**AWS DISASTER RECOVERY**

What is DR (Disaster Recovery)?

* 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.

Disaster Recovery in AWS:

* According to AWS, “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.”
* DR on AWS can significantly reduce costs (up to half the costs) as compared to a company maintaining its own redundant data centers. These costs include buying and maintaining servers and data centers, providing secure and stable connectivity and keeping them secure. The servers would also be underutilized.

**Recovery Time Objective and Recovery Point Objective:**

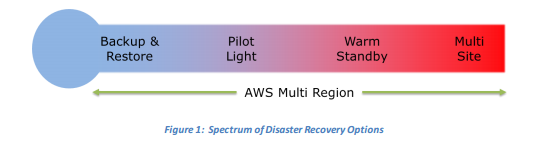
* Recovery time objective (RTO) — The time it takes after a disruption to restore a business process to its service level, as defined by the operational level agreement (OLA). For example, if a disaster occurs at 12:00 PM (noon) and the RTO is eight hours, the DR process should restore the business process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

**Disaster Recovery Scenarios with Aws:**

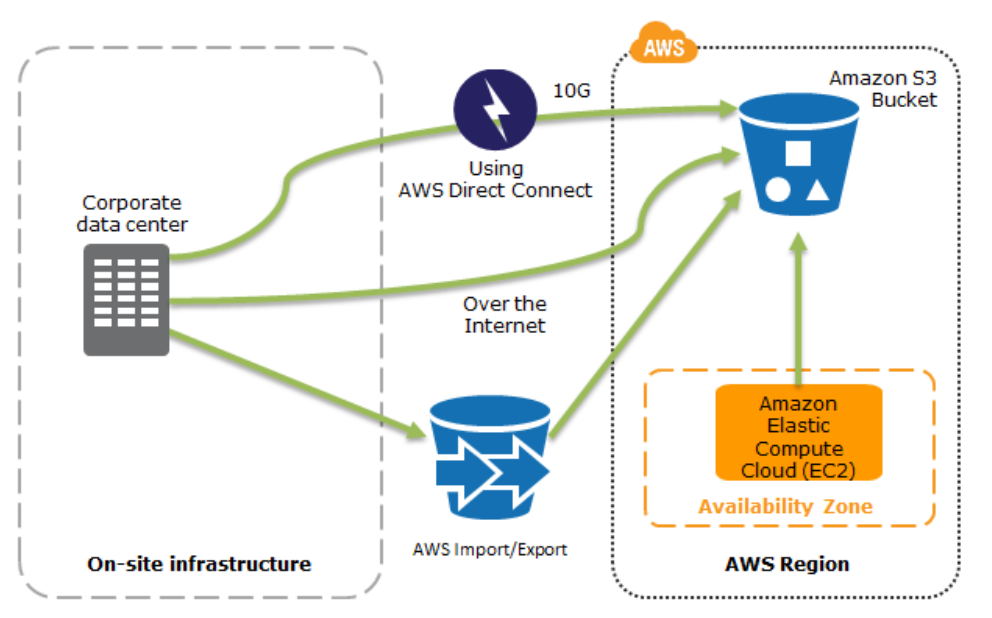
* There are four DR scenarios that highlight the use of AWS.
* The following figure shows a spectrum for the four scenarios, arranged by how quickly a system can be available to users after a DR event.



1. **Backup and Restore with AWS:**

* To recover your data in the event of any disaster, you must first have your data periodically backed up from your system to AWS. Backing up of data can be done through various mechanisms and your choice will be based on the RPO (Recovery Point Objective).
* For example, if you have a frequently changing database like say a stock market, then you will need a very high RPO. However, if your data is mostly static with a low frequency of changes, you can opt for periodic incremental backup.

**Data Backup Options to Amazon S3**



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

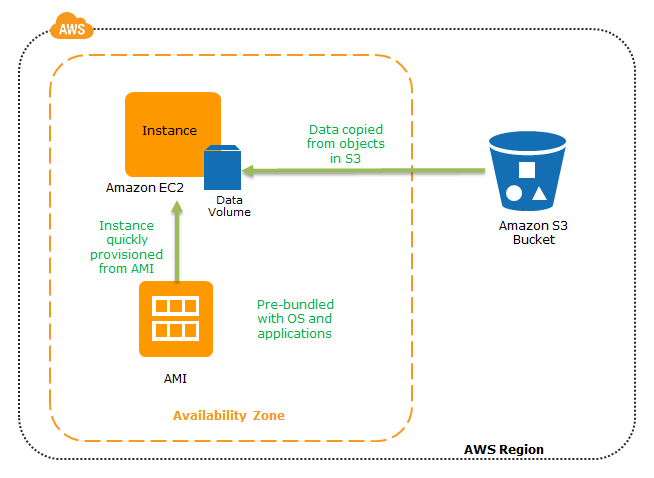
Amazon S3 is an ideal destination for backup data that might be needed quickly to perform a restore. Transferring data to and from Amazon S3 is typically done through the network.

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 for infrequent access, 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.

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.

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.

**Restoring a system from Amazon S3 backups to Amazon EC2:**

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

Once your backup mechanisms are activated you can pre-configure Amazon Machine Images (kind of like a Class while the EC2 instance is the object instantiated from the AMI class)[AMIs](http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/AMIs.html)(operating systems & application software). Now when a disaster strikes, EC2 (Elastic Compute Capacity) instances in the Cloud using EBS (Elastic Block Store) coupled with AMIs can access your data from the S3 (Simple Storage Service) buckets to revive your system and keep it going.

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.

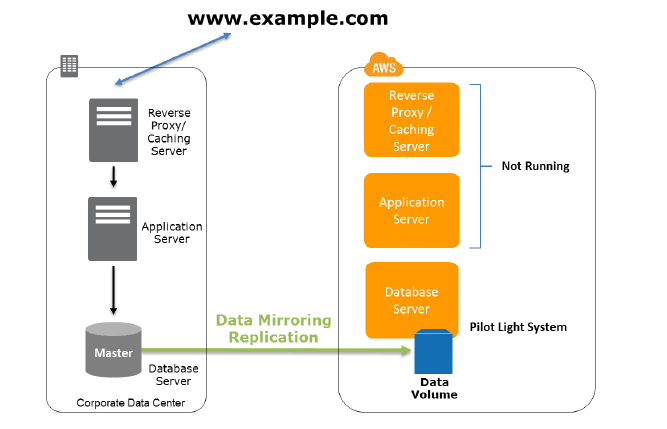
3. Ensure that appropriate security measures are in place for this data,   
 including encryption and access policies.

4. Regularly test the recovery of this data and the restoration of your system.

1. **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.
* 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 includes database servers, which would replicate data to Amazon EC2 or Amazon RDS. 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, there would be some pre-configured servers bundled as Amazon Machine Images (AMIs), which are ready to be started up at a moment’s notice (this is the furnace in the analogy). 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.
* If the on-premise system fails, then the application and caching servers get activated; further users are rerouted using elastic IP addresses (which can be pre-allocated and identified in the preparation phase for DR) which can be associated to the new instances in the ad-hoc environment on cloud. Recovery takes just a few minutes.
* The other option is to use Elastic Load Balancer (ELB) which automatically distributes incoming application traffic across multiple Amazon EC2 instances. It provides even greater fault tolerance for applications by seamlessly providing the load-balancing capacity that is needed in response to incoming application traffic. The load balancer can be pre-allocated so that its DNS name is already known and the customer DNS tables point to the load balancer.

**Pilot Light – Preparation Phase**



The figure shows the preparation phase, in which regularly changing data is replicated to the pilot light, the small core around which the full environment will be started in the recovery phase. Less frequently updated data, such as operating systems and applications, can be periodically updated and stored as AMIs.

Key steps for preparation:

1. Set up Amazon EC2 instances to replicate or mirror data.

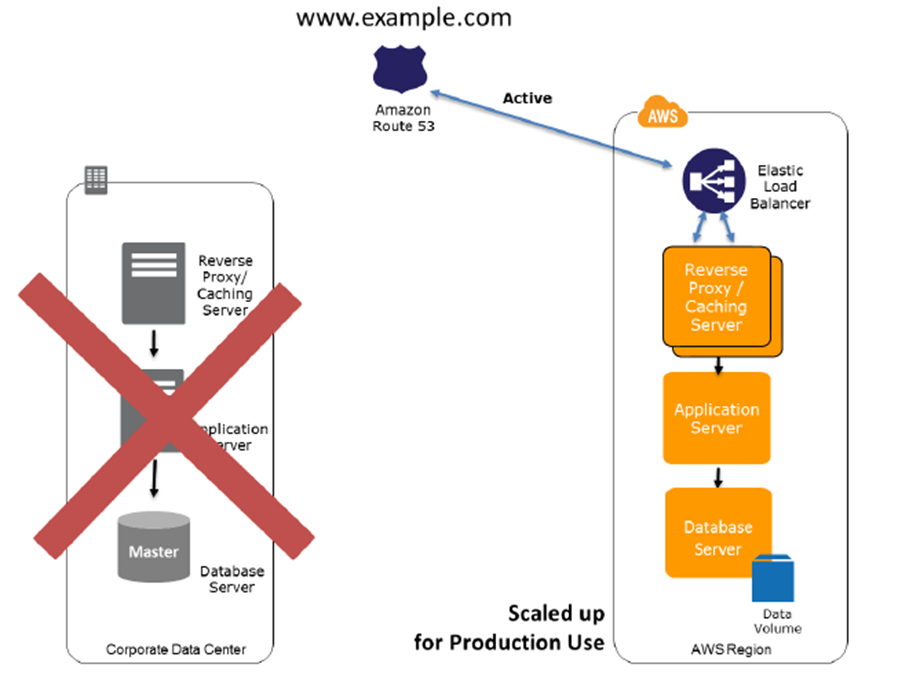
2. Ensure that you have all supporting custom software packages available in AWS.

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.

**Pilot Light – 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.

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.

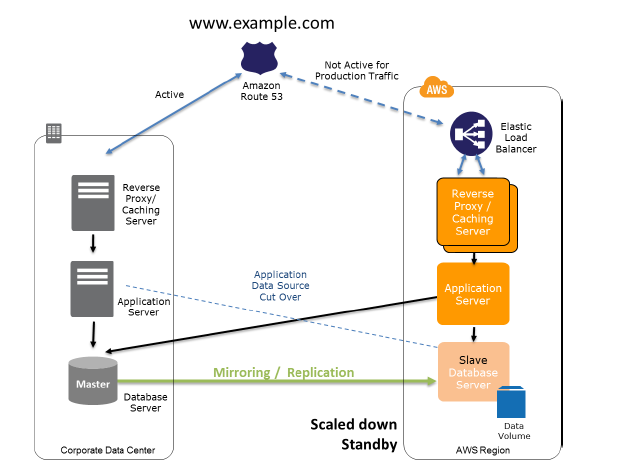
3. Add additional database/data store instances to give the DR site resilience in the   
 data tier; if you are using Amazon RDS, turn on multi-AZ to improve resilience.

4. Change DNS to point at the Amazon EC2 servers.

5. Install and configure any non-AMI based systems, ideally in an automated way.

* 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.

1. **Warm Standby – Preparation Phase:**



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

Key steps for preparation:

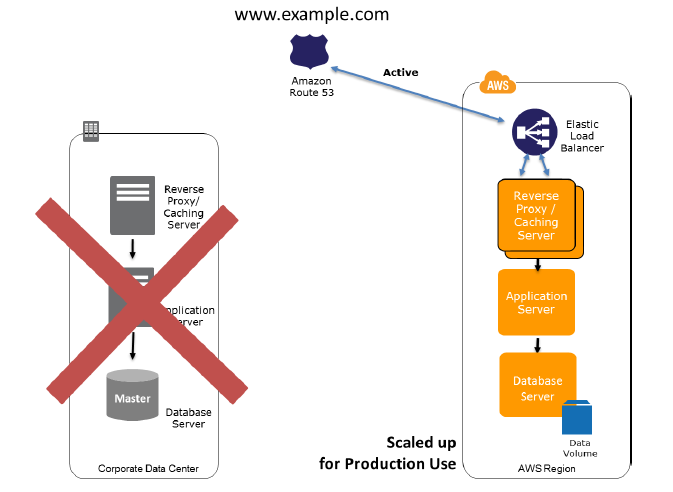
1. Set up Amazon EC2 instances to replicate or mirror data.

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.

**Warm Standby – 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.

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.

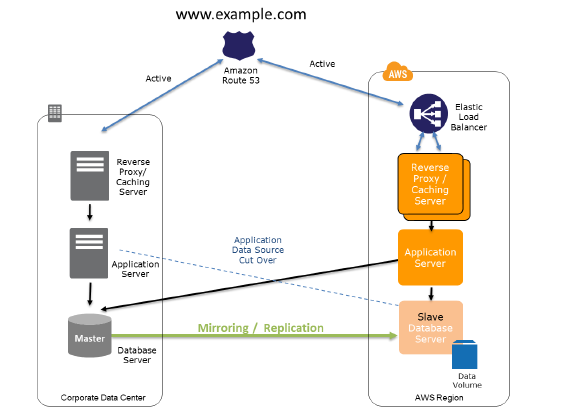
4. Consider using Auto Scaling to right-size the fleet or accommodate the   
 increased load.

5. Add resilience or scale up your database.

1. **Multi-Site Solution Deployed on AWS and On-Site:**

* This is the optimum technique in backup and DR and is the next step after warm standby. A multi-site solution runs in AWS as well as on your existing on-site infrastructure, in an active-active (or hot-hot) configuration.
* All activities in the preparatory stage are similar to a warm standby; except that the AWS backup on the cloud is also used to handle some portions of the user traffic using Route 53, a DNS service that supports weighted routing.
* When a disaster strikes, the rest of the traffic that was pointing to the on-premise servers are rerouted to AWS and using auto scaling techniques multiple EC2 instances are deployed to handle full production capacity. You can further increase the availability of your multi-site solution by using multi-AZ’s (Availability Zones).
  + 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 (data is atomically updated in multiple locations) to duplicate data in a second Availability Zone. This ensures that data is not lost if the primary Availability Zone becomes unavailable.
* 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.
* 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.

**Multi-Site Solution – Preparation Phase:**

****

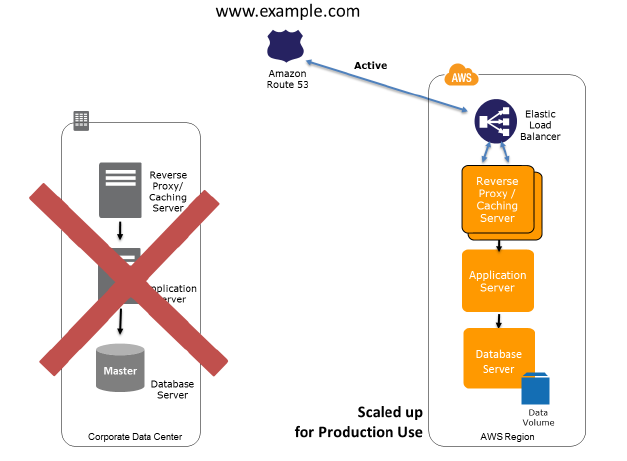
* The figure shows the use of weighted routing policy of the Amazon Route 53 DNS to route a portion of the traffic to the AWS site. The application on AWS might access data sources in the on-site production system. Data is replicated or mirrored to the AWS infrastructure.

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.

**Multi-Site Solution – Recovery Phase:**

****

The 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.

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.

Replication of Data:

* When data is replicated to a remote location, these factors need to considered:
* **Distance between the sites** — Larger distances typically are subject to more latency or jitter.
* **Available bandwidth**
* **Data rate required by your application** — The data rate should be lower than the available bandwidth.

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, this can also be extended to network file systems and data volumes.