This paper has been archived.

Using Amazon Web Services for Disaster Recovery

October 2014

Glen Robinson, Attila Narin, and Chris Elleman

For the latest information on disaster recovery, see https://aws.amazon.com/disaster-recovery/

Contents

Abstract

In the event of a disaster, you can quickly launch resources in Amazon Web Services (AWS) to ensure business continuity. This whitepaper highlights AWS services and features that you can leverage for your disaster recovery (DR) processes to significantly minimize the impact on your data, your system, and your overall business operations. The whitepaper also includes scenarios that show you, step-by-step, how to improve your DR plan and leverage the full potential of the AWS cloud for disaster recovery.

Introduction

Disaster recovery (DR) is about preparing for and recovering from a disaster. Any event that has a negative impact on a company's business continuity or finances could be termed a disaster. This includes hardware or software failure, a network outage, a power outage, physical damage to a building like fire or flooding, human error, or some other significant event.

To minimize the impact of a disaster, companies invest time and resources to plan and prepare, to train employees, and to document and update processes. The amount of investment for DR planning for a particular system can vary dramatically depending on the cost of a potential outage. Companies that have traditional physical environments typically must duplicate their infrastructure to ensure the availability of spare capacity in the event of a disaster. The infrastructure needs to be procured, installed, and maintained so that it is ready to support the anticipated capacity requirements. During normal operations, the infrastructure typically is under-utilized or over-provisioned.

With Amazon Web Services (AWS), your company can scale up its infrastructure on an as-needed, pay-as-you-go basis. You get access to the same highly secure, reliable, and fast infrastructure that Amazon uses to run its own global network of websites. AWS also gives you the flexibility to quickly change and optimize resources during a DR event, which can result in significant cost savings.

This whitepaper outlines best practices to improve your DR processes, from minimal investments to full-scale availability and fault tolerance, and shows you how you can use AWS services to reduce cost and ensure business continuity during a DR event.

Recovery Time Objective and Recovery Point Objective

This whitepaper uses two common industry terms for disaster planning:

Recovery time objective (RTO)[1] — The time it takes after a disruption to restore a business process to its service level, as defined by the operational level agreement (OLA). For example, if a disaster occurs at 12:00 PM (noon) and the RTO is eight hours, the DR process should restore the business process to the acceptable service level by 8:00 PM.

Recovery point objective (RPO)[2] — The acceptable amount of data loss measured in time. For example, if a disaster occurs at 12:00 PM (noon) and the RPO is one hour, the system should recover all data that was in the system before 11:00 AM. Data loss will span only one hour, between 11:00 AM and 12:00 PM (noon).

A company typically decides on an acceptable RTO and RPO based on the financial impact to the business when systems are unavailable. The company determines financial impact by considering many factors, such as the loss of business and damage to its reputation due to downtime and the lack of systems availability.

IT organizations then plan solutions to provide cost-effective system recovery based on the RPO within the timeline and the service level established by the RTO.

Traditional DR Investment Practices

A traditional approach to DR involves different levels of off-site duplication of data and infrastructure. Critical business services are set up and maintained on this infrastructure and tested at regular intervals. The disaster recovery environment's location and the source infrastructure should be a significant physical distance apart to ensure that the disaster recovery environment is isolated from faults that could impact the source site.

At a minimum, the infrastructure that is required to support the duplicate environment should include the following:

- Facilities to house the infrastructure, including power and cooling.
- Security to ensure the physical protection of assets.
- Suitable capacity to scale the environment.
- Support for repairing, replacing, and refreshing the infrastructure.
- Contractual agreements with an Internet service provider (ISP) to provide Internet connectivity that can sustain bandwidth utilization for the environment under a full load.
- Network infrastructure such as firewalls, routers, switches, and load balancers.
- Enough server capacity to run all mission-critical services, including storage appliances for the supporting data, and servers to run applications and backend services such as user authentication, Domain Name System (DNS), Dynamic Host Configuration Protocol (DHCP), monitoring, and alerting.

[1] From http://en.wikipedia.org/wiki/Recovery_time_objective
[2] From http://en.wikipedia.org/wiki/Recovery_point_objective

AWS Services and Features Essential for Disaster Recovery

Before we discuss the various approaches to DR, it is important to review the AWS services and features that are the most relevant to disaster recovery. This section provides a summary.

In the preparation phase of DR, it is important to consider the use of services and features that support data migration and durable storage, because they enable you to restore backed-up, critical data to AWS when disaster strikes. For some of the scenarios that involve either a scaled-down or a fully scaled deployment of your system in AWS, compute resources will be required as well.

When reacting to a disaster, it is essential to either quickly commission compute resources to run your system in AWS or to orchestrate the failover to already running resources in AWS. The essential infrastructure pieces include DNS, networking features, and various Amazon Elastic Compute Cloud (Amazon EC2) features described later in this section.

Regions

Amazon Web Services are available in multiple regions around the globe, so you can choose the most appropriate location for your DR site, in addition to the site where your system is fully deployed. AWS has multiple general purpose regions in the Americas, EMEA, and Asia Pacific that anyone with an AWS account can access. Special-use regions are also available for government agencies and for China. See the full list of available regions here.

Storage

Amazon Simple Storage Service (Amazon S3) provides a highly durable storage infrastructure designed for mission-critical and primary data storage. Objects are redundantly stored on multiple devices across multiple facilities within a region, designed to provide a durability of 99.999999999% (11 9s). AWS provides further protection for data retention and archiving through versioning in Amazon S3, AWS multi-factor authentication (AWS MFA), bucket policies, and AWS Identity and Access Management (IAM).

Amazon Glacier provides extremely low-cost storage for data archiving and backup. Objects (or archives, as they are known in Amazon Glacier) are optimized for infrequent access, for which retrieval times of several hours are adequate. Amazon Glacier is designed for the same durability as Amazon S3.

Amazon Elastic Block Store (Amazon EBS) provides the ability to create point-in-time snapshots of data volumes. You can use the snapshots as the starting point for new Amazon EBS volumes, and you can protect your data for long-term durability because snapshots are stored within Amazon S3. After a volume is created, you can attach it to a running Amazon EC2 instance. Amazon EBS volumes provide off-instance storage that persists independently from the life of an instance and is replicated across multiple servers in an Availability Zone to prevent the loss of data from the failure of any single component.

AWS Import/Export accelerates moving large amounts of data into and out of AWS by using portable storage devices for transport. AWS Import/Export bypasses the Internet and transfers your data directly onto and off of storage devices by means of the high-speed internal network of Amazon. For data sets of significant size, AWS Import/Export is often faster than Internet transfer and more cost effective than upgrading your connectivity. You can use AWS Import/Export to migrate data into and out of Amazon S3 buckets and Amazon Glacier vaults or into Amazon EBS snapshots.

AWS Storage Gateway is a service that connects an on-premises software appliance with cloud-based storage to provide seamless and highly secure integration between your on-premises IT environment and the storage infrastructure of AWS.

AWS Storage Gateway supports three different configurations:

Gateway-cached volumes — You can store your primary data in Amazon S3 and retain your frequently accessed data locally. Gateway-cached volumes provide substantial cost savings on primary storage, minimize the need to scale your storage on-premises, and retain low-latency access to your frequently accessed data.

Gateway-stored volumes — In the event that you need low-latency access to your entire data set, you can configure your gateway to store your primary data locally, and asynchronously back up point-in-time snapshots of this data to Amazon S3. Gateway-stored volumes provide durable and inexpensive off-site backups that you can recover locally or from Amazon EC2 if, for example, you need replacement capacity for disaster recovery.

Gateway-virtual tape library (gateway-VTL) — With gateway-VTL, you can have an almost limitless collection of virtual tapes. You can store each virtual tape in a virtual tape library (VTL) backed by Amazon S3 or a virtual tape shelf (VTS) backed by Amazon Glacier. The virtual tape library exposes an industry standard iSCSI interface that provides your backup application with on-line access to the virtual tapes. When you no longer require immediate or frequent access to data contained on a virtual tape, you can use your backup application to move it from its VTL to your VTS to further reduce your storage costs.

Compute

Amazon Elastic Compute Cloud (Amazon EC2) provides resizable compute capacity in the cloud. Within minutes, you can create Amazon EC2 instances, which are virtual machines over which you have complete control. In the context of DR, the ability to rapidly create virtual machines that you can control is critical. To describe every feature of Amazon EC2 is outside the scope of this document; instead; we focus on the aspects of Amazon EC2 that are most relevant to DR.

Amazon Machine Images (AMIs) are preconfigured with operating systems, and some preconfigured AMIs might also include application stacks. You can also configure your own AMIs. In the context of DR, we strongly recommend that you configure and identify your own AMIs so that they can launch as part of your recovery procedure. Such AMIs should be preconfigured with your operating system of choice plus appropriate pieces of the application stack.

Availability Zones are distinct locations that are engineered to be insulated from failures in other Availability Zones. They also provide inexpensive, low-latency network connectivity to other Availability Zones in the same region. By launching instances in separate Availability Zones, you can protect your applications from the failure of a single location. Regions consist of one or more Availability Zones.

The Amazon EC2 VM Import Connector virtual appliance enables you to import virtual machine images from your existing environment to Amazon EC2 instances.

Networking

When you are dealing with a disaster, it's very likely that you will have to modify network settings as your system is failing over to another site. AWS offers several services and features that enable you to manage and modify network settings.

Amazon Route 53 is a highly available and scalable Domain Name System (DNS) web service. It gives developers and businesses a reliable, cost-effective way to route users to Internet applications. Amazon Route 53 includes a number of global load-balancing capabilities (which can be effective when you are dealing with DR scenarios such as DNS endpoint health checks) and the ability to failover between multiple endpoints and even static websites hosted in Amazon S3.

Elastic IP addresses are static IP addresses designed for dynamic cloud computing. However, unlike traditional static IP addresses, Elastic IP addresses enable you to mask instance or Availability Zone failures by programmatically remapping

your public IP addresses to instances in your account in a particular region. For DR, you can also pre-allocate some IP addresses for the most critical systems so that their IP addresses are already known before disaster strikes. This can simplify the execution of the DR plan.

Elastic Load Balancing automatically distributes incoming application traffic across multiple Amazon EC2 instances. It enables you to achieve even greater fault tolerance in your applications by seamlessly providing the load-balancing capacity that is needed in response to incoming application traffic. Just as you can pre-allocate Elastic IP addresses, you can pre-allocate your load balancer so that its DNS name is already known, which can simplify the execution of your DR plan.

Amazon Virtual Private Cloud (Amazon VPC) lets you provision a private, isolated section of the AWS cloud where you can launch AWS resources in a virtual network that you define. You have complete control over your virtual networking environment, including selection of your own IP address range, creation of subnets, and configuration of route tables and network gateways. This enables you to create a VPN connection between your corporate data center and your VPC, and leverage the AWS cloud as an extension of your corporate data center. In the context of DR, you can use Amazon VPC to extend your existing network topology to the cloud; this can be especially appropriate when recovering enterprise applications that are typically on the internal network.

Amazon Direct Connect makes it easy to set up a dedicated network connection from your premises to AWS. In many cases, this can reduce your network costs, increase bandwidth throughput, and provide a more consistent network experience than Internet-based connections.

Databases

For your database needs, consider using these AWS services:

Amazon Relational Database Service (Amazon RDS) makes it easy to set up, operate, and scale a relational database in the cloud. You can use Amazon RDS either in the preparation phase for DR to hold your critical data in a database that is already running, or in the recovery phase to run your production database. When you want to look at multiple regions, Amazon RDS gives you the ability to snapshot data from one region to another, and also to have a read replica running in another region.

Amazon DynamoDB is a fast, fully managed NoSQL database service that makes it simple and cost-effective to store and retrieve any amount of data and serve any level of request traffic. It has reliable throughput and single-digit, millisecond latency. You can also use it in the preparation phase to copy data to DynamoDB in another region or to Amazon S3. During the recovery phase of DR, you can scale up seamlessly in a matter of minutes with a single click or API call.

Amazon Redshift is a fast, fully managed, petabyte-scale data warehouse service that makes it simple and cost-effective to efficiently analyze all your data using your existing business intelligence tools. You can use Amazon Redshift in the preparation phase to snapshot your data warehouse to be durably stored in Amazon S3 within the same region or copied to another region. During the recovery phase of DR, you can quickly restore your data warehouse into the same region or within another AWS region.

You can also install and run your choice of database software on Amazon EC2, and you can choose from a variety of leading database systems.

For more information about database options on AWS, see Running Databases on AWS.

Deployment orchestration

Deployment automation and post-startup software installation/configuration processes and tools can be used in Amazon EC2. We highly recommend investments in this area. This can be very helpful in the recovery phase, enabling you to create the required set of resources in an automated way.

AWS CloudFormation gives developers and systems administrators an easy way to create a collection of related AWS resources and provision them in an orderly and predictable fashion. You can create templates for your environments and deploy associated collections of resources (called a stack) as needed.

AWS Elastic Beanstalk is an easy-to-use service for deploying and scaling web applications and services developed with Java, .NET, PHP, Node.js, Python, Ruby, and Docker. You can deploy your application code, and AWS Elastic Beanstalk will provision the operating environment for your applications.

AWS OpsWorks is an application management service that makes it easy to deploy and operate applications of all types and sizes. You can define your environment as a series of layers, and configure each layer as a tier of your application. AWS OpsWorks has automatic host replacement, so in the event of an instance failure it will be automatically replaced. You can use AWS OpsWorks in the preparation phase to template your environment, and you can combine it with AWS CloudFormation in the recovery phase. You can quickly provision a new stack from the stored configuration that supports the defined RTO.

Security and compliance

There are many security-related features across the AWS services. We recommend that you review the Security Best Practices whitepaper. AWS also provides further risk and compliance information in the AWS Security Center. A full discussion of security is out of scope for this paper.

Example Disaster Recovery Scenarios with AWS

This section outlines four DR scenarios that highlight the use of AWS and compare AWS with traditional DR methods. The following figure shows a spectrum for the four scenarios, arranged by how quickly a system can be available to users after a DR event.

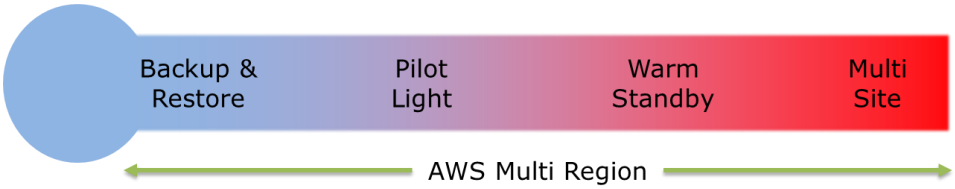

Figure 1: Spectrum of Disaster Recovery Options

AWS enables you to cost-effectively operate each of these DR strategies. It's important to note that these are just examples of possible approaches, and variations and combinations of these are possible. If your application is already running on AWS, then multiple regions can be employed and the same DR strategies will still apply.

Backup and Restore

In most traditional environments, data is backed up to tape and sent off-site regularly. If you use this method, it can take a long time to restore your system in the event of a disruption or disaster. Amazon S3 is an ideal destination for backup data that might be needed quickly to perform a restore. Transferring data to and from Amazon S3 is typically done through the network, and is therefore accessible from any location. There are many commercial and open-source backup solutions that integrate with Amazon S3. You can use AWS Import/Export to transfer very large data sets by shipping storage devices directly to AWS. For longer-term data storage where retrieval times of several hours are adequate, there is Amazon Glacier, which has the same durability model as Amazon S3. Amazon Glacier is a low-cost alternative starting from $0.01/GB per month. Amazon Glacier and Amazon S3 can be used in conjunction to produce a tiered backup solution.

AWS Storage Gateway enables snapshots of your on-premises data volumes to be transparently copied into Amazon S3 for backup. You can subsequently create local volumes or Amazon EBS volumes from these snapshots.

Storage-cached volumes allow you to store your primary data in Amazon S3, but keep your frequently accessed data local for low-latency access. As with AWS Storage Gateway, you can snapshot the data volumes to give highly durable backup. In the event of DR, you can restore the cache volumes either to a second site running a storage cache gateway or to Amazon EC2.

You can use the gateway-VTL configuration of AWS Storage Gateway as a backup target for your existing backup management software. This can be used as a replacement for traditional magnetic tape backup.

For systems running on AWS, you also can back up into Amazon S3. Snapshots of Amazon EBS volumes, Amazon RDS databases, and Amazon Redshift data warehouses can be stored in Amazon S3. Alternatively, you can copy files directly into Amazon S3, or you can choose to create backup files and copy those to Amazon S3. There are many backup solutions that store data directly in Amazon S3, and these can be used from Amazon EC2 systems as well.

The following figure shows data backup options to Amazon S3, from either on-site infrastructure or from AWS.

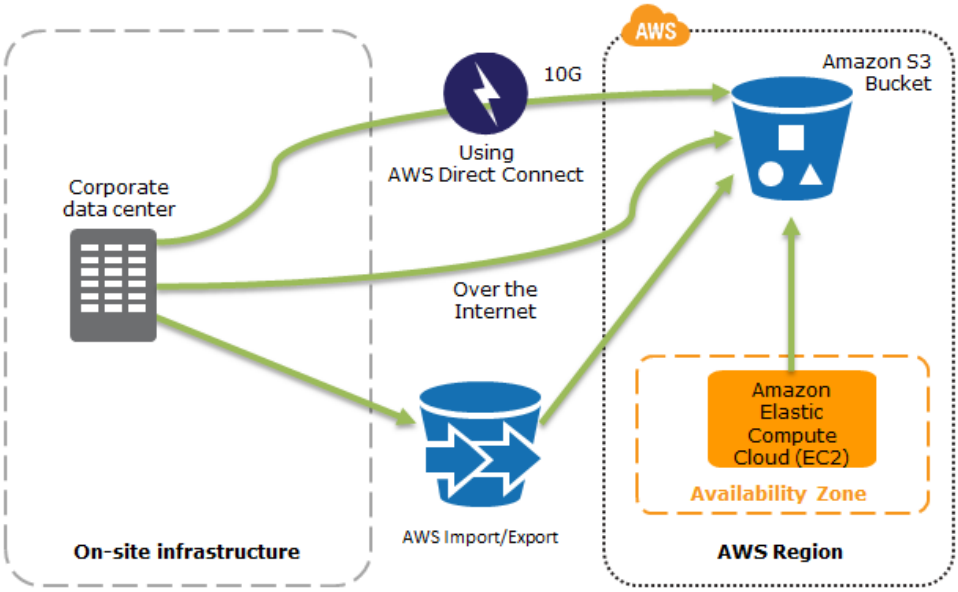

Figure 2: Data Backup Options to Amazon S3 from On-Site Infrastructure or from AWS.

Of course, the backup of your data is only half of the story. If disaster strikes, you'll need to recover your data quickly and reliably. You should ensure that your systems are configured to retain and secure your data, and you should test your data recovery processes.

The following diagram shows how you can quickly restore a system from Amazon S3 backups to Amazon EC2.

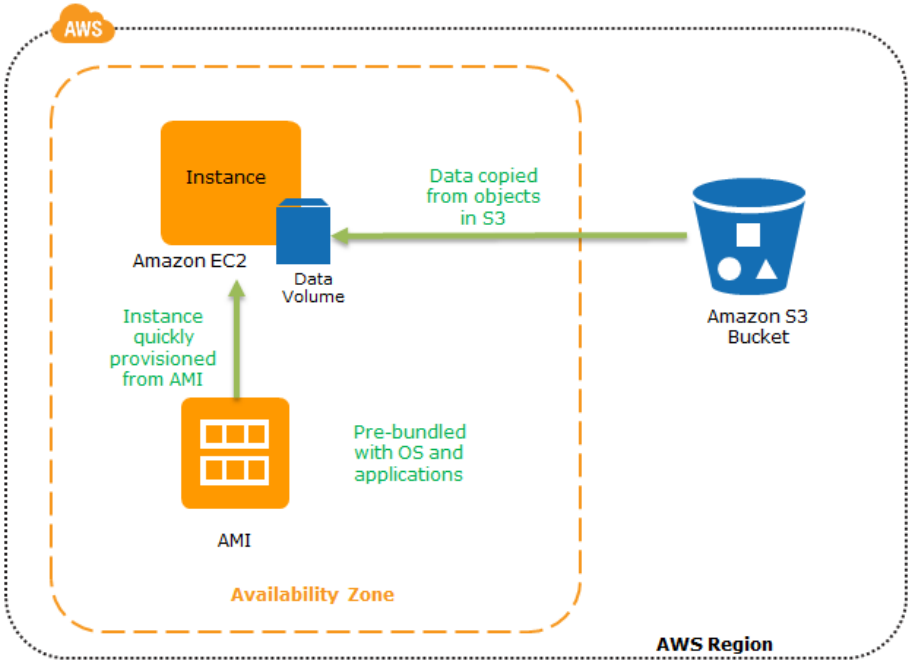

Figure 3: Restoring a System from Amazon S3 Backups to Amazon EC2

Key steps for backup and restore:

1. Select an appropriate tool or method to back up your data into AWS.
2. Ensure that you have an appropriate retention policy for this data.
3. Ensure that appropriate security measures are in place for this data, including encryption and access policies.
4. Regularly test the recovery of this data and the restoration of your system.

Pilot Light for Quick Recovery into AWS

The term *pilot light* is often used to describe a DR scenario in which a minimal version of an environment is always running in the cloud. The idea of the pilot light is an analogy that comes from the gas heater. In a gas heater, a small flame that's always on can quickly ignite the entire furnace to heat up a house.

This scenario is similar to a backup-and-restore scenario. For example, with AWS you can maintain a pilot light by configuring and running the most critical core elements of your system in AWS. When the time comes for recovery, you can rapidly provision a full-scale production environment around the critical core.

Infrastructure elements for the pilot light itself typically include your database servers, which would replicate data to Amazon EC2 or Amazon RDS. Depending on the system, there might be other critical data outside of the database that needs to be replicated to AWS. This is the critical core of the system (the pilot light) around which all other infrastructure pieces in AWS (the rest of the furnace) can quickly be provisioned to restore the complete system.

To provision the remainder of the infrastructure to restore business-critical services, you would typically have some pre-configured servers bundled as Amazon Machine Images (AMIs), which are ready to be started up at a moment's notice. When starting recovery, instances from these AMIs come up quickly with their pre-defined role (for example, Web or App Server) within the deployment around the pilot light. From a networking point of view, you have two main options for provisioning:

- Use Elastic IP addresses, which can be pre-allocated and identified in the preparation phase for DR, and associate them with your instances. Note that for MAC address-based software licensing, you can use elastic network interfaces (ENIs), which have a MAC address that can also be pre-allocated to provision licenses against. You can associate these with your instances, just as you would with Elastic IP addresses.
- Use Elastic Load Balancing (ELB) to distribute traffic to multiple instances. You would then update your DNS records to point at your Amazon EC2 instance or point to your load balancer using a CNAME. We recommend this option for traditional web-based applications.

For less critical systems, you can ensure that you have any installation packages and configuration information available in AWS, for example, in the form of an Amazon EBS snapshot. This will speed up the application server setup, because you can quickly create multiple volumes in multiple Availability Zones to attach to Amazon EC2 instances. You can then install and configure accordingly, for example, by using the backup-and-restore method.

The pilot light method gives you a quicker recovery time than the backup-and-restore method because the core pieces of the system are already running and are continually kept up to date. AWS enables you to automate the provisioning and configuration of the infrastructure resources, which can be a significant benefit to save time and help protect against human errors. However, you will still need to perform some installation and configuration tasks to recover the applications fully.

Preparation phase

The following figure shows the preparation phase, in which you need to have your regularly changi ng data replicated to the pilot light, the small core around which the full environment will be started in the recovery phase. Your less frequently updated data, such as operating systems and applications, can be periodically updated and stored as AMIs.

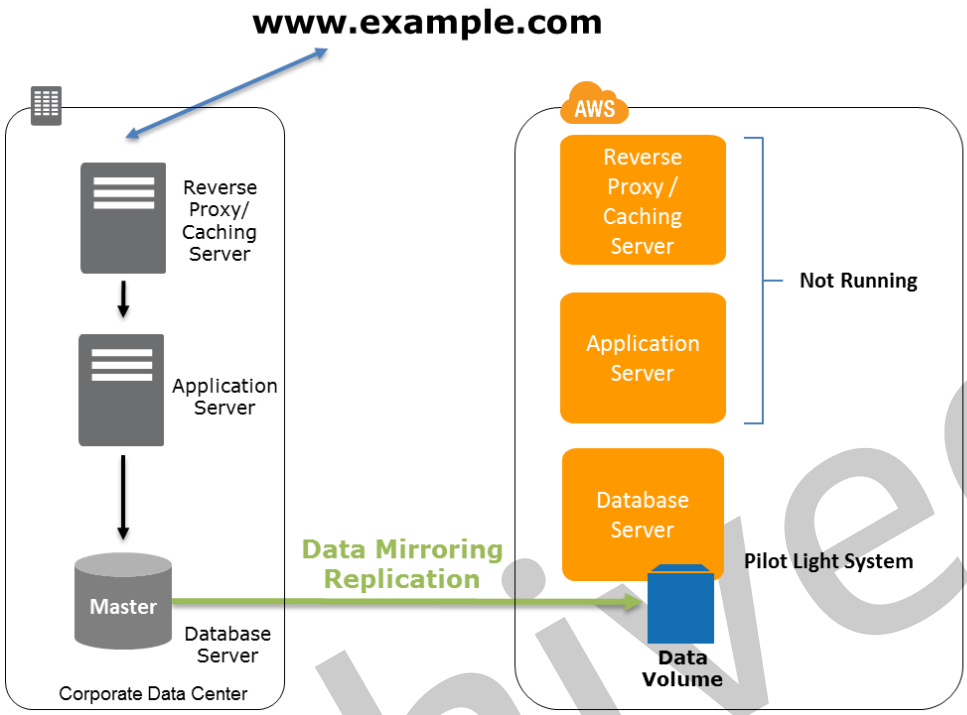

Figure 4: The Preparation Phase of the Pilot Light Scenario

Key steps for preparation:

1. Set up Amazon EC2 instances to replicate or mirror data.
2. Ensure that you have all supporting custom software packages available in AWS.
3. Create and maintain AMIs of key servers where fast recovery is required.
4. Regularly run these servers, test them, and apply any software updates and configuration changes.
5. Consider automating the provisioning of AWS resources.

Recovery phase

To recover the remainder of the environment around the pilot light, you can start your systems from the AMIs within minutes on the appropriate instance types. For your dynamic data servers, you can resize them to handle production volumes as needed or add capacity accordingly. Horizontal scaling often is the most cost-effective and scalable approach to add capacity to a system. For example, you can add more web servers at peak times. However, you can also choose larger Amazon EC2 instance types, and thus scale vertically for applications such as relational databases. From a networking perspective, any required DNS updates can be done in parallel.

After recovery, you should ensure that redundancy is restored as quickly as possible. A failure of your DR environment shortly after your production environment fails is unlikely, but you should be aware of this risk. Continue to take regular backups of your system, and consider additional redundancy at the data layer.

The following figure shows the recovery phase of the pilot light scenario.

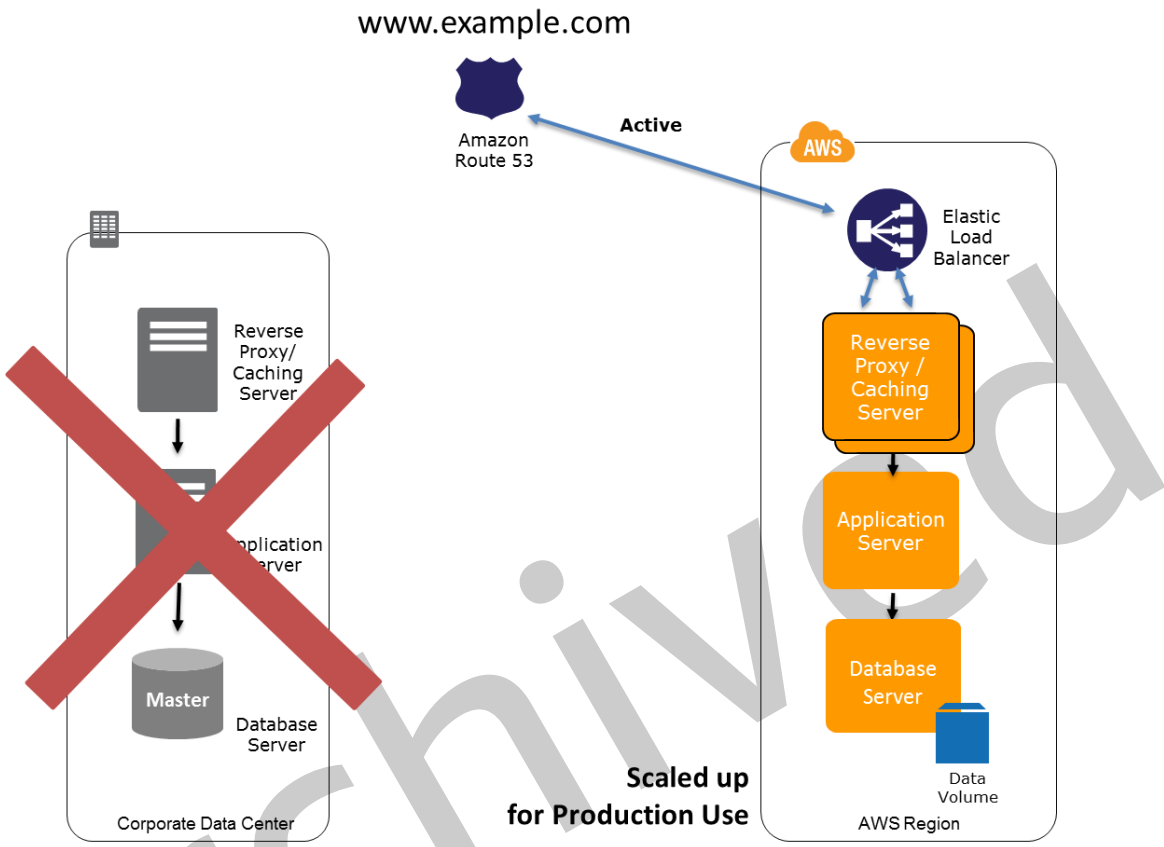

Figure 5: The Recovery Phase of the Pilot Light Scenario.

Key steps for recovery:

1. Start your application Amazon EC2 instances from your custom AMIs.
2. Resize existing database/data store instances to process the increased traffic.
3. Add additional database/data store instances to give the DR site resilience in the data tier; if you are using Amazon RDS, turn on Multi-AZ to improve resilience.
4. Change DNS to point at the Amazon EC2 servers.
5. Install and configure any non-AMI based systems, ideally in an automated way.

Warm Standby Solution in AWS

The term *warm standby* is used to describe a DR scenario in which a scaled-down version of a fully functional environment is always running in the cloud. A warm standby solution extends the pilot light elements and preparation. It further decreases the recovery time because some services are always running. By identifying your business-critical systems, you can fully duplicate these systems on AWS and have them always on.

These servers can be running on a minimum-sized fleet of Amazon EC2 instances on the smallest sizes possible. This solution is not scaled to take a full-production load, but it is fully functional. It can be used for non-production work, such as testing, quality assurance, and internal use.

In a disaster, the system is scaled up quickly to handle the production load. In AWS, this can be done by adding more instances to the load balancer and by resizing the small capacity servers to run on larger Amazon EC2 instance types. As stated in the preceding section, horizontal scaling is preferred over vertical scaling.

Preparation phase

The following figure shows the preparation phase for a warm standby solution, in which an on-site solution and an AWS solution run side-by-side.

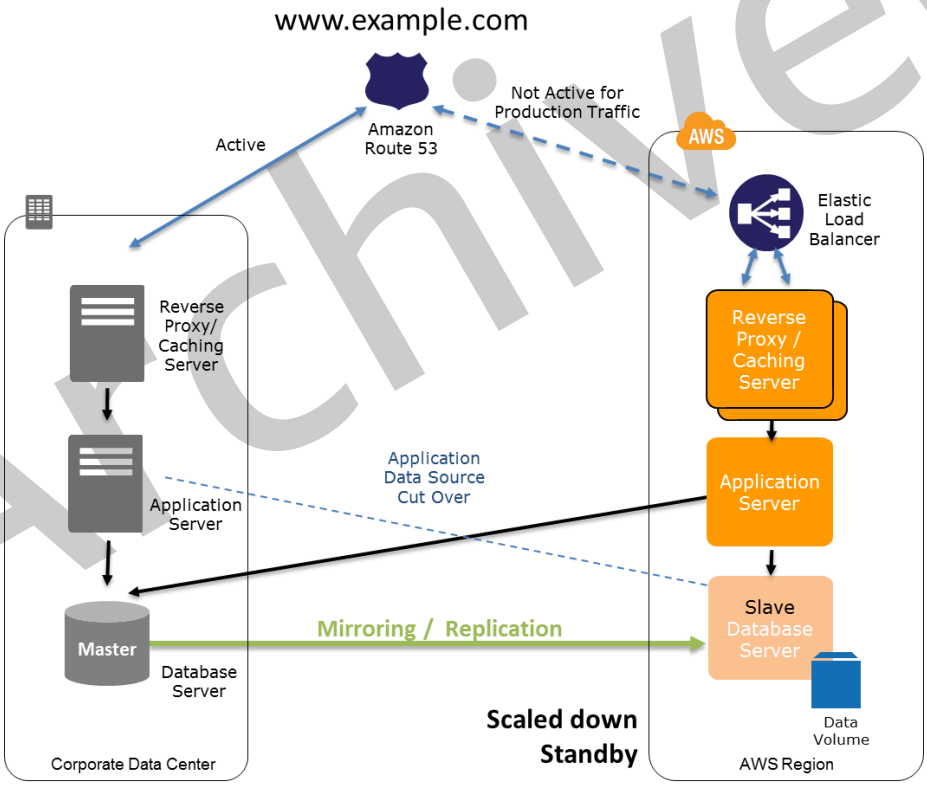

Figure 6: The Preparation Phase of the Warm Standby Scenario.

Key steps for preparation:

1. Set up Amazon EC2 instances to replicate or mirror data.
2. Create and maintain AMIs.
3. Run your application using a minimal footprint of Amazon EC2 instances or AWS infrastructure.
4. Patch and update software and configuration files in line with your live environment.

Recovery phase

In the case of failure of the production system, the standby environment will be scaled up for production load , and DNS records will be changed to route all traffic to AWS.

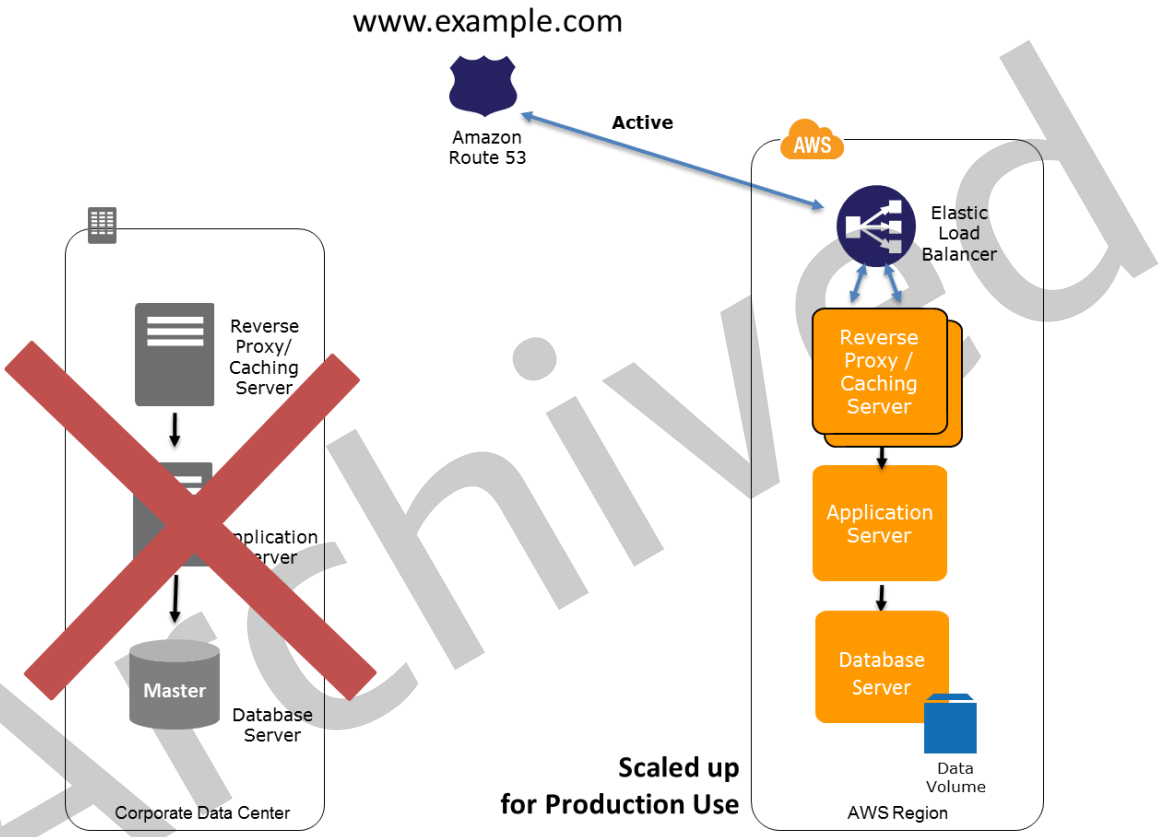

Figure 7: The Recovery Phase of the Warm Standby Scenario.

Key steps for recovery:

1. Increase the size of the Amazon EC2 fleets in service with the load balancer (horizontal scaling).
2. Start applications on larger Amazon EC2 instance types as needed (vertical scaling).
3. Either manually change the DNS records, or use Amazon Route 53 automated health checks so that all traffic is routed to the AWS environment.
4. Consider using Auto Scaling to right-size the fleet or accommodate the increased load.
5. Add resilience or scale up your database.

Multi-Site Solution Deployed on AWS and On-Site

A multi-site solution runs in AWS as well as on your existing on-site infrastructure, in an active-active configuration. The data replication method that you employ will be determined by the recovery point that you choose. For more information about recovery point options, see the Recovery Time Objective and Recovery Point Objective section in this whitepaper.

In addition to recovery point options, there are various replication methods, such as synchronous and asynchronous methods. For more information, see the Replication of Data section in this whitepaper.

You can use a DNS service that supports weighted routing, such as Amazon Route 53, to route production traffic to different sites that deliver the same application or service. A proportion of traffic will go to your infrastructure in AWS, and the remainder will go to your on-site infrastructure.

In an on-site disaster situation, you can adjust the DNS weighting and send all traffic to the AWS servers. The capacity of the AWS service can be rapidly increased to handle the full production load. You can use Amazon EC2 Auto Scaling to automate this process. You might need some application logic to detect the failure of the primary database services and cut over to the parallel database services running in AWS.

The cost of this scenario is determined by how much production traffic is handled by AWS during normal operation. In the recovery phase, you pay only for what you use for the duration that the DR environment is required at full scale. You can further reduce cost by purchasing Amazon EC2 Reserved Instances for your "always on" AWS servers.

Preparation phase

The following figure shows how you can use the weighted routing policy of the Amazon Route 53 DNS to route a portion of your traffic to the AWS site. The application on AWS might access data sources in the on-site production system. Data is replicated or mirrored to the AWS infrastructure.

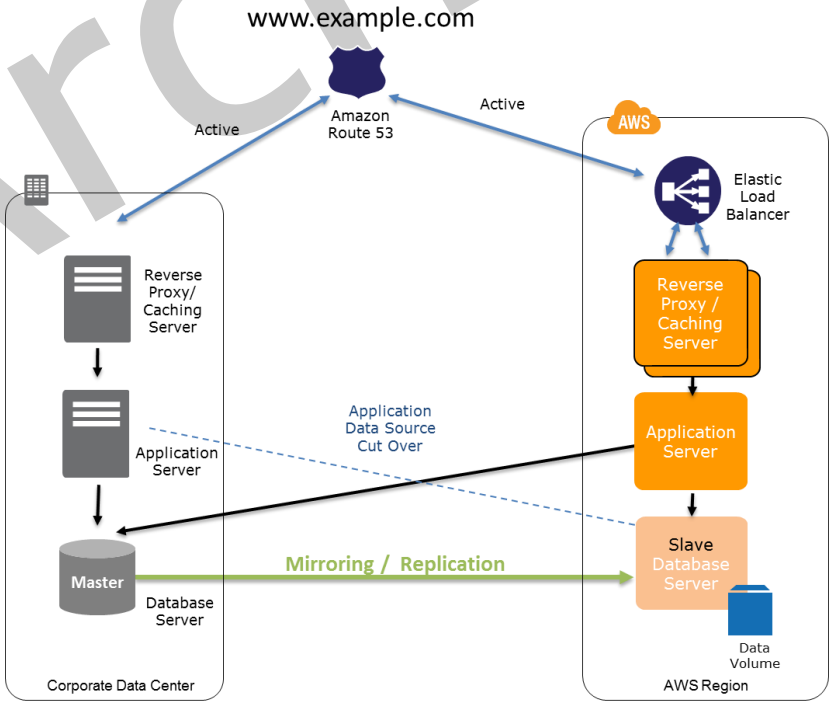

Figure 8: The Preparation Phase of the Multi-Site Scenario.

Key steps for preparation:

1. Set up your AWS environment to duplicate your production environment.
2. Set up DNS weighting, or similar traffic routing technology, to distribute incoming requests to both sites. Configure automated failover to re-route traffic away from the affected site.

Recovery phase

The following figure shows the change in traffic routing in the event of an on-site disaster. Traffic is cut over to the AWS infrastructure by updating DNS, and all traffic and supporting data queries are supported by the AWS infrastructure.

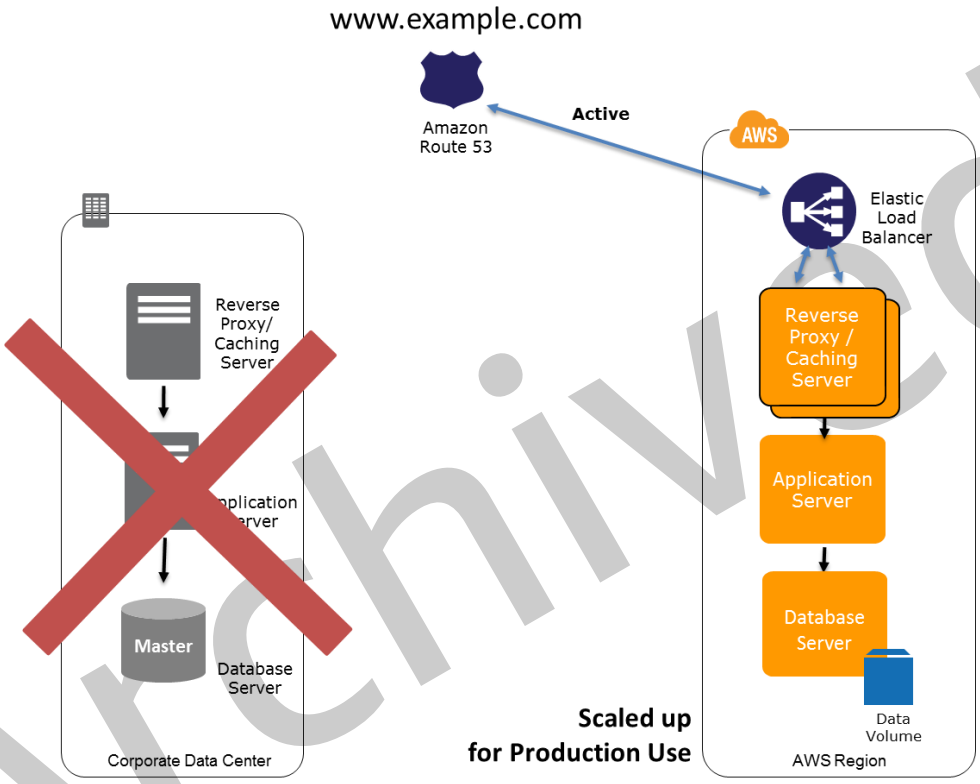

Figure 9: The Recovery Phase of the Multi-Site Scenario Involving On-Site and AWS Infrastructure.

Key steps for recovery:

1. Either manually or by using DNS failover, change the DNS weighting so that all requests are sent to the AWS site.
2. Have application logic for failover to use the local AWS database servers for all queries.
3. Consider using Auto Scaling to automatically right-size the AWS fleet.

You can further increase the availability of your multi-site solution by designing Multi-AZ architectures. For more information about how to design applications that span multiple availability zones, see the Building Fault-Tolerant Applications on AWS whitepaper.

AWS Production to an AWS DR Solution Using Multiple AWS Regions

Applications deployed on AWS have multi-site capability by means of multiple Availability Zones. Availability Zones are distinct locations that are engineered to be insulated from each other. They provide inexpensive, low-latency network connectivity within the same region.

Some applications might have an additional requirement to deploy their components using multiple regions; this can be a business or regulatory requirement.

Any of the preceding scenarios in this whitepaper can be deployed using separate AWS regions. The advantages for both production and DR scenarios include the following:

- You don't need to negotiate contracts with another provider in another region
- You can use the same underlying AWS technologies across regions
- You can use the same tools or APIs

For more information, see the Migrating AWS Resources to a New Region whitepaper.

Replication of Data

When you replicate data to a remote location, you should consider these factors:

- **Distance between the sites** — Larger distances typically are subject to more latency or jitter.
- **Available bandwidth** — The breadth and variability of the interconnections.
- **Data rate required by your application** — The data rate should be lower than the available bandwidth.
- **Replication technology** — The replication technology should be parallel (so that it can use the network effectively).

There are two main approaches for replicating data: synchronous and asynchronous.

Synchronous replication

Data is atomically updated in multiple locations. This puts a dependency on network performance and availability. In AWS, Availability Zones within a region are well connected, but physically separated. For example, when deployed in Multi-AZ mode, Amazon RDS uses synchronous replication to duplicate data in a second Availability Zone. This ensures that data is not lost if the primary Availability Zone becomes unavailable.

Asynchronous replication

Data is not atomically updated in multiple locations. It is transferred as network performance and availability allows, and the application continues to write data that might not be fully replicated yet.

Many database systems support asynchronous data replication. The database replica can be located remotely, and the replica does not have to be completely synchronized with the primary database server. This is acceptable in many scenarios, for example, as a backup source or reporting/read-only use cases. In addition to database systems, you can also extend it to network file systems and data volumes.

We recommend that you understand the replication technology used in your software solution. A detailed analysis of replication technology is beyond the scope of this paper.

AWS regions are completely independent of each other, but there are no differences in the way you access them and use them. This enables you to create DR processes that span continental distances, without the challenges or costs that this would normally incur. You can back up data and systems to two or more AWS regions, allowing service restoration even in the face of extremely large-scale disasters. You can use AWS regions to serve your users around the globe with relatively low complexity to your operational processes.

Failing Back from a Disaster

Once you have restored your primary site to a working state, you will need to restore your normal service, which is often referred to as a "fail back." Depending on your DR strategy, this typically means reversing the flow of data replication so that any data updates received while the primary site was down can be replicated back, without the loss of data. The following steps outline the different fail-back approaches:

Backup and restore

1. Freeze data changes to the DR site.
2. Take a backup.
3. Restore the backup to the primary site.
4. Re-point users to the primary site.
5. Unfreeze the changes.

Pilot light, warm standby, and multi-site

1. Establish reverse mirroring/replication from the DR site back to the primary site, once the primary site has caught up with the changes.
2. Freeze data changes to the DR site.
3. Re-point users to the primary site.
4. Unfreeze the changes.

Improving Your DR Plan

This section describes the important steps you should follow to establish a strong DR plan.

Testing

After your DR solution is in place, it needs to be tested. You can test frequently, which is one of the key advantages of deploying on AWS. "Game day" is when you exercise a failover to the DR environment, ensuring that sufficient documentation is in place to make the process as simple as possible should the real event take place. Spinning up a duplicate environment for testing your game-day scenarios is quick and cost-effective on AWS, and you typically don't need to touch your production environment. You can use AWS CloudFormation to deploy complete environments on AWS. This uses a template to describe the AWS resources and any associated dependencies or runtime parameters that are required to create a full environment.

Differentiating your tests is key to ensuring that you are covered against a multitude of different types of disasters. The following are examples of possible game-day scenarios:

- Power loss to a site or a set of servers
- Loss of ISP connectivity to a single site
- Virus impacting core business services that affects multi-sites
- User error that causes the loss of data, requiring a point-in-time recovery

Monitoring and alerting

You need to have regular checks and sufficient monitoring in place to alert you when your DR environment has been impacted by server failure, connectivity issues, and application issues. Amazon CloudWatch provides access to metrics about AWS resources, as well as custom metrics that can be application–centric or even business-centric. You can set up alarms based on defined thresholds on any of the metrics and, where required, you can set up Amazon SNS to send alerts in case of unexpected behavior.

You can use any monitoring solutions on AWS, and you can also continue to use any existing monitoring and alerting tools that your company uses to monitor your instance metrics, as well as guest OS stats and application health.

Backups

After you have switched to your DR environment, you should continue to make regular backups. Testing backup and restore regularly is essential as a fall-back solution.

AWS gives you the flexibility to perform frequent, inexpensive DR tests without needing the DR infrastructure to be "always on."

User access

You can secure access to resources in your DR environment by using AWS Identity and Access Management (IAM). With IAM, you can create role-based and user-based security policies that segregate user responsibilities and restrict user access to specified resources and tasks in your DR environment.

System access

You can also create roles for your Amazon EC2 resources, so that only users who are assigned to specified roles can perform defined actions on your DR environment, such as accessing an Amazon S3 bucket or re-pointing an Elastic IP address.

Automation

You can automate the deployment of applications onto AWS-based servers and your on-premises servers by using configuration management or orchestration software. This allows you to handle application and configuration change management across both environments with ease. There are several popular orchestration software options available. For a list of solution providers, see the AWS Partner Directory.[3]

AWS CloudFormation works in conjunction with several tools to provision infrastructure services in an automated way. Higher levels of abstraction are also available with AWS OpsWorks or AWS Elastic Beanstalk. The overall goal is to automate your instances as much as possible. For more information, see the Architecting for the Cloud: Best Practices whitepaper.

You can use Auto Scaling to ensure that your pool of instances is appropriately sized to meet the demand based on the metrics that you specify in AWS CloudWatch. This means that in a DR situation, as your user base starts to use the environment more, the solution can scale up dynamically to meet this increased demand. After the event is over and usage potentially decreases, the solution can scale back down to a minimum level of servers.

Software Licensing and DR

Ensuring that you are correctly licensed for your AWS environment is as important as licensing for any other environment. AWS provides a variety of models to make licensing easier for you to manage. For example, "Bring Your Own License" is possible for several software components or operating systems. Alternately, there is a range of software for which the cost of the license is included in the hourly charge. This is known as "License included."

"Bring your Own License" enables you to leverage your existing software investments during a disaster. "License included" minimizes up-front license costs for a DR site that doesn't get used on a day-to-day basis.

If at any stage you are in doubt about your licenses and how they apply to AWS, contact your license reseller.

Conclusion

Many options and variations for DR exist. This paper highlights some of the common scenarios, ranging from simple backup and restore to fault tolerant, multi-site solutions. AWS gives you fine-grained control and many building blocks to build the appropriate DR solution, given your DR objectives (RTO and RPO) and budget. The AWS services are available on-demand, and you pay only for what you use. This is a key advantage for DR, where significant infrastructure is needed quickly, but only in the event of a disaster.

This whitepaper has shown how AWS provides flexible, cost-effective infrastructure solutions, enabling you to have a more effective DR plan.

[3] Solution providers can be found at http://aws.amazon.com/solutions/solution-providers/

Further Reading

- *Amazon S3 Getting Started Guide*: http://docs.amazonwebservices.com/AmazonS3/latest/gsg/
- *Amazon EC2 Getting Started Guide*: http://docs.amazonwebservices.com/AWSEC2/latest/GettingStartedGuide/
- AWS Partner Directory (for a list of AWS solution providers): http://aws.amazon.com/solutions/solution-providers/
- AWS Security and Compliance Center: http://aws.amazon.com/security/
- AWS Architecture Center: http://aws.amazon.com/architecture
- Whitepaper: Designing Fault-Tolerant Applications in the AWS Cloud
- Other AWS technical whitepapers: http://aws.amazon.com/whitepapers

Document Revisions

We've made the following changes to this whitepaper since its original publication in January, 2012:

- Updated information about AWS regions
- Added information about new services: Amazon Glacier, Amazon Redshift, AWS OpsWorks, AWS Elastic Beanstalk, and Amazon DynamoDB
- Added information about elastic network interfaces (ENIs)
- Added information about various features of AWS services for DR scenarios using multiple AWS regions
- Added information about AWS Storage Gateway virtual tape libraries