CARBONITE

# 5 STEPS TO IMPROVING IT RESILIENCY

How to build a strategy that delivers on business recovery objectives.

There are five essential steps to protecting data and applications from the most common causes of data loss and downtime. IT pros who follow these steps can feel confident in their long-term organizational plan. Read this expert guide from Ben Maas, an independent consultant and system architect who has guided many companies through backup and disaster recovery (BDR) deployments.

**TIP #1**
Know your software's capabilities.

**TIP #2**
Know uptime requirements for all applications.

**TIP #3**
Properly size your environment.

**TIP #4**
Properly size and place data repositories.

**TIP #5**
Practice makes perfect.

**BEST PRACTICE #1**

# KNOW YOUR DATA PROTECTION SOFTWARE

Many people use data protection software without understanding its full capabilities and/or limitations. For instance, backup software can employ several different methods to create safesets for recovery. It can replicate at the file level, application, volume, hypervisor or operating system, or a combination of methods in order to provide multiple options for recovery. The backup software for virtual machines (VMs) is a great example. Most use snapshot technology to perform this task, though each may use different techniques to do so. Some use an agentless approach that calls VMware's native VM snapshotting technology. Others use software agents that deploy an agent on each VM.

If your backup software relies upon agents to do VM backups, it will work more directly with the VM's file system. In this case, the backup software probably uses Microsoft's Volume Shadow Copy Service (VSS) to coalesce the data to disk before it takes the snapshot of the VM.

If your backup software takes an agentless approach to snapshots, it may still partly rely upon agents to take backups. One backup software provider temporarily places a piece of software into the VM when it executes its backup to call Microsoft VSS to create a snapshot. To do so, it initiates the snapshot using the VMware APIs, which then places the software code on the VM to create the snapshot. Once it completes the snapshot, it then removes the piece of code it installed.

Even this hybrid approach to VM backup may be insufficient. In some cases, the backup software may need to integrate with specific applications such as Microsoft Exchange or SQL Server to sync the data to disk. This will create an application-consistent backup that is usable after it is recovered.

Similarly, many backup software products also use deduplication to reduce the backup size. Know your data deduplication options. Some deduplicate the data on the client, others at the media servers, and still others only deduplicate the data once it arrives on the disk storage device where the data will be stored. Some even offer options to deduplicate the data in any of these three locations or to not deduplicate data at all.

The option or options your software supports will affect the amount of bandwidth that you need to perform this operation and the amount of processing power required on the client, media server, and/or disk target to deduplicate this data. Knowing these capabilities and limitations of your backup software is important because they influence how long your backups and recoveries take and, ultimately, how reliable they are.

## BEYOND BACKUP & RECOVERY

Mission-critical applications should be online all the time, or as close as possible. This level of service requires more advanced tools than backup software can provide. Businesses that have zero tolerance for downtime should consider a high availability (HA) solution for critical systems, like DoubleTake Availability from Carbonite. HA ensures always-on services by replicating systems in real time to a remote location. If a disruption occurs in the production environment, HA lets you instantly fail over to the secondary location until you remedy the situation locally. Recovery for HA is measured in minutes or seconds, and data loss can be minimized to near zero.

**BEST PRACTICE #2**

# UNDERSTAND UPTIME REQUIREMENTS FOR APPLICATIONS

Once you understand the capabilities and limitations of your backup software, you need to understand the recovery objectives for each of your applications. Once you establish those objectives, you need to map them back to the features available in your software and even your own internal processes to make sure they align and that you can maintain availability for those applications according to business requirements.

For instance, MySQL does not have an approved method for the live snapshotting of its data. As such, there is no way for you to prove that your backup software successfully synced up data to disk at any moment in time to create a recoverable snapshot.

The only proven ways to back up MySQL is to either power off MySQL, which does not make sense for an application that requires 100 percent uptime, or to make a replica of that data and then take a snapshot of the replica. Examples like MySQL illustrate why you need to understand where your data lives and how it operates so you do not need to run a restore to find out that you are missing data or that it is corrupt.

Conversely, software such as Microsoft SQL offers APIs to provide you a better data protection experience than MySQL. Using VSS shadow copies, you avoid these issues. Here again, you need to make sure that your backup software knows how to appropriately call the APIs to verify that your data gets written to disk to minimize and ideally avoid the possibility of either data loss or corruption.

This is a very important step, especially if you are dealing with applications or regulations that require the backup software to encrypt data stored on the drive or in memory. Encryption creates an added level of protection—and you need to make sure that the backup software encrypts the data before it gets on the drive. Many providers require you to manage and keep your own encryption keys. And IT pros have a responsibility to protect those keys. If you lose your encryption keys, you lose your backups and, if you lose your backups, you lose your data.

**BEST PRACTICE #3**

# PROPERLY SIZE YOUR ENVIRONMENT

There are two types of backups that you need to consider in order to properly size your environment for backups.

1. Data center backups
2. Over-the-wire/Remote backups

**Data Center Backup**

Backups that occur in the data center are probably the easiest to quantify and size. You tend to have dedicated networks for the backups of these application servers and this backup traffic may not even go over the corporate network. Production application data may be protected by array-based snapshot technologies where the backup software initiates snapshots of the data, which are stored short term on the array and managed by the backup software. The backup software may then back up that snapshot to disk, tape, or even the cloud for long-term preservation. The more sophisticated backup software used in your corporate data center tends to make it easier to size backups of applications hosted in the data center.

It is when you start to talk about backing up applications located outside of the data center—whether they are located elsewhere in the building, on your campus, or at a remote location—that it becomes more difficult to properly size your environment for backup and recovery.

If you are doing local backups over LAN connections, you need to verify that you have access to sufficient computer resources and network bandwidth during the backup window to avoid disrupting production applications. Since backups tend to run during off-hours, this is generally not an insurmountable issue.

However, if you identify applications outside of your core data center that run 24x7 and do not have periods of low activity, you may need to upgrade the compute resources on these servers and/or provide them with additional network bandwidth to ensure their backups and recoveries can occur within the scheduled backup window. You may also want to consider more advanced backup tools, such as a high availability solution (HA). HA technology uses instant failover functionality to ensure uptime requirements for mission-critical applications and data.

# PROPERLY SIZE YOUR ENVIRONMENT *(cont.)*

**Remote Backup**

If you need to back up or recover applications running in remote locations via a WAN connection, the challenge becomes even more pronounced. Aside from making sure you have the available compute and network resources to back up and recover the data, you also need to verify that you can recover it in a timely manner; otherwise you will not meet your recovery objectives. The only real way to know is to test it in your production environment.

As you do so, consider certain variables that you might encounter in your environment when doing backups or recoveries. For instance, if you are going to run your backups or recoveries over a VPN tunnel, your throughput will go down. Also, do you need to encrypt data before you send it over a LAN or WAN link? If so, verify that the device that will encrypt the data can do it in a timely manner to meet your backup or recovery service level agreements.

One other thing to keep in mind is that the disk on which you store backup data is sufficiently fast to keep up with the demands of your backup and recovery demands. I have encountered situations where companies have had so many machines simultaneously writing or reading data that the processing slows down.

Consider the situation where you may have 50 machines that you need to recover within 24 hours. You probably are not going to try to recover them one by one by one. You are going to want to recover them in parallel. You need to make sure that the storage device from which you recover the data can handle the amount of I/O that's necessary to meet those demands. Again, there are calculators out there that can help you make these types of assessments but I have found that the only way to be sure is to test it out yourself in your environment.

**HAVING THESE REPOSITORIES PROPERLY SIZED PARTICULARLY COMES INTO PLAY WHEN DEDUPLICATING DATA DURING THE BACKUP PROCESS**

BEST PRACTICE #4

# PROPERLY SIZE AND PLACE DATA REPOSITORIES

I have encountered situations where the software provider places restrictions on how much data you can put into an individual repository. For instance, the backup software provider may impose a 2 TB limit (or some other limit on a single backup repository) that may force you to spread your backups across multiple repositories.

This comes into play if you are concurrently running multiple recovery streams. In these circumstances, you need to make sure that the repositories can read data back fast enough to meet your recovery time objectives (RTOs).

There are sizing documents out there from many vendors that are very helpful in properly sizing the repositories for your environment. You just need to make sure that you have configured enough repositories and have them available at the same time. I have found that having these repositories properly sized particularly comes into play when deduplicating data during the backup process. In cases where they are inadequately sized, the deduplication process can be slowed.

Also, be aware that vendors use backup proxies to get closer to the storage on your virtual hosts. In those circumstances, you need to make sure that you have properly sized them as well to ensure that you have enough RAM, CPU, and local storage to avoid creating a bottleneck at some point during the backup or recovery process.

I have also worked with VMs that function as database servers, which host 7 to 8 TBs of data. There were times where these size VMs would try to recover that data from a single repository. In those situations, it became a real problem because there was not enough throughput. Only after I distributed the data across multiple repositories was I able to recover in a timely fashion as the company could run restores across multiple drives at the same time.

BEST PRACTICE #5

# PRACTICE MAKES PERFECT

Practice makes perfect. This means you should run multiple tests to get it right. You never fully realize how many moving pieces there are to a recovery process until you perform one. Perhaps the most complex ones are those that involve recovering from geographically dispersed backups. In those cases, you need to run recovery tests to make sure that everything that you think is going to happen happens.

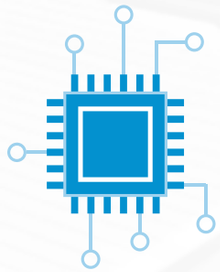
Most of the time, I run into issues during testing that I never considered as possibilities ahead of time. In one case, I encountered a problem with software licensing. After I restored the application during the test, the application software had to call home to verify its license. It calling home was not the problem. The problem surfaced after it called home. During the call-home process, the licensing software detected that the IP address of the server hosting the software had changed since I was running the application on a test server. It then invalidated the software license. While this was inconvenient, this became a production issue because it invalidated the software licenses for both the copy of the software running in test and in production. This oversight took down the production environment.

This led to making changes in how I conduct DR testing. Now when I bring up the testing environment I briefly shut down outbound internet traffic. During this period, I watch to see what traffic is outbound to make sure no software is trying to call home that could inadvertently cause outages in either the test or the production environment. That may represent a certain level of paranoia on my part and I do not necessarily tell other people to go to that extreme. However, once bitten, twice shy, and I personally have found software licensing to be a problem during recoveries.

Another good example of why you need to do tests is to ensure you can recover. One company for which I worked created an "X" drive or file share on its Microsoft SQL servers. Then once a week it would back up data to this "X" drive. However, unbeknownst to me, another guy in the company knew about this "X" drive and what it was used for so he decided to use it to do some replication between two of these SQL Server database servers, which worked fine.

After some time passed, however, the company changed its backup procedure and decided its SQL Servers no longer needed this "X" drive on these database servers. I evaluated the system, and dropped the "X" drive across the entire environment. But we wound up with one person screaming at me, "Why is the replication broken?"

In short, these situations illustrate why testing is so important. Aside from the changes that constantly occur in your environment, there are always going to be nuances in your environment, such as undocumented uses of the "X" drive, that make it extremely difficult to confidently recover your environment in the way that users expect unless you routinely perform recovery tests.



**CONFIDENTLY RECOVERING YOUR ENVIRONMENT BEGINS WITH TESTING**

Ben Maas owns and operates ALPTech. He has been an independent consultant for years, advising and helping businesses on their IT strategies including backup and DR.

**Ben Maas**

CEO and System Architect with ALPTech