Rocket® software

Best Practices for Deploying MultiValue Applications to the Cloud

Contents

← Back to contents

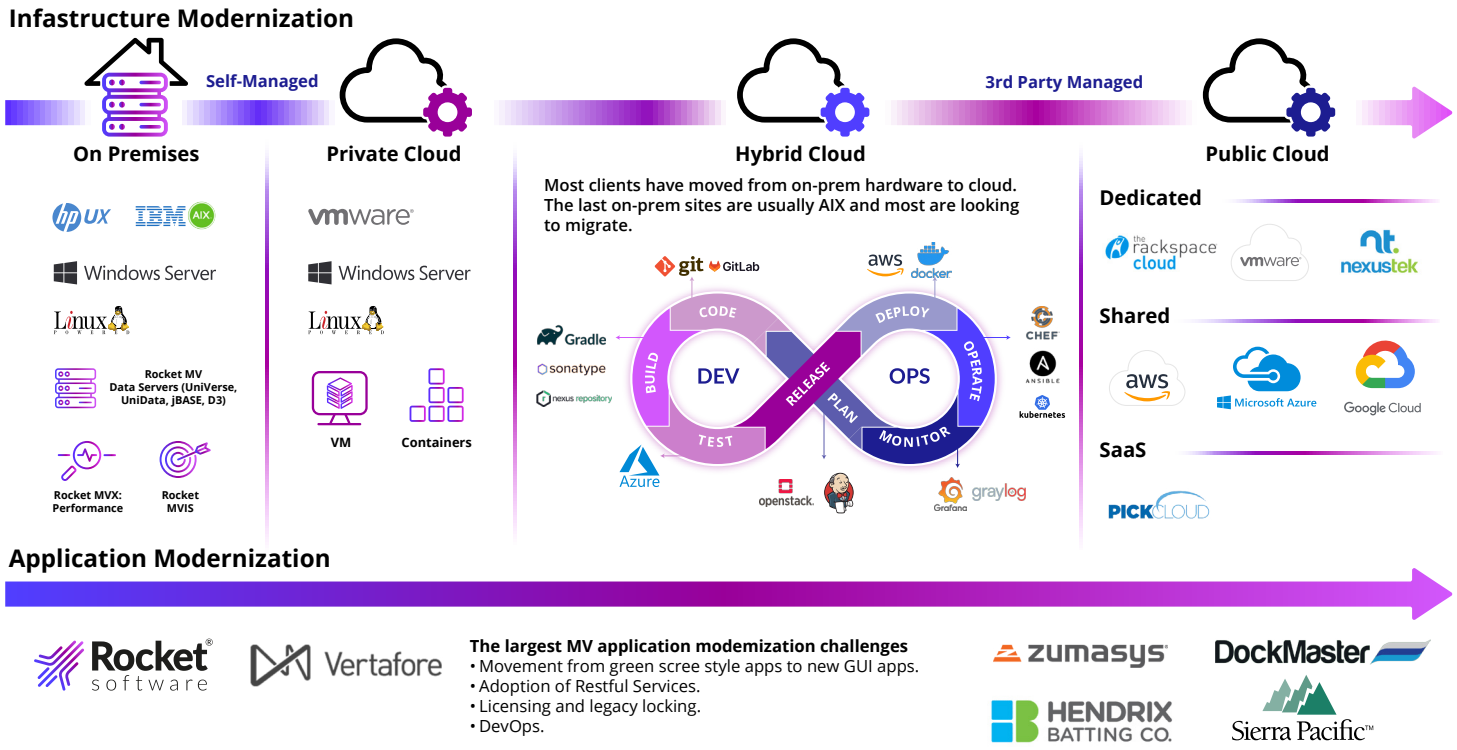
Best Practices for Cloud Deployments

What is Cloud

Cloud computing is the on-demand availability of computer system resources, especially data storage and computing power, without direct active management by the user.

This means even an on-premises datacenter with VMWare and shared San storage is Cloud. There are different cloud options which is where the definition can get confusing.

MultValue Cloud Journey



The largest MV application modemization challenges
- Movement from green scree style apps to new GUI apps.
- Adoption of Restful Services.
- Licensing and legacy locking.
- DevOps.

Most of us have already started on a Cloud Journey, usually with VMWare and our own Data Center. This is shown above as the Private Self-Managed Cloud. This initial step introduced us to the concepts of on-demand resources via Virtual Machines and VMWare.

If you are running on an AIX or other Non-Intel-based infrastructure, you may already have experience with this via technologies such as Logical partitions (LPARs) in Linux and San storage such as NetApp. With each move to the right in the graphic above, you will need to understand the hardware designs of each platform and how they can affect your existing system. Items such as Networking, storage and backups, and general performance around your app requires you to understand each Cloud's offering. Private Self-Managed clouds such as VMWare with your own NetApp is typically more flexible at matching your original process than a Public cloud offering such as Azure, where you are expected to match what those platforms are offering, primarily due to the fact that the cloud providers manage their offerings. Your MV application has no barriers that will keep you from moving it to these other platforms, but you'll need to make changes (around printing for example). In addition, many Public Cloud providers do offer more advanced options such as NetApps and dedicated hosts at a higher cost.

Operating systems (Windows VS Linux)

The most popular hosted operating system in the Cloud is Linux. Most hypervisors are based on Linux and therefore their support and performance for Linux is the best. Windows is also supported, but usually requires more resources, mainly due to the Graphical Desktop. Linux, by default, is designed to run without a Graphical Desktop and is easily managed via Console apps. A clouded system is, by default, a remote system and you'll handle administration remotely. If you're following best practices, use Linux over Windows when possible. If you want to run Windows, keep in mind you will have to allocate more resources. Go with a provider that is ready to fully support your Windows deployments. VMWare-based providers are typically better at this versus the public providers such as NexusTek and Rackspace.

Linux - Which Linux?

You want to run a flavor of Linux with support. The best supported Linux for most MV platforms is RedHat Enterprise. There are RedHat clones on the market like Rocky Linux and Alma Linux that will work and there are companies that will offer support. But if you're following best practices, go with RedHat when possible.

← Back to contents

VM Configurations

Sizing your virtual environments is very important. MV systems are both Application Servers and Database servers at the same time. You do not want to undersize or oversize your system. MV systems are very efficient and were designed to run on smaller hardware systems of the past. But as systems became more powerful your developers needed to optimize code less due to the higher performance hardware, especially around Disk and Memory performance.

Ram

You should have enough memory to hold most of your active database in memory. Disk systems have gotten faster but, as you move right on the Cloud Journey diagram shown earlier, you may find the more inexpensive offerings are not as fast as your on-premises NetApp. Your system will load as much as your database from Disk into Memory, minimizing the effects of slower Disk systems. Watch your systems swap usage and if it is in heavy usage, you should expand your memory. Some providers also offer different speeds of memory. While MV is not a compute-intensive app, it is a Database and similar recommendations for a SQL system should be used for your MV system.

Storage

Your storage system is your next-most important item. MV is a Database and the same SQL style recommendations should be used for your Storage System.

1. Use separate disks to store your Database. Do not use the O/S disk.

2. Understand Snapshot options. Most public offerings do not have performant snapshot technology as an on-premise NetApp. Some snapshots can take minutes to complete which means down time for your system (you should always pause your Database while taking Snapshots).

3. Stay away from Memory-based storage unless they have been specifically recommended for Databases.

File Systems

If you're following best practices, use a high-performance file system with good journal and snapshot abilities. On Windows you have NTFS. If you're on Linux, it's a best practice to use XFS or ZFS vs Ext3 or Ext4.

← Back to contents

NTFS – Windows

Windows comes with NTFS which is a high performance journaled file system with a built-in Volume Shadow Copy Service.

- Excellent journaling system to fix the file system after a restart
- Volume Shadow Copy Service – Snapshots
- Resilient File System (ReFS)
- Large disk support

Shapshot Support: No MV platforms at this time offer built in Shadow Copy integration and therefore you must manually script the creation of snapshots using Diskshadow. To create a consistent snapshot, you would first pause your MV database using the appropriate tool and then use DiskShadow to create your snapshot and then unpause your database.

NTFS Documentation: NTFS overview | Microsoft Learn

Volume Shadow Copy: Volume Shadow Copy Service | Microsoft Learn

Diskshadow: Diskshadow | Microsoft Learn

XFS – RedHat Linux

XFS is a file system created by Silicon Graphics and is included free in Linux Kernels and therefore most distributions. XFS is supported by Linux and is very performant and has excellent journaling capabilities. It does not have built-in snapshot capabilities and relies on LVM which is a less performant snapshot technology as it is a multi-write system.

- Less CPU usage
- Excellent Journaling via B+ trees
- Red Hat Support
- Snapshots rely on LVM

Red Hat XFS Documentation:
Chapter 3. The XFS File System Red Hat Enterprise Linux 7 | Red Hat Customer Portal

Red Hat LVM Snapshots:
Chapter 10. Snapshot of logical volumes Red Hat Enterprise Linux 8 | Red Hat Customer Portal

← Back to contents

Numa!!! – Do not overprovision!

Numa is non-uniform memory access. Numa is when a physical host has separate memory allocated to each physical CPU. This is a high-speed bus that allows Cores to run in a physical CPU to have much faster memory access. If a single machine allocated more memory than is attached to a single physical CPU, it can use memory from the other CPU via a much slower bus. For Database applications this is very dangerous and can result in performance issues on very busy systems. It is best to allocate your virtual machines to stay within a single Num Core. For example, if your host is a dual CPU box with 64 gigs of memory and 32 cores, this means you have only 32 gigs of memory and 16 cores attached to each physical CPU. If you allocate a machine with more than that (CPU or memory) it will have to use this slower bus at some point. Most operating systems are Numa aware and will do their best to keep you in a single Numa for performance, which means over-allocation provided minimal benefit. If the system does get very busy it may spread across the Numas and you may start seeing strange performance issues (typically this is caused by processes waiting on a process that is running across the slower bus). Stay within your single Numa and if you need more power add more memory/cores to your underlying hosts. When working with third-party hosts, discuss this issue with your provider and get their recommendations.

Here is a link to some Database specific articles discussing Numa.

NUMA and database headaches • JurisTech

Ulimits – Large systems need tuning based on loads

Linux systems have built in resource limits for the entire system and individual processes. These are designed to keep a single runaway process from taking all your resources and affecting all users. As your MV system gets larger, you will need to tune up these parameters based on your system requirements. You can set many of these parameters to unlimited, but that is dangerous and will allow a runaway process to consume all your resources. Examples of runaway processes can be programs not properly releasing memory or bad queries selecting your entire Database.

RedHat article on setting Ulimits: How to set ulimit values - Red Hat Customer Portal

← Back to contents

Connectivity

SSH vs Telnet

If you're following best practices, you should always use SSH for remote console connectivity. Many legacy MV apps are console based and typically use MV emulators such as Rocket® AccuTerm® or Rocket® wIntegrate®. SSH is an encrypted connection and has multiple security models for use/password authentication such as key-based, Active Directory/Pam integration, or other models. SSH servers also have multiple configuration options and have great security logging modules. As your MV application moves from Private on premises access to Public/third-party clouds, if you're following best practices, you'll need to migrate to SSH versus Telnet. Also, you can forward traffic through an SSH connection with most SSH clients. Some think that if a site is also using a VPN, that it's okay to use telnet, but this is an issue if you are sending your passwords unencrypted, meaning anybody on your private network could sniff around the network and capture other people's passwords.

AccuTerm/IO (Sockets)

A new console connectivity option is AccuTerm Web, an HTML/Javascript-based version of AccuTerm that connects to your hosts via Secure Web Sockets. Web Sockets are a bi-directional web technology that is often used by Instant Messaging software. With AccuTerm/Web your connection is secure, and the encryption keys are handled by the same technology doing your HTTPS encryption. Moving from SSH keys to Enterprise Web Certs can be more secure than some SSH techniques and may align better with your security departments.

SSO

SSO is single sign-on. This option will often fall into a few parts of your MV Application.

1. Console app login via Emulators such as AccuTerm/wIntegrate. These applications typically rely on the Host Operating system to login. On a Linux system this is typically the /etc/passwd system via PAM or on Windows Authentication. If you have an enterprise SSO system, such as Active Directory, you can integrate the host O/S to these systems. Windows is typically straight forward with AD while Linux has SSSD that integrates with AD. Other third-party providers such as Okta or Google have their own tools to integrate. Keep in mind you will have to map Group security to these systems.

2. You App. Your app may have its own security system built into it. Legacy console apps may bypass the Unix layer and prompt you (or keep both) while many GUI apps will implement some type of login security into those apps. Adding SSO to these types of apps can often be complex and beyond the scope of this document. MV apps can easily use LDAP tools and pass credentials to your SSO system for verification, but if you're following best practices this is not a good long term approach. You should be separating your password logic from your application and have third-party security software handling this. As stated earlier, this is accomplished by integrating AD into PAM. With Web and GUI software you would work with your security provider to integrate existing Authentication into your app. You often see this today with web technology when you are redirected to something like Microsoft AD to authenticate versus the application prompting you directly.

3. **Admin tools.** Often your internal admin tools will have their own use/password modules and may not offer AD/Integration options. It is best to maintain close controls on these.

    a. Limit port access via your firewalls.

    b. Use encryption if possible. If encryption is not allowed and you are accessing remotely, look at possibly using ssh port forward.

    c. If encryption is not possible, look at using VPNs or remote desktops within the protected data center network.

    d. Never expose these tools externally to the network.

    e. If passwords are clear/text or easily located on the Host, consider at using permissions to limit access.

VPNs

Virtual Private Networks are a great way to protect access to your MV system. With a VPN your connection is secured by the VPN and all traffic to your VPN gateway is encrypted. This protects all your traffic from untrusted sources such as the internet. The downsides of a VPN are:

1. They require specialized VPN software, which may restrict access to certain devices.

2. A compromised machine may have greater access to your internal network due to Network Access. It is important when setting up VPN gateways to have strict access controls and a demilitarized zone. You do not want to allow users to VPN into your main Server Network.

Web Based security controls

If you want to expose your MV application directly to the internet without a VPN, you must properly set up your web-based controls.

1. Telnet — Do not use

2. SSH — Use strong password rules

3. Have good IDS (Intrusion Detection Software) looking for hacking attempts

4. Use IP blacklists to block IDS notifications

5. Consider using IP whitelists where the SSH port is only exposed after a web authentication

6. For web apps use https

Token based/per IP based controls

Many firewalls and Linux can be configured to allow access to an exposed port on an IP-by-IP basis. This allows you to block access to your SSH port and make it look like it does not exist. A user can then use a Web app to authenticate (using stronger Web authentication protocols) and once authenticated use API tools to temporarily allow a verified IP to see the port. This keeps hackers from finding SSH servers and trying to hack in since to most users it appears that there's no SSH server.

Good access Controls

Good tracking. Watch for attacks

Use good IDS (intrusion detection software) to watch access attempts. Many firewalls will offer IDS software that will detect multiple failed attempts and automatically block the port. Linux has free offerings such as fail2ban.

Detecting Unauthorized Access With Linux Security Logs (libertycenterone.com)

Access Controls

Use good access controls on your servers. Do not use Root or Admin permissions for normal use activity. If a bad actor gains access to your system, they will have full control. Limit your exposure by properly using access controls.

Other Services

If possible, look for managed services instead of building your own. All third-party services offer excellent support and easy to use access. In addition, these services will understand security considerations.

FTP

Standard FTP is not encrypted and therefore not secure. If you must use FTP use SFTP or FTPS instead. Try not to reuse normal account passwords. Use the same logging tools to watch access attempts to your FTP server. If you must use FTP, then use a VPN to gain access. Better options for FTP are web-based applications to host files and even allow uploading. This allows you to use normal Web security controls.

Email – SendGrid/Cloud/Trellio/etc.

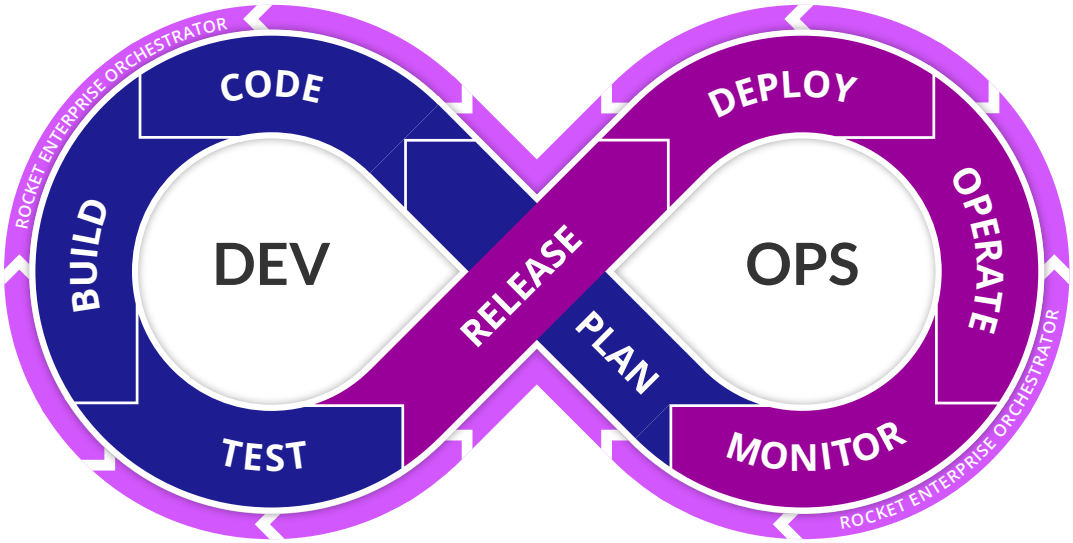
Email systems are a simple commodity today and it is best to use hosted/managed email services rather than using your own local email applications such as Postfix. Most hosted email systems such as Microsoft Outlook and Google offer email APIs to allow applications to send email. Use their encrypted offerings and if possible, move to their API/Rest based options. You can also use services such as SendGrid. Today it is easy to get marked as a SPAM sender if you do not properly identify your email which is why it is important to use managed services that can assist you in properly marking computer generated email. Most services such as SendGrid also offer services to unsubscribe or manage why a user is not receiving an email.

Storage – Usually PDFs, CSV files, etc. move away from Windows Shares

Move to normal web-style shares for shared files such as PDF and CSV files. As you move to public clouds the option to use Windows-style shares is limited. It is better to use block storage such as Azure block storage, sync files around with tools such as Rsync, etc.

Deployment techniques – Git vs Windows shares to move code around for example

Move to more modern DevOps processes to deploy code. Use modern tools such as Git and modern workflow tools. Have proper Dev and QA systems; do not develop in production. See the DevOps best practices for MV for assistance.



Web services (MVConnect, MVIS, U2 Web DE, Bluefinitty, RDM, homegrown, etc.)

Use good edge/proxy servers to manage your web access to the internet. Do not expose your MV system web servers directly to the internet. Even when using secondary servers such as MVIS it is best to put an independent Web Server/Gateway as your Edge server and proxy requests to your internal MVIS or another internal server. This allows your admin security team to separate duties and create layers of protection. There are lots of options for your Edge servers.

1. NGINX or Apache
2. HAproxy
3. Kong
4. Kemp
5. F5 Servers
6. Azure, Google, and AWS all offer API and Proxy servers
7. CloudFlare

← Back to contents

HA/DR and Backups

Backups/Restores – MV systems are databases

Snapshot technology is a supplement to your backup process and not a replacement. All MV systems offer a file-by-file backup process that also checks your files for integrity. These backups can be run while the system is running, which is a benefit. The downside of backups is that they are not a moment in time backup. As the backup runs it is backing up each file one at a time and files near the end of the backup are more up to date versus files backed up early in the process.

A restore from these MV backup tools typically will resize files giving you well sized files. Indexes are usually recreated.

Snapshots

Storage snapshot technology is a great tool to supplement your backup processes. Snapshots are NOT consistent, meaning running a snapshot without pausing writes to your database can result in files that are damaged in the snapshot due to a write happening during the snapshot. It is best practice to pause the database, make your snapshot, and then resume the database. Review your snapshot technology to determine if this is possible. Many Public Cloud default file systems have very slow snapshot technology that can take minutes to complete which is way longer than a typical site would want to pause the database. If you wish to use this technology, choose file systems/options with snapshots that can be done in less than a few seconds. NetApp Sans for example can do this and public offerings do offer upgraded storage systems with better snapshot technology, and some even offer NetApps.

HA/TJ/Restores

If you're following best practices, you should use transaction journaling in all cases. TJ will write a journal of all writes as they happen into a special journal. If you have a failure, TF allows you to restore for a point in time backup and then use your journals to recover writes between backups (or snapshots). These journals also give you another possible restore point if you have inconsistent files due to power failures during writes or snapshots without pausing the database. These journals are also often used to send updates to secondary systems. You need to make sure you have enough disk space to store database updates between your backups/snapshots. If you're following best practices, you should double and even triple your space requirements to cover longer periods of time (such as holidays or during system issues).

Logging – watching disk space, etc.

Logging is very important with backups and journals. You should have logs and alerts that watch your disk space and replication lag. Many journal systems will pause the system if disk space is used up. It is best practice to put these logs/journals on separate disks.

← Back to contents

Security

SELinux/Other kernel level tools

Many Linux systems come with Kernal security modules such as SELinux. These modules are an extra level on top of your user/group permissions securing your system. They look for unusual activity and will block processes and disk updates. You must train these modules to your MV systems' behavior and be prepared for strange behavior. If you're following best practices, run these modules in permissive modes at first. In this mode it will not stop processes/writes but will log them letting you know what it would have done. Run these tools for a few months to find out how your MV system runs and properly train the modules. You should have Admins that are very familiar with these tools and understand the training process. If you are having weird system issues, put these modules back into permissive mode and see if the issue goes away.

Windows Security tools

Windows systems have their own advanced security tools like the SELinux tools for Linux. These are typically third party. These tools may flag MV processes and either block them or slow them down as it views activity as unusual. As with SELinux make sure you have Admins that are well versed in these tools and know how to turn them off and view logs. As an example, a site had moved to Azure with Windows images with built in security tools. Web activity to the server was running very slowly while using the browser directly on the server was running fine. It was later determined that socket activity the Web used to talk to MV was flagged by security software and external access was heavily throttled. The security software was protecting the server thinking malware software was possibly sending Server Data out. The issue was only found by reloading a clean server from actual Microsoft installation disks with no security software.

User Permissions (Root/Admin vs users)

Do not run MV processes as Root or Administrator. Use user and group permissions. Running items as Root/Admin allows a compromised MV process to access your entire Operating System and possibly your entire network.

Logging – Watching activity

It is very important to have centralized logging where all your logs from all systems are sent to.

Advantages:

1. Logs on compromised systems are secured off the machine. Logs are the first items a hacker erases.
2. Reduces disk requirements on your servers. You focus your log storage on your logging server.
3. Use cloud services where possible. Some good examples are Data Dog and Splunk.
4. Ability to merge multiple logs together. With API technology you can watch a transaction through multiple systems and applications.
5. Alerting.
6. Consolidated graphing.

← Back to contents

Encrypted files

If you need encryption for Data at Rest, you have a few options. The most important item with encryption is the encryption keys and how they are stored and loaded. Based on your compliance requirements you may need to use third-party tools to store these keys. Both Windows and Linux offer encrypted disk options and usually offer access to third-party key systems (such as KIMP). If you are in a public cloud site, those sites offer encrypted disks and built in key management systems. Many MV systems offer built in encryption techniques on a file-by-file basis. Pay close attention to the key management and your compliance requirements. If you have compliance requirements keep in mind logs, journals, etc. can all have sensitive information (such as an SSN). Using full disk encryption is a good way to make sure all locations are encrypted.

Encrypted transmissions (SSH vs Telnet, https vs http, etc.)

Data in transit is typically data that is being moved around the network. An example is MV apps still reliant on Telnet/SSH apps and using tools like AccuTerm for viewing data, including API-based technology based on REST or Soap to legacy applications such as FTP. If you are sending sensitive information over these technologies such as passwords, social security numbers, or credit card numbers, you must use encrypted transmissions. Many legacy technologies such as FTP and Telnet are not encrypted and since you enter passwords and are viewing sensitive data these technologies should be deprecated. Legacy protocols such as FTP and Email (SMTP) do offer encrypted versions and usually require special clients and certs that you often have to manually manage. You can also use SSH to tunnel traffic, but this is often complicated. It is best to move to modern technologies such as SSH vs Telnet or HTTP/Web technology vs legacy API or FTP technology. AccuTerm Web for example offers a Secure Socket/Web option vs SSH allowing you to centralize your encryption technology and more easily manage your certs.

Keep software and Operating Systems up to date! DevOps

It is very important to keep your Operating System and software up to date. As you move to Public Clouds such as AWS/Azure/Google you need to keep your Operating Systems up to date as these platforms typically do not support older legacy operating system versions. All cloud systems use Hypervisors and require modern operating systems and recognize virtualization and many are often making constant security and performance improvements. Security is also very important to keep your systems safe.

Your MV and other software needs to be kept up to date to keep up with the latest operating systems. These two items work together and not keeping your software up to date will force you to rely on older operating systems.

About Rocket Software

Rocket Software partners with the largest Fortune 1000 organizations to solve their most complex IT challenges across Applications, Data and Infrastructure. Rocket Software brings customers from where they are in their modernization journey to where they want to be by architecting innovative solutions that deliver next-generation experiences. Over 10 million global IT and business professionals trust Rocket Software to deliver solutions that improve responsiveness to change and optimize workloads. Rocket Software enables organizations to modernize in place with a hybrid cloud strategy to protect investment, decrease risk and reduce time to value. Rocket Software is a privately held U.S. corporation headquartered in the Boston area with centers of excellence strategically located throughout North America, Europe, Asia and Australia. Rocket Software is a portfolio company of Bain Capital Private Equity. Follow Rocket Software on LinkedIn and X (formerly Twitter).

Rocket® software

Modernization. Without Disruption.™

Visit RocketSoftware.com ›

Talk to an expert

MAR-10150_WP_MVCloudBestPractices_V5