Using Amazon Web Services for Disaster Recovery

October 2011

Glen Robinson, Ianni Vamvadelis, and Attila Narin

Contents

Abstract

In the event of a disaster, you can quickly launch resources in Amazon Web Services (AWS) to ensure business continuity. The paper highlights relevant AWS features and services that you can leverage for your DR processes and shows example scenarios about how to recover from disaster. It further provides recommendations about how you can improve your DR plan and leverage the full potential of AWS for your Disaster Recovery processes.

Introduction

Disaster recovery (DR) is about preparing for and recovering from a disaster. Any event that has a negative impact on your business' continuity or finances could be termed a disaster. This could be 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 disaster.

To minimize the impact of a disaster on the business, companies invest time and resources to plan, prepare, rehearse, document, train, and update processes to deal with events. The amount of investment for the disaster recovery planning of a particular system can vary dramatically depending on the cost of a potential outage. This paper describes some typical approaches ranging from minimal investments to full-scale availability and fault tolerance.

Proper preparation for DR is a must, and this paper outlines some of the best practices to improve your DR plans and processes.

Disaster recovery is a continual process of analysis and improvement, as business and systems evolve. For each business service, customers need to establish an acceptable recovery point and time, and then build an appropriate DR solution.

In a traditional physical environment, a typical approach would normally involve the duplication of infrastructure to ensure the availability of spare capacity in a disaster scenario. This infrastructure needs to be procured, installed, and maintained so that it is ready to deal with the anticipated capacity requirements. Under normal operational circumstances, this infrastructure would typically be under-utilized or over-provisioned.

AWS allows you to scale up your infrastructure on an as-needed basis. You get access to the same highly scalable, reliable, secure, fast, inexpensive infrastructure that Amazon uses to run its own global network of web sites and only pay for what you use. For a disaster recovery (DR) solution, this results in significant cost savings. It also allows for more agility to change and optimize resources during a DR scenario.

Human error is the cause of a large proportion of system downtime. AWS provides tools to allow for the segregation of duties to enable a *least privilege*[1] design. AWS also enables you to automate the deployment of whole environments, enabling predictable and repeatable configurations. Test DR environments can be set up very quickly, and you can then treat them as a disposable resource. This enables organizations to test configuration changes in a duplicate environment before pushing the configuration changes into production, without the need for a dedicated full-scale test environment, which would often be underutilized.

[1] http://en.wikipedia.org/wiki/Principle_of_least_privilege

Recovery Time Objective and Recovery Point Objective

This paper uses two common industry terms for disaster planning:

Recovery time objective (RTO)[2]—This is the duration of time and the service level to which a business process must be restored after a disaster (or disruption) to avoid unacceptable consequences associated with a break in business continuity. For example, if a disaster occurs at 12:00 PM (noon) and the RTO is 8 hours, the DR process would ensure recovery to the acceptable service level would be possible by 8:00 PM.

Recovery point objective (RPO)[3]—This describes the acceptable amount of data loss measured in time. For example, if the RPO was 1 hour, after the system was recovered, it would contain all data up to a point in time that is not prior to 11:00 AM because the disaster occurred at noon.

A business typically decides on an acceptable RTO and RPO based on the financial impact to the business when systems are unavailable. The financial impact is typically assessed 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 cost-effectively provide system recovery based on the RPO within the timeline and 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.

The infrastructure needed to support the duplicate environment would include, but not be limited to 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.

Depending on the criticality of the services, the duplicate environment may be configured in a fault tolerant manner. This normally involves duplicating the entire infrastructure listed above.

[2] Taken from http://en.wikipedia.org/wiki/Recovery_time_objective
[3] Taken from http://en.wikipedia.org/wiki/Recovery_point_objective

AWS Services and Features Essential for Disaster Recovery

Before discussing 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 essential to consider the use of services and features that support data migration and durable storage because they enable you to restore back 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 here include DNS, networking features, and various Amazon Elastic Compute Cloud (Amazon EC2) features described below.

Regions

Amazon Web Services are available in multiple Regions, so you can choose the most appropriate location for your disaster recovery site, in addition to the site where your system is fully deployed. At the time of writing, AWS is available in five regions: US East (Northern Virginia), US West (Northern California), EU (Ireland), Asia Pacific (Singapore), and Asia Pacific (Tokyo).

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. AWS provides further protection for data retention and archiving via Versioning in Amazon S3, AWS Multi-Factor Authentication, bucket policies, and Identity and Access Management (IAM).

Amazon Elastic Block Store (Amazon EBS) provides the ability to create point-in-time snapshots of data volumes. Such snapshots can be used as the starting point for new Amazon EBS volumes, and to protect data for long-term durability. Once a volume is created, it can then be attached to a running Amazon EC2 instance. Amazon EBS volumes provide off-instance storage that persists independently from the life of an instance.

AWS Import/Export accelerates the moving of large amounts of data into and out of AWS using portable storage devices for transport. AWS transfers your data directly onto and off of storage devices by using Amazon's high-speed internal network and bypassing the Internet. 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 or into Amazon EBS snapshots.

Compute

Amazon Elastic Compute Cloud (Amazon EC2) provides resizable compute capacity in the cloud. Within minutes, you can create EC2 instances, which are virtual machines over which you have complete control. In the context of DR, this 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; we will 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 may also include application stacks. You can also configure your own AMIs. In the context of DR, we strongly recommended that you have your own AMIs configured and identified 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.

Amazon EC2 Reserved Instances, which are often used to receive a significant discount on the cost of running an EC2 instance, have another advantage that is especially relevant to DR. Reserved Instances help to ensure that the capacity you need is available to you when required.

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. 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 feature enables you to import virtual machine images from your existing environment to Amazon EC2 instances.

Networking

When dealing with a disaster, it's very likely that you will have to modify network settings as you are failing over to another site.

Amazon Route 53 is a highly available and scalable Domain Name System (DNS) web service. It is designed to give developers and businesses an extremely reliable and cost-effective way to route end users to Internet applications.

Elastic IP Addresses are static IP addresses designed for dynamic cloud computing. Unlike traditional static IP addresses, however, 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, seamlessly providing the amount of load balancing capacity needed in response to incoming application traffic. Just as you can pre-allocate Elastic IP addresses, you can pre-allocate your Elastic 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 Amazon Web Services 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 would enable you to create a VPN connection between your corporate datacenter and your VPC and leverage the AWS cloud as an extension of your corporate datacenter. 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 premise 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 could use Amazon RDS either in the preparation phase for DR to hold your critical data in a running database already, and/or in the recovery phase to run your production database.

Amazon SimpleDB is a highly available, flexible, non-relational data store that offloads the work of database administration. It can also be used in the preparation and the recovery phase of DR.

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

For further details about database options on AWS, please see Running Databases on AWS.

Deployment Orchestration

Deployment automation and post-startup software installation/configuration processes and tools can be used in Amazon EC2. Investments in this area are highly recommended. This can be very helpful in the recovery phase to create the required set of resources in an automated fashion.

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.

Security

There are many security related features across the AWS services. We recommend that customers 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 walks through four DR scenarios that highlight usage of AWS and compare AWS with traditional DR methods:

- Backup and Restore
- Pilot Light for Simple Recovery into AWS
- Warm Standby Solution
- Multi-site Solution

Amazon Web Services enables customers to cost effectively operate each of these example DR strategies. It's important to note that these are just examples of possible approaches, and variations and combinations of these are possible.

Backup and Restore

In most traditional environments, data is backed up to tape and sent off-site regularly. Your recovery time will be the longest using this method. Amazon S3 is an ideal destination for backup data, as it is designed to provide 99.999999999% (11 9s) durability of objects over a given year. Transferring data to and from Amazon S3 is typically done via the network, and is therefore accessible from any location. There are many commercial and open source backup solutions which backup to Amazon S3. The AWS Import/Export service enables transfers of very large data sets by shipping storage devices directly to AWS.

For systems running on AWS, customers also back up into Amazon S3. Snapshots of Elastic Block Store (EBS) volumes and backups of Amazon RDS are stored in Amazon S3. Alternatively, you can copy files directly into Amazon S3, or you can choose to create backup files and copy them to Amazon S3. There are many backup solutions which store backup data in Amazon S3, and these can be used from Amazon EC2 systems as well.

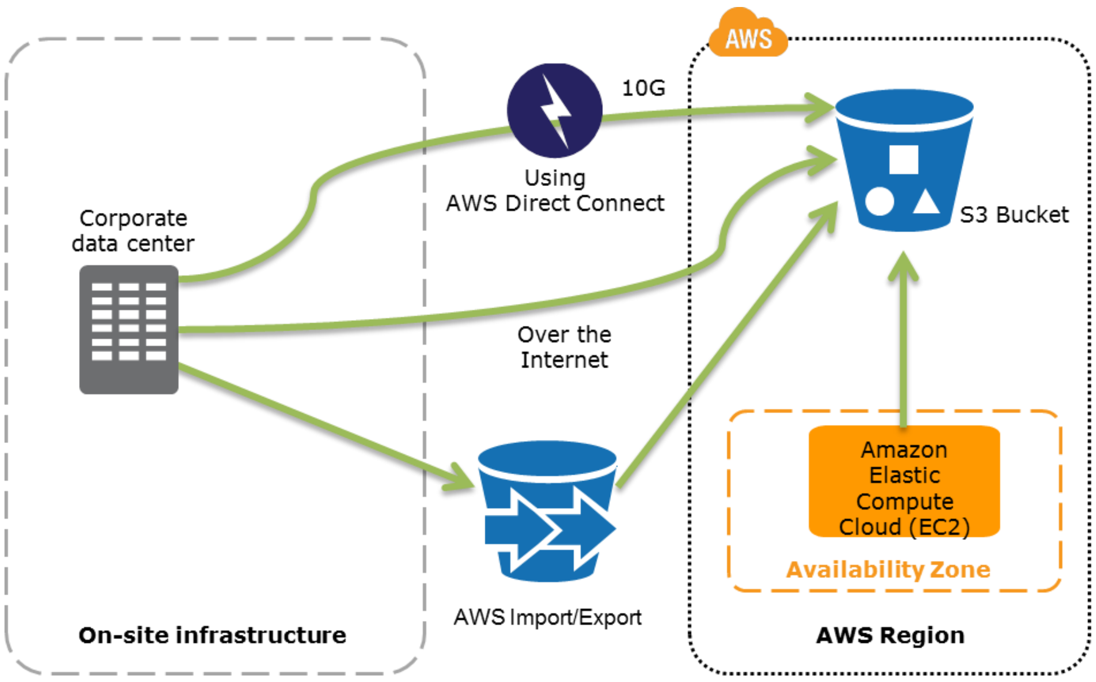

Figure 1: Data backup options to S3 from on-site infrastructure, or from AWS.

The backup of your data is only half the story. Recovery of data in a disaster scenario needs to be tested and achieved quickly and reliably. Customers should ensure that their systems are configured to appropriate retention of data, security of data, and have tested their data recovery processes.

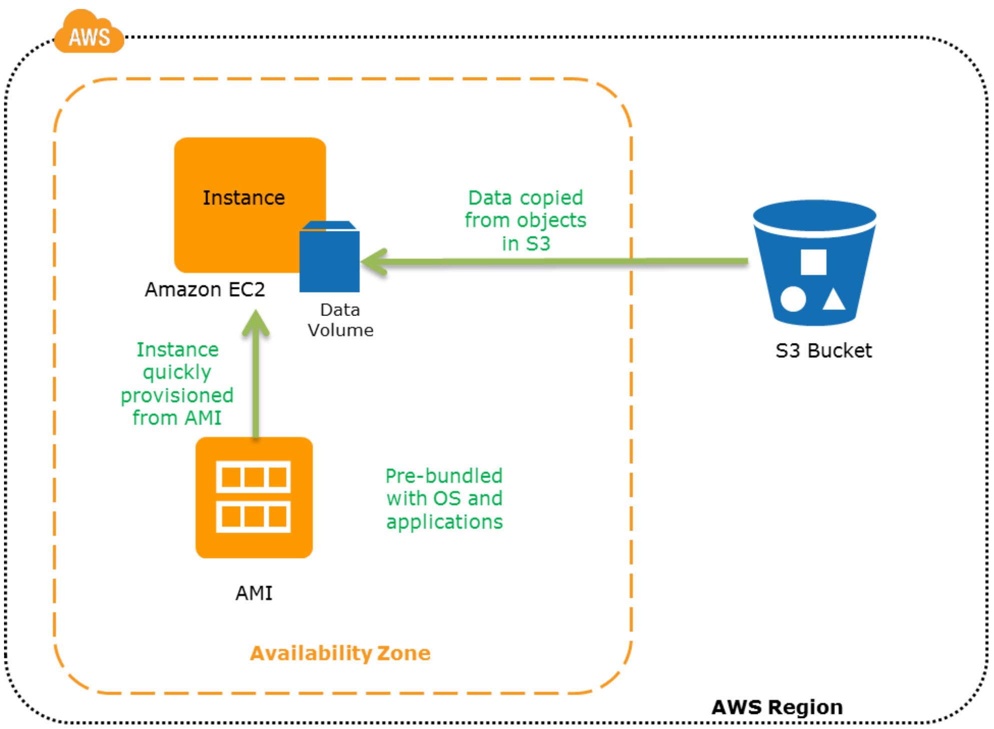
AWS
Instance
Amazon EC2
Data Volume
Data copied from objects in S3
S3 Bucket
Instance quickly provisioned from AMI
AMI
Pre-bundled with OS and applications
Availability Zone
AWS Region

Figure 2: Restoring a system from S3 backups to AWS EC2

Key steps for backup and restore:

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

Pilot Light for Quick Recovery into AWS

The idea of the pilot light is an analogy that comes from the gas heater. In a gas heater, a small idle flame that's always on can quickly ignite the entire furnace to heat up a house as needed. This scenario is similar to a Backup and Restore scenario, however, you must ensure that you have the most critical core elements of your system already configured and running in AWS (the pilot light). When the time comes for recovery, you would then 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 be replicating data to Amazon EC2. Depending on the system, there may 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 can quickly be provisioned (the rest of the furnace) 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 and find their role within the deployment around the pilot light. From a networking point of view, you can either use Elastic IP Addresses (which can be pre-allocated in the preparation phase for DR) and associate them with your instances, or use Elastic Load Balancing to distribute traffic to multiple instances. You would then update your DNS records to point at your Amazon EC2 instance or point to your Elastic Load Balancing using a CNAME.

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 EBS snapshot. This will speed up the application server setup, because you can quickly create multiple volumes in multiple Availability Zones, to attach to EC2 instances. You can then install and configure accordingly.

The Pilot Light method will give you a quicker Recovery Time than the "Backup and Restore" scenario above, because the core pieces of the system are already running and are continually kept up to date. There are still some installation and configuration tasks to fully recover the applications. 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.

Preparation Phase:

The following figure shows the preparation phase, in which you need to have your regularly changing 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 Amazon Machine Images (AMIs).

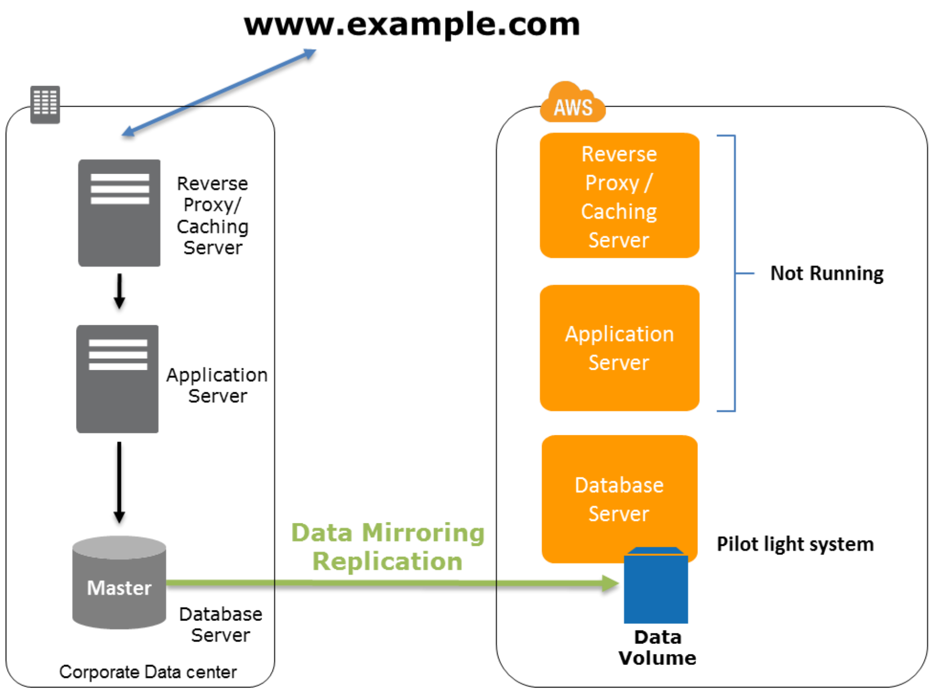

Figure 3: The preparation phase of Pilot Light scenario

Key points for preparation:

- Set up EC2 instances to replicate or mirror data.
- Ensure that you have all supporting custom software packages available in AWS.
- Create and Maintain Amazon Machine Images (AMI) of key servers where fast recovery is required.
- Regularly run these servers, test them, and apply any software updates and configuration changes.
- Consider automating the provisioning of AWS resources.

Recovery Phase:

To recover the remainder of the environment around the pilot light, you would start your systems from the Amazon Machine Images (AMIs) in 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, if possible, is often the most cost effective way and scalable approach to add capacity to a system, however, it’s also possible to pick larger EC2 instance types and thus scale horizontally. From a networking perspective, any required DNS updates can be done in parallel.

Once recovered, you should ensure that redundancy is restored as quickly as possible. While a failure of your DR environment shortly after your production environment failed is unlikely, you need to be aware of this risk. Continue to take regular backups of your system and consider additional redundancy at the data layer.

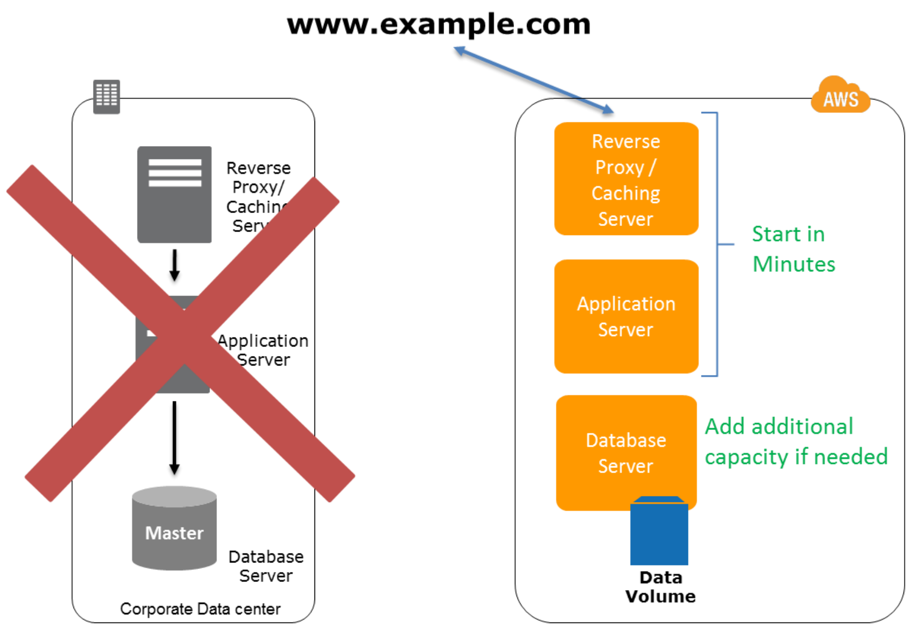

Figure 4: The recovery phase of the Pilot light scenario.

Key points for recovery:

- Start your application EC2 instances from your custom AMI’s.
- Resize and/or scale any database / data store instances, where necessary.
- Change DNS to point at the EC2 servers.
- Install and configure any non-AMI based systems, ideally in an automated fashion.

Warm Standby Solution in AWS

A warm standby solution extends the pilot light elements and preparation. It further decreases the recovery time because in this case, some services are always running. By identifying your business-critical systems, you would fully duplicate these systems on AWS and have them always on.

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

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 EC2 instance types. As stated above, horizontal scaling, if possible, is often preferred over vertical scaling.

Preparation Phase:

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

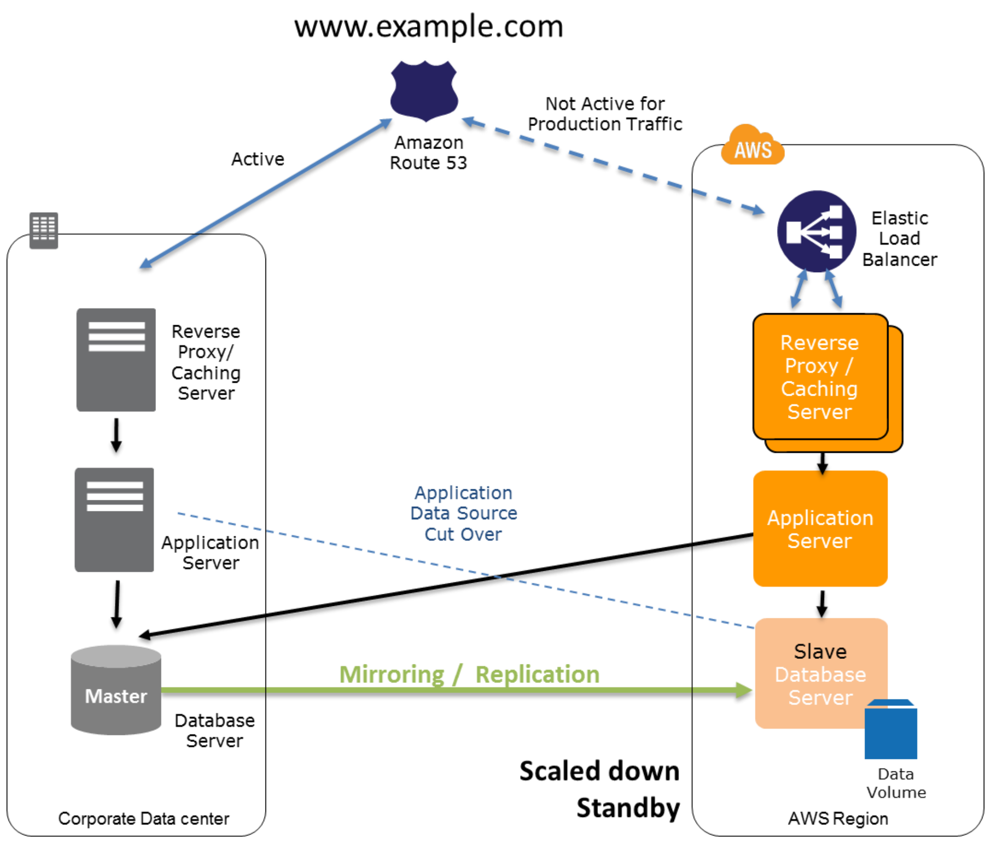

Figure 5: The preparation phase of the "warm standby" scenario.

Key points for preparation:

- Set up EC2 instances to replicate or mirror data.
- Create and maintain Amazon Machine Images (AMIs).
- Run your application using a minimal footprint of EC2 instances or AWS infrastructure.
- 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 are changed to route all traffic to AWS.

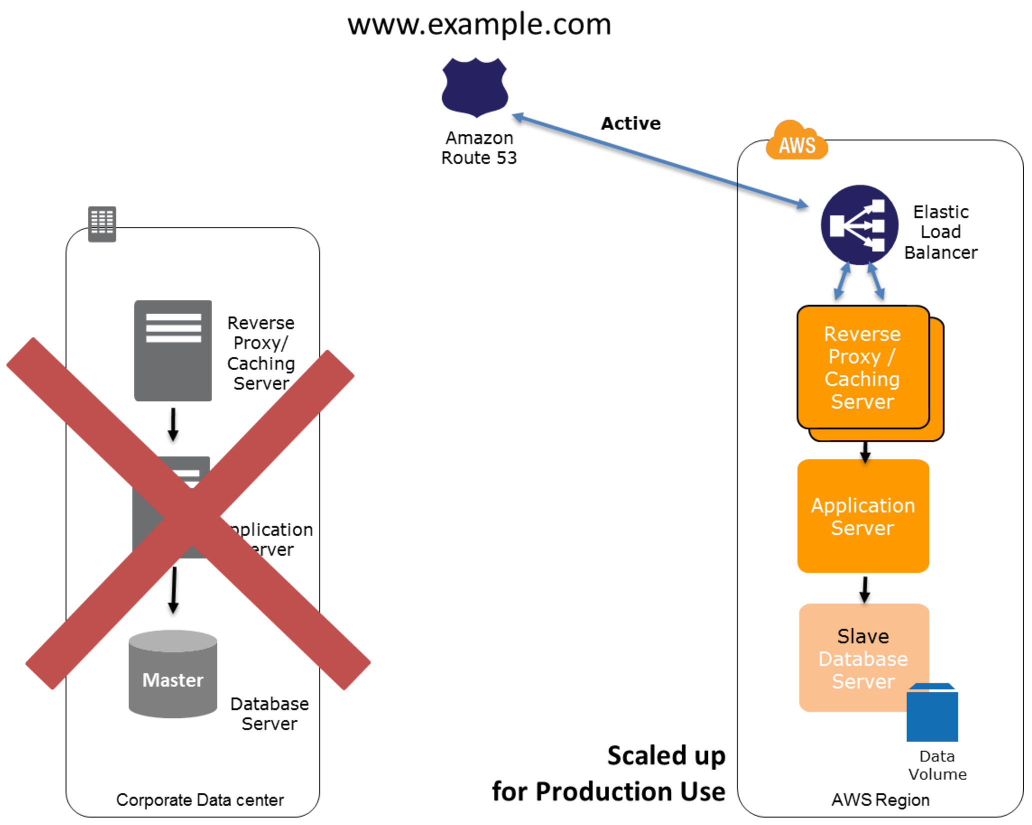

Figure 6: The recovery phase of the "warm standby" scenario.

Key points for recovery:

- Start applications on larger EC2 Instance types as needed (vertical scaling).
- Increase the size of the EC2 fleets in service with the Load Balancer (horizontal scaling).
- Change the DNS records so that all traffic is routed to the AWS environment.
- Consider using Auto scaling to right-size the fleet or accommodate the increased load.

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 (see RPO above) you choose. Various replication methods exist (see below).

A weighted DNS service, such as Amazon Route 53, is used to route production traffic to the different sites. 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. EC2 Auto Scaling can be used to automate this process. You may 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 in normal operation. In the recovery phase, you only pay for what you use in addition and for the duration that the DR environment is required at full scale. You can further reduce cost by purchasing Reserved Instances for your "always on" AWS servers.

Preparation Phase:

In the figure below, we see the use of DNS to route a portion of the traffic to the AWS site. The application on AWS may access data sources in the on-site production system. Data is replicated or mirrored to AWS infrastructure.

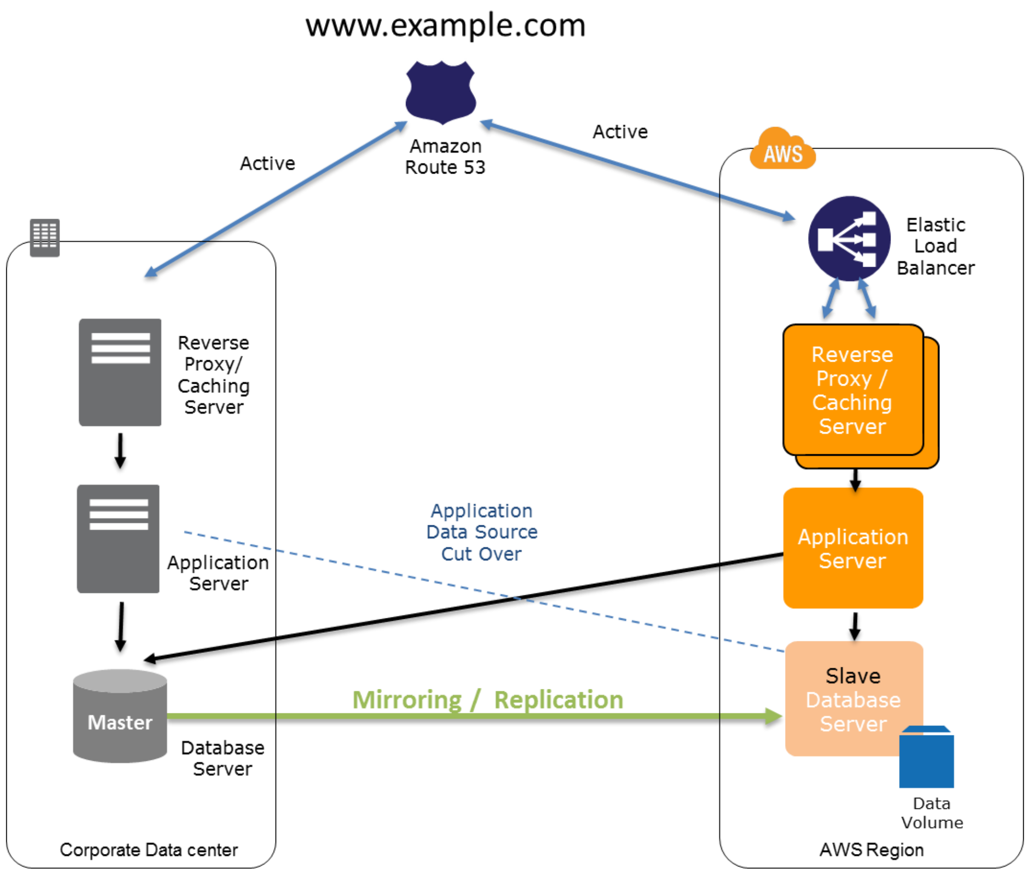

Figure 7: The preparation phase of the "Multi-Site" scenario.

Key points for preparation:

- Set up your AWS environment to duplicate your production environment.
- Set up DNS weighting or similar technology to distribute incoming requests to both sites.

Recovery Phase:

The figure below shows what happens when a disaster occurs on-site. Traffic is cut over to the AWS infrastructure by updating DNS.

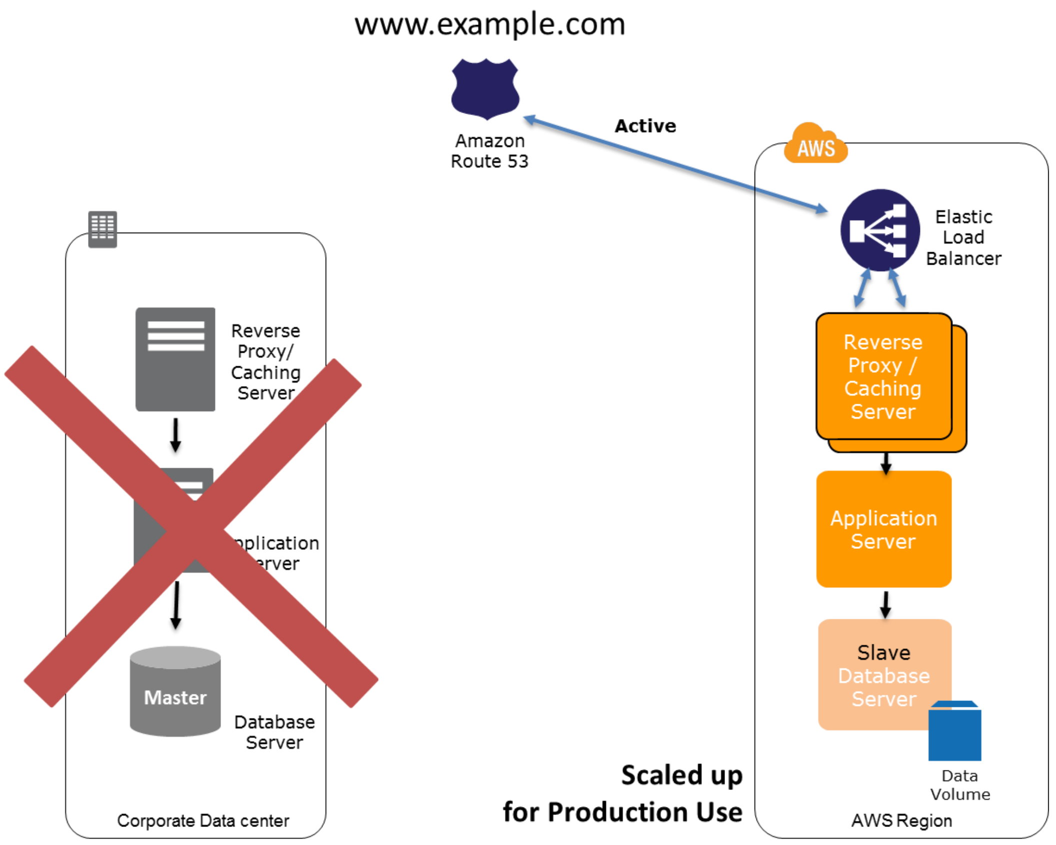

Figure 8: The recovery phase of the "multi-site" scenario involving on-site and AWS infrastructure.

Key points for recovery:

- Change the DNS weighting, so that all requests are sent to the AWS site.
- Have application logic for failover to use the local AWS database servers.
- 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 on how to design applications that span across multiple availability zones, please refer to Designing Fault-Tolerant Applications in the AWS Cloud whitepaper.

Replication of Data

When replicating data to a remote location, there are a few factors to consider.

- Distance between the sites: larger distances typically are subject to more latency and/or jitter.
- Bandwidth available: how broad and variable are the interconnections?
- Data rate required by your application: data rate should be lower than the available bandwidth.
- The replication technology should be parallel (so that it can use the network effectively).

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

Synchronous Replication

Data is atomically updated in multiple locations. This puts a dependency on network performance and availability.

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 may 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 in sync with the primary database server. This is acceptable in many scenarios, for example, as a backup source or reporting/read-only use cases.

We advise customers to understand the replication technology used in their software solution. A detailed analysis of replication technology is outside of the scope of this paper.

In AWS, Availability Zones within a Region are well connected, but physically separated. For example, when deployed in "Multi-AZ" mode, the Amazon Relational Database Service 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.

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

Improving Your DR Plan

Some important steps need to be followed in order to have a solid DR plan. This section describes some of the main steps.

Testing

After your DR solution is in place, it needs to be tested. "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, 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 example "Game Day" scenarios:

- Power loss to a site or a set of machines
- Loss of ISP connectivity to a single site
- Virus impacting core business services affecting multi-sites
- User error that caused 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. Alarms can be set up based on defined thresholds on any of the metrics and, where required Amazon Simple Notification Service messages can be sent to alert in case of unexpected behavior. You can use any monitoring solutions on AWS.

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

Once 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 do 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 Identity and AWS Access Management (IAM). This way you can create role/user based security policies that segregate user responsibilities while they work on your DR environment.

Automation

You can automate the deployment of applications onto the AWS-based servers and your on-premises servers by using configuration management or orchestration software. This will allow you to handle application and configuration change management across both environments with ease. There are several popular orchestration software options available, plus possible solution providers can be found on our solution provider's page[4]. AWS CloudFormation works in conjunction with several tools to provision the infrastructure services in an automated manner. User data can be passed into the instance on first boot and can then handed to a configuration management tool to determine the instance type or role to ensure that the correct software and configuration is deployed. The overall goal should be to have your instances end up in the final state in which you need them as automatically as possible.

Auto Scaling can be used to ensure that your pool of instances is appropriately sized to meet the demand based on the metrics you specify in 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 any other environment. Amazon 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, i.e. during a DR test.

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, and this paper highlights some of the common patterns, 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 only pay 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 paper has shown how AWS provides flexible, cost-effective infrastructure solutions, enabling you to have a more effective DR plan.

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

Further Reading

- The S3 Getting Started Guide: http://docs.amazonwebservices.com/AmazonS3/latest/gsg/
- The EC2 Getting Started Guide: http://docs.amazonwebservices.com/AWSEC2/latest/GettingStartedGuide/
- Find an AWS Solution Provider: http://aws.amazon.com/solutions/solution-providers/
- Designing Fault-Tolerant Applications in the AWS Cloud whitepaper: http://aws.amazon.com/whitepapers/
- AWS Security and Compliance Center: http://aws.amazon.com/security/
- AWS Architecture Center: http://aws.amazon.com/architecture
- AWS Technical Whitepapers: http://aws.amazon.com/whitepapers