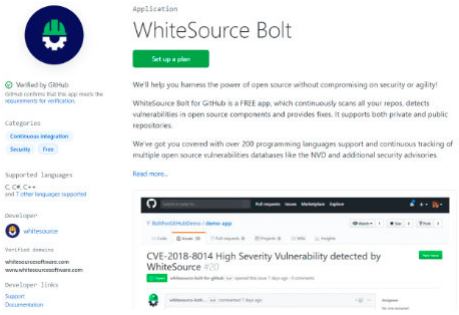
GitGuardian Shield Action
by GitGuardian

Find exposed credentials in your commits. This CLI application helps detect more than 200 types of secrets and other potential security vulnerabilities or policy breaks.

CircleCI
by CircleCI

See the status of your CircleCI workflows and related jobs all within the GitHub UI. All recent builds in one place. You can also apply filters to quickly find what you're looking for faster.



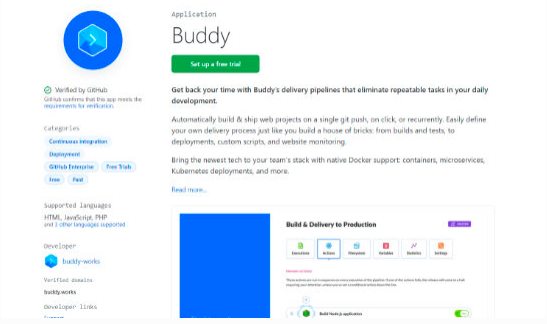


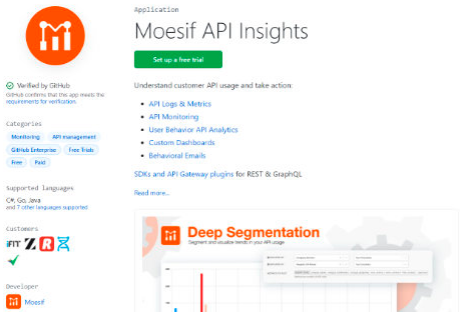
WhiteSource Bolt
by WhiteSource

Continuously scan all your repos, detect vulnerabilities in open source components and provide fixes for both private and public repositories.

Buddy
by Buddy Works

Eliminate repeatable tasks in your daily development. Automatically build & ship web projects and easily define your own delivery process.



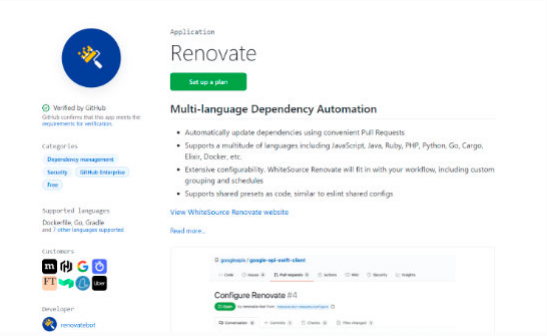


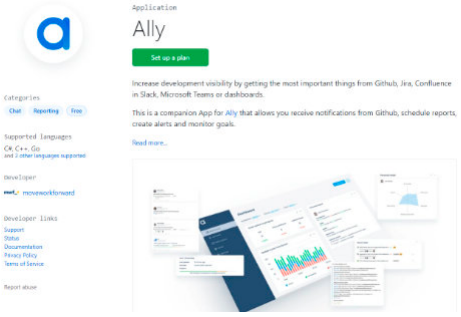
Moesif API Insights
by Moesif

Understand customer API usage and take action: API Logs & Metrics, API Monitoring, User Behavior API Analytics, Custom Dashboards and Behavioral Emails.

Renovate
by Renovate Bot/WhiteSource

Automatically update dependencies using convenient Pull Requests. Supports a multitude of languages including JavaScript, Java, Ruby, PHP, Python, Go, Cargo, etc.



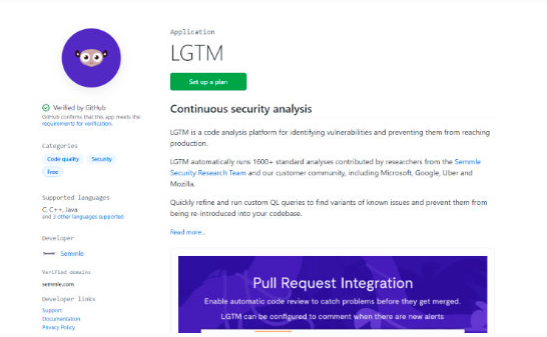


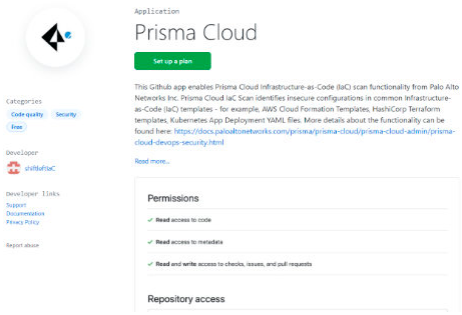
Ally
by MoveWorkForward

Increase development visibility by getting the most important things from Github, Jira, Confluence or MS Teams. Create alerts, receive notifications, schedule reports, etc.

LGTM
by Semmle

Quickly refine and run custom QL queries, find variants of known issues and prevent them from being re-introduced into your codebase.



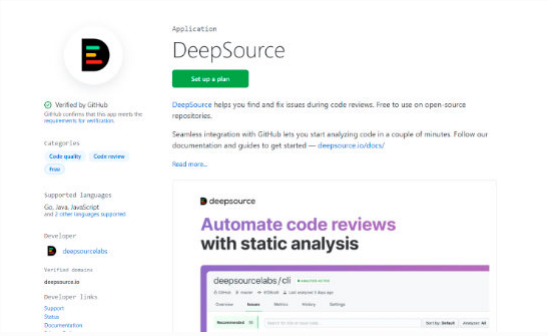


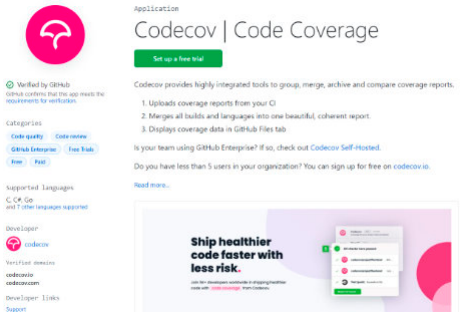
Prisma Cloud
by Palo Alto Networks

Identify insecure configurations in common Infrastructure-as-Code (IaC) templates - AWS Cloud Formation Templates, Kubernetes App Deployment YAML files, etc.

DeepSource
by DeepSource

Find and fix issues during code reviews. Free to use on open-source repositories. Seamless integration allows you start analyzing code in a couple of minutes.



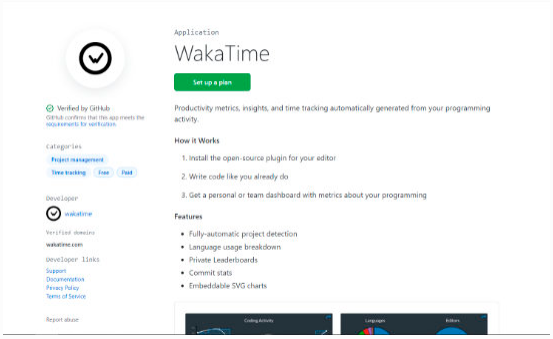


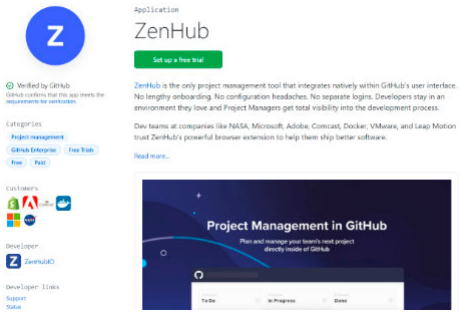
Codecov
by Codecov

Codecov provides highly integrated tools to group, merge, archive and compare coverage reports. This way you can ship healthier code much faster and with lesser risk.

WakaTime
by WakaTime

Productivity metrics, insights, and time tracking automatically generated from your programming activity. You can also set up a goal each day.



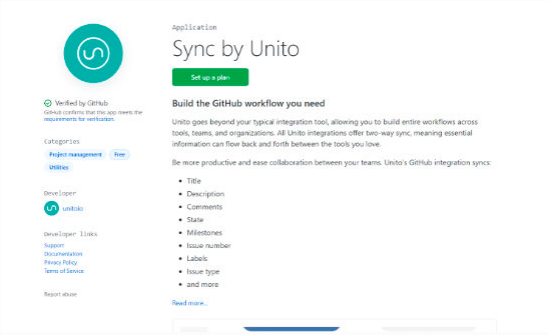


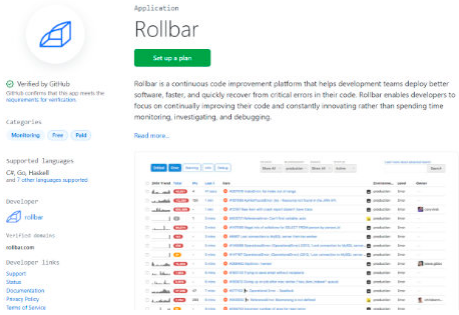
ZebHub
by ZebHub

Project management tool that integrates natively within GitHub's user interface. Turn issues into epics, add user stories, and get sprinting.

Sync
by Unito

Build entire workflows across tools, teams, and organizations. All Unito integrations offer two-way sync, meaning essential information can flow back and forth between the tools.



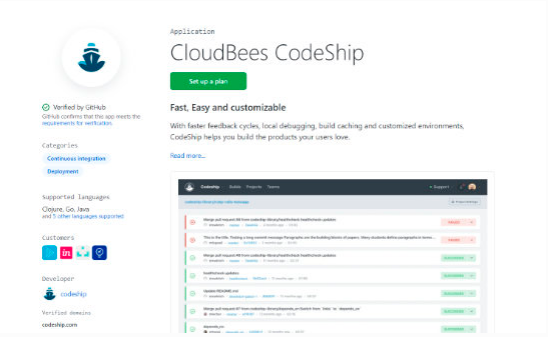


Rollbar
by Rollbar

A continuous code improvement platform that helps development teams deploy better software, faster, and quickly recover from critical errors in their code.

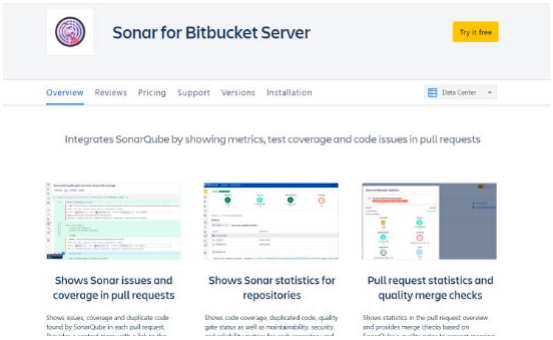
CloudBees CodeShip
by CloudBees

Fast, easy and customizable continuous integration with feedback cycles, local debugging, build caching and customized environments.



Bitbucket ECOSYSTEM

We have selected a list of tools and solutions from the Atlassian marketplace that will strengthen your data protection, facilitate management and unleash productivity.

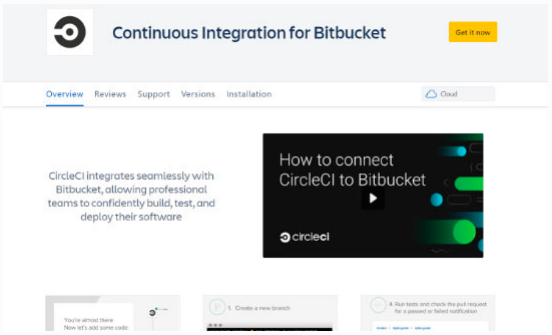


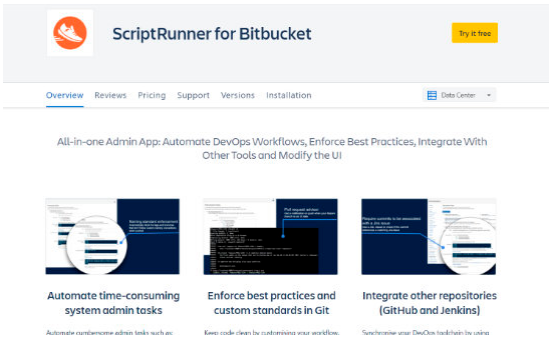
Sonar for Bitbucket Server
by Mibex Software GmbH

Integrates SonarQube by showing metrics, pull request statistics with quality merge checkstest coverage and code issues and coverage in pull requests.

Continuous Integration for Bitbucket
by CircleCI

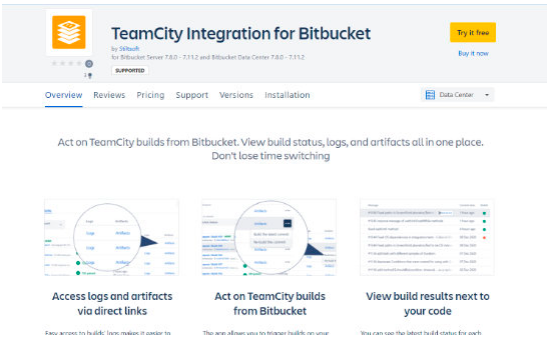
CircleCI integrates seamlessly with Bitbucket, allowing professional teams to confidently build, test, and deploy their software.





ScriptRunner for Bitbucket
by Adaptavist

All-in-one Admin App: Automate DevOps Workflows, Enforce Best Practices, Integrate With Other Tools and Modify the UI.

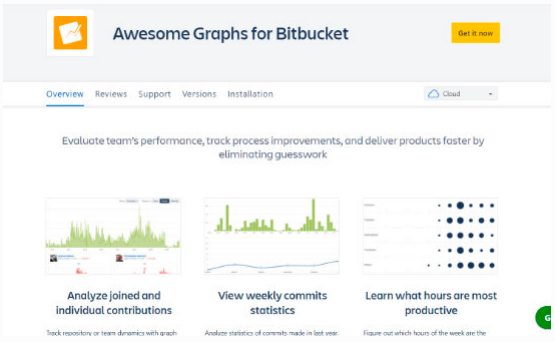


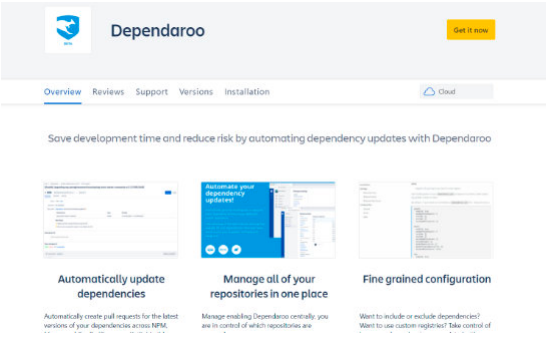
TeamCity Integration for Bitbucket
by Stiltsoft

It allows for a seamless connection of your Bitbucket to TeamCity CI/CD server, so you can monitor and configure the pipeline without losing the context.

Awesome Graphs for Bitbucket
by Stiltsoft

Awesome Graphs provides illustrative graphs and charts to visualize the contribution statistics in Git repositories. You get analytical data for monitoring, reporting, and planning.



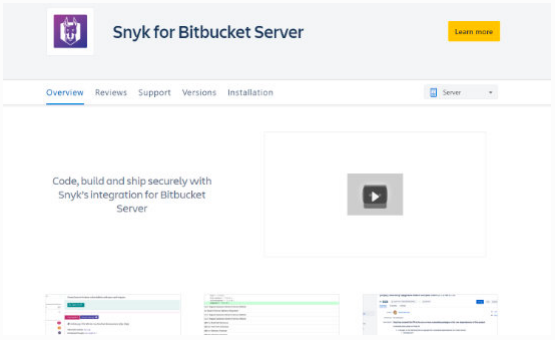


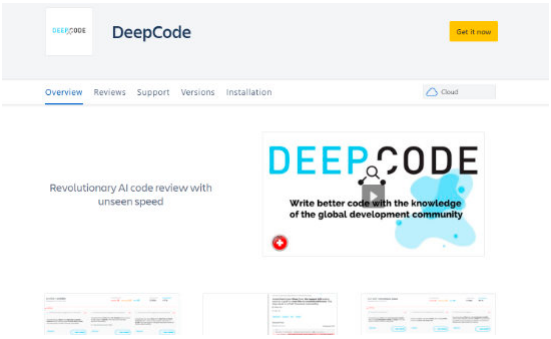
Dependaroo
by Instil Software

Save development time and reduce risk by automating dependency updates and manage all of your repositories in one place.

Snyk for Bitbucket Server
by Snyk.io

Find and fix security vulnerabilities and license issues in their open source dependencies and container images across the Bitbucket Server development workflow..



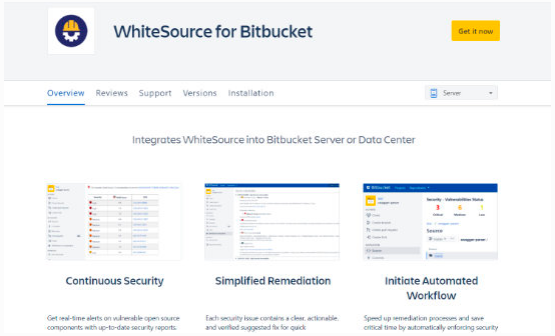


DeepCode
by DeepCode by Snyk

Innovative code review powered by AI. Live alerts of critical bugs upon every pull request pull request on Bitbucket or within your IDE.

WhiteSource for Bitbucket
by WhiteSource

WhiteSource seamlessly integrates with your repositories, IDEs, build tool, CI servers and more to secure and manage the open source components in your products.



Summary

According to official numbers, there are over 56M developers on GitHub and 60M new repositories created only in the last year. Bitbucket is used by over 10 million users and GitLab has 30 million estimated registered users! Here at Xopero Software, we believe that you guys are people who force the digital revolution. We want you to be properly secured so you can make a change peacefully. You need a reliable backup solution to make your code, intellectual property, and projects accessible and recoverable.

We hope this document will help you convince your team members and superiors to GitHub, Bitbucket, and GitLab data protection and deliver you a very solid argumentation.

If you have any questions, doubts, or insights, feel free to reach out.

We have got your back(up)!

GitProtect

by Xopero ONE