A WHITEPAPER ON HIGH AVAILABILITY

Protecting PaperCut NG & MF, printing and beyond

by PaperCut Software

Contents

Introduction

This white paper presents the PaperCut philosophy on providing High Availability to our components and the surrounding print infrastructure. It's intended for IT decision-makers and anyone exploring how they can ensure maximum uptime and protection for their PaperCut installation.

When the term High Availability (also coined as HA) is brought up in conversation, system administrators, technical buffs, and stakeholders are usually thinking about how they can ensure that their most business-critical services continue to run without interruption.

The question we need to answer is: which systems need to be protected, and at what level using the available tools of high availability?

The concept of HA can be found in all sorts of things besides computing services. For example, we want our automobiles to be highly available, and we take steps to protect some of the features that could fail, such as tires. This is why most vehicles carry a spare tire, but since there are costs to carrying multiple spares, we have alternatives such as space saver tires, run-flat tires, emergency puncture repair kits, roadside assistance, or a combination of these.

With only one spare tire, we move the single point of failure. Depending on circumstances, this might be good enough for some; however, what happens when that spare tire fails? Then we are right back to where we began. Therefore it is best practice to utilize a combination of available technologies to support the businesses' HA endeavors while avoiding a 'worst-case scenario' as best we can.

As with anything, there's always a cost involved when traversing the road to HA in both time and money. Luckily, due to the plethora of technologies, we now have multiple avenues to reach the HA destination that suits the organization's requirements.

It would be fabulous if computers never crashed or suffered outages from hard disk failures. But they do. And even if they didn't, there'd still be downtime from routine maintenance, human error, hackers, network connectivity, natural disasters, etc.

So considering which systems to protect – and at what level – is essential. When it comes to PaperCut NG and MF and the surrounding printing infrastructure, we don't want the equivalent of carrying six spare tires when one will do quite nicely.

What is High Availability (HA)?

Simply put, High Availability provides how a service is available to the user without any noticeable interruption and is achieved using the following principles:

1. Redundancy
2. Monitoring
3. Failover

Implementing a highly available solution provides the benefits of reducing incurred downtime, whether due to loss of service, a site outage, human error, or scheduled maintenance, and improving application and service performance through scalability.

Another advantage of implementing high availability is that there are a plethora of technologies to support the needs of any organization. These can range from network load balancing, service clustering at the hypervisor or the database level to service backups and replication, all depending on budget and existing infrastructure resources.

The caveat is that in most if not all cases, someone needs to implement and support the solution and technology of choice. The systems administrator will require expertise in the technology or someone else within the organization, i.e., a network load balancer expert or a paid support contract from the technology vendor.

HA always has costs in time and money, all determined by the solution and technology of choice.

Why High Availability is a good thing

HA protects resources so that in the event of a computer service outage, your business can continue with minimal interruption or a decrease in throughput. By reducing the risk of downtime, we improve business continuity.

One way that HA reduces the risk of downtime is by eliminating single points of failure. If your entire PaperCut system runs on one server, then that server is a single point of failure – and so are the components of that server like the hard drive, power supply, and network interface card, by the way. Even software such as driver updates and operating system patches can bring down the system if they contain severe defects. Therefore, it is essential to highlight the benefits of testing before going live in a production environment supported by standardized change control processes thereafter.

Making any system Highly Available is not just about how it is architected, but also the processes and procedures that are implemented, which help reduce the risk of a system failure.

All in all, HA gives you methodologies to eliminate single points of failure, reducing the risk of downtime, and increased business continuity. That's why it's a good thing.

Business Continuity Planning

Having a business continuity plan is paramount when we look to implement high availability within an organization. It provides the business with defined guidelines of how operations will continue during an unanticipated disruption of service through the means of prevention and recovery systems while maintaining service levels for the end-users.

When creating a business continuity plan, it's good practice to conduct a business impact analysis and recovery strategy initiative. This helps set expectations, define processes and guide resolutions.

Business Impact Analysis (BIA)

A (BIA) is used to identify the time-sensitive and business-critical functions that need to be maintained, the current processes, and the resources required to support them. This will allow the business to understand the operational ramifications further should the print service fail and indicate how long the company can operate without print?

Producing questionnaires, running workshops, and formulating interviews with the relevant stakeholders are some of the ways to ascertain the required information.

Recovery strategies

When generating a recovery strategy, we must - using the (BIA), identify and document the resource requirements of the business. We can recognize the gaps between recovery requirements and existing business capabilities by conducting a gap analysis exercise.

With this information, we can answer questions such as:

- What current technologies and processes exist within the organization?
- What in-house organization resources can we utilize to support the business requirements?
- What teams or individuals can we contact when a failure occurs?

Implementing a business continuity plan is hugely beneficial. It provides the organization with a roadmap and a set of processes that support making the right decisions if and when unexpected operational failures occur.

Organizational factors

The PaperCut Print Management software is commonly used to provide many of the traceability, security, and high availability features described in this whitepaper. For example, using PaperCut Find-Me printing enables high availability at a Multi-Functional Device (MFD) level.

A user simply submits a print job to the (virtual) Find-Me queue, targeting multiple MFDs. If one of these MFDs falls into a failed state, the user can easily release their job at another MFD. Both PaperCut NG and PaperCut MF can support and provide HA, printer load balancing, and security measures described in this document via a few clicks within the admin console.

Business Value

First, we must know what parts of the business need to continue in an interruption and their value to the overall organization. Running a hospital and the snack vending machine can't process payments is probably not as vital as ICU patient monitoring. However, if you're a university during finals week, that same vending system could be mission-critical.

User behavior

Another impact of proper HA planning is user behavior. Are there peak usage times which stress specific components of the computing system? Do some applications create a higher system load? Which users will need system access, even in the event of a disaster? Can the HA design scale with user demand?

System infrastructure

Are there multiple types of devices that need to be protected? Have we identified all the pieces of the system? What if we lose power to the primary data center?

Textbooks are written on techniques to protect computing services. Whether it's the servers, operating systems, databases, or power sources, we need to understand business goals and any single points of failure in the computing systems that will put them at risk.

Achieving High Availability

Achieving HA is accomplished primarily through two methods: redundancy and recovery. Both give the computing system resilience (i.e., an ability to return to full operation).

Redundancy

Redundancy solves the problem of a potential failure by having a duplicate standing by. It uses technologies like RAID, Virtual Machine (VM) images, clusters, and Network Load Balancers (NLB).

If you're using RAID and a hard disk crashes, no problem – the data's been redundantly written to other disks. If you're using virtual machines and the whole VM crashes, no problem again – just spin up the latest VM image on a new VM server, and you're back in business. Clusters and NLBs have multiple servers running and can divert computing requests away from a failed server to one that's still up. Theoretically, you can have an extremely low risk of downtime by implementing redundancy.

However, the cost is usually at least double for a redundant system. Plus, there's added time and complexity to build and maintain these systems, requiring highly trained personnel.

Serious consideration should be given to determine the type and level of redundancy. One PaperCut customer had a very sophisticated Linux HA system with clustered DNS servers and clustered database servers. It was bulletproof. However, when their system administrator left the organization, no one knew how to maintain it. System upgrades were painful and time-intensive. Eventually, they scrapped the complex, over-engineered system and implemented VM servers that could be spun up on new hardware at a moment's notice. It provided the same level of redundancy, with much less complexity.

Consequently, it is recommended to document the High/low-level designs and all test plans and run procedures. This will ensure that new administrators who take the helm have a complete understanding of system operations and contingency plans, ultimately mitigating delays in return to service.

Recovery

HA can also be accomplished with a good disaster recovery plan. This is the most basic form of HA and avoids most of the complexity of many HA techniques.

Good disaster recovery plans will have procedures to minimize downtime for critical systems. One such procedure could be taking a database backup every night and writing daily transactions to an offsite server. This should give you the ability to have the entire database back online within a short time, even if the primary database server crashes and burns.

Resilience

System resilience is the ability to ensure the continuous availability of operational services such as printing within an organization. Utilizing the PaperCut Site-server component can protect the print infrastructure from unexpected network outages or unreliable network links across multiple remote office locations.

In the event of a connectivity failure between the head office and a remote site, the PaperCut Site server will automatically take over the role of the PaperCut Primary Application server. The failover process is seamless and transparent, protecting print tracking, copy, authentication, and Find-Me printing from going offline.

Once network connectivity to the Primary Application server at the head office has resumed, the PaperCut Site server hands back the responsibility to the Primary Application server.

For more information on what the PaperCut Site server protects, please see our Offline Operations KB

What is the right level of High Availability?

Once upon a time we had another PaperCut customer with a robust installation that included clustered application servers, clustered print servers, and clustered database servers – all of which pointed to a SAN. From a system point of view, it was on the high end of HA... Until a fire tore through the data center, and then it wasn't on the high end of anything, let alone available. Forced into a redesign, they reconstructed their PaperCut installation and opted for more modern virtual machine technologies to provide the same level of HA.

This leads us to a primary consideration: what is the right level of HA? There is a significant difference in TCO between providing 99% uptime and 99.99% uptime. Is the difference necessary and worth the cost? Even if HA is a good thing for your business, and the ways to achieve it are well understood, the question still needs to be considered: what level of HA is necessary? You'll have different answers to this question for various functions in your organization. Your mission-critical systems need more HA techniques for more parts of the infrastructure if the cost of an outage would be greater than the cost of providing HA.

For example, vital systems such as order placement, user authentication, and database may need 99.99% uptime. This might be enabled with multiple HA techniques like virtual machine snapshots, clustering, synchronous replication, hot sites, and off-site backups. However, the print system might be just fine with an hour of downtime.

Our focus when considering HA should be on two objectives:

Recovery Point Objective (RPO)

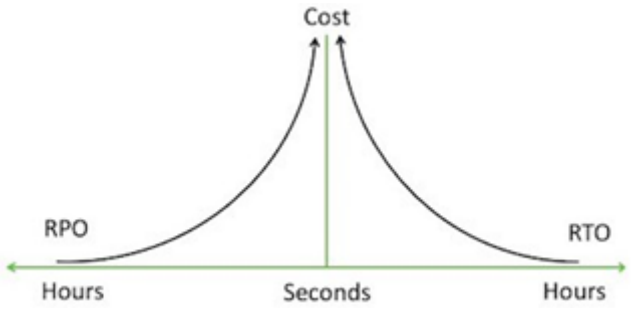
RPO is the length of time between taking snapshots of your data. It's a measure of how far back in time you must go to get a recovery point. It's also the amount of time where the business process can cope with a loss of data.

Recovery Time Objective (RTO)

RTO (aka Mean Time to Recovery) is the maximum tolerable time from the point of failure back to normal operation.

RPO and RTO need to be carefully considered because together, they determine the cost of recovery. The smaller the time for RPO and RTO, the larger the cost to recovery. If you want

recovery points measured in seconds and recovery time in minutes, then expect a very high cost to recovery.



Uptime and how is it calculated?

Living in the current era of the "always-on, always available" service, there is the ever-increasing requirement to obtain 100% uptime of business-critical services. Whether it is your print management solution or the broader supporting infrastructure, in reality, things are bound to fail no matter how hard we prepare. Therefore we must align the expectation to something more achievable, 99% uptime, by utilizing the tools available to create a HA environment.

As such, uptime is determined by the availability of resources over a year, defined using the number of minutes or seconds. Calculations start at 99% and are measured in "nines," representing the ratio of how many minutes out of the total minutes in a year service is up and running.

Availability Max Disruption (per year)

Percentage of Uptime	Allowed downtime	Business case
99%	3 days 15 hours	Batch processing, data extraction, transfer, and load jobs
99.9%	8 hours 45 minutes	Internal tools like knowledge management, project tracking
99.95%	4 hours 22 minutes	Online commerce, point of sale
99.99%	52 minutes	Video delivery, broadcast systems
99.999%	5 minutes	ATM transactions, telecommunications systems

Depending on the organization's requirements will determine what level of uptime is required versus the level of downtime that is acceptable.

Protecting PaperCut with High Availability

Now let's apply all this to a PaperCut deployment. We don't want to start with the assumption that printing systems need to be protected at the same level as other business functions. We should begin with the business objective questions. What's the maximum amount of time that print jobs (and their accounting data) could be lost and need to be reprinted (RPO)? What's the maximum time allowable from the failure of printing systems to full recovery (RTO)? And third, what's the expected total cost to recovery?

It's possible with PaperCut to improve overall system resilience at multiple points in the infrastructure. PaperCut recommends using HA technologies and methods that are most familiar to the customer and where they have trained personnel to support them. This reduces overall system complexity by avoiding introducing new tools and procedures to learn and implement in the event of an outage. This is a primary reason why PaperCut doesn't mandate HA methodologies or create HA products of our own.

Remember our clustered Linux HA customer mentioned earlier? It turns out that their single point of failure was the person who set up the complex environment using multiple HA technologies that no one else understood. PaperCut doesn't want to force customers to add yet another tool on the HA stack.

HA technologies have evolved to the point where very favorable RPO and RTO can be achieved without adding this cost and complexity into PaperCut products. The customer shouldn't have to learn our way of providing HA; they should be able to use what they already know. Even if we wanted to build all this into PaperCut, we couldn't fully protect against the most common sources of downtime: full hard disks, dead NICs, and human error. The entire printing system, including PaperCut, can be protected against downtime with off-the-shelf HA technologies.

Therefore, the discussion about adding HA for PaperCut should start with, "How does the customer defend against downtime on other servers, and in particular the printing systems?" For example, it would do us little good to defend the PaperCut server and not other print servers.

Anywhere PaperCut is running or where it utilizes a resource should be considered in the HA plan. Simply put, the main pieces that we want to evaluate protecting are the PaperCut Application Server, PaperCut Site Server, Print Provider (i.e., print servers), database, and System-Level Multi-function devices (MFDs). If a customer already employs HA methods on other servers and resources, they should include PaperCut in the same manner with very little additional training or expense.

Whatever methods you employ for PaperCut HA, "The general principle is to start light, and build over time" (Chris Dance, PaperCut CEO).

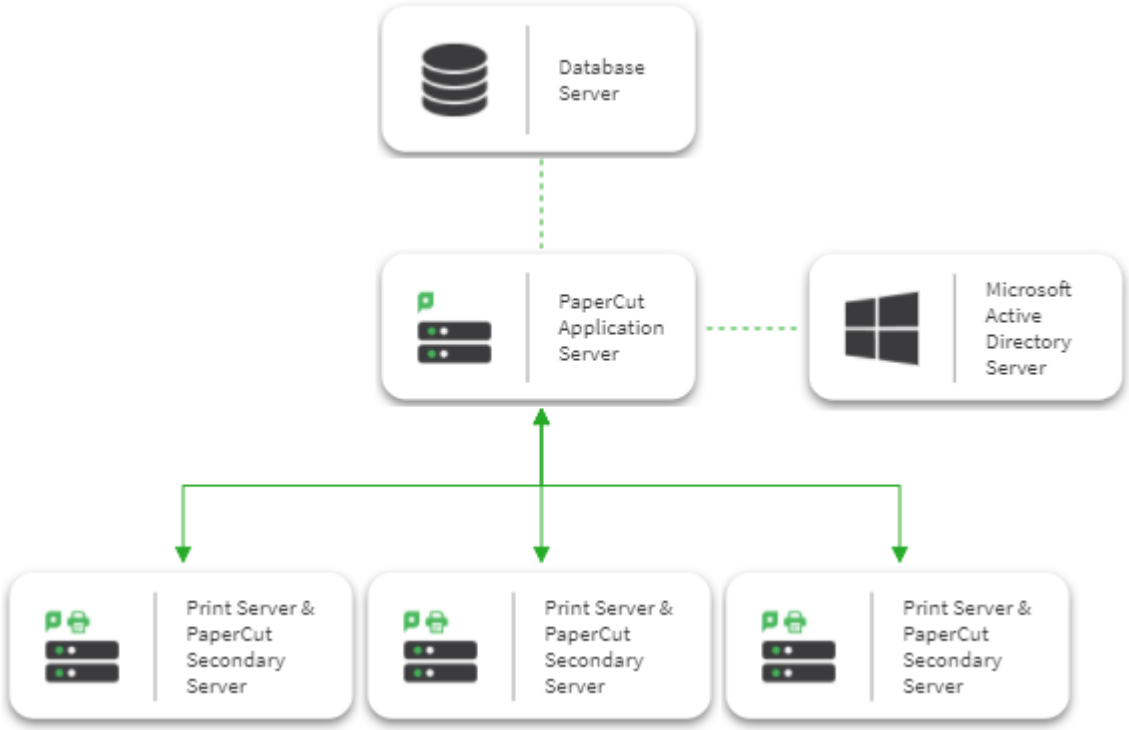
PaperCut examples

There are several proven techniques to provide HA, each with differences that may impact your decision of which one(s) to use. Technical constraints may make some techniques unadvisable (e.g., using an NLB for SQL databases). The overarching principles are:

- Use the tools with which you have a depth of experience
- Provide the level of HA that meets business objectives
- Start light and build over time

PaperCut's scalable architecture

PaperCut is designed to be a scalable system. This means that critical components (e.g., application server, database) can be located on separate servers to allow for growth and performance improvements. This also has the advantage of making these components more easily protected for HA. The diagram below shows a very common deployment with a PaperCut Application Server, a user directory, a database server, and multiple print servers. In this example, it'd be almost trivial to protect the PaperCut app server since the critical data is in the database.

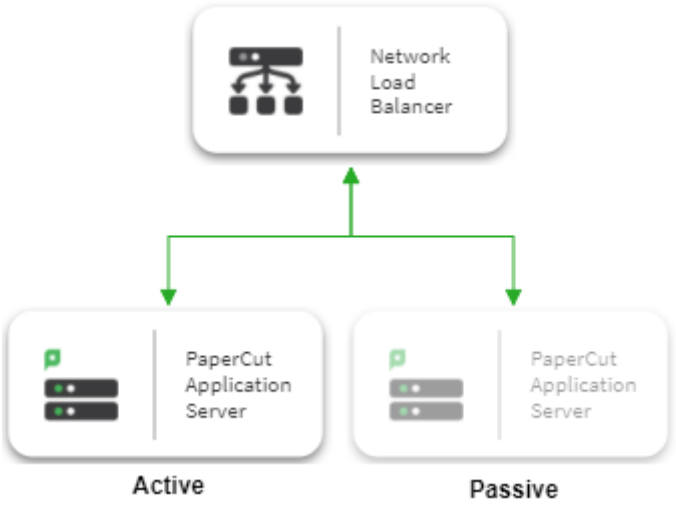


Common PaperCut Example Deployment

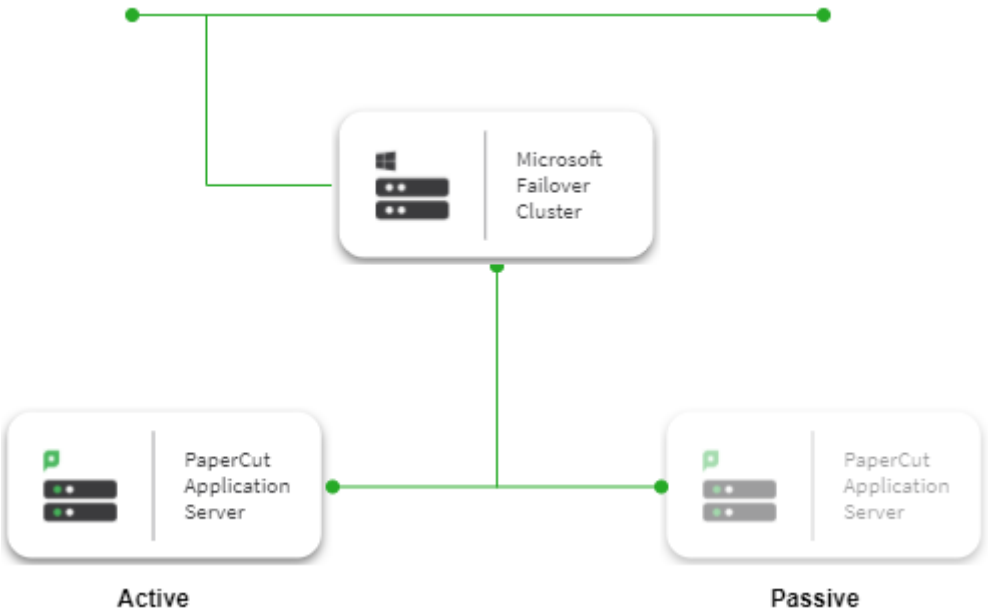
PaperCut Application Server

The PaperCut Application Server (app server) can be protected with an NLB using an Active/Passive configuration, clustering, virtual machines, simple backup, and restore.

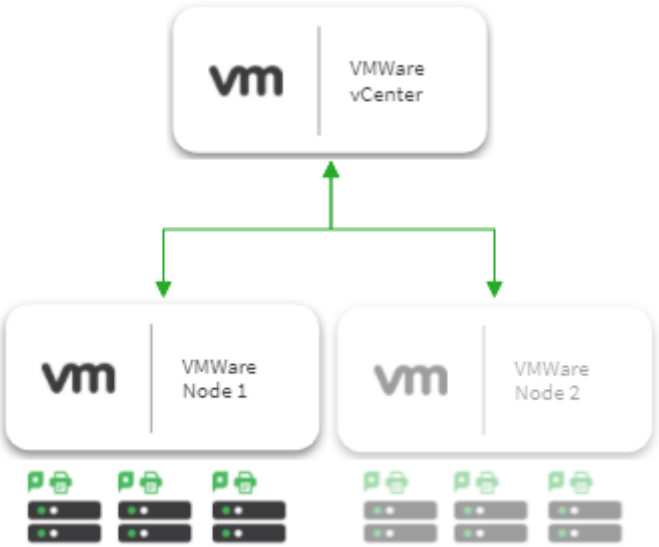
Active/Passive Load Balancing



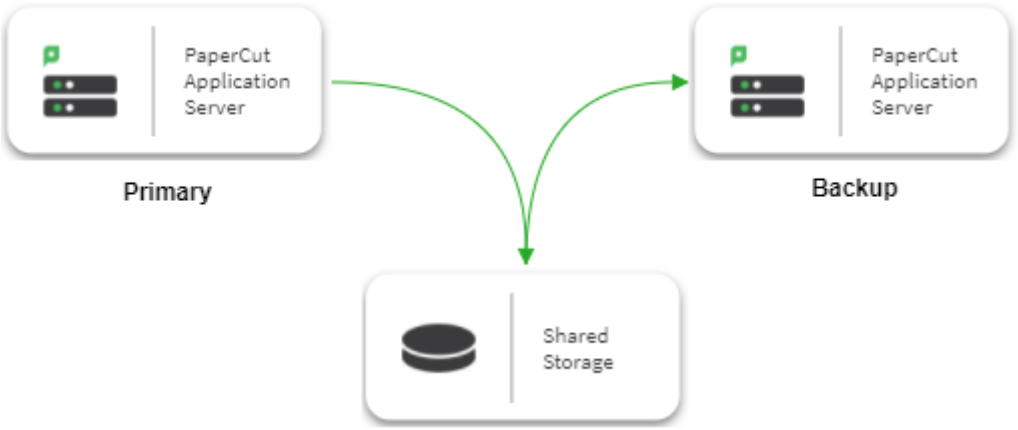
Operating System-Level Clustering



Virtual Machine High Availability



Backup and Restore

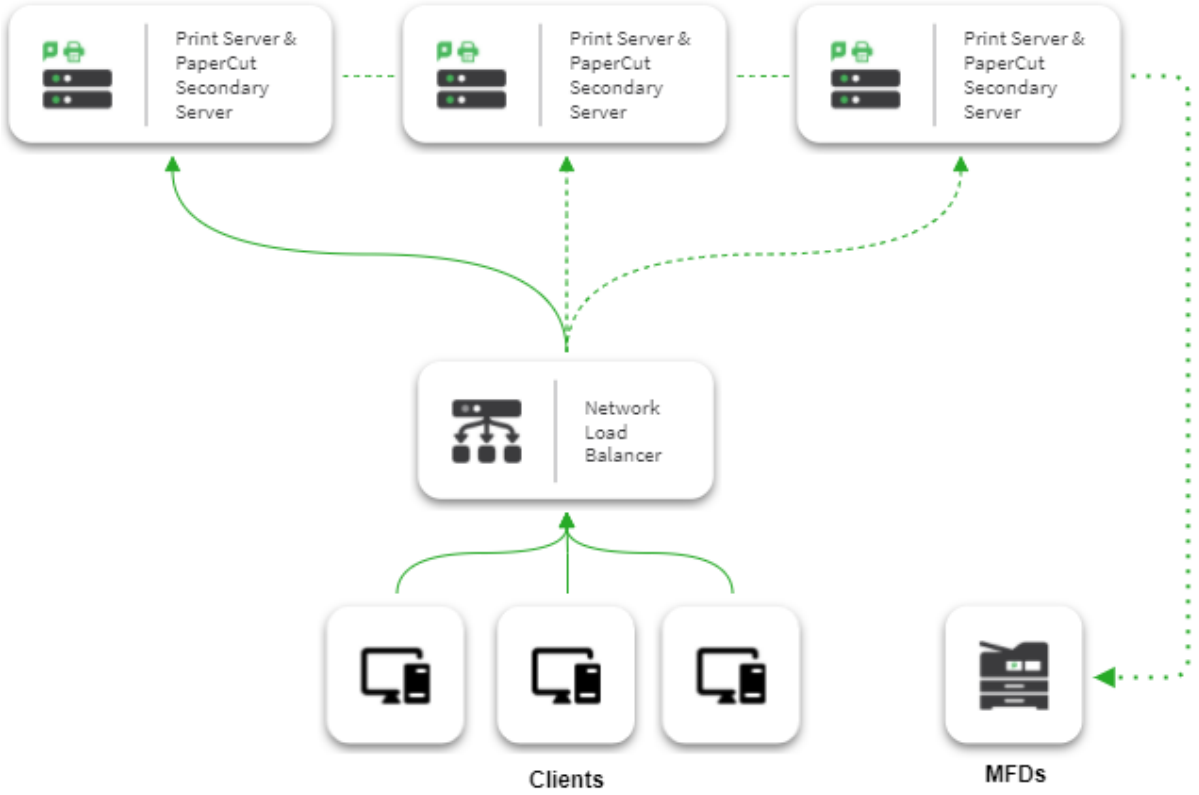


PaperCut Site Server

The PaperCut Site Server (site server) provides resilience to the printing system if the connection to the app server becomes unavailable. The site server itself should also be protected with clustering or a virtual machine. We will discuss the PaperCut Site Server later in detail, if simply can't wait to get to the good bit feel free to jump to Site Server resiliency in detail

Print providers and print servers

The PaperCut Print Provider runs on print servers and can employ all the aforementioned methods and Network Load Balancers (NLB). An NLB distributes application traffic (print jobs in this case) across several servers. On top of load-balancing print jobs, an NLB will bypass a server that isn't responding, which adds protection for a failed print server.



External database

An external database can be protected with clustering and VMs and some database-specific techniques, such as synchronous replication and off-site transaction logs. Check with the customer and the database manufacturer for the best choice. In the meantime, please check out The ultimate guide to High Availability methods for Microsoft SQL Server article for more information.

Network Load Balancers in detail

Where to use an NLB

Configuring PaperCut software behind a load balancer provides the capability to protect both the PaperCut Application Server and Print Server layer using the functions of:

- High availability - ensuring services are always online through redundancy. This applies to both the Application and Print Server layers.
- Load balancing - for connection distribution and scalability of services applicable to the Print Server layer.
- Failover - mitigating service interruption when a server has entered an unhealthy state unbeknownst to the user.
- Security - by removing user direct IP access to the servers within the server pool.
- Persistence and session affinity - ensuring that user data is forwarded to the same server they initially connected to within a specified timeframe.

Using the function of load balancing, the NLB divides network traffic (i.e., service requests) among multiple servers that can respond to the demand from clients. One main advantage is that all the clients make their requests to the NLB, not the individual servers adding a layer of security. This technique is used extensively on the internet. The most practical way for websites to respond to increased load is by utilizing an NLB. Otherwise, we'd all need to browse to google2.com if google.com was busy – and if they're both busy, we randomly try google42.com. Not fun.

Using an NLB allows everyone to browse to the same internet site (e.g., google.com) because the NLB decides which actual server is available to respond using various server health monitoring techniques called health checks. Now the term NLB makes more sense; it balances the network load across multiple servers. The NLB achieves this by using various balancing algorithms such as *round-robin*, whereby the load balancer will send each new connection to each server within the pool in a round-robin fashion, or *least connection* where the load balancer sends new connections to the server that has the least connection load. Pretty cool.

When implemented, load balancing provides system administrators with the advantage of scalability, meaning if the server pool is hitting the resource threshold and user demand increases, we can simply add more servers to the pool. Easy P-easy!

This works the other way around. Using a school as an example, imagine the semester is over, students are home, and printing requirements are low. Fewer students means fewer print jobs. Rather than having unused servers running, consuming resources and power, we can scale down by removing servers without impacting the service of any staff who are still hard at work preparing for the start of the next academic year, meanwhile reducing infrastructure running costs.

Ok, we can agree that a load balancer can provide a simple way to scale our services, check! So let's talk about how the NLB allows for easier maintenance of individual servers, using the function of high availability. If you have sixteen servers behind your NLB and one of them needs a new disk drive, you can remove it from the NLB, replace the drive and then put the server back in the load-balanced server pool.

Through server health monitoring (health checks), the NLBs can protect against an individual server failure when servers are deemed unhealthy. Using the scenario above, what if that bad-mannered disk drive crashed before you got around to replacing it? You still have fifteen other servers to bear the load. While the sysadmin is sourcing the replacement disk, the NLB has the intelligence to confirm the health of the failed server and automatically remove it from the load-balanced pool, avoiding any users connecting to it. Once that disk is replaced, either the sysadmin or the NLB can bring the server back online - via automation.

Disaster averted.

The PaperCut Application Server, for example, is a stateful application; it keeps track of connected clients to make communicating more efficient, so in this case, the NLB doesn't have to. This is achieved using a combination of shared storage and an external database. So, with a range of available technologies, the app server can be protected with clustering, virtual machines, backup and restore, or an NLB using an Active/Passive configuration.

How does a Network Load Balancer help with printing?

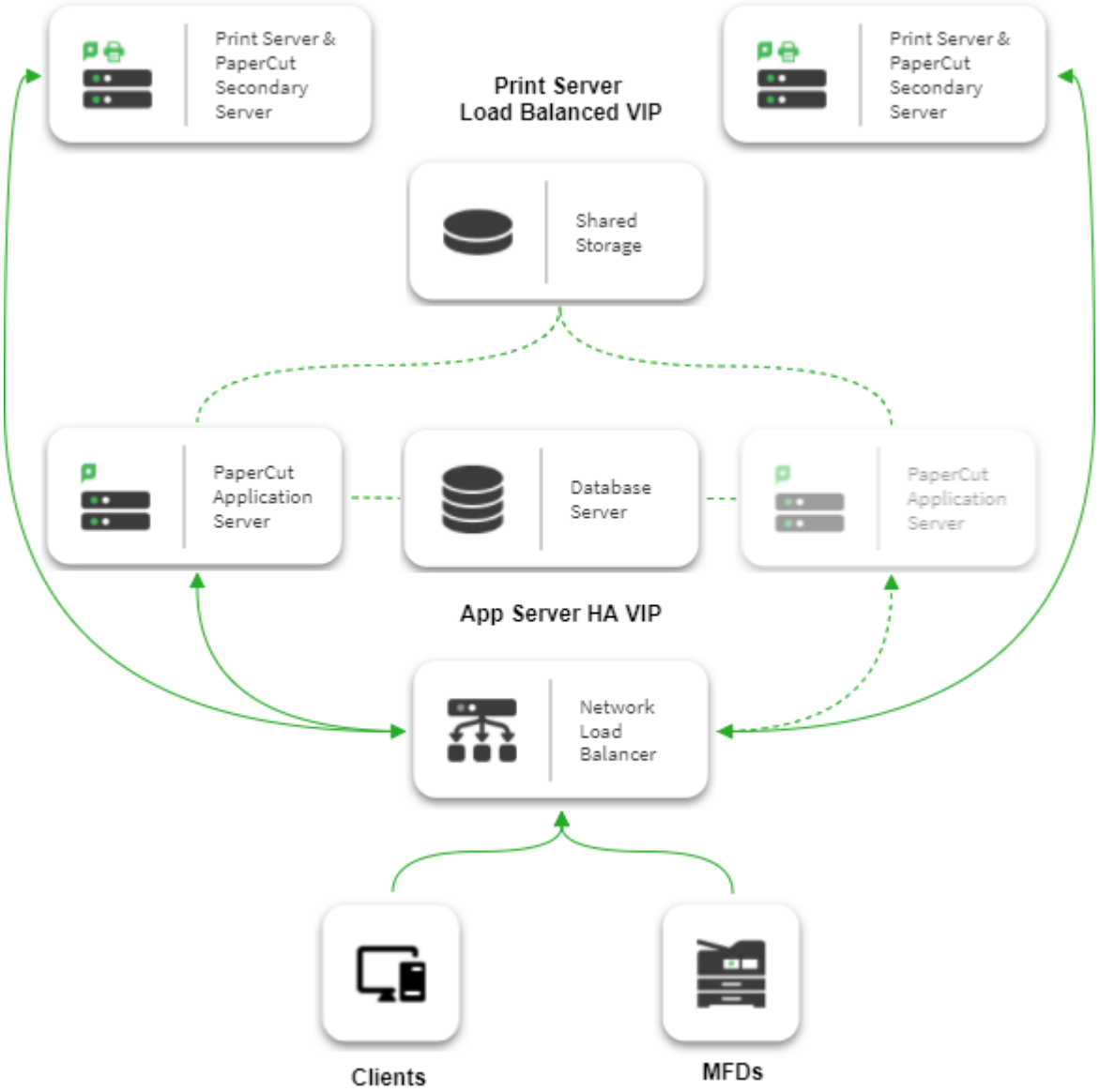
Just as a web server farm can allow multiple requests to be split across multiple web servers, we can achieve the same effect with print jobs. It's possible to configure a load balancing device (e.g. Loadbalancer.org, Kemp, F5, NetScaler) to accept incoming print requests and split them across multiple print servers.

NLBs can be helpful in a customer environment where there are large numbers of users constantly printing or where large print jobs are being frequently produced. Load balancing gives system administrators a more straightforward method for print administration and a flexible approach to the print environment. There are a couple of different ways in which PaperCut can operate with these devices.

As you might already know, PaperCut can be installed as a primary server, secondary server, and site server. The primary server is the installation of the main PaperCut application, which is where the software is managed.

As of version 20+, the primary server can be placed behind an NLB within an active/passive configuration. This means that only one primary server receives connections until a failure occurs or a sysadmin enacts scheduled maintenance. The NLB will then disable the failed primary server and promote the passive primary server to an active state.

This is what the basic setup for load balancing will look like:



For detailed information on how to configure multiple Papercut Primary Application Servers behind an NLB, please look at our Application Server failover knowledge base article.

The PaperCut secondary print servers are installed behind the NLB.

Each PaperCut secondary print server is connected to the app server and configured with all available print queues.

For example, if you have five printers in the network (Printer1, Printer2, Printer3, Printer4, Printer5), then each PaperCut secondary print server would have a print queue for each of these five printers.

The NLB can be configured to accept print traffic from client machines and redirect them to any PaperCut secondary print server.

In most cases, you'll configure a DNS name that references the IP address of the NLB.

Clients will use that DNS name to reference the printers. For example, if we created a DNS name for the NLB of 'NLB-Uno,' clients would connect to NLB- Uno\Printer1 and NLB-Uno\Printer2.

When a user prints to one of these print queues, the job will be redirected by the NLB to one of the PaperCut secondary print servers. And this means that the job may end up on any of the four PaperCut secondary servers (depending on how the NLB has been configured).

The app server will see each of the PaperCut secondary servers and the associated print queues individually and will release the job to the appropriate printer as required.

If secure print release is being used, the app server will redirect the job to the correct printer when the user logs on to the device. Otherwise, the job won't be held at the server and will be directly sent to the printer.

Does network load balancing work with Find Me Printing?

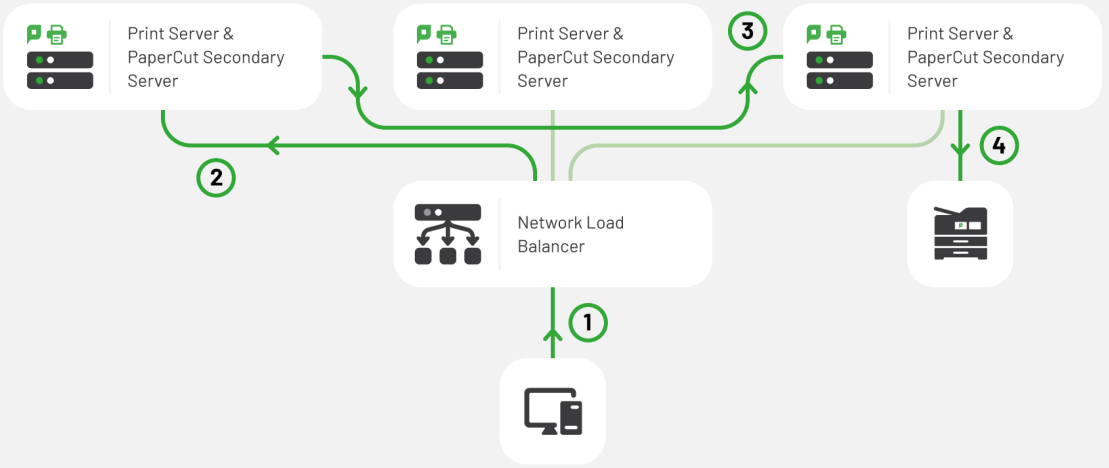
It sure does. If you're using Find Me Printing, you need to configure an additional queue on the PaperCut secondary print servers. This queue will be your virtual Find Me queue. Make sure you replicate this across all PaperCut secondary servers.

The end user's queue would be NLB\Find Me. The NLB can be configured to redirect the incoming Find Me queue to one of the four print servers.

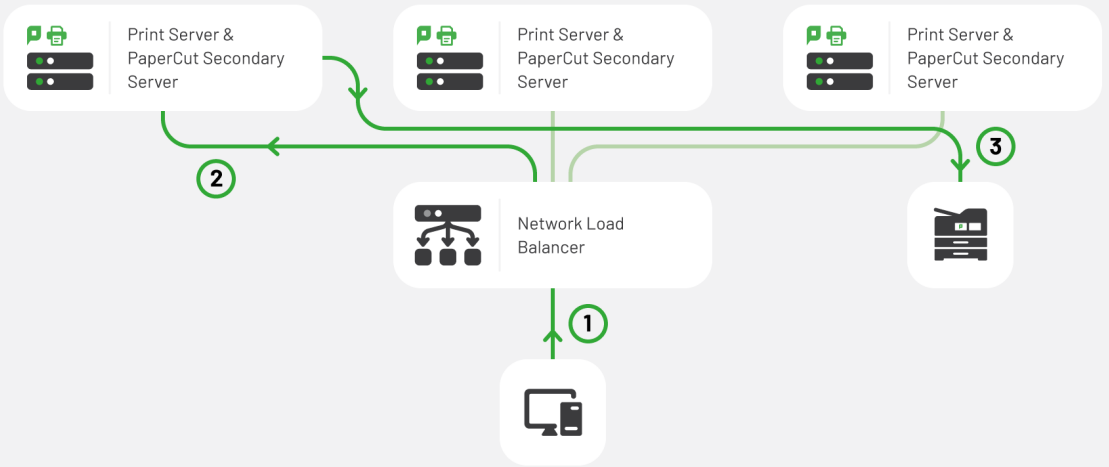
You'll see four Find Me queues from the app server, and PaperCut will report on each one individually. Since each queue is a separate instance, we can take our solution one step further and configure server affinity for the print queues. This ensures the server that receives the print job will be the same server that sends the job to the printer.

We configure server affinity because it reduces network traffic. It stops jobs from being received by one server (e.g., Server1), and then transferred to another server (e.g., Server4). You can see what happens to a print job with and without server affinity below.

Print Server Network Load Balancing: *without server affinity*



Print Server Network Load Balancing: *with server affinity*



For information on how to configure PaperCut Secondary Servers behind an NLB, please see our Print Server Network Load Balancing knowledge base article.

Virtual machine clustering in detail

What can Virtual Machines do?

Virtual Machine (VM) technologies, such as VMWare and Microsoft Hyper-V, provide great flexibility in deploying servers within an organization. VM implementations can also provide HA using VM clusters. When a VM is running in a highly available VM cluster, any physical hardware failure has minimal impact on the running VM, which is simply transferred to another node in the cluster.

Implementing HA using VM infrastructure is a viable alternative to the built-in operating system and application clustering support. This allows you to set up PaperCut in the same way you would on a physical server but enables the VM infrastructure to provide the HA.

Clustering at the VM level offers advantages over other traditional clustering setups, such as:

- Your software, drivers, and settings only need to be installed and configured once in a single VM image
- Depending on your VM infrastructure, the VM can be shifted to another node with little downtime when a physical node fails.
- Simplified backup processes
- Disaster Recovery (DR)

VM hypervisors detect when a VM becomes unresponsive. You should consider whether you'll augment this with application-level monitoring. Although the VM might be running normally, the underlying application may have problems, and application-level monitoring can detect this. Ways to perform application-level monitoring include:

- Loading an Application Server URL to test the server is running
- IP pings
- Checking that PaperCut services are running.

Defining your Virtual Machine clustering environment

There are many VM deployment strategies you can leverage, depending on the VM platform you're using. This includes VMs hosted in different physical boxes or even on different sites. This also offers DR options.

When selecting a VM product, it's particularly important to consider the following features:

- Fault Tolerance (FT) – provides continuous availability for VMs by automatically creating and maintaining a secondary VM identical to the primary VM

- High Availability – lets you make a new VM available to minimize downtime if an existing VM fails. Generally, FT is a functionality added on top of HA that provides seamless switch-over if there's no loss of state.
- Application HA – lets you make a new VM available to minimize downtime if the application fails. Your VM product must offer application-level monitoring.
- Data replication and backup – lets you backup and restore in-memory and application data. Replication of in-memory data in the case of FT or DR features might also play an important role. Specific proprietary algorithms for the replication of memory segments usually reduce the bandwidth needed and are very efficient.

Defining your Virtual Machine clustering setup

There are many ways in which you can deploy PaperCut on VM infrastructure.

Consider the following for implementing a VM-based clustered PaperCut installation:

Mode 1: Clustering at the print layer

Configure print servers as required in your VM environment and configure them for HA. Install the PaperCut secondary server to monitor printing on the print servers.

Mode 2: Clustering at the Application Server layer

Configure a new server in your VM infrastructure to host the Application Server. Configure this VM with HA. Install the PaperCut Application Server. You can then choose to set up your print servers for HA.

Microsoft Failover Cluster Manager in detail

What is Microsoft Failover Cluster Manager?

Since Windows 2012, Microsoft has reconfigured how you can manage HA with print services. They introduced the ability to use Hyper-V and failover clustering to make your print server a highly available VM. This solution provides full server failover options and can be implemented with the PaperCut servers.

How to set up Microsoft Failover Cluster Manager

Don't worry; we've got you covered. For a nicely detailed article on setting up PaperCut servers on Microsoft Failover Cluster Manager, check out the Microsoft Failover Cluster Manager section in the PaperCut manual right here.

Site Server resiliency in detail

Resiliency and redundancy considerations

For customers with distributed deployments, considerations such as redundancy and resilience to network outages are often a high priority.

A robust solution should defend critical points of failure, allowing an organization to continue operating while a network is under duress. For PaperCut, this means a robust deployment should ensure the availability of printing over unreliable network links.

The installation of PaperCut Site Servers gives customers peace of mind because access to printing resources won't be interrupted by unexpected network dropouts.

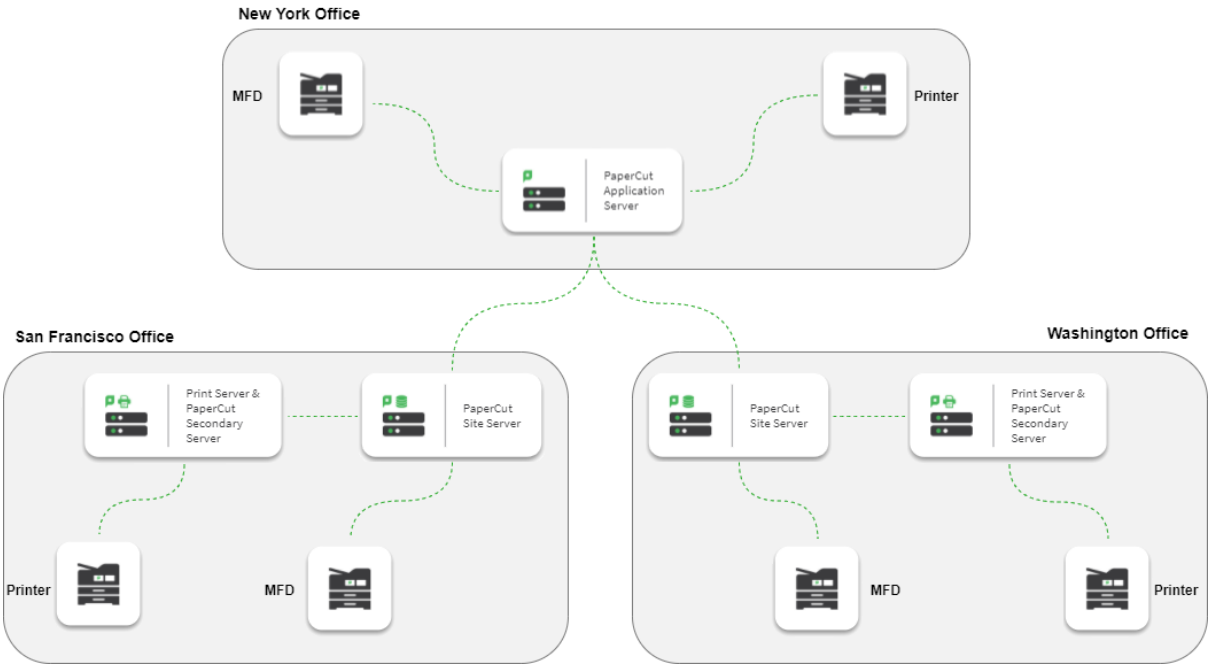
The Site Server duplicates the critical features of a PaperCut primary server to a local site during an outage. MFDs are configured to connect to a Site Server as if it were the primary server to remove their reliance on WAN

links. PaperCut secondary servers (Print Providers) are also aware of their local Site Server, providing a failover server if the primary server can't be contacted.

This simple but effective design delivers HA to MFDs and support for Secure Print Release, including Find-Me printing.

The Site Server installs in minutes with minimal configuration steps and no ongoing administration. Installers and Administrators need no specialist skills in database management or replication to provide business continuity. The Site Server ensures it's kept up to date with the current state of the primary server, transparently performing the role of the primary server when needed. Once a connection to the primary server can be re-established, the merging of local Site Server logs and transactions back to the primary server is also seamlessly managed by the Site Server.

The close relationship between the Site and primary servers allows the support of the same set of operating systems and databases for installations. It's perfectly valid to promote an existing PaperCut secondary server to a PaperCut Site Server to improve a site's resiliency.



Conclusion

We have reached the end of our High Availability journey. We hope you now have a better idea of the PaperCut best practices and recommendations on protecting your PaperCut environment.

The essential items to take into consideration would be:

The PaperCut Application Server can be placed behind a load balancer in an Active/Passive configuration as of version 20 and above.

Having the ability to monitor system health and quickly adapt to changes when a service failover occurs through resilience.

Where possible, ensure the HA design is kept simple as overly complicated HA designs can spawn overly complex HA problems, and nobody wants that.

"With great power comes great responsibility!" Thus **"With great HA designs comes great documentation!"** especially if the design is complex; trust me, your colleagues will be thankful.

In addition to documentation, adopting a change control procedure can help reduce intervention and provide the necessary audit trail if and when things don't go to plan, and a configuration rollback is required. This can be very handy when performing system upgrades.

The final, if not the most important, thing to remember is to test, Test and TEST again before going live in the production environment!!!

"Expect the best, plan for the worst and prepare to be surprised." Denis Waitley

Thank you

If you have any questions on any topics covered in this document, please let me know.

Authors

David O'Hara (Lead Solutions Architect, PaperCut Software)

Matthew Lee (Lead Solutions Architect, PaperCut Software)

Neil Sabine (Systems Architect, PaperCut Software)

Imannuel Bilson-Graham (Solutions Architect, PaperCut Software)

PaperCut HQ

sales@papercut.com

www.papercut.com

Support

www.papercut.com/support