BackUpScale - Restic As a Service

Backups have never been a more vital component of a company's data integrity planning. Cyberattacks have become so easy to deploy that, for example, ransomware as a service provides a viable business model for malicious actors, even those without technical skills. If you don't have a backup process in place that recognizes the changing landscape of attacks, your data may be at risk.

Strong backups lie at the heart of any data recovery strategy. However, if incorrectly implemented, they too can be targeted by the current approaches to malware and ransomware. If your backups are not correctly isolated from the system they are protecting, they may be swept up in an attack. The last thing you need is to attempt to restore, only to discover that your backups have also been held for ransom.

Any backup solution that you choose for your organization should have several key characteristics. At a minimum, backups need to be automated, secured, and restorable. The historic ideal has been an "air gap," which is to say that the files are stored on a physical medium with no connection to the original servers. This is time consuming, requires off site storage, and demands regular manual intervention, but it is still the gold standard for large enterprises with the necessary resources.

The goal of BackUpScale is to provide smaller companies and organizations with a less labour-intensive automated solution. We emulate the benefits of remote storage by exporting the backups into a highly-secured environment.

For this purpose, we have developed an integrated and automated implementation of Restic, which is one of the most advanced backup systems available.

Restic's architecture is designed for ease of use, high throughput, verification of backups, security, and efficiency of storage. It uses chunking and deduplication to reduce file size. Its ability to connect to the cloud storage of your choice, notably object storage services (which are significantly cheaper), allows you to keep backup storage costs contained. Additionally, everything is client-side encrypted and transferred via TLS, so even if the backups were stolen, they would be of no use to anybody without the key.

As a final step in the BackUpScale implementation, all backups are set to append-only mode, which means that your backups cannot be overwritten or deleted, even in the event that your initial system is breached.

Security Considerations for Unattended Backups

For backups to be reliable, they need to run consistently, at scheduled intervals tuned to the particular requirements of the organization's data, which are sensitive to:

- Frequency of changes
- How critical the data is to the organization
- Regulatory requirements
- Costs of potential data loss

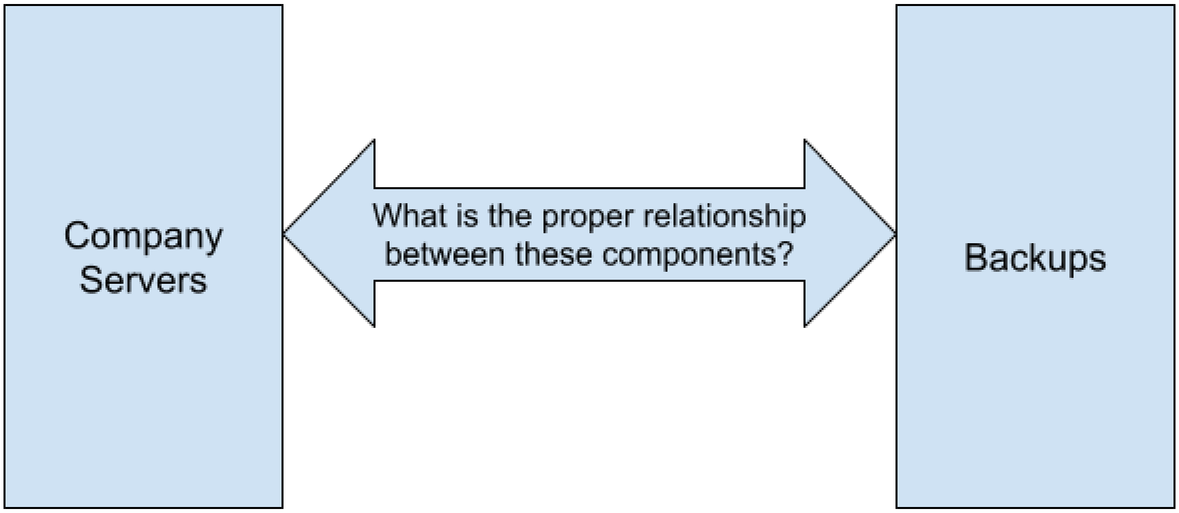
Leaving the recurring tasks to humans, the manual method, is simply not reliable. Data is missed, it is backed up infrequently, or the wrong things are prioritized and the security and integrity of the overall system is left to chance.

Decisions about frequency and completeness are strategic, and are likely to vary, even within an organization. A full assessment of the risks and how to prioritize the prevention of data loss is a major undertaking that should be done relatively infrequently. The backups themselves, however, may need to be current to the most recent week, day, or hour. *Running* them, once the strategy has been developed, is an ideal task for automation.

Without the luxury of an air-gap, though, this automation goal introduces a new consideration for backup systems. Every connection between computers becomes a possible source of intrusion and malicious software. The BackUpScale architecture is designed to minimize the attack surface while still maintaining the ability to let your backups run silently in the background, confident that they will be intact when you need them.

The Problem in a Nutshell

How do you protect the integrity of the backed-up data from potentially compromised systems you're backing up? And how do you protect your systems from a potentially compromised backup service, which could have access to all of your servers?



This diagram lays out the beginning of the question. How, in a connected world, do we

1. keep the Servers and their Backups in sync with each other,

2. minimize the risk that either end is compromised
3. Isolate compromised components from one another

If the company servers are attacked, we need the backups to be

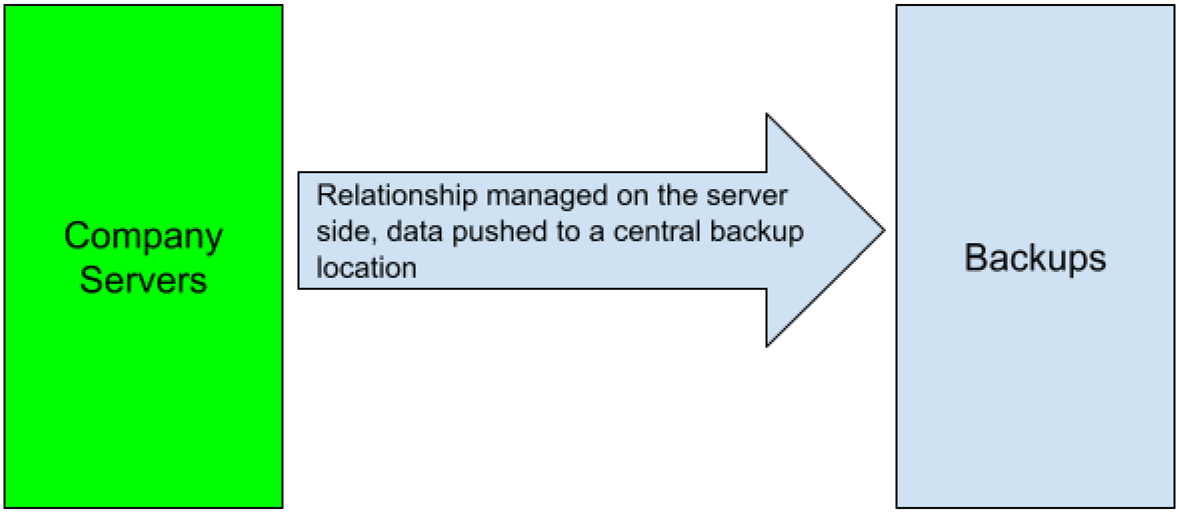
- easily recovered,
- relatively current, and
- Intact

Let's review the classic strategies for backing up server data.

The "Push" Method

The "push" method involves pushing data from individual servers to a central backup server.

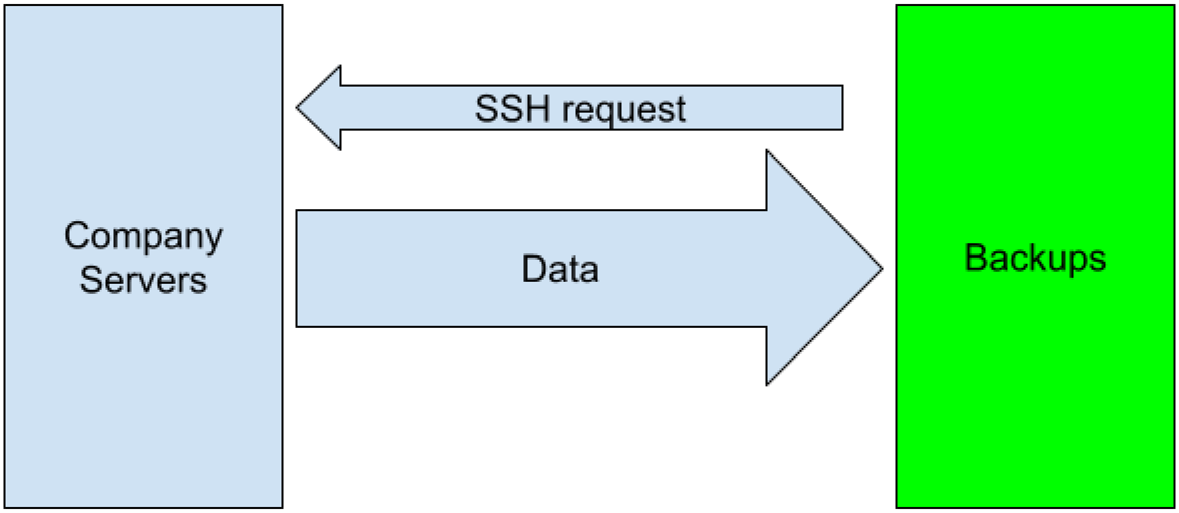
The benefit here is that a compromised backup server doesn't have access to authoritative data on the servers it backs up because *they* push to *it*, not the other way around. We're assuming, of course, that your enterprise's firewall rules are configured to only allow servers to talk to each other as required.


Company Servers
Relationship managed on the server side, data pushed to a central backup location
Backups

The problem, if you can see it already, is that each individual server has access to the backup server. If *any* of them get compromised, say due to a ransomware attack, the attackers can do the same to the backup server. You would then lose not only your authoritative data, but your backups as well.

The "Pull" Method

With the "pull" method, the individual servers don't get access to the backup server. The backup server is provided with read-only access to the servers to be backed up, pulls the data, and then saves it. This method protects the backups: if the servers with authoritative data are compromised by an attack, they can't access the backup server themselves, and they are therefore unable to initiate an attack on the backups.



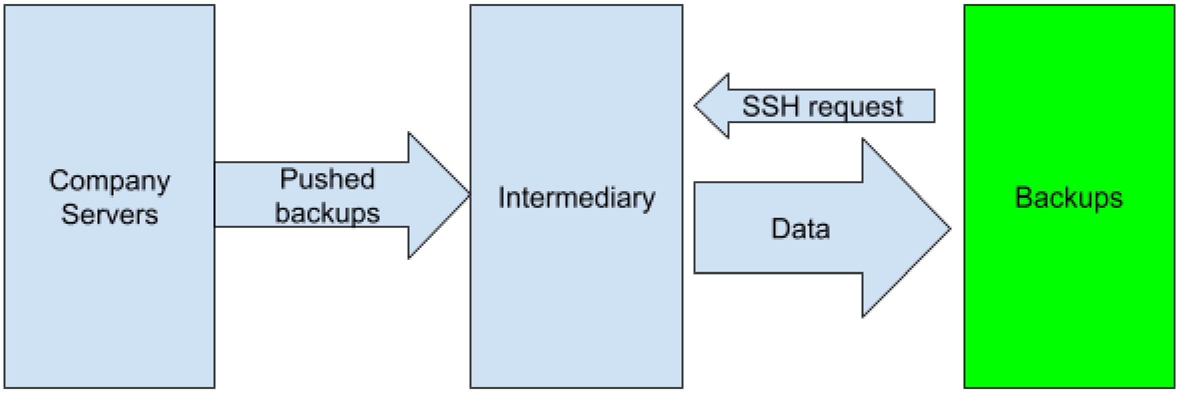
The problem here is that a compromised backup server can access its clients, and compromise them as well. Additionally, there is no way for the backup server to detect whether the data on the main server has been tampered with, so if there is an attack and the data is pulled on schedule, it can still damage the backups.

This method isn't ideal either.

A Hybrid architecture

As we can see, neither of the above methods provide good security. In either case, an infection can spread.

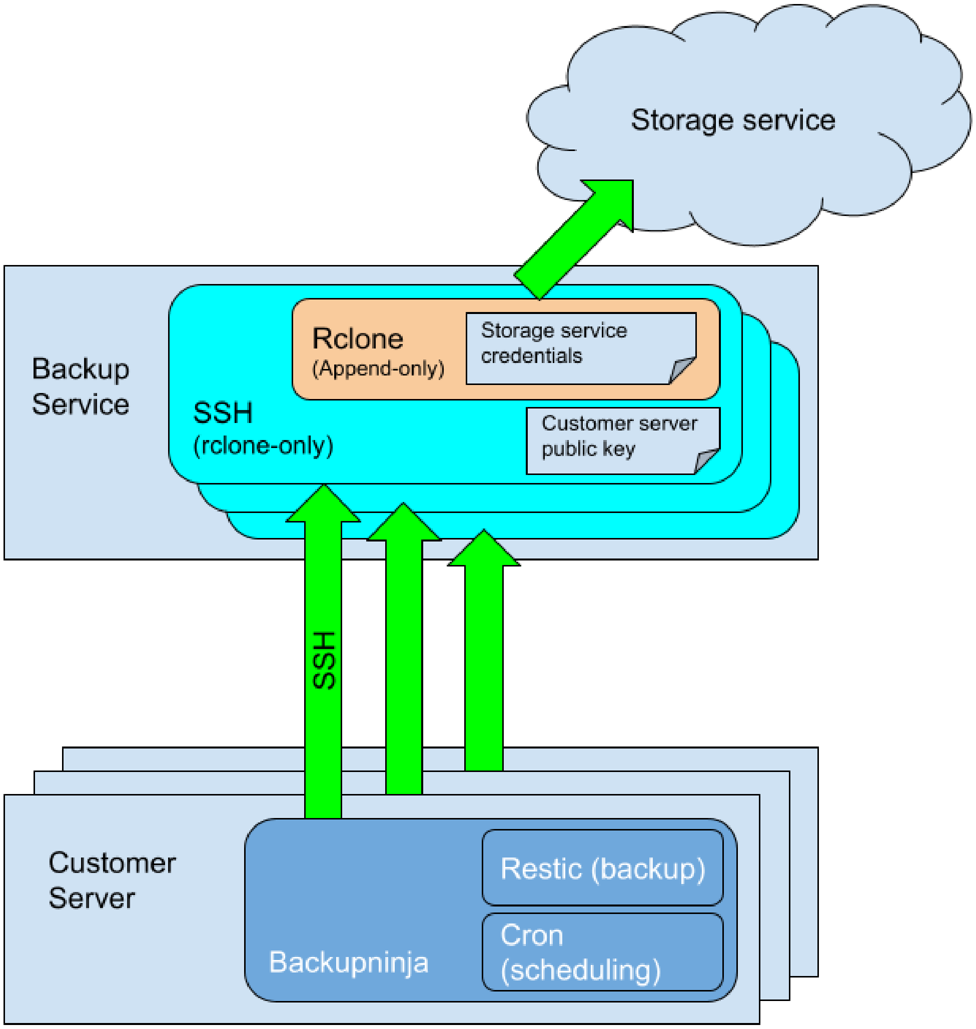
One way around this is to combine both strategies. Push backups from data servers to the backup server, and then use a third system to pull data from there. The backed up servers won't have access to the third system, and the third system can't access them. This method is described in a GitHub issue.



In this case, no connection is ever established between the backup server and the company servers, so they are isolated from one another, and any infection is contained.

While it's sound from a security perspective, it requires yet another system, so one would actually need two different kinds of backup servers. It is possible, however, to get the same security with just one.

Architecture



Backup Service

By introducing an intermediate stage between the company server and the backup storage, BackUpScale provides a barrier between the two systems without the burden of installing yet another kind of backup server. Instead, we pipe encrypted backups from company servers to an instance of Rclone, which is the cloud-storage engine for Restic. Rclone preserves timestamps and verifies checksums, ensuring the integrity of the backup files. It allows the use of a wide range of different cloud storage solutions with simple configuration changes.

This intermediate server will house *only* the credentials required for communicating with the storage service. As such, it will not need to store application data locally. It is also a highly locked-down server, accepting only one form of incoming connection. The firewall on the backup service is configured to allow incoming SSH connections from the servers that wish to be backed up, and reject all other incoming connections.

The server will make Rclone available over SSH, with key-access only. The enforced shell command that runs on successful logins is the Rclone executable itself, to prevent client connections from running any other commands, and locked down even further with an append-only configuration. This will prevent connecting users from altering the existing stored data, or removing it entirely.

This idea was discussed in the Restic feature request Support asymmetric backups, and is documented in Preparing a new repository: Other Services via rclone. It gives us exactly what we want:

- A single backup server,
- the inability of the backup server to connect to the clients, and
- a guarantee that the clients can only append data to the backup service when writing to it; they cannot alter existing data.

Preparing Customer Servers

Customer servers, data servers that will be clients of the backup server, will be pushing their backup data. As such, they need to have the appropriate software installed and set up for each server in the system.

A complete installation includes:

- the Restic application
- Backupninja which handles scheduling and maintains the list of files users wish to back up
- SSH keys that have been authorized on the other end
- outgoing SSH connections to the backup server enabled on the firewall

There is a manual step in this process for setting up the lists of files and databases to be backed up. Once that is completed, Backupninja runs Restic at the configured times, sending the desired snapshots to remote storage. It can also perform any necessary preprocessing, such as dumping databases.

Restic performs all preprocessing on the servers being backed up, so it is most appropriate for larger machines that are powerful enough to do the extractions and then run the deduplication and encryption processes.

Robust Backups. Automated. At Scale

Building on cutting-edge open-source technologies, BackUpScale eliminates many of the barriers to implementing a robust off-site backup solution. Backups are secured using an append-only strategy that preserves your backups as unalterable tamper-proof objects in a highly locked-down environment. BackUpScale also integrates with the remote storage of your choice for flexibility, cost containment, and ease of use.

We are currently accepting applicants for a beta round.

For more information and a quote, please contact us https://backupscale.com/contact/