GitHub Backup

GUIDE

How to backup and restore GitHub repositories with metadata to stay compliant and ensure safe DevSecOps processes.

GitProtect
by Xopero ONE

Table of contents

Introduction

If your organization uses GitHub you are probably 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, Head of IT, IT manager, or CISO and Security Architect – 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, 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 those factors can limit access to your repositories and metadata on GitHub, 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 a reliable company as GitHub, 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 GitHub Cloud 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 some arguments that will back you up during discussions with your superiors, team members, and even developers and convince them that the GitHub backup solution is something crucial for your organization.

Or if you are simply curious about GitHub data protection - please feel welcome.

Chapter 1

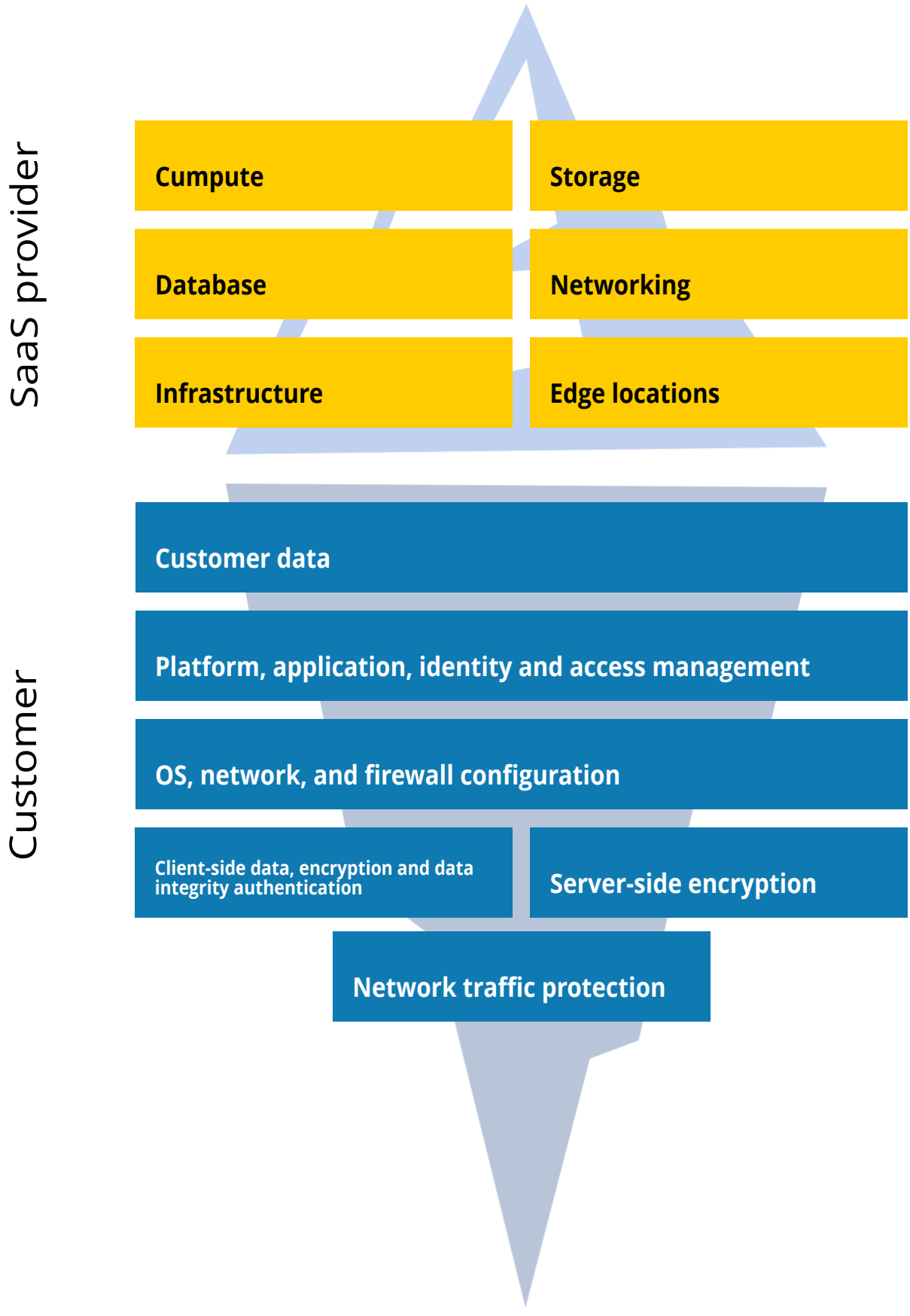
Why backup GitHub - the risk of data loss

Reason #1

Shared Responsibility Model

As most SaaS providers, also GitHub (and Microsoft) rely on shared responsibility models that define which security duties are handled by the service provider, which belong to your organization as a user/customer and which duties you both share. 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.

In short: GitHub takes care of operating systems, network controls, applications, physical hosts, network and data centers, and partly for the identity and directory infrastructure. They ensure compliance with standards like SOC 2, ISO/IEC 27001:2013, GDPR, and more. But when it comes to data, they are data processors - you are the owner and it's your responsibility to ensure data confidentiality, accessibility, and recoverability. You are responsible to manage the information within your accounts, the users, access to your data, and control what apps you install and trust. Finally, it's your duty to ensure your company meets compliance requirements. Just like in the picture on the next page. Probably that is why cloud service providers recommend their users to have reliable third-party backup software, such as GitProtect.


SaaS provider
Cumpute
Storage
Database
Networking
Infrastructure
Edge locations
Customer
Customer data
Platform, application, identity and access management
OS, network, and firewall configuration
Client-side data, encryption and data integrity authentication
Server-side encryption
Network traffic protection

Reason #2

Outages

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

Even the beginning of 2022 wasn't the best for GitHub. In February, GitHub's worldwide outage prevented access to the website, issuing commits, cloning projects, or performing pull requests. Then, in March 2022 there were 4 outages that occurred within a week.

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 GitHub backup by GitProtect.io 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

According to scientific research, an employee can make from 3 to 6 mistakes an hour. Thus, human errors are an eternal problem because humans are humans and when it comes to cybersecurity incidents, it is one of the most common reasons. It's the problem we can never underestimate. 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

Some say that the most high-cost threat for a business is ransomware. Moreover, every 11 seconds somewhere in the world there can be an attack and it leads to the company losing billions of dollars. In 2019 there were reported attacks on different repositories, and GitHub was among their targets.

No wonder attackers intend to get to GitHub data, as it is the most valuable SaaS in every technology-related IT company. 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.

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, and downloading it, you 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 GitHub repository and metadata backup which is not so obvious, but of equal importance.

Managing your own scripts - pros & cons

Managing backups in-house obligate 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 costs 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 for 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 repo backup - pros & 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.io enables you to protect all GitHub repositories and metadata - wiki, issues, issue comments, deployment keys, pull requests, webhooks, labels, milestones, pipelines/actions, tag, LFS, releases, commits, branches, projects and more.

Such dedicated GitHub repository backup makes you sure you use years of experience of a backup service provider on the 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, and 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, specification and pipeline)
- a wide range of recovery options (including granular, point-in-time recovery, cross-user recovery, and disaster recovery technologies)
- Security level that meet SOC 2 and ISO27K requirements.

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 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

As we said before, GitHub relies on the shared responsibility model. In short: it is responsible for the accessibility and availability of its 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 like GitProtect.io you share this concern - now also an external company is responsible for keeping your data safe - just take a look at the below table to check how it can support you in fulfilling shared responsibility duties.

Area of responsibility	GitHub's responsibility share	User's responsibility share	GitProtect.io responsibility share
Infrastructure	Uptime of GitHub service	Access and control of the data in the GitHub repository	Automatic backup and data protection, the implementation of the 3-2-1 rule
Compliance	SOC 1, SOC 2, ISO/IEC 20071, GDPR, FedRAM LI-SaaS ATO (data processor)	Meeting legal compliance, local compliance and industry regulations (data owner)	Encryption, integrity, data restore, disaster recovery - we help you meet all your compliance requirements (data guards)
Storage	GitHub cloud storage, at-rest encryption	Keeping multiple copies	On-premise, public/hybrid - store your data anywhere you want
Retention	Basic retention (up to 90-days in the public repository, up to 400 days in the private repository)	Long-term retention	Flexible or unlimited enterprise-grade data retention (and advanced retention schemes - Basic, GFS as well as granular & point-in-time recovery options
Restore	Restore the entire platform only	Own-written scripts, snapshots, clones	Point-in-time restore of repositories and metadata, Crossover recovery (and easy migration between platforms), Disaster Technologies, Restore to your local device

Chapter 3:

GitHub backup best practices

Now let’s get deep into the best practices for protecting your GitHub data and make every line of the source code accessible and recoverable so you can be sure your team can work uninterruptedly even during serious GitHub outages and you never lose access to your Intellectual Property, hours of work (and money) as well as reputation and customer trust.

PART I
BACKUP PERFORMANCE

Backup all repositories with related metadata

To be sure that all of your GitHub environment is reliably protected make sure to backup all repositories with related metadata. Whether you use GitHub or GitHub Enterprise your copies should include:

- Repositories,
- wiki,
- issues,
- issue comments,
- deployment keys,
- pull requests,
- pull request comments,
- webhooks,
- labels,
- milestones,
- pipelines,
- projects,
- LFS backup.

Your backup software should enable you to create many custom backup plans to adjust your data protection policy to your organization's needs, structure, and workflow.

The best practice is to create a backup plan for critical repositories and metadata that change on the daily basis (or even more frequently) for example using recommended Grandfather-Father-Son (GFS) rotation scheme and another backup plan for unused repositories that you need to keep for any future reference. This kind of backup is required more for GitHub archive goals and due to unlimited retention, you can store your copies for as long as you need - even infinitely. Moreover, you can even delete those repositories from your GitHub account and keep the copy on storage to bypass GitHub limits.

Incremental and differential backups that save your storage space

Your backup software should include only changed blocks of your GitHub data since the last copy to reduce the backup size on your storage, speed up backup and limit

bandwidth. Moreover, in the perfect scenario, you should be able to define different retention and performance schemes for every type of copy (full, incremental, and differential).

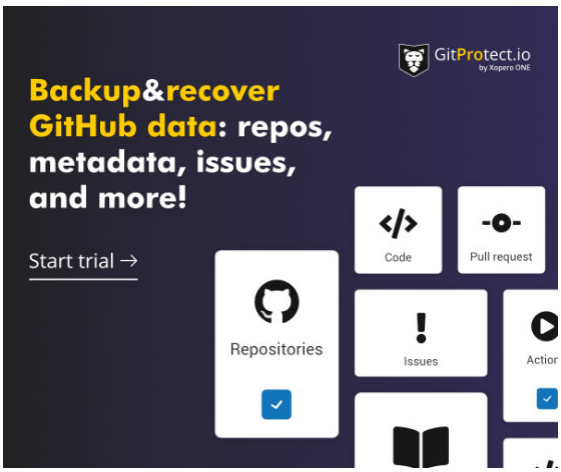
SaaS or On-Premise deployment

Whether you use GitHub or GitHub Enterprise you might want to run your backup software on the cloud or self-host it on your private infrastructure. The basic difference is the place where the backup service is installed and running.

To deploy it in a SaaS model, you are not obligated to allocate any additional device that could be used as a local server - the service runs within the provider's cloud infrastructure. You do not have to worry about its maintenance or administration, and the continuity of operation is guaranteed by the service provider.

On-Premise deployment means you install the software on a machine of your provision and control so it works in your environment locally. It is good to have the possibility to install it on any computer (Windows, Linux, macOS), or even on popular NAS devices. In this deployment model, you will avoid any issues that may occur within connectivity to the network, and the copies will be made using the local network, thanks to which the backup process will be faster and more efficient.

Please note that the deployment model should be independent of data storage compatibility.



All GitProtect.io Cloud PRO and Cloud Enterprise, as well as On-Premise Enterprise, give you the possibility to use GitProtect unlimited cloud storage which is always included in the license. In the Enterprise plans, you can also bring your own storage - cloud or on-premise. GitProtect.io supports AWS S3, Wasabi Cloud, Backblaze B2, Google Cloud Storage, Azure Blob Storage, and any public cloud compatible with S3,on-premise storage (NFS, CIFS, SMB network shares, local disk resources), as well as, hybrid and multi-cloud environments.

Adding multiple storage instances and fulfilling the 3-2-1 backup rule

Your GitHub backup software should enable you to add an unlimited number of storage instances - on-premise or cloud (preferred both) to replicate backups between storages, eliminate any outage or disaster risk and meet the 3-2-1 backup rule. It says that you should have at least 3 copies on 2 different storage instances with at least 1 in the cloud.

GitProtect.io is a multi-storage system. It allows you to store your data:

- in the cloud (GitProtect Cloud, AWS S3, Wasabi Cloud, Backblaze B2, Google Cloud Storage, Azure Blob Storage, and any public cloud compatible with S3).
- locally (NFS, CIFS, SMB network shares, local disk resources),
- in hybrid environment/multi-cloud

Regardless of the type of license, you always get GitProtect Unlimited Cloud Storage for free so you can start protecting your repositories immediately.

Now, let's see how this multi-storage system works in practice.

Let's assume that your Security and Compliance department forces you to store your data on Google Cloud Storage but you also have a backup plan that sends your copies to your local server. Then the huge Google outage occurs and you need to instantly restore your copy from two weeks ago. If so, simply login to your GitProtect.io account, choose a backup plan assigned to your local server, choose the copy from two weeks ago, and restore it - to the same or new GitHub account, to your local machine, or cross-over to another git hosting platform. 5 mins and you have to no longer stress out with the ongoing Google outage.

Backup replication

One of the most important features which you should consider when choosing backup software is backup replication. It permits you to keep consistent copies in multiple locations to follow the 3-2-1 backup rule, enabling redundancy and business continuity. You should have the possibility to replicate from any to any data store - cloud to cloud, cloud to local, or locally with no limitations.

How does it work in GitProtect.io? In the menu of the central management console, you will find a replication plan. All you need to do is to indicate the source and target storage, agent, simple schedule, and... voilà!

Flexible retention - up to unlimited

Retention settings are one of the potential game-changers when it comes to choosing the right GitHub backup software. You need to make sure that the features it offers meet your legal, compliance, and industry requirements. There are organizations that need to keep some data for years - it all depends on what kind of data you store in your repositories, for how long you have to keep them, and from what period that data should be restored in the event of failure.

Default 30 up to 365 days of retention that most vendors offer is definitely not enough. Forget it. Especially, when you consider backup software for archiving old, unused repository purposes or meeting your compliance.

You should be able to set different retention for every backup plan by:

- indicating the number of copies you want to keep,
- indicating the time of each copy to be kept in the storage (those parameters should be set separately for the full, differential, and incremental backup),
- disabling rules and keeping copies infinitely (to use it for GitHub archive purposes).

GitHub backup as a part of the CI/CD process

When building software, it is very important to take care of every stage of the deployment process so that delivery is predictable and flawless. Automation plays a key role in ensuring that each release goes through a specified list of steps. The desirable DevSecOps approach highlights the need to include security in this process. Of course, CI/CD steps can be customized according to your workflow, type of project, team, etc. but please make sure that it includes backup and ensures your team with peace of mind and source code recoverability from every point in time. Help them move faster and focus on delivering outstanding features without worrying about any potential mistakes and breaking anything.

GitProtect.io through API gives you the possibility to customize backup so that it works based on your company's workflow - for example, you can set an automatic backup after each pull request.

The biggest advantage of automated backup as a part of your CI/CD process is a guarantee that every change in the code is reliably protected and easy to recover from any point in time and roll back to previous versions in the event of failure or a human mistake.

Monitoring center - email and Slack notifications, tasks, advanced audit logs

You might not be directly responsible for managing backup software, but sure you want to easily monitor backup performance, check on statuses, and just in case - check on who exactly is responsible for a specific change in the settings to control your admins - in short - you need to have a complex, customized monitoring center.

One of the easiest ways to stay up to date without even the need to log in is custom email notifications. You should be able to configure:

- recipients (so you don't even have to have an account in the backup software to stay informed about backup statuses),
- backup plan summary details such as successfully finished tasks, tasks finished with warnings, failed tasks, canceled tasks, and tasks not started.
- choosing a language might be a plus.

Ideally, you should have notifications sent directly to the software you and your team use as a daily routine - Slack notifications. Then you get a 100% guarantee that you won't miss any important information.

You should be able to check the status of ongoing tasks and historical events. The tasks section gives you a clear view of actions in progress with detailed information, so it's just a glance to check on running operations.

Finally, your GitHub backup software should provide you with advanced audit logs. Logs contain allinformation about the work of applications, services, created backups, and restored data. Moreover, you see which actions are performed by each admin and can prevent any intentional malicious activity.

To make monitoring even easier and non-engaging you should have the possibility to attach those audit logs to your external monitoring systems and remote management software (i.e. PRTG) via webhooks and API.

All of this should be accessible through one central management console where you manage backup, restore, monitoring, and all the system settings. Powerful visual statistics, data-driven dashboards, and real-time actions should combine ease of use, and precise management to save your time.

GitProtect.io is the only GitHub backup and recovery software on the market that ensures you with an all-in-one solution managed in one central management console.

Create a dedicated GitHub user account only for backup reasons to bypass throttling

The best practice for big, enterprise users is to create a dedicated GitHub user account that will be connected to GitHub backup software and responsible only for backup purposes (ie. backup@companyname.com). It is due to two reasons - but security first. It means that this user should have access only to repositories it aims to protect. It also helps to bypass throttling - each GitHub user has his own pool of requests to the GitHub API - so every application associated with this account operates on the same number of requests. Thus, the separate user enables them to bypass these limits and perform backup smoothly without any queuing or delay.

If you manage a big organization and numerous repositories it is good to have even several GitHub users dedicated to backup purposes within your GitHub account - when the first one exhausts the number of requests to the API, the next one is automatically attached, and so on. Then the backup of even the biggest GitHub environment performs uninterruptedly.

PART II
BACKUP SECURITY

GitHub backup software for SOC2, ISO 27001 compliance

For most companies, security has proven to be a major concern nowadays. Source code is probably the most sensitive data for any IT-related organizations. That's why your repository and metadata backup should provide you with numerous security features that can significantly ensure data accessibility and recoverability, improve your security posture, and help you meet your shared responsibility duties. In short: it should allow you to empower your teams while staying on top of regulatory standards.

The software provider and Data Center your service is hosted should have all world-class security measures, audits, and certificates in place.

What security issues are worth paying special attention to?

- AES encryption with your own encryption key (more)
- In-flight or at rest encryption (more)
- Long-term, unlimited, flexible retention (more)
- Possibility to archive old, unused repositories due to legal requirements (more)
- Easy monitoring center (more)
- Multi-tenancy, possibility to add additional admins and assign privileges (more)
- Data Center strict security measures (more)
- Ransomware protection (more)
- Disaster Recovery technologies (more)

User AES encryption in-flight and at rest

We can not talk about data protection at all, without proper, reliable encryption. Your data should be encrypted on every stage - in-flight so on your device, before it even leaves your machine, during the transfer, and finally at rest in the repository. Only then you get the guarantee that no matter when data

might be intercepted, they cannot be decrypted.

Make sure that your software has Advanced Encryption Standard (AES). AES is a symmetric-key algorithm, meaning the same key is used for both encrypting and decrypting the data. AES is considered unbreakable and is widely used by many governments and organizations.

In the perfect scenario, you should have a choice on encryption strength and level, considering:

Low - Forces the AES algorithm to work in OFB (OUTPUT FEEDBACK) mode with an encryption key of 128 bits.

Medium - as in the case of 'Low' encryption strength, the AES algorithm is run in OFB mode, but the key used is the encryption encryptor is twice longer - it consists of 256 bits.

High - with this option selected, AES will work in CBC (CIPHER-BLOCK CHAINING) mode, and the encryption key is 256 bits long.

It is worth adding that depending on the selected encryption method, the backup time will vary and the load on the end device or selected functionalities may be limited - thus you should have a choice. As you see, all levels refer to AES encryption, still considered unbreakable.

During the encryption configuration, you must provide a string of characters on the basis of which the encryption key will be built. This string should be known only by you and saved in the password manager preferably.

The essence of strong encryption is your own encryption key. Most providers create encryption keys to secure user data. GitProtect goes one step ahead – to enforce your data security our solution enables you to create custom encryption keys.

Moreover, you will be able to use your own Vault to provide us with your key only while performing a backup and make sure you have even stronger control over your access and credentials.

Zero-knowledge encryption

Your device should not have any information about the encryption key - it should receive it only when performing a backup so no one, except the key owner (you by default), is able to decrypt it. This approach in the security industry is called zero-knowledge encryption. When checking for a reliable backup

software make sure that it has all AES data encryption with your own encryption key and zero-knowledge infrastructure.

Data Center region of choice

For every security-oriented business, it is critical to know how your data is stored and managed. The Data Center location of your backup software provider should be relevant to you - it might impact coverage, application availability, and uptime. Thus, you should have a choice where you want to host your software and store your data alternatively. GitProtect.io gives you this choice from the very beginning - after signup, you will be asked to decide whether to store your management service in an EU or US-based Data Center.

However, whichever Data Center you choose, make sure it is compliant with strict security guidelines and meets standards and certifications such as ISO 27001, EN 50600, EN 1047-2 standard, SOC 2 Type II, SOC 3, FISMA, DOD, DCID, HIPAA, PCI-DSS Level 1and PCI DSS, ISO 50001, LEED Gold Certified, SSAE 16.

What else should you pay attention to? Physical security, fire protection and suppression, regular audits, and 24x7 technical and network support.

Sharing the responsibility for managing the backup system

In every business area sharing responsibility among employees helps you perform faster, increases team morale, and lets you focus on a big picture. Your GitHub backup software should let you add new accounts, set roles, and privileges to delegate responsibilities to your team and administrators, and have more control over access and data protection.

It is only possible with a central management console and easy monitoring. You should know exactly what actions are performed in the system and who made concrete changes - and thus you should have access to insightful and advanced audit logs.

Ransomware protection

Backup is the final line of defense against ransomware so it should be ransomware-proof itself. What does it mean? Please pay attention to how the backup vendor processes your data. GitProtect.io compresses and encrypts your data which keeps it unexecutable on the storage. It means that even if ransomware will hit your backed-up data, it can not be executed and spread on the storage.

The authorization data for storage and GitHub are stored in Secure Password Manager, and in the case of on-premise instances, the agent receives them only for the duration of the backup. So if ransomware hits the machine our agent is on, it won't have access to authorization data and storage.

Finally, even if ransomware will encrypt your GitHub data, you should be able to restore a chosen copy from the exact point in time and get back to coding immediately.

Take notice if a backup vendor offers you immutable, WORM-compliant storage technology that writes each file only once and reads it many times. It prevents data from being modified or erased and makes them ransomware-proof.

Part III
DISASTER RECOVERY

Disaster Recovery - use cases & scenarios

When choosing the right backup and recovery software for your GitHub repositories and metadata you have to make sure that its Disaster Recovery technology responds to every possible data loss scenario. Most of the vendors give you some kind of data recoverability only in the case when GitHub is down - but there are more dangerous situations. Let's check on how GitProtect.io prepares you for every scenario possible but first, take a quick look at data restore options.

Recovery features in a nutshell:

- Point-in-time restore
- Granular recovery (of repositories and only selected metadata)
- Restore to the same or new repository/organization account
- Cross-over recovery to another Git hosting platform (ie. from GitHub to Bitbucket and conversely)
- Easy data migration between platforms
- Restore to your local device

Unlike other vendors, to restore the data you don't need any additional app - GitProtect.io is a complete backup & recovery software for your DevOps ecosystem with one central

Use case #1

What if GitHub is down?

GitHub outage is one of those situations in which you urgently need to recover your data to ensure the continuity of your team's work. In such a situation, you can instantly restore your entire GitHub environment from the last copy or a chosen point in time to your local machine as .git, to your GitHub local instance, or cross over to another git hosting platform - GitLab or Bitbucket and keep your team working uninterruptedly.

Use case #2

What if your infrastucture is down?

The best, indisputable backup practice is the 3-2-1 backup rule which becomes a widely-adopted standard in data protection. It states that you should have at least 3 copies on 2 different storage instances, including at least 1 in the cloud. GitProtect.io is a multi-storage system that enables you to add an unlimited number of storage instances (on-premise, cloud, hybrid, or multi-cloud) and make backup replication among them. Moreover, it offers you free cloud storage in case you need a reliable, second backup target. Then you are sure that even if your backup storage is down (ie. there is an AWS outage) you can restore all or chosen data from any point in time from your second storage.

Use case #3

What if GitProtect's infrastructure is down?

We live from data protection - that's why we need to be prepared for every potential outage scenario - especially the one harming our infrastructure. When our environment is down, we will share with you the installator of your on-premise application. All you need to do is to log in, assign your storage where your copies are stored so you have access to all your backed up data, and use all data restore and Disaster Recovery options mentioned above.

Restore multiple git repositories at a time

Downtime? Service outage? There are many situations when you might need to instantly restore your entire Git environment. Restore and Disaster Recovery technologies are decision-makers. After all, backup is done so that in the event of a failure, it can be quickly restored. The easiest way to accomplish this is the possibility to restore multiple GitHub repositories at a time. So you can just choose repositories you want to restore, see the most recent copies or assign them manually and restore them to your local machine, or recover cross over to another hosting service provider and make your Disaster Recovery plan easy, fast and efficient.

Point-in-time restore - don't limit yourself to the last copy

As you probably know, human errors are one of the most common reasons for cybersecurity risks and data losses. It is no different in the

case of git repository backup - from in- or unintentional repository or branch deletion to HEAD overwrite - you never know when and where the risk is hidden. Once you define the exact state and date you want to roll out, it would be crucial to have the possibility to restore your GitHub backup from a very specific and defined moment in time. Please note that most backup vendors offer you to restore only the last copy or the copy from up to 30 days prior.

What if you realize some serious change in your source code after let's say 50 days after occurring? You would have to go back to some extra time prior. Then you need to make sure that your GitHub backup software offers you point-in-time restore together with unlimited retention options (over default 30 days or 365 days periods) - here you can find it is described. Then you can use such software for GitHub archive reasons, overcome GitHub storage limitations, ensure legal compliance, and have data recoverability no matter when the threat or mistake is disclosed.

Restore directly to your local machine

Even if you work on GitHub in SaaS at some point you might want to restore copies to your local machine. Cloud infrastructure downtime, service outage, or weak internet connection - just to name a few. So among other restore possibilities, your GitHub backup software should provide you with the option to restore your entire git environment to the local machine.

On the other hand, please note if your software provides you with some additional options like restoring to the same or new GitHub repository as well as crossover recovery (to another git hosting service - ie. GitLab or Bitbucket). You can never predict all scenarios to make sure which option is the best for your organization - thus, better have them all.

No overwriting of repositories during the restore process

When you want to restore your repository from the copy, it is good to have it restored as a new repository instead of overwriting the original one. First of all, you might want to use the original one to track changes or keep it for future reference. But first of all, it is important from the security point of view. Moreover, you have full control over your data and you are a decision-maker when it comes to keeping or deleting your repositories.

Summary

According to official numbers in June 2022, GitHub reported having over 83 million developers and more than 200 million repositories (including at least 28 million public repos). Here at GitProtect.io, 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 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