EVIDEN

Evidian SafeKit, High Availability Software Real-time replication Load balancing Failover

Contents

The ideal product for a software publisher

“SafeKit is the ideal application clustering solution for a software publisher. We currently have deployed more than 80 SafeKit clusters worldwide with our critical TV broadcasting application.”

The product very easy to deploy for a reseller

”SafeKit is a professional solution making easy the redundancy of Milestone video surveillance server. The solution is easy to deploy, easy to maintain and can be added on existing installation.”

The product to gain time for a system integrator

”Thanks to a simple and powerful product, we gained time in the integration and validation of our critical projects like the supervision of Paris metro lines (the control rooms).”

High availability, business continuity, disaster recovery

Every computer-system-based activity, both big and small, is one day or the other faced with the problem of computer failure. Unfortunately, the day the failure occurs, a small problem may turn into a general crisis if no high availability solution has been implemented.

10 reasons to choose the SafeKit clustering software

1. Software-only high availability solution

Evidian SafeKit is a software-only high availability solution. This solution secures easily and quickly the 24x7 operation of your critical applications.

While traditional high availability solutions are focused on the hardware failover of physical servers, SafeKit has chosen to focus on the hardware and software failover of critical applications.

2. High availability which targets all types of failures

The unavailability of an application can be due to 3 types of problems:

- Hardware and environment: including the complete failure of a computer room (20%).
- Software: regression on software update, overloaded service, software bug (40%).
- Human errors: administration error and inability to properly restart a critical service (40%).

SafeKit addresses these issues, which are all essential to ensure the high availability of critical applications.

The 3 best use cases of software clustering

After over 20 years of 24x7 experience, SafeKit is the preferred clustering solution on the market in three cases:

1. A software publishing company can add SafeKit to its application suite as a software OEM high availability option.
2. A distributed enterprise can deploy a high availability solution on standard hardware without the need for specific IT skills.
3. A data center can provide high availability for multiple applications with a uniform solution on Windows or Linux and with load balancing, real time data replication and failover between two remote sites.

4. Unique on the market: 3 products in 1

Traditionally, three different products are necessary to create an application cluster:

- load balancing network boxes,
- disk bays replicated synchronously on a SAN for data availability,
- high-availability toolkits for application failure recovery.

SafeKit offers these three features within the same software product.

To further reduce implementation costs, SafeKit runs on your existing physical or virtual servers and with the standard editions of OS and databases: Windows, Linux, Microsoft SQL Server, Oracle, Firebird, MariaDB, PostgreSQL or other databases or flat files... and even with Windows editions for PCs!

5. A solution suited for Cloud environments

Application high availability with SafeKit can be deployed in AWS, Azure and Google clouds as well as on premise on physical or virtual machines. Redundancy of Docker applications is also supported.

6. Full virtual machines replication and failover

SafeKit also offers replication and failover of full virtual machines between two active Hyper-V or KVM physical servers . The solution is simple and economical because it requires no shared disk.

7. Plug and play deployment of a software cluster

Once a failover module is configured and tested for an application, deployment requires no specific IT skills . Just install the application, the SafeKit software and the failover module on two standard Windows or Linux servers.

8. Rich choice of application integration inside a software cluster

SafeKit proposes different types of software clusters . Cluster configuration for a given application is rich and is made with one or several application modules . SafeKit proposes mirror modules (primary/ secondary with replication and failover), farm modules (network load balancing and failover), and mixed of several modules that can be implemented on the same cluster or on different clusters.

A module is configured with the server IP addresses for heartbeats, the virtual IP address of the cluster, the load balancing rules for a farm module, the file directories to replicate for a mirror module, the hardware and software failure detectors and the service to restart in case of failure.

9. User-friendly administration to avoid human error

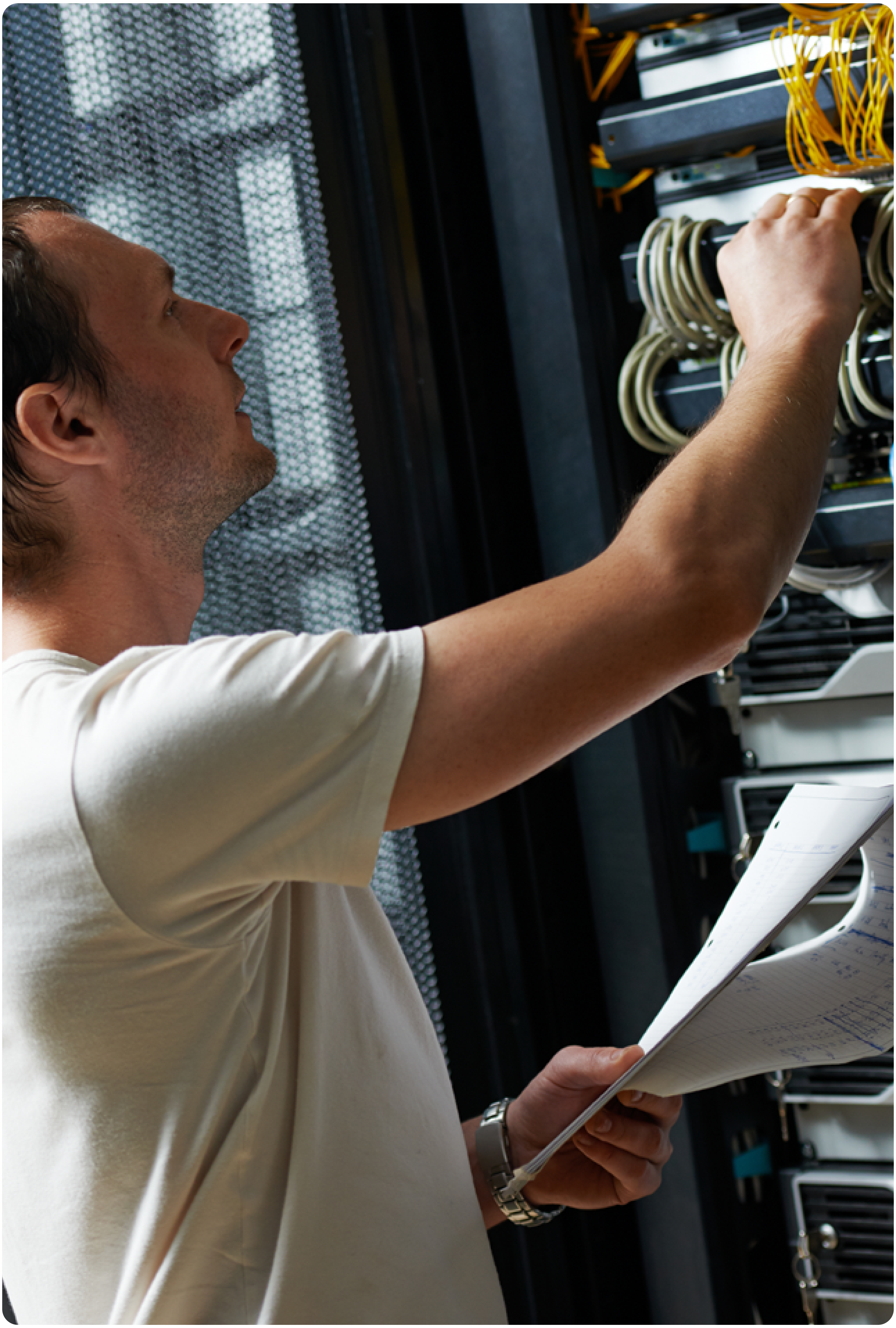
SafeKit provides a centralized administration web console. An administrator can remotely monitor status of applications on different clusters and act with simple buttons (start, stop).

You can test SafeKit for free. In less than 1 hour, you can implement your first software cluster on two virtual or physical machines thanks to the administration console.

10. Synchronous replication for transactional applications

SafeKit's synchronous real time replication function strengthens high availability and prevents data loss. With this mechanism, a data committed on a disk by a transactional application is replicated on the secondary machine.

Application servers can be located in geographically remote computer rooms through an extended LAN to withstand the loss of a full room.



Integration – Deployment - Architectures

Integration via application modules

An application module is a customization of SafeKit for an application. There are two types of modules: the mirror module with real-time data replication and failover and the farm module with load balancing and failover.

1. In practice, an application module is a .safe file (zip type) including:
   - the configuration file userconfig.xml which contains:
   - names or physical IP addresses of the servers,
   - name or virtual IP address of the cluster,
   - file directories to replicate in real time (for a mirror module),
   - network load balancing criteria (for a farm module),
   - configuration of software and hardware failures detectors,
2. the scripts to start and stop the application.

Architectures: the different software clusters

SafeKit offers two basic clusters:

- Mirror cluster built by deploying a mirror application module on 2 servers,
- Farm cluster built by deploying a farm application module on 2 servers or more.

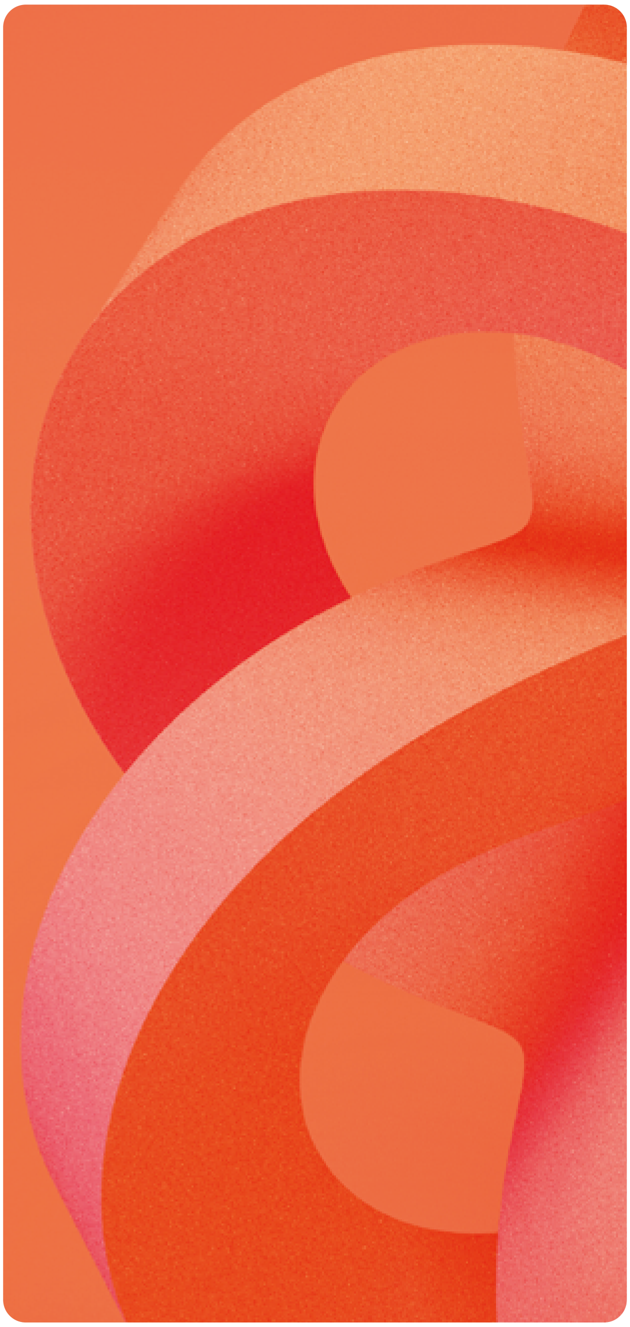
Several application modules can be deployed on the same cluster. Thus, advanced clustering architectures can be implemented:

- Farm+mirror cluster with the deployment of one farm module and one mirror module on the same cluster,
- Active/active cluster with the deployment of several mirror modules on 2 servers,
- Hyper-V or KVM cluster with replication and failover of full VMs between 2 active physical servers,
- N-1 cluster with the deployment of N mirror module on N+1 servers.

Plug and play deployment

Once an application module is configured and tested with an application, deployment requires no specific IT skills:

1. install the application on 2 standard servers (physical or virtual),
2. install the SafeKit software on both servers,
3. install the application module on both servers.



The SafeKit mirror cluster

High availability cluster with real time file replication and application failover

The mirror software cluster is an active-passive high-availability solution. The application runs on a primary server and is restarted automatically on a secondary server if the primary server fails.

The mirror cluster can be configured with or without file replication. With its file-replication function, this architecture is particularly suited to providing high availability for back-end applications with critical data to protect against failure.

Microsoft SQL Server.safe, PostgreSQL.safe and Oracle.safe are examples of «mirror» type application modules. You can write your own mirror module for your application, based on the generic module Mirror.safe.

A demonstration of a mirror cluster with Microsoft SQL Server is presented here.

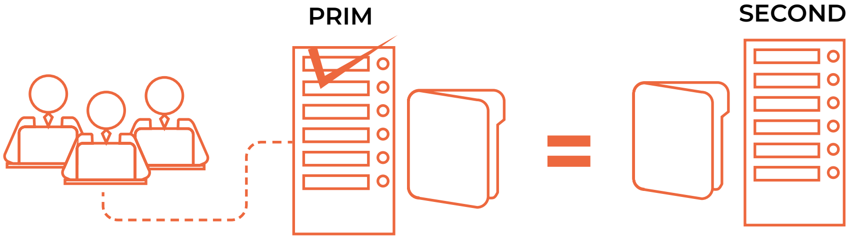
The mirror software cluster works as follows.

Step 1. Normal operation

For replication, only names of file directories are configured in SafeKit. There are no pre-requisites on disk organization for the two servers. Directories to replicate may be located in the system disk.

Server 1 (PRIM) runs the application.

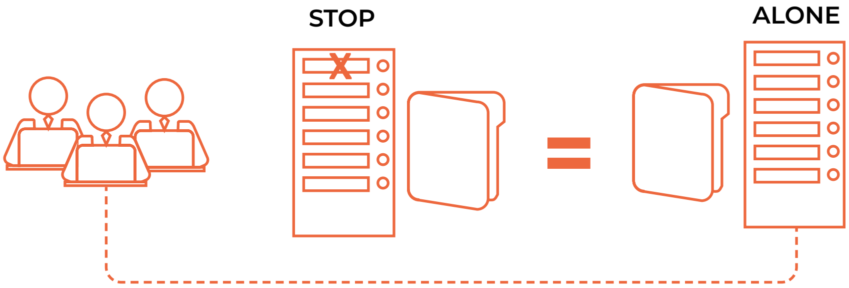
SafeKit replicates files opened by the application. Only changes made by the application in the files are replicated in real time across the network, thus limiting traffic.



Step 2. Failover

When Server 1 fails, Server 2 takes over. SafeKit switches the cluster's virtual IP address and restarts the application automatically on Server 2. The application finds the files replicated by SafeKit uptodate on Server 2, thanks to the synchronous replication between Server 1 and Server 2. The application continues to run on Server 2 by locally modifying its files that are no longer replicated to Server 1.

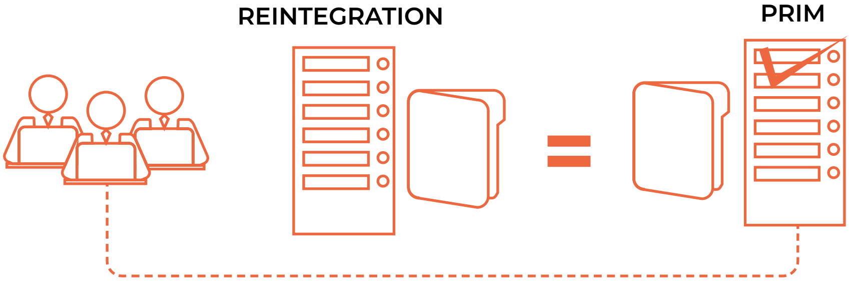
The switch-over time is equal to the fault-detection time (set to 30 seconds by default) plus the application start-up time. Unlike disk replication solutions, there is no delay for remounting file systems and running recovery procedures.



Step 3. Failback and reintegration

Failback involves restarting Server 1 after fixing the problem that caused it to fail. SafeKit automatically resynchronizes the files, updating only the files modified on Server 2 while Server 1 was halted.

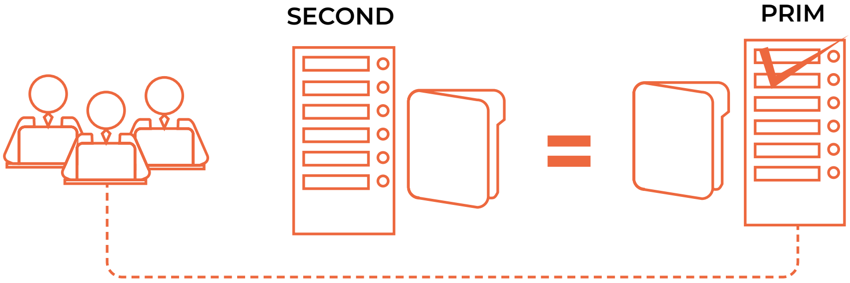
This reintegration takes place without disturbing the applications, which can continue running on Server 2. This is a major feature that differentiates SafeKit from other solutions, which require you to stop the applications on Server 2 in order to resynchronize Server 1.



Step 4. Return to normal operation

After reintegration, the files are once again in mirror mode, as in step 1. The system is back in high-availability mode, with the application running on Server 2 and SafeKit replicating file updates to the secondary Server 1.

If the administrator wishes the application to run on Server 1, he/she can execute a "swap" command either manually at an appropriate time, or automatically through configuration.



Synchronous replication versus asynchronous replication

There is a significant difference between synchronous replication, as offered by the SafeKit mirror solution, and asynchronous replication traditionally offered by other file replication solutions.

With synchronous replication, when a disk IO is performed by the application or by the file cache system on the primary server inside a replicated file, SafeKit waits for the IO acknowledgement from the local disk and from the secondary server, before sending the IO acknowledgement to the application or to the file system cache. This mechanism is essential for recovery of transactional applications.

The bandwidth of a LAN between the servers is required to implement synchronous data replication, possibly with an extended LAN in two geographically remote computer rooms.

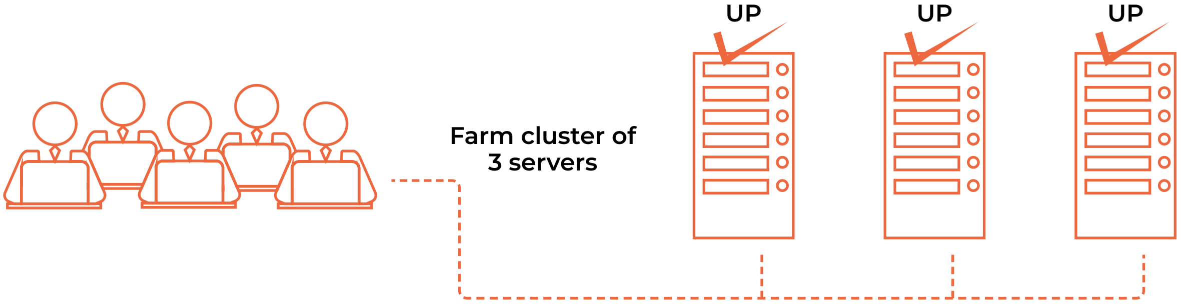
With asynchronous replication implemented by other solutions, the IOs are placed in a queue on the primary server but the primary server does not wait for the IO acknowledgments of the secondary server. So, all data that did not have time to be copied across the network on the second server is lost if the first server fails.

In particular, a transactional application loses committed data in case of failure. Asynchronous replication can be used for data replication through a low-speed WAN, to back up data remotely.

SafeKit provides a semi-synchronous solution, implementing the asynchrony not on the primary machine but on the secondary one. In this solution, SafeKit always waits for the acknowledgement of the two machines before sending the acknowledgement to the application or the system cache. But on the secondary, there are 2 options asynchronous or synchronous. In the asynchronous case, the secondary sends the acknowledgement to the primary upon receipt of the IO and writes to disk after. In the synchronous case, the secondary writes the IO to disk and then sends the acknowledgement to the primary. The synchronous mode is required if we consider a simultaneous double power outage of two servers, with inability to restart the former primary server and requirement to restart on the secondary.

The SafeKit farm cluster

Scalability and high availability with network load balancing and application failover



The farm software cluster provides both network load balancing, through transparent distribution of network traffic, and software and hardware failover. This architecture offers a simple solution for increasing system load.

The same application runs on each server, and the load is balanced by the distribution of network activity on the different servers of the farm.

Farm clusters are suited to front-end applications like web services.

Apache_farm.safe and Microsoft IIS_farm.safe are examples of farm application modules. You can write your own farm module for your application, based on the generic module Farm.safe.

A demonstration of a farm cluster with Apache is presented here.

Principle of a virtual IP address with network load balancing

The virtual IP address is configured locally on each server of the farm. The input traffic for this address is split among them at low level by a filter inside each server's kernel.

The load balancing algorithm inside the filter is based on the identity of the client packets (client IP address, client TCP port). Depending on the identity of the client packet input, a single filter instance in a server farm transmits the packet to the upper network layers; the other filter instances in other servers drop it. Once a packet is accepted by the filter on a server, only the CPU and memory of this server are used by the application that responds to the request of the client. The output messages are sent directly from the application server to the client.

If a server fails, the SafeKit membership protocol reconfigures the filters in the farm to re-balance the traffic on the remaining available servers.

Load balancing for stateful or stateless web services

With a stateful server, there is session affinity. The same client must be connected to the same server on multiple TCP sessions to retrieve its context on the server. In this case, the SafeKit load balancing rule is configured on the client IP address. Thus, the same client is always connected to the same server on multiple TCP sessions. And different clients are distributed across different servers in the farm. This configuration is used when there are session affinities.

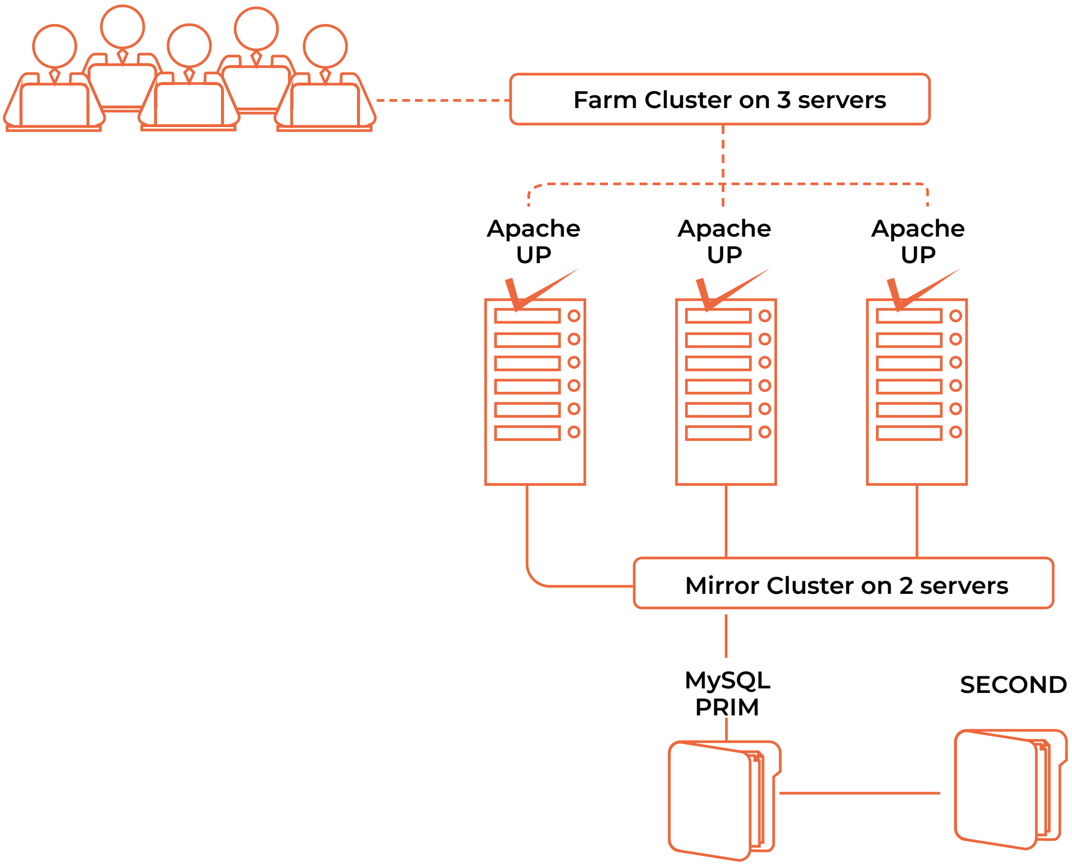
With a stateless server, there is no session affinity. The same client can be connected to different servers in the farm on multiple TCP sessions; because there is no context stored locally on a server from one session to another. In this case, the SafeKit load balancing rule is configured on the TCP client session identity. This configuration is the one which is the best for distributing sessions between servers, but it requires a TCP service without session affinity.

The SafeKit farm+mirror cluster

Network load balancing, file replication and application failover

You can mix farm and mirror application modules on the same cluster of servers.

This option allows you to implement a multi-tier application architecture, such as Apache_farm.safe (farm architecture with load balancing and failover) and MySQL.safe (mirror architecture with file replication and failover) on the same application servers.

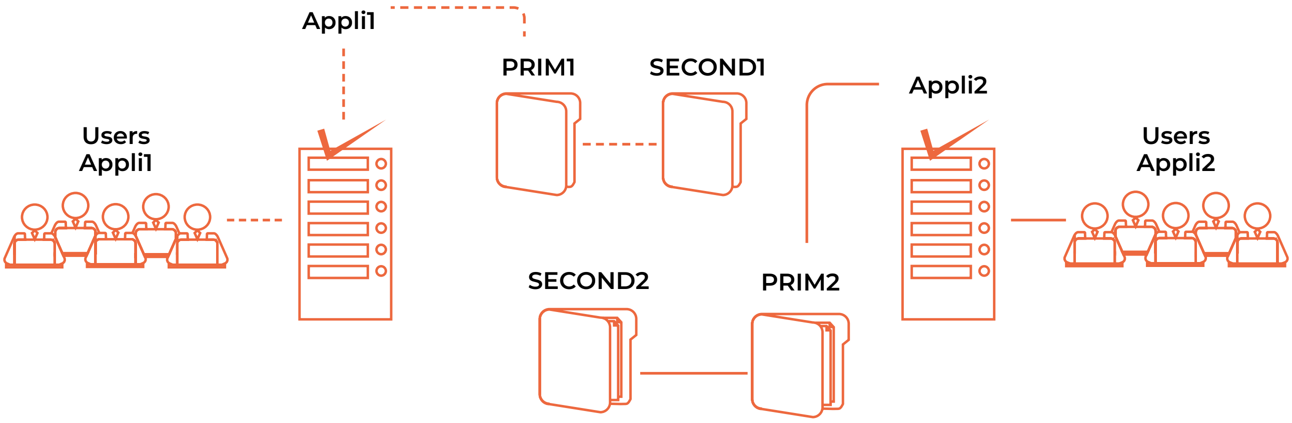


As a result, load balancing, file replication and failover are managed coherently on the same servers. Specific to SafeKit, this mixed cluster is unique on the market!

The SafeKit active/ active cluster

Crossed replication and mutual takeover

In an active / active cluster, there are two servers and two mirror application modules in mutual takeover (Appli1.safe and Appli2.safe). Each application server is backup of the other server.



If one application server fails, both applications will be active on the same physical server. After restart of the failed server, its application will return to run on its default primary server.

A mutual takeover cluster is a more economical solution than two separate mirror clusters, because there is no need to invest in backup servers that will spend most of their time sitting idle waiting for the primary server to fail. Note that during a failure, the remaining server has to be able to handle the combined workload of both applications.

Note that:

- the 2 applications Appli1 and Appli2 must be installed on each server for application failover,
- this architecture is not reduced to 2 applications: N application modules can be deployed on 2 servers,
- each mirror module will have its own virtual IP address, its own replicated file directories and its own recovery scripts.

The SafeKit Hyper-V or KVM cluster

Load balancing, replication and failover of full virtual machines

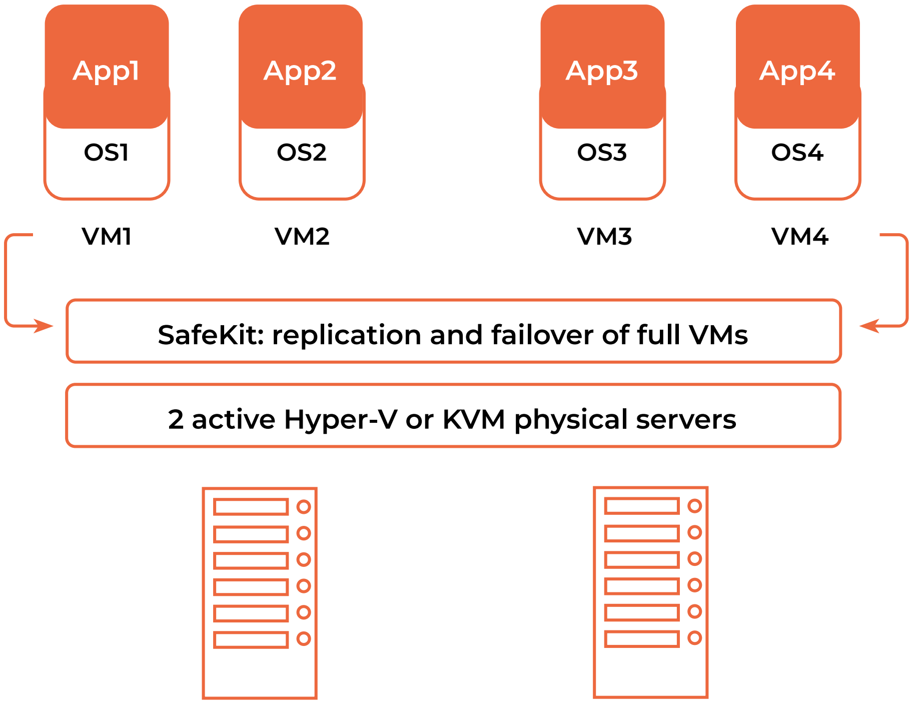
The Hyper-V or KVM cluster is an example of an active-active cluster with several mirror modules. Each virtual machine is integrated inside a mirror module. The solution has the following features:

- a full synchronous real-time replication of a virtual machine with failover,
- a load balancing of virtual machines between 2 Hyper-V or KVM servers with crossed replication,
- a centralized and ergonomic console to manage failover of VMs,
- a very interesting offer for a reseller with zero integration with applications,
- an interesting architecture for HA solutions which cannot be integrated at the application level.

A demonstration of the Hyper-V cluster with SafeKit is presented here.

A demonstration of the KVM cluster with SafeKit is presented here.

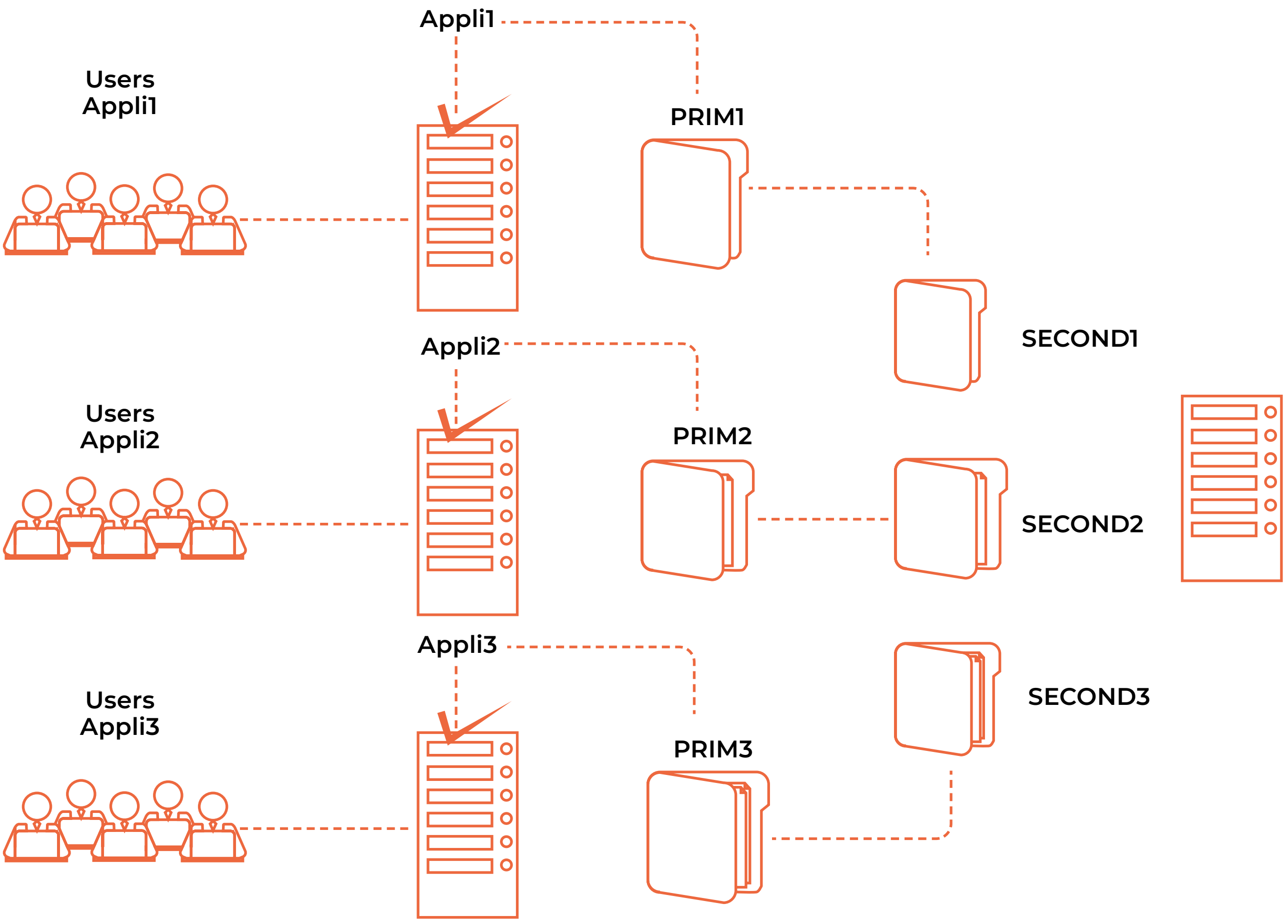
Example of a SafeKit Hyper-V or KVM cluster with 4 virtual machines


App1
OS1
VM1
App2
OS2
VM2
App3
OS3
VM3
App4
OS4
VM4
SafeKit: replication and failover of full VMs
2 active Hyper-V or KVM physical servers

The SafeKit N-1 cluster

Replication and application failover from N servers to 1

In an N-1 cluster, there are N mirror application modules installed on N primary servers and one backup server.

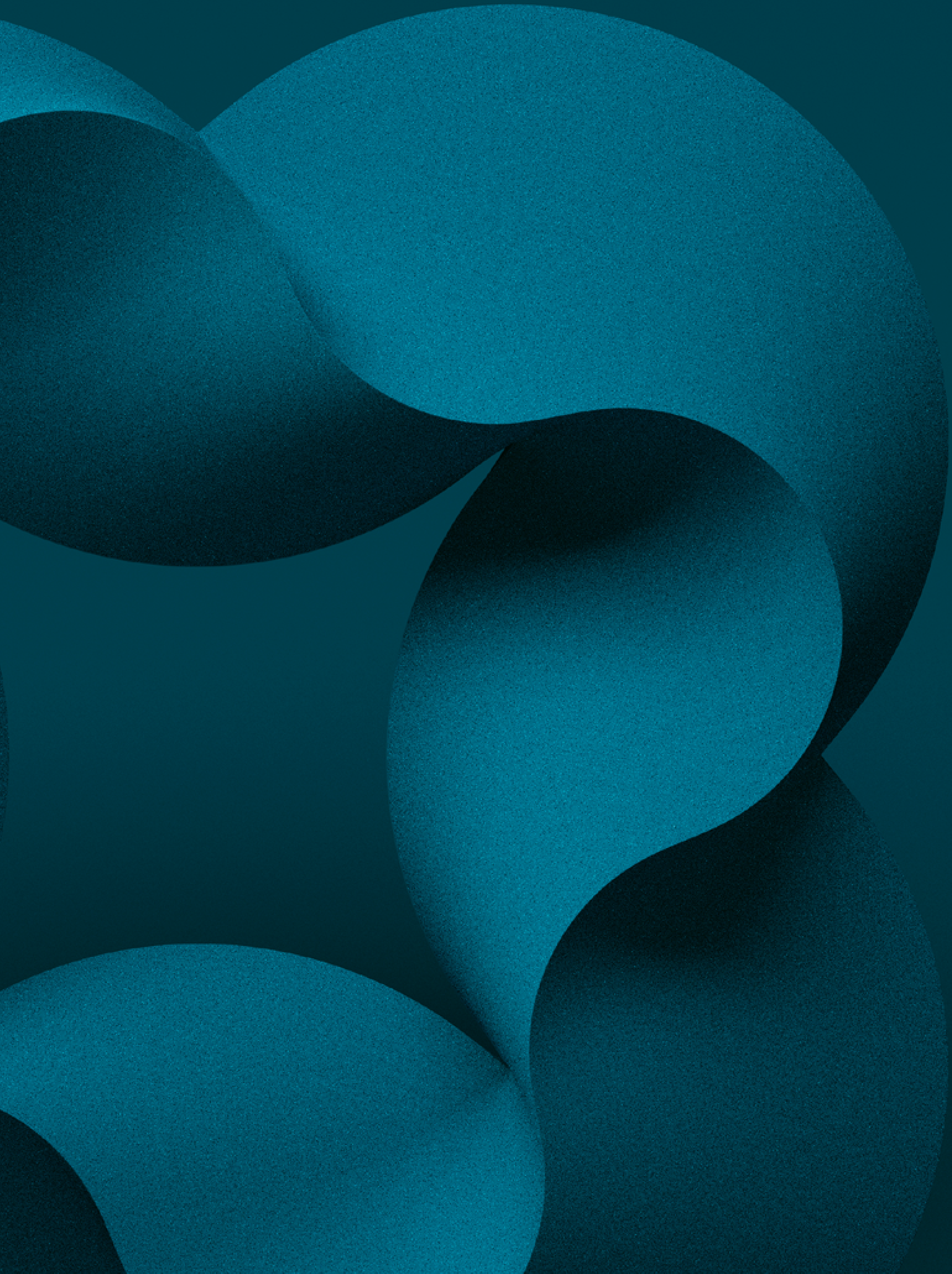


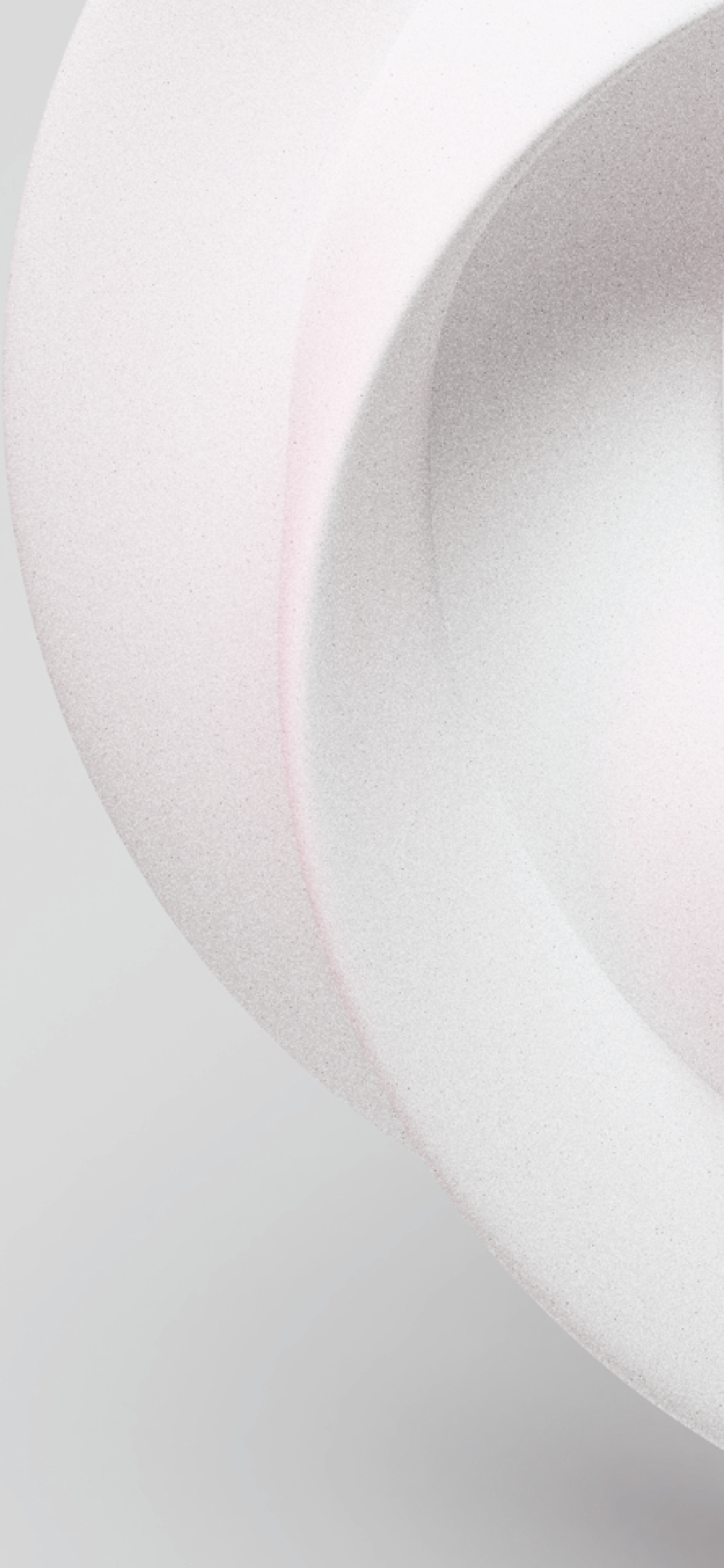
If one of the N active servers fails, the single backup server restarts the module of the failed server. Once the problem is fixed and the failed server is restarted, the application switches back to its original server.

In case of failure, unlike the active/active cluster, the backup server doesn't have to handle a double workload when a primary server fails. This assumes there is only one failure at a time - the solution can support multiple primary server failures at the same time, but in this case the single backup server will have to handle the combined workload of all the failed servers.

Note that in a N-1 cluster:

- all applications (Appli1, Appli2, Appli3) must be installed on the single backup for application failover,
- each mirror module will have its own virtual IP address, its own replicated file directories and its own recovery scripts.





Connect with us

eviden.com

CARTONS
ET EMBALLAGE
PAPIER À TRIER