Using AWS for Disaster Recovery of On-premises Applications

The following section outlines the benefits of using AWS for disaster recovery, explores various AWS disaster recovery solutions, and explains how to assess and implement a disaster recovery solution.

Topics:

Benefits of Using AWS for Disaster Recovery

Using AWS for disaster recovery offers services that have the following benefits:

- **Elasticity** – Disaster recovery-related AWS services are normally billed per usage. This provides the flexibility to utilize AWS as a disaster recovery site by paying only for the resources used, rather than committing to a long-term contract or set number of servers.
- **TCO** – AWS offers a lower TCO than traditional, non-cloud solutions. AWS-based disaster recovery uses minimal resources in customers' AWS accounts – primarily low-cost storage for replicating data. Customers are billed only for fully provisioned servers when launched at the time of recovery or drill.
- **RTO and RPO** – Ability to achieve RTOs of minutes by launching the disaster recovery site on demand and RPOs of seconds using continuous data replication.

- **Source infrastructure support** – Supports most applications running on x86 architecture (including physical and virtual).
- **Hypervisor support** – Supports any hypervisor (including physical servers, when there is no hypervisor at all).
- **Wide OS support** – AWS supports a large variety of Linux distributions and Windows versions.
- **Environment isolation** – Ability to create the disaster recovery site in an isolated environment so that drills do not impact the source site.
- **Application support** – Atomicity, consistency, isolation, durability (ACID)-compliant applications are supported, including any ACID-compliant database, such as Microsoft SQL Server, Oracle Database, and SAP HANA.
- **Automation** – Ability to fully automate all of the disaster recovery-related operations.
- **Ease of use** – Ability to add the disaster recovery capabilities to working applications with no need for redesign or re-architecture work.

AWS Services Related to Disaster Recovery

AWS can be used as the infrastructure running your disaster recovery site. Additionally, AWS offers a number of services that can help you replicate your source applications into the AWS infrastructure and recover them in the case of a disaster.

AWS Elastic Disaster Recovery

AWS Elastic Disaster Recovery (AWS DRS) is designed for cloud-based disaster recovery of virtual and physical servers. Elastic Disaster Recovery continuously replicates applications and databases from any supported source to AWS using block-level replication of the underlying server. The service allows you to use AWS as a disaster recovery site for on-premises applications comprising servers (physical and virtual), including databases. During normal operations, Elastic Disaster Recovery continuously replicates changes to the source data to a staging area subnet in your AWS account. The staging area design uses affordable storage and minimal compute resources to maintain ongoing replication. When you initiate a failover or drill, the staged resources are used to

automatically create a full-capacity deployment in your Amazon Virtual Private Cloud (VPC), which is used as the disaster recovery site. You can launch recovery instances on AWS within minutes, using the most up-to-date server state or a previous point in time. After the disaster, you can use Elastic Disaster Recovery to fail back to your primary site.

AWS DataSync

AWS DataSync is an online data transfer service that simplifies, automates, and accelerates moving data between on-premises storage systems and AWS storage services, and also between AWS storage services. DataSync can copy data between Network File System (NFS), Server Message Block (SMB) file servers, self-managed object storage, AWS Snowcone, Amazon Simple Storage Service (Amazon S3) buckets, Amazon Elastic File System (Amazon EFS), and Amazon FSx for Windows File Server file systems.

If you have large network-attached storage (NAS) appliances, AWS recommends using AWS DataSync for replicating them to Amazon S3, Amazon S3 Glacier, Amazon EFS, or Amazon FSx for Windows File Server.

Amazon Route 53 Application Recovery Controller

Amazon Route 53 Application Recovery Controller gives you insights into whether your applications and resources are ready for recovery, and helps you manage and coordinate failover using readiness check and routing control features. These features continually monitor your application's ability to recover from failures, and enable you to control your application recovery across multiple AWS Regions, Availability Zones, as well as on premises. These capabilities make application recoveries simpler and more reliable by eliminating the manual steps required by traditional tools and processes.

How to respond in the event of a disaster

The following section explains how to respond in the event of a disaster, including how to perform a failover and how to perform a failback and return to normal operations.

Performing a Failover

An actual failover is very similar to a drill. The two main differences are that during an actual failover, your users will be redirected to the disaster recovery site, and that a failback may be needed when the disaster is over. In the event of planned or unplanned downtime, begin the failover process by using the Elastic Disaster Recovery console to launch recovery instances on AWS from the latest state or a point in time you select.

1. Launch recovery instances for a single source server or multiple source servers. This action is recorded in AWS CloudTrail.
2. Perform the actual failover (directing traffic to your recovery instances) using the tool or the means you use for directing traffic. For a Domain Name System (DNS) redirect, AWS recommends using Amazon Route 53 Application Recovery Controller.
3. After a successful failover, and after the downtime is over, you can prepare for failback.

Performing a Failback and Returning to Normal Operations

After performing a successful failover, verify that any data that was written to your recovery systems is replicated back to your original systems before you perform the actual failback and redirect users to your primary systems. This can be data from changes that occurred while the disaster recovery site was active and that needs to merged back into the source applications, or all the data may need to be copied back in case the data of the source site cannot be recovered after the disaster is over.

To fail back to the source servers (or to new servers, if the original source servers are no longer available after the disaster is over), follow these steps:

1. Start by replicating the data from the disaster recovery site back to the source site (when using Elastic Disaster Recovery, replicate the data from your recovery instances on AWS back to your source servers using the Failback Client).

2. Continue using the disaster recovery site while the data is being replicated. When using Elastic Disaster Recovery, you can track failback replication progress from the console.
3. Choose a failback window that minimizes the impact on users (because during failback a short amount of downtime may occur).
4. During the failback window, go to the disaster recovery site and stop the application.
5. Make sure that all the data finishes replicating to the source site.
6. Start the application in the source site.
7. Make sure the application launches correctly.
8. Once you have verified that the application is running properly, redirect users back to the application in the source site.

Credit to: AWS Documentation