AWS Architecture Blog

Disaster Recovery Solutions with AWS managed services, Part 3: Multi-Site Active/Passive

by Brent Kim and Dhruv Bakshi | on 10 MAR 2023 | in Amazon Elastic Kubernetes Service, Amazon OpenSearch Service, Amazon Route 53, Architecture, AWS Managed Services, Best Practices, RDS for PostgreSQL | Permalink | Share

Welcome to the third post of a multi-part series that addresses disaster recovery (DR) strategies with the use of AWS-managed services to align with customer requirements of performance, cost, and compliance. In part two of this series, we introduced a DR concept that utilizes managed services through a backup and restore strategy with multiple Regions. The post also introduces a multi-site active/passive approach.

The multi-site active/passive approach is best for customers who have business-critical workloads with higher availability requirements over other active/passive environments. A warm-standby strategy (as in Figure 1) is more costly than other active/passive strategies, but provides good protection from downtime and data loss outside of an active/active (A/A) environment.

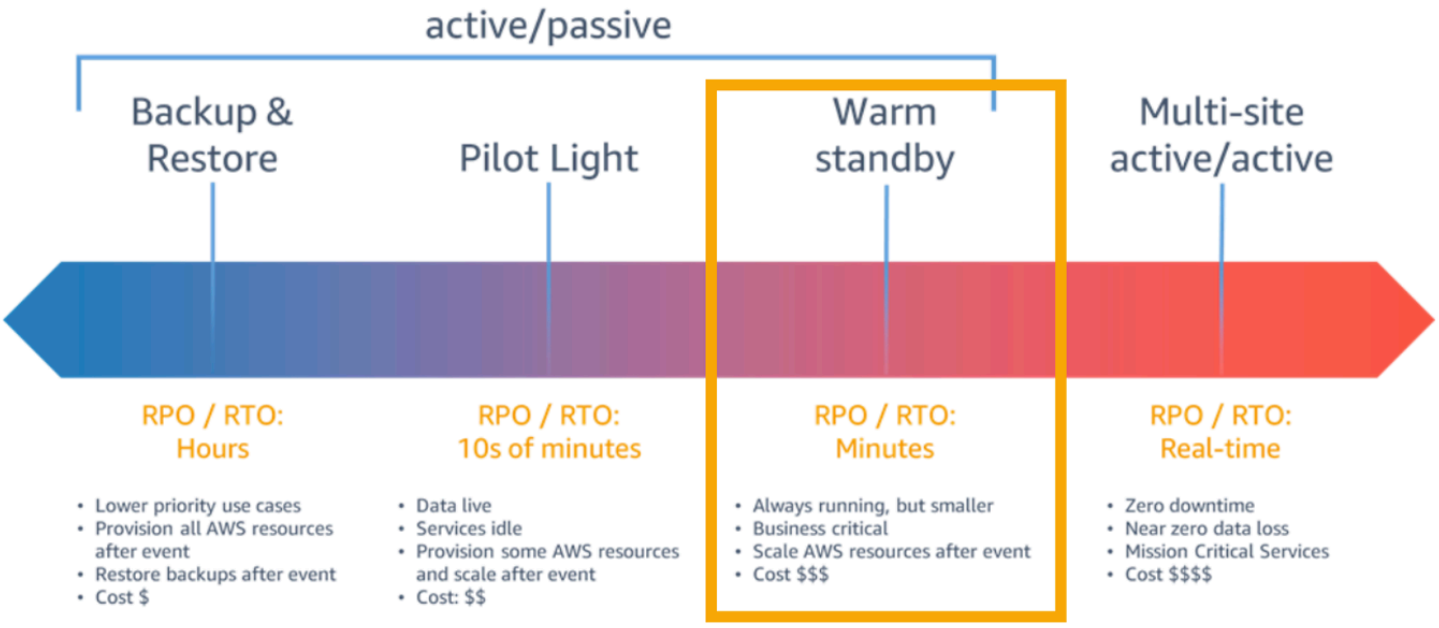

Figure 1. Warm standby

Implementing the multi-site active/passive strategy

By replicating across multiple Availability Zones in same Region, your workloads become resilient to the failure of an entire data center. Using multiple Regions provides the most resilient option to deploy workloads, which safeguards against the risk of failure of multiple data centers.

Let's explore an application that processes payment transactions and is modernized to utilize managed services in the AWS Cloud, as in Figure 2.

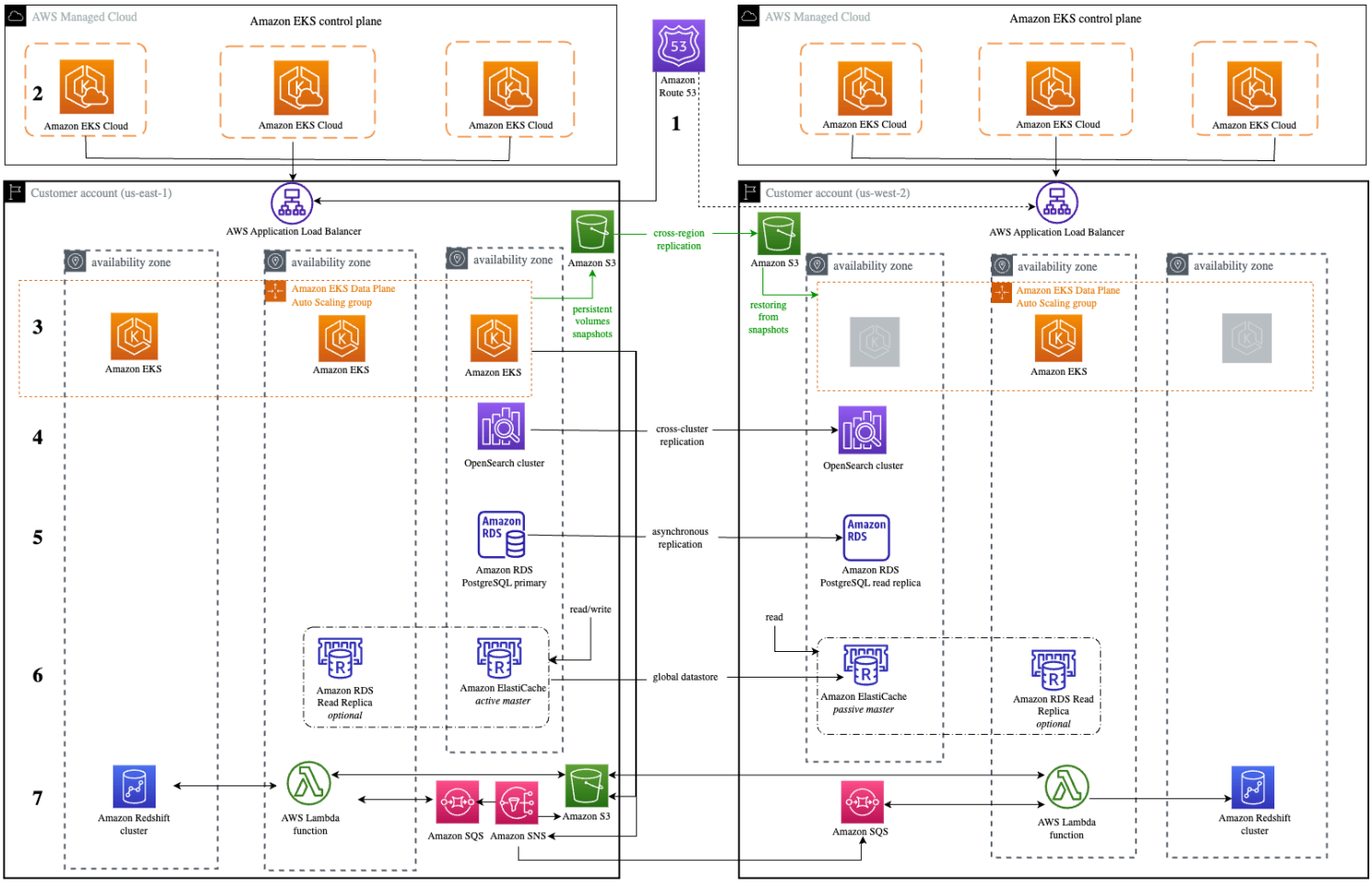

Figure 2. Warm standby with managed services

Let's cover each of the components of this application, as well as how managed services behave in a multisite environment.

1. Amazon Route53 – Active/Passive Failover: This configuration consists of primary resources to be available, and secondary resources on standby in the case of failure of the primary environment. You would just need to create the records and specify failover for the routing policy. When responding to queries, Amazon Route 53 includes only the healthy primary resources. If the primary record configured in the Route 53 health check shows as unhealthy, Route 53 responds to DNS queries using the secondary record.

2. Amazon EKS control plane: Amazon Elastic Kubernetes Service (Amazon EKS) control plane nodes run in an account managed by AWS. Each EKS cluster control plane is single-tenant and unique, and runs on its own set of Amazon Elastic Compute Cloud (Amazon EC2) instances. Amazon EKS is also a Regional service, so each cluster is confined to the Region where it is deployed, with each cluster being a standalone entity.

3. Amazon EKS data plane: Operating highly available and resilient applications requires a highly available and resilient data plane. It's best practice to create worker nodes using Amazon EC2 Auto Scaling groups instead of creating individual Amazon EC2 instances and joining them to the cluster.

Figure 2 shows three nodes in the primary Region while there will only be a single node in the secondary. In case of failover, the data plane scales up to meet the workload requirements. This strategy deploys a functional stack to the secondary Region to test Region readiness before failover. You can use Velero with

Portworx to manage snapshots of persistent volumes. These snapshots can be stored in an Amazon Simple Storage Service (Amazon S3) bucket in the primary Region, which is replicated to an Amazon S3 bucket in another Region using Amazon S3 cross-Region replication.

During an outage in the primary Region, Velero restores volumes from the latest snapshots in the standby cluster.

4. Amazon OpenSearch Service: With cross-cluster replication in Amazon OpenSearch Service, you can replicate indexes, mappings, and metadata from one OpenSearch Service domain to another. The domain follows an active-passive replication model where the follower index (where the data is replicated) pulls data from the leader index. Using cross-cluster replication helps to ensure recovery from disaster events and allows you to replicate data across geographically distant data centers to reduce latency.

Cross-cluster replication is available on domains running Elasticsearch 7.10 or OpenSearch 1.1 or later. Full documentation for cross-cluster replication is available in the OpenSearch documentation.

If you are using any versions prior to Elasticsearch 7.10 or OpenSearch 1.1, refer to part two of our blog series for guidance on using APIs for cross-Region replication.

5. Amazon RDS for PostgreSQL: One of the managed service offerings of Amazon Relational Database Service (Amazon RDS) for PostgreSQL is cross-Region read replicas. Cross-Region read replicas enable you to have a DR solution scaling read database workloads, and cross-Region migration.

Amazon RDS for PostgreSQL supports the ability to create read replicas of a source database (DB). Amazon RDS uses an asynchronous replication method of the DB engine to update the read replica whenever there is a change made on the source DB instance. Although read replicas operate as a DB instance that allows only read-only connections, they can be used to implement a DR solution for your production DB environment. If the source DB instance fails, you can promote your Read Replica to a standalone source server.

Using a cross-Region read replica helps ensure that you get back up and running if you experience a Regional availability issue. For more information on PostgreSQL cross-Region read replicas, visit the Best Practices for Amazon RDS for PostgreSQL Cross-Region Read Replicas blog post.

6. Amazon ElastiCache: AWS provides a native solution called Global Datastore that enables cross-Region replication. By using the Global Datastore for Redis feature, you can work with fully managed, fast, reliable, and secure replication across AWS Regions. This feature helps create cross-Region read replica clusters for ElastiCache for Redis to enable low-latency reads and DR across AWS Regions. Each global datastore is a collection of one or more clusters that replicate to one another. When you create a global datastore in Amazon ElastiCache, ElastiCache for Redis automatically replicates your data from the primary cluster to the secondary cluster. ElastiCache then sets up and manages automatic, asynchronous replication of data between the two clusters.

7. Amazon Redshift: With Amazon Redshift, there are only two ways of deploying a true DR approach: backup and restore, and an (A/A) solution. We'll use the A/A solution as this provides a better recovery time objective

(RTO) for the overall approach. The recovery point objective (RPO) is dependent upon the configured schedule of AWS Lambda functions. The application within the primary Region sends data to both Amazon Simple Notification Service (Amazon SNS) and Amazon S3, and the data is distributed to the Redshift clusters in both Regions through Lambda functions.

Amazon EKS uploads data to an Amazon S3 bucket and publishes a message to an Amazon SNS topic with a reference to the stored S3 object. S3 acts as an intermediate data store for messages beyond the maximum output limit of Amazon SNS. Amazon SNS is configured with primary and secondary Region Amazon Simple Queue Service (Amazon SQS) endpoint subscriptions. Amazon SNS supports the cross-Region delivery of notifications to Amazon SQS queues. Lambda functions deployed in the primary and secondary Region are used to poll the Amazon SQS queue in respective Regions to read the message. The Lambda functions then use the Amazon SQS Extended Client Library for Java to retrieve the Amazon S3 object referenced in the message. Once the Amazon S3 object is retrieved, the Lambda functions upload the data into Amazon Redshift.

For more on how to coordinate large messages across accounts and Regions with Amazon SNS and Amazon SQS, explore the Coordinating Large Messages Across Accounts and Regions with Amazon SNS and SQS blog post.

Conclusion

This active/passive approach covers how you can build a creative DR solution using a mix of native and non-native cross-Region replication methods. By using managed services, this strategy becomes simpler through automation of service updates, deployment using Infrastructure as a Code (IaaC), and general management of the two environments.

Related information

Want to learn more? Explore the following resources within this series and beyond!

- Disaster Recovery with AWS Managed Services, Part 1: Single Region
- Disaster Recovery with AWS Managed Services, Part 2: Multi-Region/Backup and Restore
- Disaster Recovery Series blog posts

TAGS: Disaster recovery, Disaster Recovery with AWS Managed Services series