THE EXPERT'S VOICE® IN SHAREPOINT

Pro SharePoint 2010 Disaster Recovery and High Availability

BE PREPARED FOR WHEN BAD THINGS HAPPEN TO GOOD PORTALS

Stephen Cummins

Apress®

Pro SharePoint 2010 Disaster Recovery and High Availability

Stephen Cummins

Apress®

Pro SharePoint 2010 Disaster Recovery and High Availability

ISBN-13 (pbk): 978-1-4302-3951-2

ISBN-13 (electronic): 978-1-4302-3952-9

President and Publisher: Paul Manning
Lead Editor: Jonathan Hassell
Development Editor: Chris Nelson
Technical Reviewer: Jeff Sanders
Editorial Board: Steve Anglin, Mark Beckner, Ewan Buckingham, Gary Cornell, Morgan Ertel, Jonathan Gennick, Jonathan Hassell, Robert Hutchinson, Michelle Lowman, James Markham, Matthew Moodie, Jeff Olson, Jeffrey Pepper, Douglas Pundick, Ben Renow-Clarke, Dominic Shakeshaft, Gwenan Spearing, Matt Wade, Tom Welsh
Coordinating Editor: Jennifer L. Blackwell
Copy Editor: Mary Behr
Compositor: Bytheway Publishing Services
Indexer: BIM Indexing & Proofreading Services
Artist: SPI Global
Cover Designer: Anna Ishchenko

Distributed to the book trade worldwide by Springer Science+Business Media, LLC., 233 Spring Street, 6th Floor, New York, NY 10013. Phone 1-800-SPRINGER, fax (201) 348-4505, e-mail `orders-ny@springer-sbm.com`, or visit `www.springeronline.com`.

For information on translations, please e-mail `rights@apress.com`, or visit `www.apress.com`.

Apress and friends of ED books may be purchased in bulk for academic, corporate, or promotional use. eBook versions and licenses are also available for most titles. For more information, reference our Special Bulk Sales–eBook Licensing web page at `www.apress.com/bulk-sales`.

Any source code or other supplementary materials referenced by the author in this text is available to readers at `www.apress.com`. For detailed information about how to locate your book’s source code, go to `http://www.apress.com/source-code`.

To Jane and Lia
—Stephen

Contents at a Glance

Contents

About the Author

Stephen Cummins was one of the earliest established experts on SharePoint. He set up the first blog about the platform (`www.spsfaq.com`) in 2001 and has continued to share what he's learned since then in many forms of media. Since 2001, he has worked with more than 50 clients on three continents and learned a great deal about a great many businesses and people along the way. He has also learned many lessons about what makes the SharePoint platform adopted, valuable, and resilient. He has eight SharePoint Most Valuable Professional (MVP) awards as well as four Microsoft Certified Technical Specialist (MCTS), two Microsoft Certified Professional (MCP) and two Microsoft Certified IT Professional (MCITP) certificates. He is currently preparing to become a Microsoft Certified Master (MCM).

He lives in Ireland with his wife, daughter, three dogs, and an ever-changing number of goldfish. Hobbies-wise he writes science fiction, surfs and plays the ukulele. He likes to write about himself in the third person, too.

About the Technical Reviewer

■ **Jeff Sanders** is a published author, technical editor, and accomplished technologist. He is currently employed with Avanade in the capacity of a Group Manager/Senior Architect. He is very happy to have been involved in this project and feels it's a long overdue subject for addressing.

Jeff has years of professional experience in the field of IT and strategic business consulting, leading both sales and delivery efforts. He regularly contributes to certification and product roadmap development with Microsoft and speaks publicly on Microsoft enterprise technologies. With his roots in software development, Jeff's areas of expertise include collaboration and content management solutions, operational intelligence, digital marketing, distributed component-based application architectures, object-oriented analysis and design, and enterprise integration patterns and designs.

Jeff is also the CTO of DynamicShift, a client-focused organization specializing in Microsoft technologies, specifically Office365/BPOS, SharePoint Server, StreamInsight, Windows Azure, AppFabric, Business Activity Monitoring, BizTalk Server, and .NET. He is a Microsoft Certified Trainer and leads DynamicShift in both training and consulting efforts.

He enjoys non-work-related travel as well as spending time with his wife and daughter, and wishes he had more time for both. He may be reached at `jeff.sanders@dynamicshift.com`.

Acknowledgments

Acknowledgement must first go to my wife Jane and daughter Lia who stayed with me patiently through the process of writing this book.

I would also like to acknowledge Jonathan, Jennifer, and especially Chris and Jeff for guiding me through the writing maze. I learned a lot and your enthusiasm kept me going at times.

Introduction

I wrote this book to share what I have learned about high availability and disaster recovery for SharePoint at this point in time. It is certainly an interesting time. In the past 10 years, SharePoint has gone from a compiled application that just looked superficially like a web application into a more fully fledged cloud platform. The process is far from over, however, and SharePoint will likely look very different in 10 years time. But there is no doubt in my mind that it will still be in use in some form. It will be interesting for me look back on this book and see what's the same and what's different. I tried to focus on general principles in this book so that even as the technology changes, the principles still apply.

The main risk with any information recording system is that once you use it, you become dependent on it. If that information becomes unavailable for any number of reasons, it has a detrimental effect on your organization. We are just as subject to whims of Mother Nature as we ever were, and now technology has become complex enough that it is difficult for anyone but the most specialized to know enough about it to know how to make it resilient, redundant, and recoverable. In relation to SharePoint, this book will give you the knowledge and guidance to mitigate this risk.

Who This Book Is For

If you worry about what would happen to your organization if the data in your SharePoint farm was lost, this book is for you! It is a technical book in parts, but most of it is about the principles of good planning and stories of how things have gone right and wrong in the field. My intention is that it should be instructive and entertaining for anyone whose organization has begun to rely on SharePoint to function.

How This Book Is Structured

Each chapter describes practical steps that can be taken to make your system more resilient and give you the best range of options when a disaster hits your SharePoint farm. Reading, however, is not enough. I offer pointers to inspire you to take what you have learned here and apply it in the real world. After you read each chapter, put into practice what you have learned! At the very least, take notes of your thoughts on what to do so you can do it later.

Chapter 1: Steering Away from Disaster

To protect your content, you must know your technology and realize its importance to your organization. Roles must be assigned and responsibility taken. Moreover, there should be a way to record near-misses so they can be captured and addressed. SharePoint is not just a technology platform; it's partly owned by the users, too. They and management must play a part in its governance.

Chapter 2: Planning Your Plan

Before you can write a plan you will need to lay a foundation. You will first need stakeholder and management buy-in. You will also need to do a business impact assessment. You may need to plan different SharePoint architectures that have different RTO/RPOs and different cost levels relative to the importance of the data within them. You will also need to create a good SLA and plan how to coordinate a disaster.

Chapter 3: Activating Your Plan

Many processes and procedures have to be in place before you can put your SharePoint disaster recovery plan into action. These are not abstract things on paper; they are actual tasks that defined roles have to perform. This chapter details who is going to do what and when, knowing the interdependencies, accessing the plan, and making sure in advance the plan contains what it should.

Chapter 4: High Availability

High availability is something achieved not just through meeting a percentage of uptime in a year. It is a proactive process of monitoring and change management to ensure the system does not go down. It is also about having high quality hardware. Finally, it is about having redundancy at every level of your architecture from the data center down to the components of the individual service applications.

Chapter 5: Quality of Service

The main ways to improve your quality of service are WAN optimization, designing your farm so that content is near the people who need to see it, and caching infrequently changed pages. WAN acceleration can only help so far with the limitations of latency, but there are options in SharePoint 2010 to get a cost-effective compromise between user satisfaction and a not overly complex architecture.

Chapter 6: Back Up a Step

Your farm is a unique and constantly changing complex system. When focusing on how to back up and restore it successfully, you will need clearly documented and tested steps. You can't fully rely on automated tools, partly because they can't capture everything and partly because they can only capture what you tell them to and when.

Chapter 7: Monitoring

SharePoint must be monitored at the Windows and application levels. The SharePoint application is so dependent on the network infrastructure that anything wrong with SQL Server, Windows, or the network will affect SharePoint. The information in this chapter gives you the guidance and direction you need to watch what needs watching.

Chapter 8: DIY DR

This chapter shows that the task of maintaining backups of valuable content need not be the exclusive domain of the IT staff. Giving users the responsibility for and means to back up their own content is an excellent idea from an organizational point of view as it is likely to save resources in both backup space and IT man-hours.

Chapter 9: Change Management and DR

Change management is a collaborative process where the impact of change has to be assessed from a business and a technical perspective. Change is the life-blood of SharePoint; without it the system succumbs to entropy, becomes less and less relevant to user needs, and becomes a burden rather than a boon to the business.

Chapter 10: DR and the Cloud

Analyze the additional problems and opportunities presented by off-premises hosting. There is still a great deal of planning involved in moving to the cloud. This chapter looks at the process by which SharePoint developed into its current form, how cloud architecture options come down to cost and control, and how multi-tenancy and planning federation are key aspects of SharePoint in the cloud.

Chapter 11: Best Practices and Worst Practices

When it comes to best and worst practices in SharePoint, there is no such thing as perfection and no implementation is all bad. But it is possible to improve and to avoid obvious pitfalls. Primarily, you have to avoid the easy path of short term results, the quagmires of weak assumptions, a reactionary approach to change, and an irresponsible approach to governance. Those four principles will get your SharePoint platform off to a good start and keep it on course.

Chapter 12: Final Conclusions

This chapter brings together the key principles contained in this book. The approach has been to create a guide that can be used in any circumstance rather than to define only one approach. Principles are more universal and can be applied to any version of SharePoint irrespective of changes in the underlying technology. Even as SharePoint transitions to the cloud, there are still lessons than can be applied from the four previous versions of SharePoint, and high availability and disaster recovery in general.

CHAPTER 1

Steering Away from Disaster

On my very first SharePoint job back in 2001, I spent hours backing up, copying and restoring the SharePoint installation from an internal domain to the one accessible to users from the Internet. This was not a backup strategy; it was a crude way to get content to the Internet while keeping the intranet secure. But it made the system very vulnerable to failure. Every time content was updated, I had to manually overwrite the production SPS 2001 with the updated staging SPS 2001 out of hours so users could see the changes the next day. This started to become a nightly occurrence. I still remember the feeling of fear every time I had to run the commands to overwrite the production farm and bring it up to date. I would stare at that cursor while it made up its mind (far too casually, I thought) to bring everything in line. I would sigh with relief when it worked and I was able to see the changes there. I still feel the sense of mild panic when it didn't work and I had to troubleshoot what went wrong. It was usually an easy fix—some step I missed—but sometimes it was a change to the network or the Exchange server where the data was stored or a Windows security issue.

Disaster was always only a click away and even back then I knew this way was not the best way to do what I was doing. It made no sense, but I did it every day anyway. The process had been signed off by management, who thought it looked secure and prudent on paper, but in reality it was inefficient and a disaster waiting to happen. Eventually, I left for a better job. Perhaps that's how they still do content deployment there.

Maybe you are in a similar situation now: you know that the processes and procedures your organization is using to protect itself are just not realistic or sustainable. They may, in fact, be about to cause the very thing they are supposed to protect against. Or perhaps the disaster has already occurred and you are now analyzing how to do things better. Either way, this book is designed to focus your thinking on what needs to be done to make your SharePoint farm as resistant to failure as possible and to help you plan what to do in the event of a failure to minimize the cost and even win praise for how well you recovered. The ideal scenario is when a disaster becomes an opportunity to succeed rather than just a domino effect of successive failures. Can you harness the dragon rather than be destroyed by it?

This chapter addresses the following topics:

- The hidden costs of IT disasters.
- Why they happen.
- Key disaster recovery concepts: recovery time objective and recovery point objective.
- Key platform concepts: networks, the cloud, IaaS, and SaaS.
- Roles and responsibilities.

- Measures of success.
- Some applied scenarios, options, and potential solutions.

The Real Cost of Failure

This book focuses on two different but related concepts: high availability (HA) and disaster recovery (DR). Together they are sometimes referred to as Service Continuity Management (SCM). While SCM focuses on the recovery of primarily IT services after a disaster, as IT systems become more crucial to the functioning of the business as a whole, many businesses also assess the impact of the system failing on the organization itself.

No matter what your core business, it is dependent on technology in some form. It may be mechanical machinery or IT systems. IT systems have become central to many kinds of businesses but the business managers and owners have not kept up with the pace of change. Here's an example of how core technology has become important for many types of companies.

Starbucks recently closed all its U.S. stores for three hours to retrain baristas in making espresso. It cost them $65 million in lost revenue. Was that crazy? They did it on purpose; they realized the company was sacrificing quality in the name of (store) quantity. They had expanded so fast that they were losing what made the Starbucks brand famous: nice coffee in a nice coffee shop. They anticipated their seeming success in the short term would kill them in the long term. They had more stores, but less people were coming in. The short term cost of closing for three hours was far less than what they would lose if they did not improve a core process in their business. Making espresso seems a small task, but it's one performed often by their most numerous staff members. If those people couldn't make a quality espresso every time, the company was doomed in the longer term. Focusing on this one process first was a step in improving business practices overall. It was a sign that Starbucks knew they need to improve, not just proliferate, in order to survive.

In this case, falling standards of skill was a seen as reason to stop production. It was planned but it underlines the cost when a business can't deliver that they produce. Your SharePoint farm produces productivity. It does this by making the user activity of sharing information more efficient. SharePoint is worthless if the information in it is lost or the sharing process is stopped. Worse than that, it could seriously damage your business's ability to function.

Perception is reality, they say. Even if only a little data or a small amount of productive time is lost, some of an organization's credibility can be lost as well. A reputation takes years to build but it can be lost in days. If increasingly valuable information of yours or your customers is lost or stolen from your SharePoint infrastructure, the cost can be very high indeed. Your reputation might never recover.

Poor perception leads to brand erosion. IT systems are now an essential part of many businesses' brand, not just hidden in a back room somewhere. For many companies, that brand depends on consumer confidence in their technology. Erosion can mean lost revenues or even legal exposure. The attack on Sony's PlayStation Network where 100 million accounts were hacked (the fourth biggest in history) will cost Sony a lot of real money. One Canadian class action suit on behalf of 1 million users is for $1 billion. What might the perceived antenna problems with iOS4 have cost Apple if they had not reacted (after some initial denial) swiftly to compensate customers?

Large companies like Starbucks, Sony, and Apple know technology is not just part of what they sell, it is core to who they are. If you neglect the core of your business, it will fail. The cost of total failure is much higher than the cost of understanding and investing in the technology that your staff relies on every day. SharePoint has become more than a useful place to put documents in order to share them with other users. It is now the repository for the daily tasks of many users. It has become the core technology platform in many businesses and it should be treated as such.

Why Disasters Happen and How to Prevent Them

In IT there is a belief that more documentation, processes, and procedures means better documentation, processes, and procedures—like the idea that more Starbucks meant Starbucks was doing better. In fact, the opposite is true. Processes around HA and DR (indeed all governance) should follow the principle that perfection is reached not when there is nothing left to add, but when there is nothing left to take away. Good practice requires constant revision and adjustment. Finally, the people who do the work should own the processes and maintain them. In too many businesses the people who define the policies and procedures are remote from the work being done and so the documents are unrealistic and prone to being ignored or causing failures.

Success/Failure

SharePoint farms are like any complex system: we can't afford to rely on the hope that haphazard actions will somehow reward us with a stable, secure collaboration platform. But the reality is most of our processes and procedures are reactive, temporary stop-gap solutions that end up being perpetuated because there's no time or resources to come up with something better. We would, in fact, be better off with "Intelligent Design" than with Evolution in this case because we are in a position to interpret small events in a way that lets us anticipate the future further ahead than nature. At the same time, near misses dangerously teach us something similar but opposite: if you keep succeeding, it will cause you to fail. So who is right and how can we apply this to the governance of our SharePoint architectures?

There is some research from Gartner that has been around for a few years that says that we put too much emphasis on making our platforms highly available only through hardware and software, when 80% of system failures are caused by human error or lack of proper change management procedures. So, what are the thought processes that lead us to ignore near-misses and think that the more success we have, the less likely we are to fail?

If we're not careful, success can lead to failure. We think that because we were lucky not to fail before, we will always be lucky. Our guard goes down and we ignore the tell-tale signs that things will eventually go wrong in a big way, given enough time.

Research shows that for every 30 near misses, there will be a minor accident, and for every 30 of those, one will be serious. SharePoint farms have monitoring software capturing logs, but they only capture what we tell them to; we have to read and interpret them. The problem is that not enough time is allocated to looking for small cracks in the system or looking into the causes of the near misses.

But a more pernicious cause of failure is the fact that when processes are weak, the people who monitor the system are continuously bailing out the poor processes. Those who have responsibility for the processes are not reviewing the processes continually to keep them up to date. The people who don't own the process are not escalating the problems; instead they are coming up with quick fixes to keep things going in the short term. Sooner or later, they will get tired or frustrated or bored or they'll leave before things really go wrong. Then it is too late to prevent the real big FUBAR.

Thus, management must not ignore the fact that staff on the ground are working at capacity and keeping things going but it will not last. Likewise, staff on the ground must step up and report situations that will lead to system failure and data loss.

Is failure necessary for success? I think that every process has to be the best it can be with the realization that it must be tested and improved continuously. This is the essence of governance: people taking ownership of change and reacting to it constructively. The constant evolution of policies is needed.

Your SharePoint Project: Will It Sink or Float?

Let's use an analogy—and it's one I will revisit throughout this book. Your SharePoint project is like the voyage of a cruise liner. Will it be that of a safe, modern vessel or the ill-fated Titanic? Your cruise ship company has invested a lot of money into building a big chunk of metal that can cross the Atlantic. Your SharePoint farm is like that ship. The farm can be on-premise, in the cloud, or a hybrid of both. You have a destination and high ambitions as to what it will achieve. You know for it to succeed you will need an able crew to administer it plus many happy paying passengers.

This analogy is assuming something inevitable. The ship will sink. Is it fair to say your SharePoint implementation will fail? Of course not, but you should still plan realistically that it could happen. Not being able to conceive of failure is bound to make you more vulnerable than if you had looked at everything that could go wrong and what should be done if it happened. This is why ships have lifeboat drills—because they help prevent disaster. Acknowledging the fact that disasters do happen is not inviting them. In fact, it does the opposite; it makes them less likely to happen as it helps reveal weaknesses in the infrastructure and leads to realistic plans to recover more quickly when disasters do happen.

Figure 1-1 is of a typical SharePoint 2010 farm. Note that more than half of the servers are redundant. The farm could still function if one web front end, one application server, and one SQL server stayed functioning. Let's return to the Titanic metaphor. It was engineered with a hull with multiple compartments; the builders said that the ship could still float if many of these were breached. In fact, ships had hit icebergs head on and survived because of this forethought in the design.

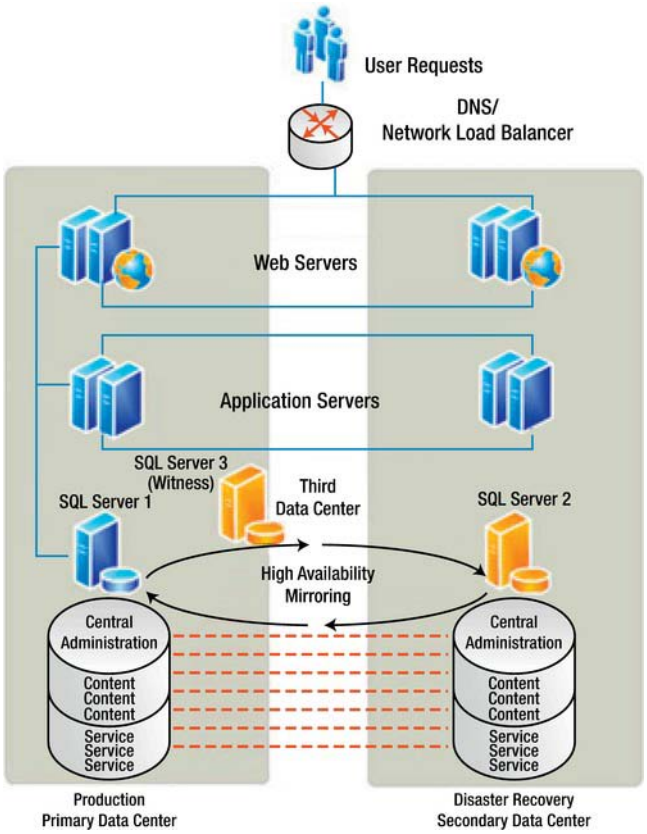

Figure 1-1. *Typical highly available on-premises SharePoint farm*

So technology convinced experts that very large ships were beyond the laws of physics. It somehow became widely believed that not only was this the biggest, most luxurious liner on the sea, but it was also virtually unsinkable. And we all know how that turned out. The story was very sensational news at the time and still is. The press today is no different from the press 100 years ago; they love big stories. The Titanic was such a compelling story because it was the world's biggest passenger ship on its maiden voyage full of the rich and the poor—a metaphor for modernity and society.

Perhaps your SharePoint deployment will be watched by the press, too, and you will want it to go well for the same reasons. Perhaps it will only be watched by internal audiences, but its success or failure will still be very visible as it involves all kinds of users in your organization. This is certainly a good argument for piloting and prototyping, but the real full-scale system still has to go live and set sail someday.

High Availability: The Watertight Compartments

High availability is the IT terminology for the efforts made to ensure your SharePoint Farm will not sink, no matter what happens to it—its resilience and quality can handle the damage and still keep afloat. Automatic systems that kick in when things go wrong are referred to as *failover systems*. In the case of my

analogy, they would be like the bulkhead doors that close to make the compartments watertight (see Figure 1-2). These could be triggered manually but would also kick in automatically if water rose to a certain level in the compartments. In SharePoint, on-premises, clustering, load balancing, and mirroring provide this failover and resilience. But they can be overwhelmed.

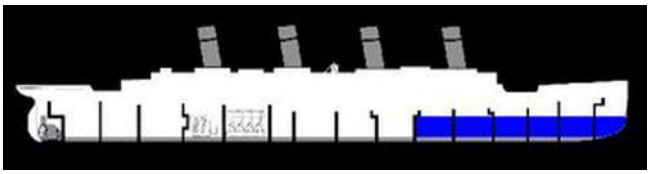

Figure 1-2. *High availability on R.M.S Titanic*

In most IT systems, it's too easy to provide the minimum or even recommended level of resilience without much active thought. In the Titanic, the 16 compartments exceeded the Board of Trade's requirements; the problem was that 16 watertight cubes in a ship are inconvenient for the crew (administrators) and passengers (business users). There were many doors between the compartments so that people could move freely through these barriers. As a result, safety was trumped by convenience. This is a common reason for the failure of high availability systems in SharePoint, too. The failure is usually in the rush to apply updates and routine improvements to the system. The more complex the high availability systems, the more moving parts there are that can fail.

In a SharePoint on-premise farm, you can achieve high availability through a number of options. A combination of the following is common:

- *SQL mirroring:* Synchronously maintaining a copy of your databases. *Synchronously* means the data is always the same at the same time.
- *SQL clustering:* Spreading a SQL instance over multiple machines. An *instance* is a group of servers that appears as one SQL server.
- *SQL log shipping:* Backing up to file the data and restoring to another SQL instance asynchronously. *Asynchronously* means the data is not exactly the same at the same time. There is a delay of hours in moving the logs from one instance to the other.
- *Multiple data centers (DCs):* This means locating your server farms in independent premises in different geographical locations. For example, Office 365 for EMEA is in Dublin, but there is also another DC in Amsterdam.
- *Load balancing:* Software or hardware, more than one server seems to have the same IP address as they have virtual IP addresses.
- *Stretched farm:* Hosting some servers in your farm in different data centers.
- *SAN replication:* Synchronously maintaining a copy of your data.
- *Redundant disaster recovery farm:* A second farm in another location ready to take the place of the production farm.
- *Availability zones and regions:* Used in Amazon Web Services, these are analogous to servers and data centers.

Disaster Recovery

Disaster recovery is what to do when something has already gone wrong. With a SharePoint Farm, it's the point when users start to lose access, performance, or data. It can also be when security is compromised. Basically, it's when the integrity of the system is compromised. You've hit the iceberg. With the Titanic, the disaster recovery process was the lifeboat drill and the lifeboats themselves. With a SharePoint farm, it's the processes, policies, and procedures related to preparing for and undergoing a recovery from a disaster. Thus, it is the planning that goes into what to do from the point the problem is detected. Note that it may not be the exact time the problem started to occur—only when it is detected. Error detection and reporting will examined in further detail in a later chapter.

On the Titanic there were not enough lifeboats because it was believed that the ship was unsinkable due to its watertight compartments. Also, it was believed that it would take the crew too long to load all the lifeboats in the event of a sinking (the Titanic had a capacity of over 3,500 souls, although there were only about 2,500 on board when it sunk). Finally, the regulations were out of date at the time; the ship was legally compliant, but in actuality had less than half the capacity needed, even if the lifeboats had been full. Relying too much on documentation and the recommended approach is not always enough.

Recovery Time Objective and Recovery Point Objective

Two metrics commonly used in SCM to evaluate disaster recovery solutions are recovery time objective (RTO), which measures the time between a system disaster and the time when the system is again operational, and recovery point objective (RPO), which measures the time between the latest backup and the system disaster, representing the nearest historical point in time to which a system can recover. These will be set in the Service Level Agreement (SLA), which is the legal document the provider has to follow. For example, SharePoint Online as part of Office 365 has set an RPO and RTO in the event of a disaster as the following:

> *"12-hour RPO: Microsoft protects an organization's SharePoint Online data and has a copy of that data that is equal to or less than 12 hours old.*
>
> *24-hour RTO: Organizations will be able to resume service within 24 hours after service disruption if a disaster incapacitates the primary data center."*

Networks and the Cloud

Think of your network or the cloud as the ocean. It's big, unpredictable, and full of dangerous things, most of which the administrator can't control. There are denial-of-service attacks, human error, hardware failures, acts of God, and all manner of things that can happen to compromise your system. Later I will describe the kinds of events that can compromise the integrity of your system and how to mitigate them.

IaaS vs. SaaS

Infrastructure as a Service (IaaS) and Software as a Service (SaaS) emphasize high availability over disaster recovery. Naturally, it makes more sense to keep the system working rather than recover from it failing. With IaaS, high availability is more in the hands of the tenant. With SaaS, like SharePoint Online in Office 365, you are more reliant on the provider to keep the system working. My analogy is that IaaS is

like being a crew member; you have training and responsibility to keep the passengers safe. With SaaS, you are more like a passenger, reliant on the provider to keep you safe.

For example, in the case of an IaaS provider like Amazon Web Services (AWS), there is the ability of the tenant to place instances in multiple locations. These locations are composed of regions and availability zones. Availability zones are distinct locations that are engineered to be insulated from failures in other availability zones and provide inexpensive, low latency network connectivity to other availability zones in the same region. Think of these as your watertight compartments.

By launching instances (in your case, your SharePoint servers) in separate availability zones, you can protect your applications from the failure of one single location. There are also regions. These consist of one or more availability zones, are geographically dispersed, and are in separate geographic areas or countries. By spreading your instances across these, you have greater resilience.

With SaaS examples like Office 365, if there is a problem with the platform, you have less control over reacting to that problem. Think of this as a passenger bringing his or her own lifejacket. I will go into more detail later on how to have more control.

SharePoint in the Cloud

The IT world is shifting to where computing, networking, and storage resources are migrating onto the Internet from local networks. SharePoint is a good candidate for cloud computing because it is already web-based. From a setup and administration point of view, it has a growing complex service architecture. Also, many companies would gladly do without the cost of having the skills in house to administer it, not to mention the opportunity to move Exchange to the cloud. This will not happen all at once, but it does mean that hosting your SharePoint farms on-premises is no longer the only option. For that reason I will outline the new cloud options for those unfamiliar with them.

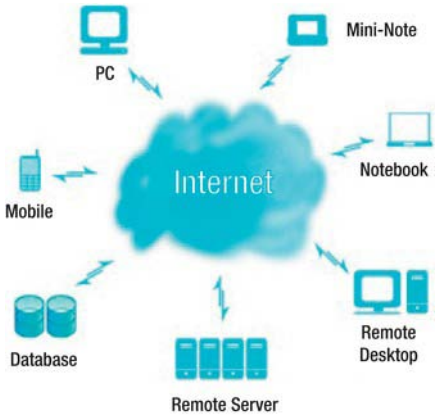

Figure 1-3. *The cloud was a metaphor for the Internet.*

Once upon a time a picture of a cloud was used on network diagrams to denote the Internet (see Figure 1-3). This is why we use the term *the cloud* now. It had a "here be dragons" feel about it. (Prior to the Europeans discovering big chunks of the world, large areas on maps were labeled "here be dragons," as shown in Figure 1-4. It was a way to fill an empty space that could not be understood. With this lack of knowledge comes fear; hence pictures of dragons.) In the context of this metaphor, the dragon is complacency—a false bravado born of fear. The cloud is full of positive benefits for businesses. It will soon be seen as a New World to be discovered and explored, not an unknown danger.

Figure 1-4. *Dragons were a metaphor for the uncharted parts of the map.*

By moving your SharePoint infrastructure or software into the cloud, there is a danger that too much trust is placed in the platform provider to automatically take care of all the high availability and disaster recovery options. They do, in most cases, provide excellent tools to manage your infrastructure, but you must still know how to use them. The truth is the final responsibility still rests with the owner of the data to understand the options and choose the best ones for their needs and budget.

Instead of some nice, healthy fear, there is dangerous complacency that comes from a reluctance to take control of the infrastructure. It is easier just to assume someone else it taking care of it. I take it, dear reader, that you bought this book because you don't want to get swallowed up by the great chewing complacency.

Why Is Infrastructure Moving to the Cloud?

We live in a more connected world. Wi-Fi, Smartphones, tablets, notebooks, and laptops allow workers to be more mobile and connection options more plentiful. People can access so much and communicate so easily through the Internet they now expect to be able to access their work data from any location with any device with the same ease.

Another major factor in the arrival of the cloud for businesses is technologies like virtualization and cheap hardware that allow for the commoditization of resources to the point that they are like any other utility, such as power, water, or gas. SharePoint 2010 needs a lot of hardware and capacity. The standard build is three farms: Development, Testing, and Production. SharePoint also requires a lot of software and licenses if you want in each farm, for example, three web front-end servers, two application servers, and a SQL cluster.

SharePoint Online (SPO) makes paying for access much simpler. There is no need for a large upfront investment in hardware, software, and licenses. Organizations can just sign on and pay monthly per user. They can even invite users from outside their network; this just requires a LiveID account like Hotmail or an existing Office365 account. This makes collaborating beyond your network with partners or customers so much simpler. This also makes starting small and adding users gradually much easier—and the costs of user licenses up front much lower. It is much easier to remove user licenses, too, because each user has to re-authenticate once every 30 days; thus, once the 30-day license has expired, you no longer have to pay if you don't want to. There's no requirement to buy and configure a number of servers and work out what server and software licenses you will need. This has always been an overly complex and arcane art and any simplification here is very welcome. It is true there are still a range of user licenses to choose from, but the options are clearer and it's easier to identify what you want.

Licenses are also priced differently. They are now per user and not per device with SharePoint Online. The Client Access Licenses (CALs) for SharePoint 2010 are per device, so if you access from home, office, and mobile, you need three licenses, in theory, which is not something most organizations plan for. With SPO, a user can connect with up to five devices but it counts as only one device—a more