

Disaster recovery with AWS managed services, Part 2: Multi-Region/backup and restore

by Dhruv Bakshi and Brent Kim | on 23 MAY 2022 | in [Amazon CloudWatch](#), [Amazon Route 53](#), [Amazon Simple Notification Service \(SNS\)](#), [AWS Lambda](#) | [Permalink](#) | [Share](#)

In [part 1](#) of this series, we introduced a disaster recovery (DR) concept that uses managed services through a single AWS Region strategy. In part two, we introduce a multi-Region backup and restore approach. With this approach, you can deploy a DR solution in multiple Regions, but it will be associated with longer [RPO/RTO](#). Using a backup and restore strategy will safeguard applications and data against large-scale events as a cost-effective solution, but will result in longer downtimes and greater loss of data in the event of a disaster as compared to other strategies as shown in Figure 1.

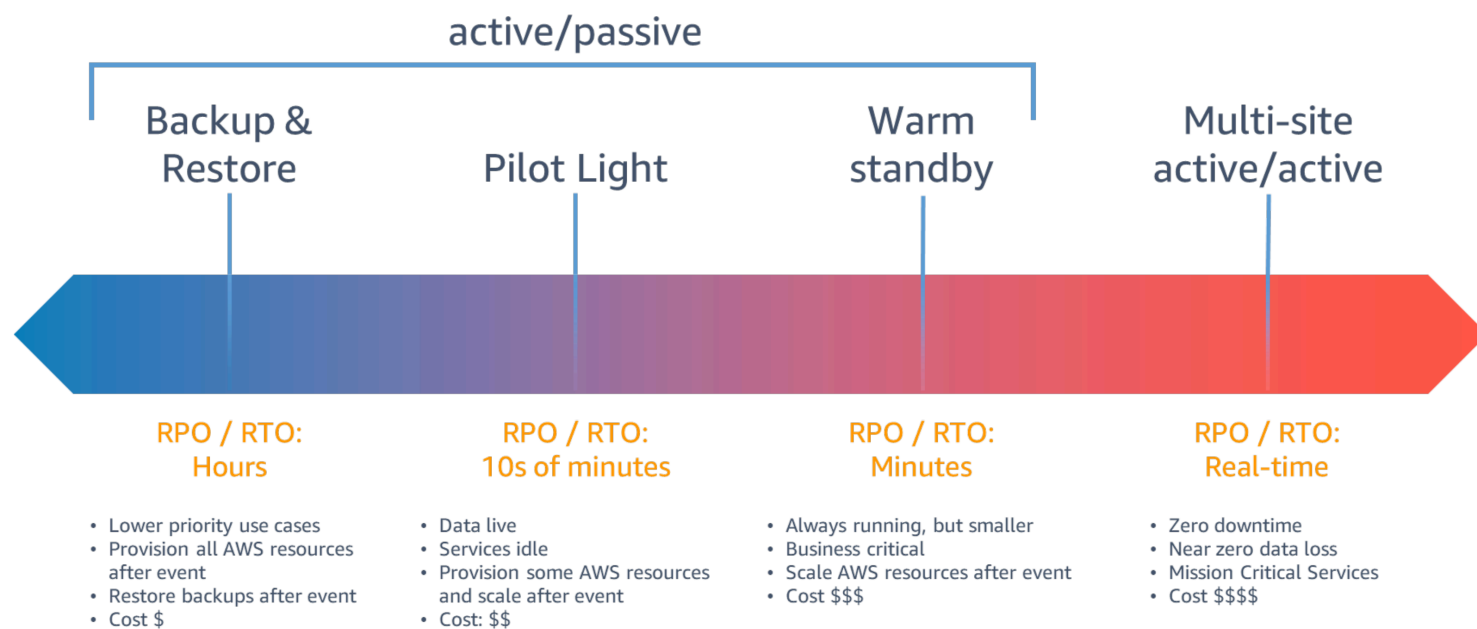


Figure 1. DR Strategies

Implementing the multi-Region/backup and restore strategy

Using multiple Regions ensures resiliency in the most serious, widespread outages. A secondary Region protects workloads against being unable to run within a given Region, because they are wide and geographically dispersed.

Architecture overview

The application diagram presented in Figures 2.1 and 2.2 refers to an application that processes payment transactions, which was modernized to utilize managed services in the AWS Cloud. In this post, we'll show you which AWS services it uses and how they work to maintain multi-Region/backup and restore strategy.

These figures show how to successfully implement the backup and restore strategy and successfully fail over your workload. The following sections list the components of the example application presented in the figures, which works as follows:

- [Amazon Route 53](#) health checks monitor application endpoints
- If the Route 53 health check fails, an [Amazon CloudWatch](#) alarm prompts an [Amazon Simple Notification Service \(Amazon SNS\)](#) topic
- This SNS topic invokes an [AWS Lambda](#) function, which will invoke the infrastructure pipeline to initiate cluster provision in the secondary Region.

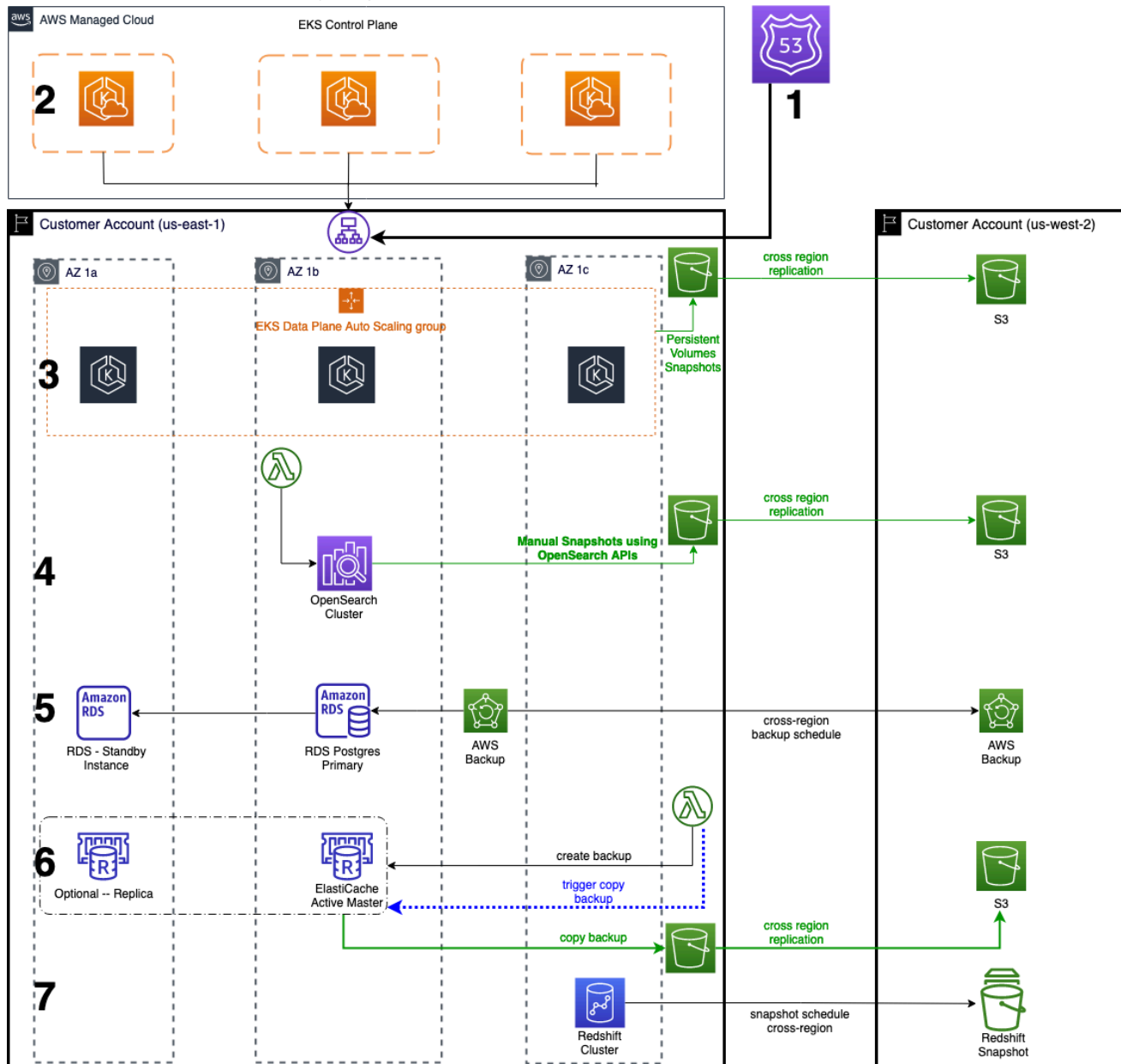


Figure 2.1. Multi-Region backup

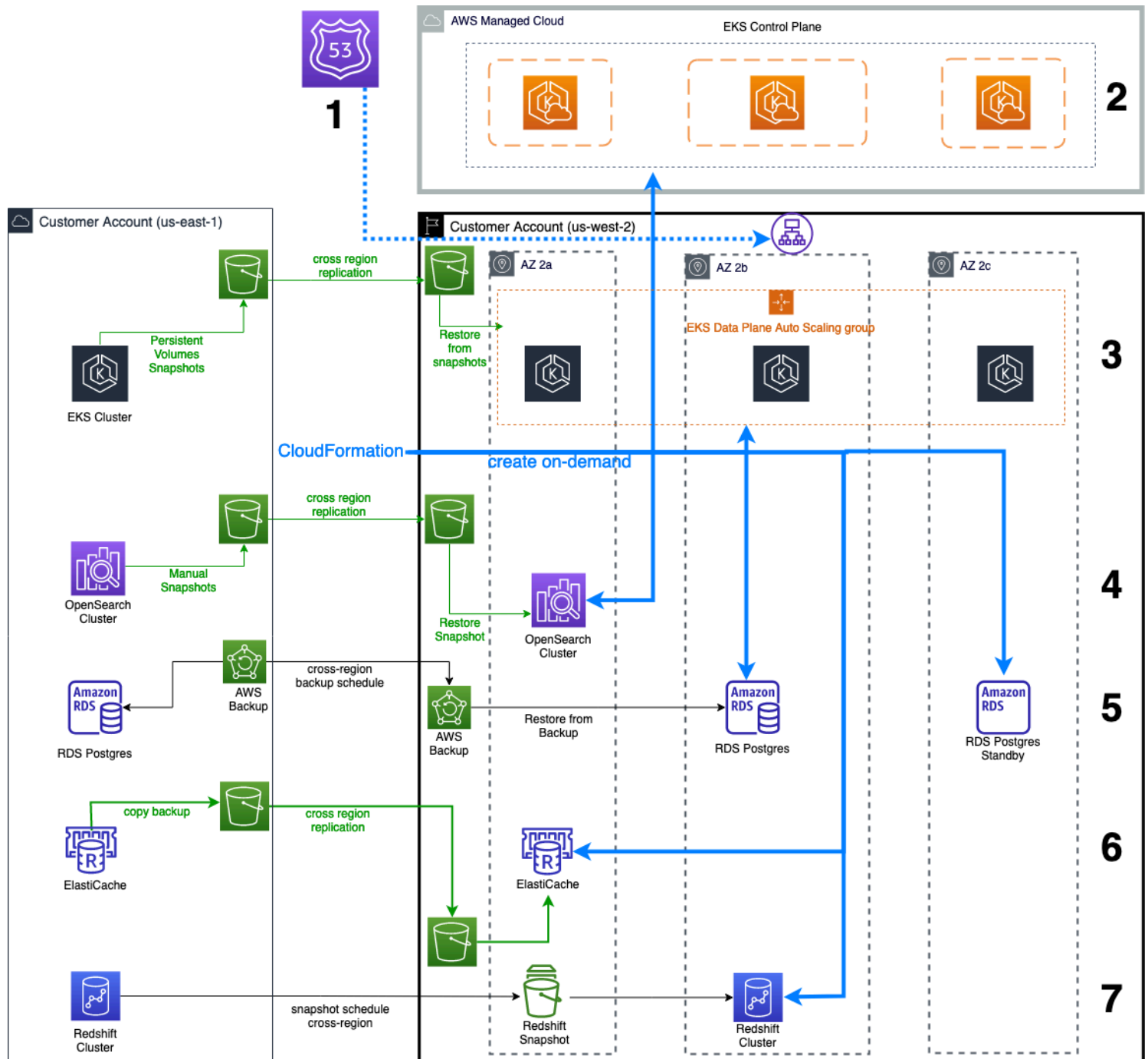


Figure 2.2. Multi-Region restore

1. Route 53

Route 53 health checks monitor the health and performance of your web applications, web servers, and other resources. Health checks are necessary for configuring DNS failover within Route 53. Once an application or resource becomes unhealthy, you'll need to initiate a manual failover process to create resources in the secondary Region. In our architecture, we use CloudWatch alarms to automate notifications of changes in health status.

Please check out the [Creating Disaster Recovery Mechanisms Using Amazon Route 53](#) blog post for additional DR mechanisms using Amazon Route 53.

2. Amazon EKS control plane

[Amazon Elastic Kubernetes Service \(Amazon EKS\)](#) automatically scales control plane instances based on load, automatically detects and replaces unhealthy control plane instances, and restarts them across the Availability Zones within the Region as needed. Because on-demand clusters are provisioned in the secondary Region, AWS also manages the control plane the same way.

3. Amazon EKS data plane

It is a best practice to create worker nodes using [Amazon Elastic Compute Cloud \(Amazon EC2\) Auto Scaling](#) groups instead of creating individual EC2 instances and joining them to the cluster. This is because Amazon EC2 Auto Scaling groups automatically replace any terminated or failed nodes, which ensures that the cluster always has the capacity to run your workload.

The Amazon EKS control plane and data plane will be created on demand in the secondary Region during an outage via Infrastructure-as-a-Code (IaC) such as [AWS CloudFormation](#), Terraform, etc. You should pre-stage all networking requirements like virtual private cloud (VPC), subnets, route tables, gateways and deploy the Amazon EKS cluster during an outage in the primary Region.

As shown in [the Backup and restore your Amazon EKS cluster resources using Velero](#) blog post, you may use a third-party tool like [Velero](#) for managing snapshots of persistent volumes. These snapshots can be stored in an [Amazon Simple Storage Service \(Amazon S3\)](#) bucket in the primary Region, which will be replicated to an S3 bucket in another Region via cross-Region replication.

During an outage in the primary Region, you can use the tool in the secondary Region to restore volumes from snapshots in the standby cluster.

4. OpenSearch Service

For domains running [Amazon OpenSearch Service](#), OpenSearch Service takes hourly automated snapshots and retains up to 336 for 14 days. These snapshots can only be used for cluster recovery within the same Region as the primary OpenSearch cluster.

You can use OpenSearch APIs to create a manual snapshot of an OpenSearch cluster, which can be stored in a registered repository like Amazon S3. You can do this manually or create a scheduled Lambda function based on their RPO, which prompts creation of a manual snapshot that will be stored in an S3 bucket. Amazon S3 cross-Region replication will then automatically and asynchronously copy objects across S3 buckets.

You can restore OpenSearch Service clusters by creating the cluster on demand via CloudFormation and using OpenSearch APIs to restore the snapshot from an S3 bucket.

5. Amazon RDS Postgres

[Amazon Relational Database Service \(Amazon RDS\)](#) can copy continuous backups cross-Region. You can configure your Amazon RDS database instance to replicate snapshots and transaction logs to a destination Region of your choice.

If a continuous backup rule also specifies a cross-account or cross-Region copy, [AWS Backup](#) takes a snapshot of the continuous backup, copies that snapshot to the destination vault, and then deletes the source snapshot. For continuous backup of Amazon RDS, AWS Backup creates a snapshot every 24 hours and stores transaction logs every 5 minutes in-Region. The Backup Frequency setting only applies to cross-Region backups of these continuous backups. Backup Frequency determines how often AWS Backup:

- Creates a snapshot at that point in time from the existing snapshot plus all transaction logs up to that point
- Copies snapshots to the other Region(s)
- Deletes snapshots (because it only was created to be copied)

For more information, refer to the [Point-in-time recovery and continuous backup for Amazon RDS with AWS Backup](#) blog post.

6. ElastiCache

You can export and import backup and copy API calls for [Amazon ElastiCache](#) to develop a snapshot and restore strategy in a secondary Region. You can either prompt a manual backup and copy of that backup to S3 bucket or create a pair of Lambda functions to run at a schedule to meet the RPO requirements. The Lambda functions will prompt a manual backup, which creates a .rdb to an S3 bucket. Amazon S3 cross-Region replication will then handle asynchronous copy of the backup to an S3 bucket in a secondary Region.

You can use CloudFormation to create an ElastiCache cluster on demand and use CloudFormation properties such as [SnapshotArns and SnapshotName](#) to point to the desired ElastiCache backup stored in Amazon S3 to seed the cluster in the secondary Region.

7. Amazon Redshift

[Amazon Redshift](#) takes automatic, incremental snapshots of your data periodically and saves them to Amazon S3. Additionally, you can take manual snapshots of your data whenever you want.

To precisely control when snapshots are taken, you can create a snapshot schedule and attach it to one or more clusters. You can also configure [cross-Region snapshot copy](#), which will automatically copy all your automated and manual snapshots to another Region.

During an outage, you can create the Amazon Redshift cluster on demand via CloudFormation and use CloudFormation properties such as [SnapshotIdentifier](#) to restore the new cluster from that snapshot.

Note: You can add an additional layer of protection to your backups through [AWS Backup Vault Lock](#), [S3 Object Lock](#), and [Encrypted Backups](#).

Conclusion

With greater adoption of managed services within the cloud, there is a need to think of creative ways to implement a cost-effective DR solution. This backup and restore approach offered in this post will lower costs through more

lenient RPO/RTO requirements, while providing a solution to utilize AWS managed services.

In the next post, we will discuss a multi-Region active/active strategy for the same application stack illustrated in this post.

Other posts in this series

- [Disaster Recovery with AWS Managed Services, Part I: Single Region](#)

Related information

- [Disaster Recovery \(DR\) Architecture on AWS, Part II: Backup and Restore with Rapid Recovery](#)

Looking for more architecture content? [AWS Architecture Center](#) provides reference architecture diagrams, vetted architecture solutions, [Well-Architected](#) best practices, patterns, icons, and more!

TAGS: [Disaster recovery](#), [Disaster Recovery with AWS Managed Services series](#)