Brought to you by:

VEEAM

AWS Backup

for dummies®
A Wiley Brand

Understand the threats facing your data

Build a solid backup regimen

Create a robust recovery plan

Veeam Special Edition

David Clinton

About Veeam

Veeam® is the leader in backup solutions delivering Cloud Data Management™, serving more than 365,000 organizations across the globe. Founded in 2006, Veeam quickly grew to become an industry leader in data center backup and recovery. But, as the importance of data has grown to drive every aspect of the digital business, so has the need for solutions that can do far more than ensure its availability. With its strong partner ecosystem and breadth of integrations, Veeam has the most complete software platform to meet today's new expectations for data.

Veeam automates the provisioning and management of the massive, constant flow of data running across on-premises and multi-cloud infrastructures, both today and tomorrow. Veeam solutions receive the highest customer-satisfaction scores in the industry, 3.5 times the industry average, for their simplicity, flexibility, and reliability. As hundreds of thousands of customers have said, "It Just Works!"

AWS Backup

Veeam Special Edition

by David Clinton

for dummies®
A Wiley Brand

AWS Backup For Dummies®, Veeam Special Edition

Published by
John Wiley & Sons, Inc.
111 River St.
Hoboken, NJ 07030-5774
www.wiley.com

LIMIT OF LIABILITY/DISCLAIMER OF WARRANTY: THE PUBLISHER AND THE AUTHOR MAKE NO REPRESENTATIONS OR WARRANTIES WITH RESPECT TO THE ACCURACY OR COMPLETENESS OF THE CONTENTS OF THIS WORK AND SPECIFICALLY DISCLAIM ALL WARRANTIES, INCLUDING WITHOUT LIMITATION WARRANTIES OF FITNESS FOR A PARTICULAR PURPOSE. NO WARRANTY MAY BE CREATED OR EXTENDED BY SALES OR PROMOTIONAL MATERIALS. THE ADVICE AND STRATEGIES CONTAINED HEREIN MAY NOT BE SUITABLE FOR EVERY SITUATION. THIS WORK IS SOLD WITH THE UNDERSTANDING THAT THE PUBLISHER IS NOT ENGAGED IN RENDERING LEGAL, ACCOUNTING, OR OTHER PROFESSIONAL SERVICES. IF PROFESSIONAL ASSISTANCE IS REQUIRED, THE SERVICES OF A COMPETENT PROFESSIONAL PERSON SHOULD BE SOUGHT. NEITHER THE PUBLISHER NOR THE AUTHOR SHALL BE LIABLE FOR DAMAGES ARISING HEREFROM. THE FACT THAT AN ORGANIZATION OR WEBSITE IS REFERRED TO IN THIS WORK AS A CITATION AND/OR A POTENTIAL SOURCE OF FURTHER INFORMATION DOES NOT MEAN THAT THE AUTHOR OR THE PUBLISHER ENDORSES THE INFORMATION THE ORGANIZATION OR WEBSITE MAY PROVIDE OR RECOMMENDATIONS IT MAY MAKE. FURTHER, READERS SHOULD BE AWARE THAT INTERNET WEBSITES LISTED IN THIS WORK MAY HAVE CHANGED OR DISAPPEARED BETWEEN WHEN THIS WORK WAS WRITTEN AND WHEN IT IS READ.

For general information on our other products and services, or how to create a custom *For Dummies* book for your business or organization, please contact our Business Development Department in the U.S. at 877-409-4177, contact info@dummies.biz, or visit www.wiley.com/go/custompub. For information about licensing the *For Dummies* brand for products or services, contact BrandedRights&Licenses@Wiley.com.

ISBN 978-1-119-68713-9 (pbk); ISBN 978-1-119-68714-6 (ebk)

Manufactured in the United States of America

10 9 8 7 6 5 4 3 2 1

Publisher's Acknowledgments

We're proud of this book and of the people who worked on it. Some of the people who helped bring this book to market include the following:

Project Editor: Martin V. Minner

Editorial Manager: Rev Mengle

Associate Publisher: Katie Mohr

Business Development Representative: Karen Hattan

Production Editor: Mohammed Zafar Ali

IN THIS CHAPTER

- » **Knowing what can go wrong with infrastructure**
- » **Understanding what can go wrong with backups**

Chapter 1
Understanding the Threats Facing Your Cloud Data

So it turns out that data doesn't protect itself. And despite providing what might well be the most secure and reliable compute platform the Universe has ever seen, Amazon Web Services (AWS) can't guarantee that you'll never lose data, either. To understand why that is, you'll need to face your worst nightmares while visualizing all the horrifying things that can go badly wrong, and then boldly adopting some best-practice solutions as you map out a plan to protect yourself.

Before I begin, though, you should understand how the AWS Shared Responsibility Model plays a role in the way you build your cloud applications. Amazon, according to the model, guarantees the viability of its cloud, and you accept responsibility for everything you deploy *into* the cloud. The security and functionality of all the servers, storage devices, and network switching and cabling — along with things like building security and cooling — are managed by Amazon. Amazon also stands behind the software infrastructure driving AWS *managed* services (like AWS Elastic Beanstalk or AWS Lambda). But all the data, configuration, and account administration settings that you add to the mix are your problem.

When Cloud Infrastructure Fails

With that in mind, it's worth spending some time talking about the unavoidable limits built into Amazon's platform.

First, take a moment to appreciate the big difference between availability and durability. When AWS guaranties that a particular service will be *available* for, say, 99.99 percent of the time, you and your customers should expect to be able to access your data for all but 0.01 percent (around nine hours) of a given year. But that doesn't mean there's a 0.01 percent chance you'll permanently *lose* your data. Just because a networking or hardware failure in one or more Amazon data centers temporarily blocks access doesn't mean your data has disappeared. The odds are that multiple copies of your data are still safe and sound, and those copies will be used to restore your access within minutes.

Those "odds" are described in terms of how *durable* your data is. For the Amazon Simple Storage Service (S3), that would be 99.999999999 percent of the time. For all intents and purposes, it would be statistically impossible for your data to be permanently lost anytime between now and the year 12,020 (10,000 years from now). So I won't be staying up too many nights worrying about my Amazon S3 buckets.

But not everything on AWS is as simple as Amazon S3. Amazon Elastic Compute Cloud (EC2) instance store volumes, for example, have a much lower durability rating. And I'll bet you can figure out why AWS recommends you replicate many elements of your infrastructure across multiple Availability Zones. The threat of failure is a part of life. Ignore it at your peril.

WARNING

AWS protects your applications from distributed denial-of-service (DDoS) attacks by default through AWS Shield, but even they're vulnerable. One such attack took out some of Amazon's own Route 53 DNS service for a staggering eight hours in October 2019!

When Your Configurations Fail

All things considered, AWS vulnerabilities — few and far between as they tend to be — are the good news. The bad news is your end of the Shared Responsibility Model. Unfortunately, infrastructure configuration vulnerabilities are not only common, they're often catastrophic and occasionally the stuff of legend.

Where are the mistakes most likely to get you into trouble? Arguably, you only need to remember two things when planning your configuration:

- Following the principle of least privilege = double-plus good.
- Allowing single points of failure = double-plus "ungood."

Here's why. First of all, the principle of *least privilege* is an administration design pattern that provides the members of your team with just enough access to your organization's resources to effectively do their jobs — but not one inch more. What's a few extra inches between friends? You don't want to come across as unreasonable, do you?

Well, perhaps you do. For lots of reasons, every unnecessary layer of access risks exposing another avenue of attack:

- If you use your AWS root (or a full admin) account for day-to-day administration of your account resources, you're increasing the potential damage should that account be hijacked by hackers.
- If you don't immediately remove account access for users who have left your organization, you're opening yourself to a revenge attack that can begin *behind* your firewall.
- If you hard code remote authentication information into your scripts, you're making it way too easy for hackers to get access to your resources — especially if you save your code to public repos on sites like GitHub.

WARNING

In 2016, a British company called Voova fired a worker after a brief period of employment. The fellow then used stolen credentials to delete the company's AWS instances, causing significant damage.

Now what about that single point of failure thing? A point of failure is anything that can go wrong. A *single* point of failure is anything that, if it goes wrong, leaves you without any plan B. That's very bad.

All hardware (and most software) will fail. It's just a question of when. Odds are that your "when" will come at the worst possible time — for example, just as demand for your application is starting to take off. Sure, well-managed cloud resources are far less likely to fail, leaving you high and dry. But, as you've probably discovered on your own, it happens:

- If you provision and launch a public-facing e-commerce website running on a single server, what happens when that server fails?
- If you provision and launch a public-facing e-commerce website running on *two* servers that live within a single AWS Availability Zone, what happens when that zone goes offline?
- If you route the traffic aimed at a database running in a private subnet through a single NAT host, what happens when that NAT host goes down?

It's almost like you're on your knees and begging — pleading! — for something unspeakably awful to happen.

WARNING

Remember Code Spaces? I'll bet you don't. Back in 2014 Code Spaces was competing with GitHub. Then one day it wasn't. What happened? A malicious actor gained access to the company's Amazon EC2 management console — presumably through a weak or exposed admin password — set about creating backup logins, and eventually deleted all of the Amazon S3 buckets, Amazon Machine Images (AMIs), and Amazon Elastic Block Store (EBS) resources. By the time the company regained control it was too late; everything was gone.

When Your Local Infrastructure Fails

AWS has you good and covered for the hardware and networking that make up the AWS part of the cloud. But you're likely still running at least part of your IT stuff locally. Perhaps you have a central office where your development and staging servers run. Or maybe your developers and admins still work together in a single complex of rooms.

Now, what if you all showed up for work one morning to see the smoking ruins of your office building surrounded by fire trucks? Could you recover?

I remember asking the folks at one company that question. After a few moments of thoughtful silence, I was told that nearly all of their code was replicated on off-site repos and they had copies of server images that could be retrieved in an emergency. What no one mentioned — but everyone was thinking — was that it would take weeks to reassemble and relaunch their applications based on those resources, assuming they could even get everything to work. Who knew? After all, nothing like that scenario had ever been tested.

Fires aren't your only worries. Think about an extended power outage, a disruption to your Internet access, or the catastrophic destruction left by a hurricane or tornado.

Nature? Politics? Meh. That's all minor league. Just think about how much fun you'll have trying to put things back together after a ransomware attack leaves your application data encrypted. Imagine having to choose between paying a million-dollar ransom that *might* inspire the attackers to decrypt your data; and facing the prospect of not having a reliable backup. I don't want to give anyone evil ideas, but there's no reason ransomware attacks can't be launched against poorly protected AWS accounts (assuming they haven't already).

No data backups? You're out of luck.

When Stuff Gets Destroyed by Accident

Accidents happen. People click the wrong button or mess up important configuration settings, though AWS does its best to help protect us from ourselves. You can enable termination protection for your Amazon EC2 instances, for example, but, if you're anything like me, you'll still find ways to break things.

What sorts of mistakes are you likely to face?

- Accidental deletion of running resources along with their associated data volumes
- Misconfiguration of automated infrastructure that unintentionally brings down containers or other resources

- Improperly setting life cycle configurations for Amazon S3 buckets causing data to be archived and, eventually, deleted before it's appropriate
- Accidental corruption of production databases

Still no data backups? I guess you're still out of luck.

When You Can't Get Your Backup Back Up

The best insurance against the dark menaces I describe in this chapter is a solid and reliable backup regimen. But not all backups are created equal. To be useful, backups need to be:

- **Current:** Will the data you retrieve be similar enough to the data you lost to be useful?
- **Accessible:** How long will it take you to get to your data?
- **Readable:** Are you sure the backup media are reliable and the archives themselves aren't corrupted?
- **Tested:** Are you sure your restoration plan will work?

Most of all, you need to be able to move your backup archive into production quickly. What good is a perfectly preserved archive if it won't come online as fast as you need it to?

Building an application environment that's secure, resilient, and reliable takes planning and investment. In Chapter 2, you learn how that works.

IN THIS CHAPTER

- » **Designing applications for high availability**
- » **Incorporating backups into business continuity planning**
- » **Optimizing and controlling backup costs**

Chapter 2 Telling Your Data How Much You Love It

Thinking about everything that could go wrong can turn into depression. But don't despair! With some moderate planning and preparation, you can build yourself a robust, wall-to-wall backup and recovery solution that's the next best thing to bulletproof.

Backing up your data is a big deal in its own right, but it's really only one piece of the infrastructure puzzle. You should also take steps to avoid failure altogether. Then, after developing a firm sense of how long your organization can survive a downed application, figure out how to recover quickly in the event of a complete, unexpected meltdown.

This chapter introduces all the bits and pieces you'll need.

Planning to Stick Around

The magic of AWS service and network integration makes it possible to greatly reduce the chances your application will go down. But your design must take the cloud's unique architectural features into account. The next two sections describe best practices that make high availability a particularly achievable goal.

Using multiple Availability Zones

Stuff happens. But the odds of the same stuff happening at the same time in two or three locations separated by hundreds of miles are negligible. So AWS recommends you replicate your compute instances and other resources across multiple Availability Zones. That way, even if one Availability Zone suffers an outage, your application remains accessible through the resources you launched in other zones.

This is true of the single Amazon EC2 instance that could fail without warning, taking your application down with it. But even if you wisely decided to launch multiple instances in multiple Availability Zones, you don't want all of those instances to have to depend on a single *database* instance. The same principle applies to load balancers and other networking resources.

REMEMBER

The bottom line: Protect your data as if your life depended on it (because it likely does).

Designing loosely coupled applications

Thinking about storing dynamic application data on your Amazon EC2 instances or within an AWS Lambda execution? What will happen to that data when the instance fails? How will the instance that's launched to replace it pick up the pieces of half-finished operations and transactions? The better solution is an application that writes its data to a highly resilient and stable platform — like an Amazon S3 bucket.

This kind of design is considered *loosely coupled* because the instances handling transactions require little or no independent knowledge of the current state of the operation. An instance launched to replace one that just failed can simply query the Amazon S3 bucket to retrieve all the data it needs to complete its task.

Cloud systems can manage workload failures by configuring automated failover. Should monitoring processes detect trouble with an application, incoming client requests can be directed away from the primary infrastructure to an exact copy that's been patiently waiting in the wings for the call to service.

The reserve infrastructure stack can be running all along at full speed, requiring only a quick change to the routing rules. This is known as a *hot failover* and is the quickest — and most expensive — solution. Alternatively, you can run a reduced version of your stack full-time that can, when necessary, be scaled up to meet demand once it's pressed into service. This is known as *warm failover* or *pilot light failover*. Finally, you might decide to automatically launch your full failover stack from scratch based on a configuration template (using a service like AWS CloudFormation) that will call on existing resources like pre-built AMIs. In all cases, a loosely coupled application is easier to work into a failover system because your newly launched instances will already know exactly where to go for access to up-to-date data.

But here's a thought: What will happen when you lose a complete infrastructure stack? What kind of backup will give you back your *complete* application, including all the loosely coupled, failover-happy, high-availability goodness I've just described for you? And another thought: How quickly can you get all those resources up and running? This is where the kind of end-to-end solution a company like Veeam can provide comes in.

Creating a Business Continuity/ Recovery Plan

Before settling on the backup plan of your dreams, you need to understand exactly what your organization's operations demand. How long can an outage last and how damaging can it be before it becomes impossible to get the business back on its feet again? You calculate this using these two metrics:

- **The recovery time objective (RTO):** What's the maximum application downtime your organization can endure before you *must* be up and running again . . . or else?
- **The recovery point objective (RPO):** How much transaction and operational data (or how many customer transactions) can you afford to lose during an outage and still bounce back?

Using those figures — along with some business operation-specific insights — as an absolute baseline, you can begin to build your recovery plan. Your next step will be to familiarize yourself with the options for developing a data protection strategy that meets your desired RTOs and RPOs, as well as the length of retention desired or required:

- **Amazon EBS snapshots:** Most Amazon EC2 instances are run using a storage volume carved out of Amazon EBS. Because Amazon EBS volumes host your instance's operating system — along with any application data you maintain locally — it's the first thing you should target for backup. Veeam's AWS-native backup solution — Veeam Backup *for AWS* — enables you to automate the frequency and retention rules you'd like applied to your snapshots.
- **Amazon S3:** Although Amazon EBS snapshots are ideal for frequent data protection and faster recoveries, scheduling frequently and retaining for long periods of time can escalate your AWS bill quickly. Consider how copying snapshots to more cost-effective Amazon S3 object storage enables you to achieve long-term retention while saving on storage costs. Better yet, automate that process, too.

But here's the issue: As powerful and flexible as those tools are, there's little chance they'll get you close to where you need to be to meet your RTO and RPO. After all, having a reliable backup of a data volume is *not* the same thing as having a replacement instance ready to launch. What about all the networking configurations, service integrations, and application architecture you rely on? Are you always going to be sitting in front of your workstation, logged into your AWS account with your finger poised above the Enter key, ready to launch a manual recovery? Don't you ever sleep?

TIP

For all but the simplest cloud deployments, you need a significantly more robust backup solution. Something that covers your entire cloud, hybrid, and on-premises resource stack and offers a higher level of control and automation. Something, like Veeam Backup & Replication for AWS.

Optimizing Your Backup Operations

Anything as critical and, potentially, as complex as backing up multi-tier application data is going to have loads of moving parts. As you put your solution together, you'll face conflicting priorities and more than one approach to solving problems. Like someone in bed under a blanket that's just a bit too short, you'll always have something sticking out. From time to time during the process, you should step back and take a big-picture view. Your goal is optimization.

Addressing risk

Allow me a moment of stark, screaming paranoia. I happen to be a system administrator by trade, so it comes naturally. Just imagine that your organization's AWS account has been compromised. (You remembered to configure multi-factor authentication for your users, right?) You could, as you see earlier in this chapter, lose complete access to all the resources in that account.

"All the resources." Wouldn't those include the backups you so cheerfully stored in Amazon S3? Whoops. What happens now?

Perhaps that isn't the question you should be asking. Instead, think about what you *could have done* from the start to avoid the whole mess.

One possible approach would involve isolating the resources used by your development, staging, and production layers into completely separate AWS accounts. That way, even if one account is brought down, you'll have the resources maintained within the others from which to rebuild.

Won't that add complexity and inconvenience to your backups? Possibly. But a well-designed third-party backup solution like Veeam can smoothly orchestrate the whole thing from a single control panel with relatively minimal effort and reliable results.

Controlling backup costs

However you end up organizing your AWS accounts, you can save yourself significant money by carefully managing the storage lifecycles your backups will use. Data that you might need to access in a hurry should be kept as Amazon EBS snapshots. Older data that, perhaps, has to be kept available in case you're audited, could be stored in the much less expensive Amazon S3 to control costs.

Backup cost estimation tools like those found in Veeam Backup *for AWS* help you proactively understand the financial impact of your backup policies before they surprise you when that monthly bill comes in.

TIP

Of course, don't forget to completely delete archives once they're no longer needed.

Will you remember to transition data between its higher- and lower-cost stages over its years-long life cycle? Probably not. Automate is your best friend.

IN THIS CHAPTER

» Incorporating encryption security in your infrastructure

» Understanding your relationship to your cloud platform

Chapter 3
Ten Cloud Data Things to Remember

Securing your data to ensure it isn't lost in the chaos of cloud life can be a full-time occupation. But here's a helpful checklist of items that are worth reviewing. If you can answer "already nailed it" for all of these, then you're in good shape:

- **Automate, or assume it wasn't done.** You spend time devising and documenting a protocol for protecting your data that involves running a process at regular intervals. And your team does it, once or twice. But do you expect human beings to stick to a routine over the long haul? No. The solution: Automate your data protection processes.
- **Integrate monitoring with automated alerts.** The beauty of cloud virtualization is that complex and powerful things can happen without direct human intervention. But that advantage can also become a threat. The cost of forgetting to cancel unused running resources, misconfiguring service deployments, or unauthorized use of your account can grow exponentially the longer you remain unaware of a problem. So get friendly with cloud and data protection monitoring tools. Better yet, automate your monitoring and alerts.
- **Encrypt.** Except where necessary, don't leave your data in plain text formats that can be read by all who wander by.

Whether through the Let's Encrypt project or Amazon's Key Management Service (KMS), encrypting data both in transit and at rest is relatively simple. Do it.

- **Use multi-factor authentication and roles.** The stronger your authentication process, the better your data will be protected. Requiring users to undergo multi-factor authentication (MFA) before accessing your AWS resources adds serious gatekeeping credibility. In addition, open up programmatic and scripted access only through roles and policies.
- **Be aware of your provider's service-level agreement (SLA).** All responsible cloud providers publish comprehensive SLAs outlining the built-in limits to their service. It's your responsibility to find out how many hours each year you can expect a particular resource to be off-line. Plan accordingly. (See: `https://aws.amazon.com/compute/sla/`.)
- **Patch, patch, patch.** Some AWS resources are "managed," meaning Amazon ensures that the underlying software is properly patched and updated. But it's up to you to care for the operating systems and applications on, say, your Amazon EC2 instances. Your servers rely on you: please keep them well fed.
- **Audit your cloud infrastructure monthly.** Visibility is a challenge when you're dealing with virtual services running across a global platform like AWS. So you should invest time at regular intervals to look at the big picture: How much are you spending? What resources are running? Is anything running that doesn't seem to fit into any ongoing projects? Leverage AWS services like CloudWatch, CloudTrail, AWS Config, and Cost and Usage Reports to get a good view.
- **Match your cloud strategy to your needs.** Just because everyone tells you "go serverless" or "Kubernetes is king" doesn't mean those solutions will work for you. The planning process includes crafting a customized approach to solving your needs — even if that means ignoring popular trends.
- **A single backup is never enough.** Frankly, two backups aren't going to be perfect either. Make sure copies of your data and environment configurations are safely maintained in multiple formats, on media using multiple technologies, and across multiple physical locations.
- **Avoid platform lock-in.** Your business needs might change over time. One day they might be better met by technologies found elsewhere. So, keep your applications as generic and platform-agnostic as possible.

aws

Protect your cloud data

Data doesn't protect itself. And despite providing what might well be the most secure and reliable compute platform the universe has ever seen, Amazon Web Services (AWS) can't guarantee that you'll never lose data, either. This book is your guide to understanding the threats facing your cloud data and mapping out a plan to protect yourself.

Inside...

- Understand how infrastructure can fail
- Protect against accidental data loss
- Design apps for high availability
- Incorporate backups into business continuity planning
- Optimize and control backup costs

VEEAM

David Clinton is a system administrator and a widely-published technology author and course creator. He takes comfort in the thought that dummies sometimes make the best teachers: Who better understands how hard it can be to learn a new technology? Find him at bootstrap-it.com.

ISBN: 978-1-119-68713-9
Not for resale

9 781119 687139

for dummies®
A Wiley Brand