Disaster Recovery using Hybrid Pilot Light - Active/Active And Placement Strategy

MSc Research Project
Cloud Computing

Sachin Dhanaji Ingale
Student ID: 22144528

School of Computing
National College of Ireland

Supervisor: Ahmed Makki

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Sachin Dhanaji Ingale
Student ID:	22144528
Programme:	MSc in Cloud Computing
Year:	2023
Module:	MSc Research Project
Supervisor:	Ahmed Makki
Submission Due Date:	14/12/2023
Project Title:	Disaster Recovery using Hybrid Pilot Light - Active/Active And Placement Strategy
Word Count:	6209
Page Count:	22

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	
Date:	14th December 2023

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Disaster Recovery using Hybrid Pilot Light - Active/Active And Placement Strategy

Sachin Dhanaji Ingale
22144528

Abstract

This research describes a unique hybrid pilot light and active-active disaster recovery technique for ensuring the robustness and high availability of AWS-deployed workloads. Fueled by the critical need for strong disaster recovery solutions in the face of unplanned disasters, the report investigates the development and analysis of a hybrid system comprising the primary as well as the secondary regions. The architecture makes use of Amazon Web Services (AWS) CloudFormation, RDS, Elastic Beanstalk, and S3 to create two different environments, each with its own Virtual Private Cloud (VPC). The primary region acts as a production environment, whereas the secondary region acts as a failover backup. This research paper thoroughly describes the methods for provision of both primary as well as secondary environments, pointing out the need for already configured CloudFormation templates. Two evaluations, based on the Recovery Time Objective (RTO) and the Recovery Point Objective (RPO), demonstrate the infrastructure's ability to fulfill the targeted recovery objectives. The secondary region has a significantly lower RTO than the primary, indicating the higher fault tolerance of the proposed hybrid method.

Table of Contents

1 Introduction

Disaster recovery planning is critical for ensuring the continuity and resilience of modern cloud infrastructures. The necessity for strong disaster recovery techniques becomes more obvious as organizations increasingly rely on cloud-based systems to power their important applications. The constantly evolving nature of possible interruptions, ranging from natural catastrophes to cyber attacks, needs a thorough examination of innovative strategies for minimizing downtime and ensuring rapid recovery. This study conducts a thorough assessment of the hybrid disaster recovery technique, concentrating on the combination of the pilot light and active/active strategies. As organizations battle with the problems provided by traditional recovery approaches in terms of RTO and capital commitments, a hybrid strategy appears as a possible alternative. This technique aims for maximum fault tolerance by maintaining two separate environments, an active primary and a secondary standby, minimizing the effect of unexpected disasters. Figure 1 shows the high level disaster recovery overview.

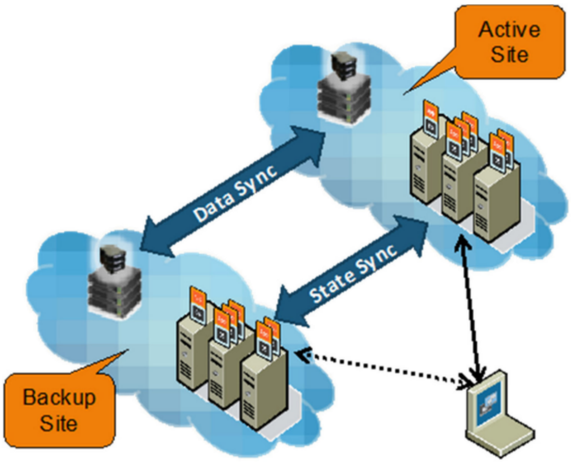


Figure 1: Disaster Recovery Overview Tomás et al. (2020)

1.1 Motivation

The potential impact of interruptions on corporate operations has attracted increasing attention as organizations transfer their applications to cloud environments. Cloud service outages, natural disasters, and cyber assaults may all cause considerable downtime, resulting in economic losses and reputational harm. The need for robust disaster recovery solutions has never been greater.

Motivated by the need to improve disaster recovery quality, this research project digs into the complicated workings of a hybrid disaster recovery strategy designed for cloud-based applications. Using insights from current literature on disaster recovery techniques in cloud computing, the author highlights the gaps and issues that drive the need for a hybrid strategy. The key research aim of this study is to evaluate the efficiency of the hybrid pilot light and active/active method in minimizing recovery time and maximizing fault tolerance in cloud based systems.

1.2 Research Question

Mentioned Below is the author's research question that needs to be accomplished and evaluated:

What would be the impact of using hybrid pilot light - active/active and placement strategy disaster recovery approach on RPO, RTO, Latency and Cost?

Objectives:

1. To work towards achieving better RTO and RPO for a production level environment.
2. Provisioning secure cloud environments using AWS CloudFormation.
3. To cut down on the cost involved using the hybrid approach.

1.3 Contribution to Scientific Literature

This research paper adds to the scientific literature by proposing and analysing an improved disaster recovery technique that is compatible with the dynamic nature of cloud computing. The hybrid pilot light and active/active method provides a sophisticated solution that capitalises on the benefits of both techniques to optimise resource use and accelerate recovery processes.

On the other hand the research attempts to bridge the gap between conceptual disaster recovery frameworks and practical implementation by offering specific insights into disaster recovery plan formulation and execution. This paper aims to build a benchmark for effective disaster recovery in cloud-based applications by analyzing the suggested technique using performance measures such as Recovery Time Objective and Recovery Point Objective.

1.4 Report Structure

The rest of this paper is structured as mentioned here: Section 2 provides a detailed literature review of existing disaster recovery strategies and the limitations they pose. Section 3 delves into the methodology adopted for the hybrid pilot light and active/active approach, Section 4 explain the proposed design specification for hybrid model. Section 5 details the implementation of the proposed strategy, offering insights into the primary and secondary environments. Section 6 evaluates the performance of the proposed disaster recovery plan through measures such as RTO and RPO and also discusses the results and proposes potential improvements. At the end, Section 7 concludes the research paper with a summary of the major results and outlines areas for future research and enhancement of disaster recovery strategies.

2 Related Work

This section contains information related to the research papers that helped the author of this paper gain knowledge about cloud disaster recovery which indirectly helped in conducting this research. The related work mostly contains details regarding disaster recovery strategies and technologies, advanced techniques and models in disaster recovery, and also about security, automation and future trends in disaster recovery. The goal was to find gaps and look for future work from previous research in order to achieve a disaster recovery plan that has an exceptionally better recovery time objective, recovery point objective and cost involved.

2.1 Disaster Recovery Strategies and Technologies

The research done by Alhazmi and Malaiya (2012) investigates the tradeoffs of onsite, colocation, and cloud disaster recovery alternatives. It employs analytical methodologies to assist future disaster recovery planning and maintenance. The authors propose that chief information officers use a quantitative method to compare DRP systems and choose the best one. They recommend that organizations detect probable catastrophic occurrences and assess their impact. Their study's findings suggest that employing trustworthy quantitative indicators, decision makers may be objective in feasibility analysis and business needs research. More data is still required for mathematical optimization models. The quantitative analysis and practical effects of the work are among its strengths.

Chen and Shang (2017) looks into disaster recovery technologies for cloud-based online systems, emphasising the need of having a thorough, multi-level backup plan in place to safeguard system security and ensure quick data recovery. To collect information from academic publications and books, the research used a literature review technique. Backup techniques, recovering transactional logs, pages, files, and record groups, and information recovery exercises are all included in the research. It emphasises the significance of recovering data and fault drills in a timely manner. The report includes a complete review of disaster recovery technologies for cloud-based systems, as well as helpful tips for building a backup plan and restoring data in the event of a disaster. Its shortcomings, however, include an absence of scientific data and case studies, which means that its conclusions may not be relevant to all systems that use clouds.

To avoid single point failures, Sabbaghi et al. (2017) recommends a cloud disaster recovery model with network redundancy as a corporate contingency plan. To collect data from multiple institutions, the research adopts a quantitative approach that includes surveys and interviews. SPSS statistics software is used to assess and validate the framework. The findings of author indicate that the suggested framework is economical and provides an excellent chance for small and medium-sized enterprises (SMEs) to establish disaster recovery. Along with that the study emphasizes the significance of disaster recovery in the domain of cloud computing. Still the limited sample size and absence of extensive cost analysis of the framework's implementation may restrict its generalizability.

As per the research done by Mendonça et al. (2018) the author of the paper evaluates cloud-based disaster recovery systems using an integrated model-experiment method, including stochastic petri nets and fault-injection experiments to measure availability metrics such as steady-state availability and downtime. The findings of the author suggests that using a disaster recovery solution considerably boosts system availability and reduces downtime costs. The method's strengths include its application of modelling and testing, which results in more precise and trustworthy outcomes. It also analyses varied failure/repair behaviours and component dependability. However, it may not be appropriate to all disaster recovery systems or IT environments and may need substantial time and money to execute.

Shahzadi et al. (2018) talks about the goal to provide a self-contained, frictionless, and robust carrier cloud brokerage solution for disaster recovery. It uses a proof-of-concept method and tests its performance for each cloud service using use-case scenarios. The solution allows for the seamless migration of a full IaaS, lowers capital cost, and ensures dependable data access during catastrophic events. The paper covers the challenge of constructing robust live/real-time catastrophe recovery methods and evaluates performance using use-case scenarios. Unfortunately, it is restricted to a single organization's private cloud and needs real-world testing to confirm its usefulness. The study's strength comes from its attention on crucial challenges in disaster recovery.

The research performed by Baginda et al. (2018) analyses the RTO and RPO parameters of a service available on AWS and GCE by using a disaster recovery evaluation strategy, disaster recovery actions, and system and application testing. According to the findings, AWS offers a lower RTO and RPO versus GCE, making it a better alternative for businesses that require high application availability. The research offers a thorough comparison of RTO and RPO parameters between two major cloud providers, as well as an implementation design strategy for businesses to evaluate overall disaster recovery strategies. The article only reviews two providers and does not give a pricing analysis, which may be an essential consideration for businesses when selecting a cloud provider.

2.2 Advanced Techniques and Models in Disaster Recovery

Utilizing an integrated model-experiment method, Mendonça et al. (2019) assesses a Backup-as-a-Service (BaaS) setting for disaster recovery (DR) instances. Key parameters including availability, downtime, RTO, and RPO are examined in a real-world BaaS context using analytic models as well as fault-injection experiments. The study discovers that backup interval, mean time to recover from a catastrophe, and mean time to disaster for the primary data center are the most relevant criteria for RPO, RTO, and availability. The study's advantages include the application of analytical models and fault-injection tests for a thorough evaluation of DR key-metrics. Whereas limitations include the controlled setting, failures that were not included in the disaster monitor, and the research only examining one type of DR solution, which may not adequately reflect real-world conditions.

Tamimi et al. (2019) investigates several disaster recovery approaches in cloud computing and also compared and evaluated them using his previous literature review results. The author analyzed research papers, journals, and conference papers to determine their benefits and drawbacks, and then evaluated them based on points such as cost savings, data duplication, and security perspective. According to the survey, disaster recovery is becoming a vital part of organizations, and deploying it as a service can mitigate calamities and recover data at a reasonable cost. Some approaches, such as Parity Cloud Service, are very secure and efficient, while others, such as Cold Backup Service, offer longer recovery times and are more cost-effective. The study's strength is in providing a thorough review of various methodologies, emphasizing their relevance in organizations, and providing insights on how to execute disaster recovery as a service. It lacks in rigorous cost-benefit analysis and actual data to back up its assertions.

On the other hand Patel and Keerthana (2019) emphasises the importance of disaster recovery across business continuity management and provides suggestions for organisations to protect data security and privacy. They used a qualitative research technique to identify typical reasons of data loss, including hardware malfunction, human mistake, and cyber threats, and suggests that cloud computing be used to assist disaster recovery efforts. To secure stored data, the authors recommends using data encryption and access limitations. While it gives a thorough overview, it lacked original research and data analysis, as well as an in-depth evaluation of the costs and advantages of various disaster recovery solutions, which might be valuable for organisations formulating data backup and recovery plans.

Tsubaki et al. (2020) talks about ways to reduce data loss from natural catastrophes and network congestion, this paper provides a disaster recovery strategy for edge computing based on distributed TCOs. It compares traditional approaches to disaster prediction data to choose TCOs for data backup. According to the authors findings, the proposed strategy is more successful in reducing the overall number of hops for data backup. It does not, despite this give real-world implementation outcomes and only focuses on natural calamities strengthening is its usefulness and thoroughness.

Abualkishik et al. (2020) supervised the research who's purpose is to examine prior studies on disaster recovery in cloud computing, assessing their strengths and flaws, current data recovery issues, and current developments in the industry. It provides a detailed overview of the area and suggests future research prospects using his literature review studies. The comprehensive review technique and in-depth examination of present and future developments are among the study's strengths. It does have shortcomings, such as a concentration on past research and a lack of actual evidence to back up its claims.

2.3 Security, Automation, and Future Trends in Disaster Recovery

Emejeamara (2020) conducted a research whose goal was to solve security problems in multi-cloud technologies while also providing effective techniques for controlling automation and monitoring. It looked at threat detection, intrusion detection, data security, and vendor lock-in. The authors investigated numerous automation and monitoring recommendations, such as multi-cloud monitoring in cyber physical applications, smart domotics, slipstream automation in regulating large volume data infrastructure, disaster recovery as well as secret protection. The study's strengths include a detailed examination of security risks as well as an investigation of automation and monitoring ideas. It has some drawbacks, including a lack of concrete proof and a limited reach.

Yu et al. (2022) directed a rigorous research whose goal is to create and deploy a software disaster recovery solution for cloud computing-based aircraft ground systems. The service attempts to increase corporate security and continuity in the case of a calamity. The service operates in main and standby modes, combining data centre resources based on a software warehouse. Extensive simulation trials are used to assess the service's performance. The results suggest that the service is successful at detecting software flaws, guaranteeing business continuity, and addressing them in a timely manner. The service also increases the overall security of the cloud infrastructure. The report emphasises the advantages of providing a reliable disaster recovery solution for both data and software in cloud data centres. Extensive simulation tests and addressing the weaknesses of present disaster recovery approaches in aircraft ground systems are among the study's highlights. The research does not give a thorough examination of the service's limits and scalability in bigger systems.

Trovato et al. (2019) supervised a research that looks at several disaster recovery alternatives, such as cloud-based, co-location, onsite, and hybrid options. It evaluates these possibilities, which include public or private clouds DR, DR as a service (DRaaS), and hybrid models, using previous case studies and a structured manner. According to the author, there is no one size fits all disaster recovery solution, and organizations must examine their individual business environment as well as requirements before selecting a solution. The research provides advice on selecting the optimum solution according to cost, compatibility, and organizational constraints. The drawback of this study is that it still remains dependent on secondary data and lacks a full technical study.

Stamenkov (2022) presented a tiered business continuity and disaster recovery (BC/DR) approach to combine diverse BC/DR strategies. It analyses similarities and contrasts between four types of plans which are vital infrastructure safeguarding plan, DR plan, system contingency plan, and business continuity plan. The model is built on a qualitative research strategy, especially using previous case studies, and it analyses four plans provided by a Macedonian government agency. According to the author, the layered BC/DR approach can assist organisations simplify strategies as well as enhance their BC/DR capabilities. The research's advantages include its focus on real difficulties encountered by organisations.On the other hand one of the study's shortcomings is its restricted emphasis on four kinds of plans from a single organisation, as well as its lack of quantitative data. Despite these constraints, the study offers useful insights into BC/DR planning and emphasises the significance of combining several strategies.

Solis et al. (2021) proposed risk mitigation and cloud based disaster recovery strategy for small and medium-sized organisations. It has two parts: Qubes OS and the capacity to restore baseline or application images from the cloud. The Qubes OS component enables users to maintain templates as well as application machines based on templates, and the cloud-based component enables users to acquire and restore systems from the cloud, having recovery time varying according to circumstances. The framework provides various enhancement options, including the use of an SSH command, the implementation of an FTP server as a storage place, and the scripting of backups. It also allows you to deal with Linux Shell and Amazon AWS with ease. The research lacks a full examination of its limits, as well as a cost analysis, which might be valuable for organisations thinking about applying the framework.

3 Methodology

This section of the research paper consists of data relating to the methodology of the disaster recovery approach. After reading all the research papers carefully, I came across all the standard disaster recovery strategies out there, i.e. backup and restore strategy, pilot light strategy, warm standby strategy, and active/active strategy, as shown in Figure 2. Of course, all of the advantages we get from standard approaches are not enough in terms of recovery time objectives and capital involved these days. Thus a hybrid pilot light and active-active strategy come into play, as discussed in Trovato et al. (2019).

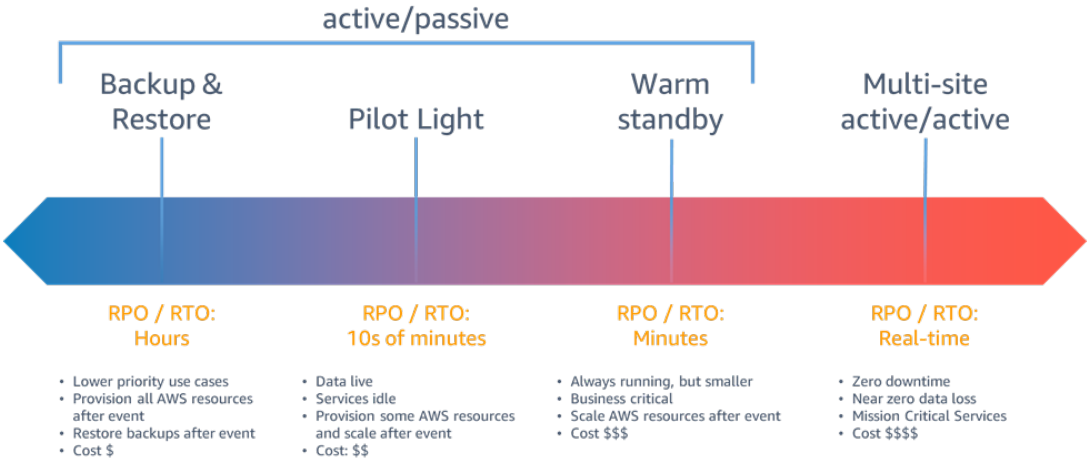

active/passive
Backup & Restore
Pilot Light
Warm standby
Multi-site active/active
RPO / RTO: Hours
RPO / RTO: 10s of minutes
RPO / RTO: Minutes
RPO / RTO: Real-time
• Lower priority use cases
• Provision all AWS resources after event
• Restore backups after event
• Cost $
• Data live
• Services idle
• Provision some AWS resources and scale after event
• Cost: $$
• Always running, but smaller
• Business critical
• Scale AWS resources after event
• Cost $$$
• Zero downtime
• Near zero data loss
• Mission Critical Services
• Cost $$$$

Figure 2: Standard Disaster Recovery Strategies [1]

3.1 Hybrid Approach

The hybrid approach consists of maintaining two separate environments, namely an active or primary environment and another secondary environment in different region for maximum fault tolerance. As the name suggests, the active environment will have all the resources in an active state, as it's a primary environment. The secondary environment will maintain all the heavy services, which take the most time to be created in an active state. This approach won't keep the resources in a standby state rather, these resources will stay fully active. In case of a disaster, if our primary region's both availability zones goes down make use of our secondary region's AWS CloudFormation templates to allocate the remaining resources so that the application is again back to a fully functional state. Figure 3 shows the proposed hybrid DR flow and its components.

3.2 Disaster Recovery Plan

Most of the research papers studied included strategies for creating a more reliant disaster recovery plan. All those combined strategies studied helped in identifying a series of steps for a hybrid approach. In the case of the hybrid approach creating a disaster recovery plan (DRP) for a customer's application on the cloud involves several phases, which are data collection, developing AWS CloudFormation templates, replication procedures, downtime analysis, DR execution steps, and other considerations.

[1] `https://shorturl.at/drAF8`

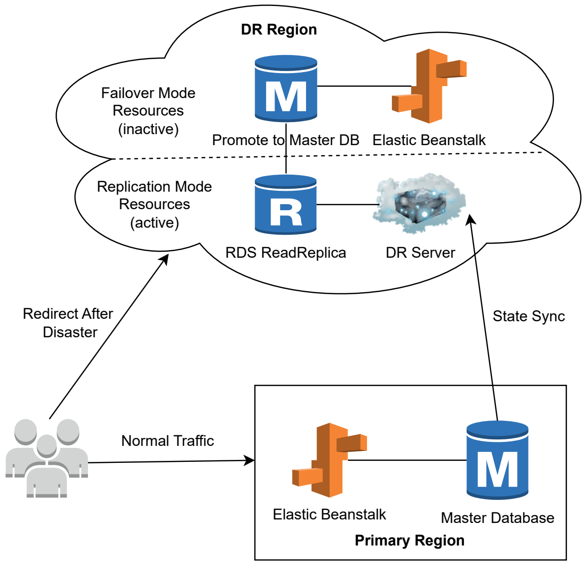


Figure 3: Hybrid DR Flow

3.2.1 Data Collection

The primary aim in data collecting is to have a thorough understanding about the customer's application. Going beyond specific components such as databases and servers helped to decipher the complicated network of connections and links that include the application's architecture. Each component's importance is determined by its involvement in company operations. Also knowing about the specifics like the volume of data, the frequency with which it is updated, the type of resource, and any unique rules or restrictions that must be followed. obtaining vital insights for the adaptation of hybrid technique by studying preexisting disaster recovery plans, if they exist. This deep understanding serves as the foundation for developing a disaster recovery strategy customized to the application's specific demands and complexities Prabantoro and Aji (2021).

3.2.2 Developing AWS CloudFormation Templates

This step involves progressing from a theoretical understanding to practical plans during the development of AWS CloudFormation templates. These templates serves as thorough blueprints for the application's complete infrastructure. From networking setup to data storage as well as computing resources, everything is provisioned using code as AWS CloudFormation is an Infrastructure as a Service (IaaS). AWS CloudFormation allows to reuse prior templates in present models Kartheeyayini et al. (2022). The software also allows to alter prior templates that are appropriate for the current stack. The hybrid approach is smoothly implemented inside these templates, resulting in an active environment maintaining a state of availability for all resources in the primary region whereas in the secondary region only the resources that take the most time to be provisioned are kept active and other resources are brought into action using separate templates whenever there is a need. Thorough testing is required to assure the reliability and effectiveness of these templates, validating their ability to quickly and seamlessly recreate the needed infrastructure during a disaster.

3.2.3 Database replication procedures

A comprehensive method is used in the RDS (Relational Database Service) replication processes to assure data consistency and availability across multiple regions. As illustrated in Figure 4 the primary region consists of a master database along with a read replica, both of which have been designed with multi-AZ (Availability Zone) capabilities for increased fault tolerance. On top of that, snapshots have been turned on for enabling the most recent and accurate data backups. The primary region's master database acts as the main source of data, whereas the read replica, stored inside the same region, serves as a secondary copy. The multi-AZ setup ensures that if one Availability Zone fails, the other zone automatically takes over, minimising downtime and improving overall system resilience. The RDS configuration in the secondary region is identical to that in the primary region, which consists of a cross-region read replica pointing to its master database in the primary region. This configuration allows for continuous data replication from the primary region to the secondary region, guaranteeing the information is always up to date. The read replica in the secondary region is critical for keeping a real-time copy of the master database via the primary region. This replication procedure guarantees that what is stored in the secondary region is an exact and current replica of the master database in the main region. Along with this snapshots are produced in both the primary as well as the secondary regions for recording the present status of the databases.

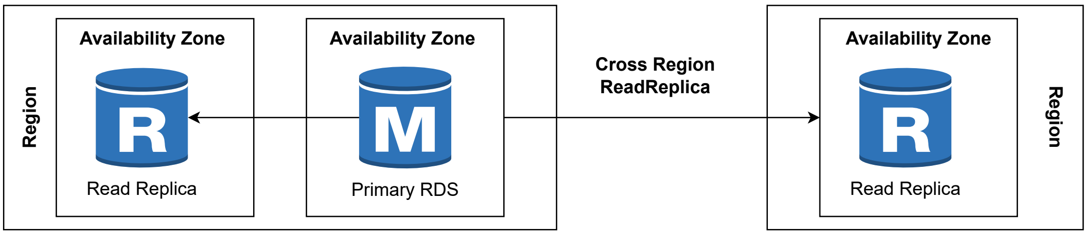


Figure 4: RDS Cross Region Replication

This snapshot feature provides a point-in-time recovering option and adds to the overall disaster recovery plan by allowing the restoration of databases to a specified state in the event of unanticipated catastrophes Paul (2023).

3.2.4 Downtime & Recovery Analysis

Downtime analysis investigates the impact of possible disruptions on corporate activities during disaster recovery, moving beyond just measuring downtime to consider elements such as data transfer times as well as application startup durations. Two critical metrics are taken into account: Recovery Time Objective (RTO), which specifies the maximum time required for system restoration, and Recovery Point Objective (RPO), which specifies the bearable data loss. Careful prioritisation of important parts correlates with both RTO and RPO goals. By establishing effective communication channels, stakeholders and users should be informed about scheduled downtime, promoting transparency, and managing expectations during the recovery process. This phase acknowledges the importance of addressing technological and human-centric components together before implementing DR steps Baginda et al. (2018).

4 Design Specification

The suggested system design shown in Figure 5 attempts to create a resilient as well as highly available infrastructure to serve an AWS-deployed application by utilizing a hybrid disaster recovery method. The architecture is divided into two distinct regions, primary and secondary, each having its own Virtual Private Cloud (VPC). The primary region serves as the current production environment, whereas the secondary region serves as a backup for disaster recovery. Amazon S3 is used for CloudFormation templates and application artifacts, AWS Elastic Beanstalk is used for simplified deployment of the application with a load balancer, and Amazon RDS is setup with multi-AZ along with read replicas for database resilience. Both regions maintain independent VPCs in the hybrid disaster recovery arrangement, each with its own Amazon S3 for reliable storage of CloudFormation templates as well as application artifacts. Except for Elastic Beanstalk, which is available for immediate activation following a disaster, the secondary environment carefully preserves the active condition of all resources. Other than that the secondary environment runs an active ReadReplica of Amazon RDS from the primary region enabling continuous database synchronisation Tomás et al. (2017).

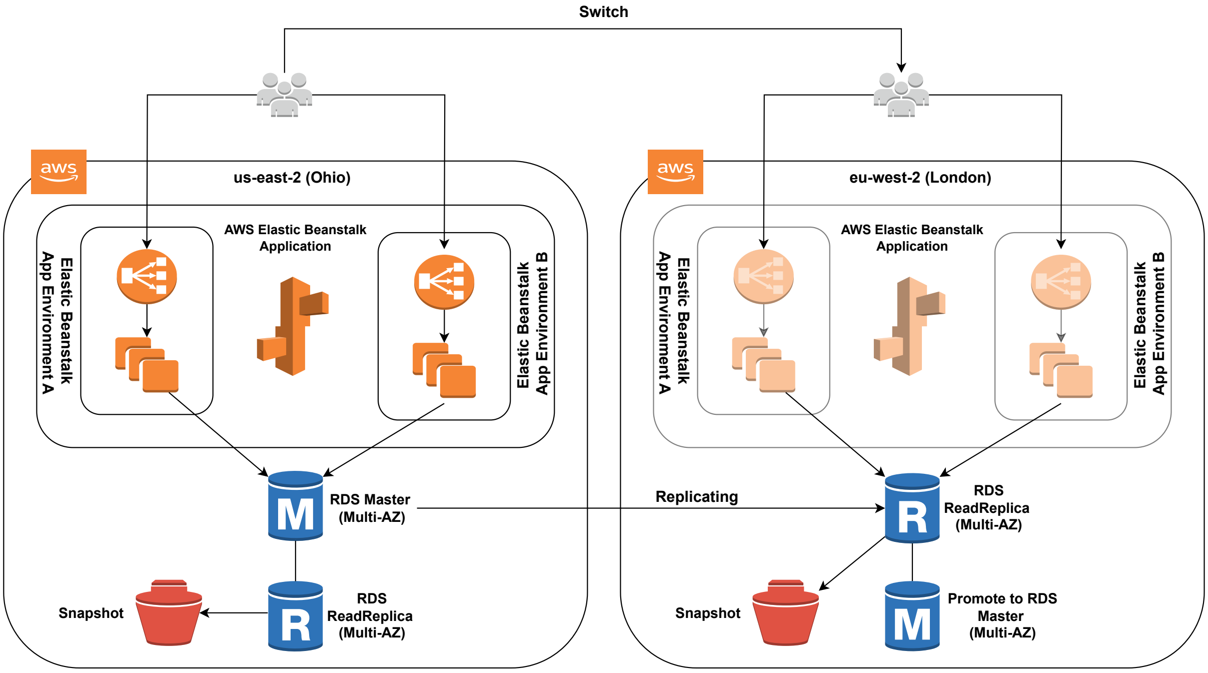
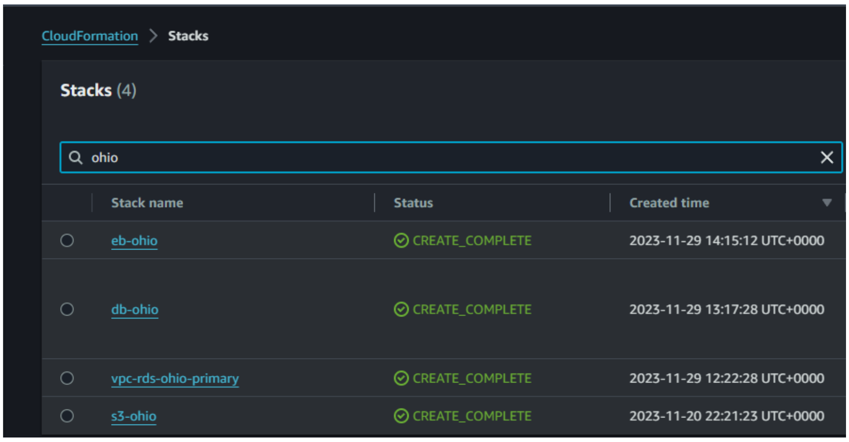

Switch
aws
us-east-2 (Ohio)
AWS Elastic Beanstalk Application
Elastic Beanstalk App Environment A
Elastic Beanstalk App Environment B
RDS Master (Multi-AZ)
Snapshot
RDS ReadReplica (Multi-AZ)
Replicating
aws
eu-west-2 (London)
AWS Elastic Beanstalk Application
Elastic Beanstalk App Environment A
Elastic Beanstalk App Environment B
RDS ReadReplica (Multi-AZ)
Snapshot
Promote to RDS Master (Multi-AZ)

Figure 5: Infrastructure Diagram for Primary and Secondary Environment

4.1 Primary Environment

For research purpose, the author has used us-east-2 (Ohio) as the primary region. All the AWS infrastructure is built using an IaaC service called AWS CloudFormation. In the primary region, AWS S3 is used to store the CloudFormation templates and the application war file for AWS Elastic Beanstalk application deployment. The primary region consists of one VPC which contains four subnets in total of which two are public and two are private. The primary region also contains an AWS RDS instance which is Multi-AZ enabled along with its own ReadReplica which is also Multi-AZ enabled. The RDS instance is configured to create snapshots daily with a retention of one day, which is economical as well. To deploy the application, the author has used AWS Elastic Beanstalk to support automatic deployment, load balancing, and auto scaling. To access the primary environment, AWS Elastic Beanstalks environment domain is used. Figure 6 shows the stacks created in AWS CloudFormation to provision the primary environment.



Stacks (4)

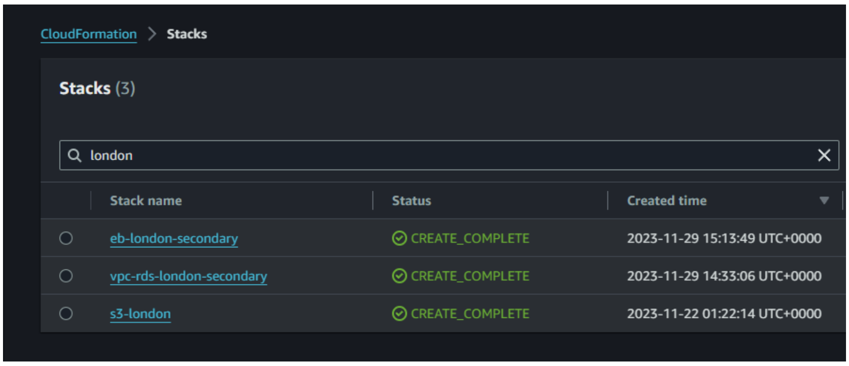
ohio

	Stack name	Status	Created time
○	eb-ohio	CREATE_COMPLETE	2023-11-29 14:15:12 UTC+0000
○	db-ohio	CREATE_COMPLETE	2023-11-29 13:17:28 UTC+0000
○	vpc-rds-ohio-primary	CREATE_COMPLETE	2023-11-29 12:22:28 UTC+0000
○	s3-ohio	CREATE_COMPLETE	2023-11-20 22:21:23 UTC+0000

Figure 6: AWS CloudFormation templates are executed to create primary environment.

4.2 Secondary Environment

The secondary environment is maintained for consistent fault tolerance in the overall infrastructure. Over here, the author of this research paper has used eu-west-2 (London) as the secondary region. Similarly all the AWS infrastructure is built using IaaC service which is AWS CloudFormation. Same as the primary region AWS S3 bucket is used to store the CloudFormation templates and the application war file for AWS Elastic Beanstalk application deployment. Fresh VPC is created for this environment along with four subnets, of which two are public and two are private. The secondary region also contains AWS RDS ReadReplica, which is Multi-AZ enabled. The master database for the secondary region is the same as the primary region therefore it is configured to point to the master database in the primary region. AWS Elastic Beanstalk CloudFormation



Stacks (3)

london

	Stack name	Status	Created time
○	eb-london-secondary	CREATE_COMPLETE	2023-11-29 15:13:49 UTC+0000
○	vpc-rds-london-secondary	CREATE_COMPLETE	2023-11-29 14:33:06 UTC+0000
○	s3-london	CREATE_COMPLETE	2023-11-22 01:22:14 UTC+0000

Figure 7: AWS CloudFormation templates are executed to create secondary environment.

stack is not created here beforehand, and it is only created when the primary environment is under disaster or is not working as expected for some odd reason. To get the application again in the normal state when the primary environment is not working the author will run the Elastic Beanstalk yaml template stored in S3 Bucket, which will deploy the application again and bring it back to life. Figure 7 shows the stacks created in AWS CloudFormation to provision the secondary environment.

5 Implementation

In the previous section author discussed the overall infrastructure of the hybrid system using Figure 5. In this section, an effort is made to explain the implementation of the proposed strategy and also an explanation for the disaster recovery plan (DRP) in case a real disaster hits the primary region. Interested people can find the implemented source code on GITHUB.

It is expected to have the primary environment running at all times with zero downtime but unpredictable circumstances like natural or man-made disasters can interfere with the normal workflow of the primary infrastructure as shown in Figure 8. The author has developed separate CloudFormation templates for both primary and secondary region infrastructures. But just running the CloudFormation templates on AWS is not sufficient to achieve the required results nor does it justify the author's overall implementation, setting up each component taking part is also a key requirement for which the configuration manual will be the best document to go through. The main idea behind creating these CloudFormation templates is to provision all the services pre-configured before you attach the application and all its dependencies to it. Only a few parameters will be required to be given to have a unique service namespace.

5.1 Provisioning Primary Environment

In order to provision primary envrionment here are the steps given below:

1. Login to AWS management console and switch to **us-east-2 (Ohio)** region. Make sure to use primary region CloudFormation templates.

2. Go to AWS CloudFormation Service and hit on create stack with new resources. From the primary region CloudFormation template files, select S3-ohio.yaml file to create the S3 bucket. Enter stackname and click on submit. This will create S3 bucket for you. Upload all the primary regions yaml file onto the recently created S3 bucket along with the application war file.

3. Click on create stack with new resources again and give the S3 url of the VPC.yaml file this time and go next, enter the required parameter names and hit submit. This will create a VPC along with four subnets, subnet group, public and private route table, NAT gateway, Internet gateway, RDS Master database, RDS ReadReplica and Database ec2 security group. This makes sure that VPC and RDS database is created together in one go. But in case just one service goes down there are separate templates for each service available to be executed.

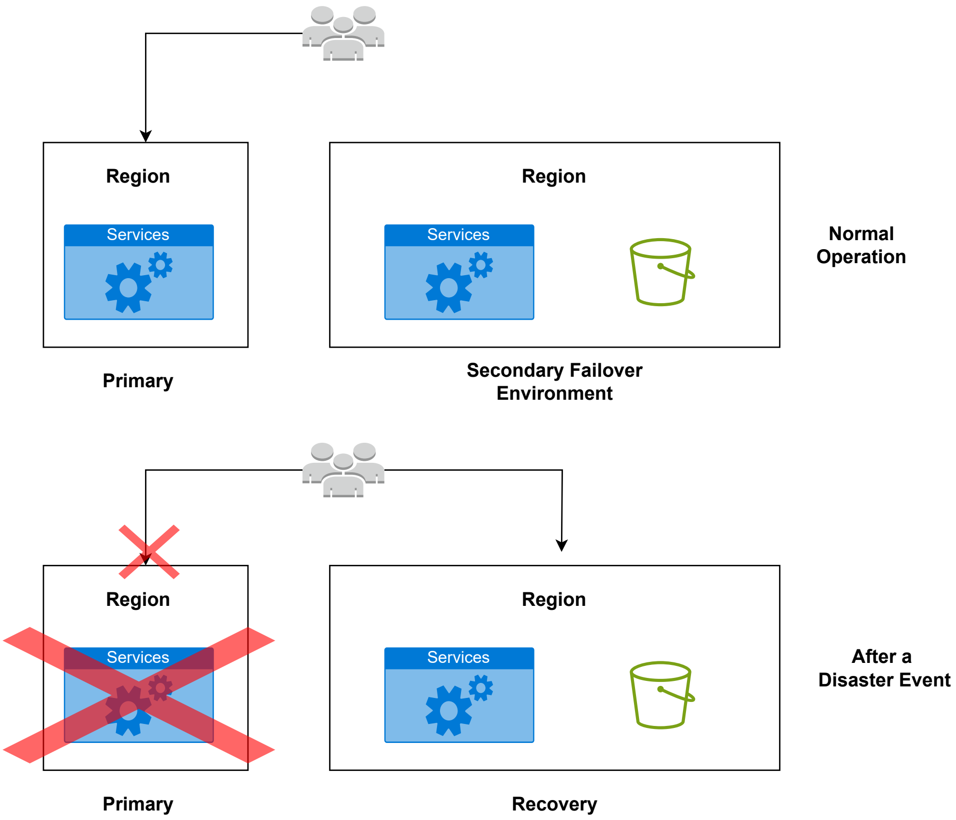


Figure 8: High level Infrastructure Diagram

4. Next step is to copy both public, private subnet ids and also VPC id just created by VPC.yaml template and pass the subnet ids into the EB-ohio.yaml code to deploy the application on AWS Elastic Beanstalk which uses the updated EB-ohio.yaml file to create a stack with new resources make sure that the Elastic load balancer uses public subnet ids and EC2 servers use private subnet ids only.

5. Once the Elastic Beanstalk stack is created the application is ready to be used, we can use the Elastic Beanstalk DNS to access the deployed application [2].

5.2 Failover Process to Secondary Envrionment

AWS failover maintains uninterrupted service availability by routing traffic to a secondary environment in the event of a breakdown, ensuring high dependability and reducing downtime for services and applications. To provision a secondary environment here are the steps given below:

1. Login to AWS management console and switch to **eu-west-2 (London)** region. Make sure to use secondary region CloudFormation templates.

2. Go to AWS CloudFormation Service and hit on create stack with new resources. From the primary region CloudFormation template files, select S3-london.yaml file

[2] `https://docs.aws.amazon.com/whitepapers/latest/disaster-recovery-workloads-on-aws/disaster-recovery-options-in-the-cloud.html`

to create the S3 bucket. Enter stackname and click on submit. This will create S3 bucket for you. Upload all the secondary regions yaml files onto the recently created S3 bucket along with the application war file.

3. Edit VPC2.yaml file so that the RDS ReadReplica points to the master RDS Database instance ARN. Once done click on create stack with new resources again and give the S3 url of the updated VPC2.yaml file this time and go next, enter the required parameter names and hit submit. This will create a VPC along with four subnets, subnet group, public and private route table, NAT gateway, Internet gateway, RDS ReadReplica and Database ec2 security group. This makes sure that VPC and RDS database is created together in one go. But in case just one service goes down there are separate templates for each service available to be executed.

4. **Important Key Step:** This step is only executed when primary region goes down. Copy both public and private subnet IDs and also VPC ID just created by VPC2.yaml template and pass it into the EB-london.yaml to deploy the application on AWS Elastic Beanstalk which uses the updated EB-london.yaml file to create a stack with new resources make sure that the Elastic load balancer uses public subnet ids and EC2 servers use private subnet ids only.

5. Once the Elastic Beanstalk stack is created the application is ready to be used, we can use the Elastic Beanstalk DNS to access the deployed application [3].

5.3 Tools & Services Used

Tools and Technologies	
Cloud Provider	Amazon Web Services
Source Code Editor	VS Code
IaaC Service	AWS CloudFormation
Programming Language used	YAML
Database Service	AWS RDS
AWS RDS Instance type	db.t2.micro gp2 5 GiB
Orchestration Service	AWS Elastic Beanstalk
AWS Elastic Beanstalk Platform	Tomcat 10 running on 64bit Amazon Linux
CloudFormation Template Storage Service	AWS S3

Table 1: Tools and Technologies Used.

6 Evaluation

This section contains all the results of evaluations conducted during the research. There are total two evaluations done during the research which are mentioned below.

[3] `https://docs.aws.amazon.com/whitepapers/latest/disaster-recovery-workloads-on-aws/disaster-recovery-options-in-the-cloud.html`

6.1 Performance measures: RTO

Every company has their targeted recovery time objective fixed. It indicates the maximum threshold of time a company can have their application down without having much financial loss. From business-critical applications to small-scale applications, everyone has a different RTO. For the purpose of evaluation the author of this paper decided to have a targeted RTO of a maximum of 1 hour for both regions. While calculating RTO it is important to note that we will have a unique RTO for each service, so in this case, the below bar chart in Figure 9 shows RTO for each service in primary region and also overall RTO to provision entire infrastructure in the primary region. As per the results, it shows that the RTO is well below the targeted RTO.

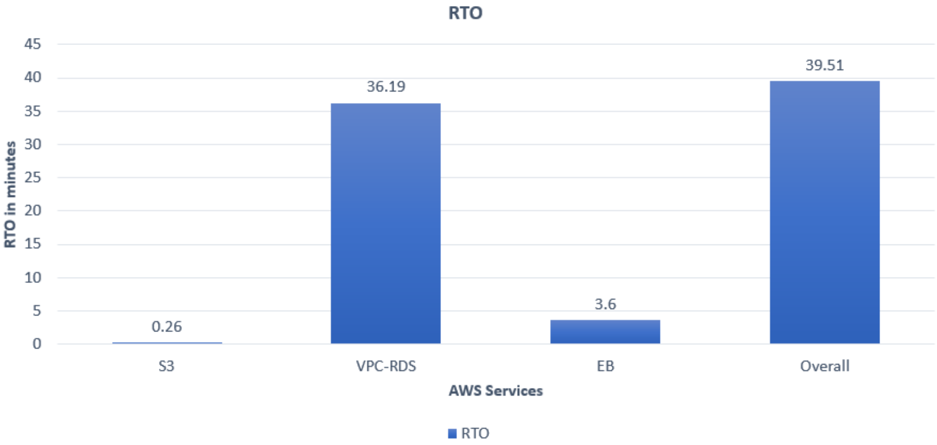


Figure 9: RTO for Primary Region Stacks

Looking at the RTO results of the secondary or failover region in Figure 10 it is clear that it has much lower RTO than that of the primary region. For the overall stack

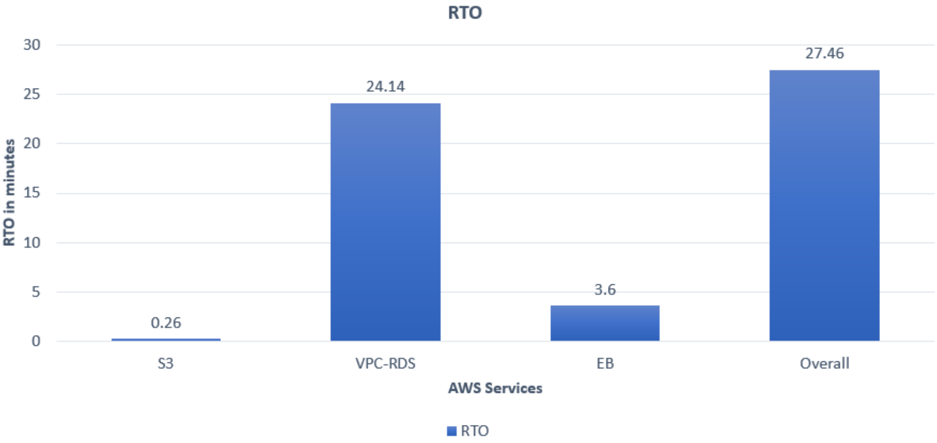


Figure 10: RTO for Secondary Region Stacks

creation in secondary region it takes 27.46 minutes as compared to primary region which takes 39.51 minutes.

6.2 Performance measures: RPO

The recovery point objective refers to the quantity of data a cloud application can afford to lose. As the hybrid disaster recovery infrastructure is using Multi-AZ with two Read-Replica, a new master database will be ready to serve within 35 seconds of disaster, which makes our RPO 35 seconds. Failover time also depends on the duration of ReplicaLag and the latency between the two regions. From the Figure 11 we can tell the ReplicaLag in primary regions is 0 seconds with few levels of network fluctuation [4].

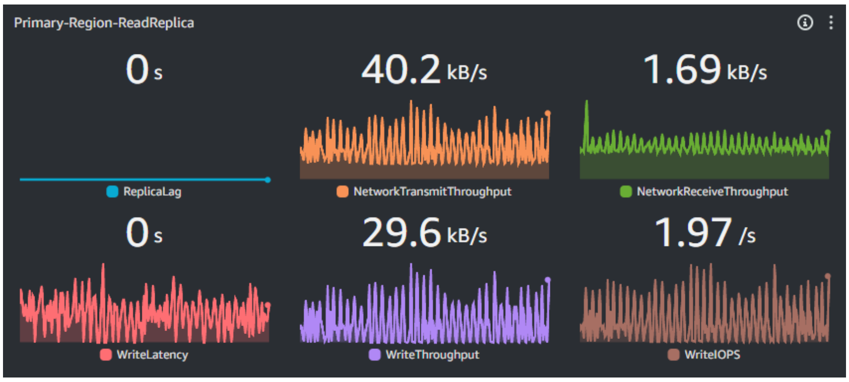


Figure 11: RPO for Primary Region

Similarly, Looking at the metrics in Figure 12 it will be right to say that the ReadReplica in the secondary region is also of 0 seconds with some network conditions involved.

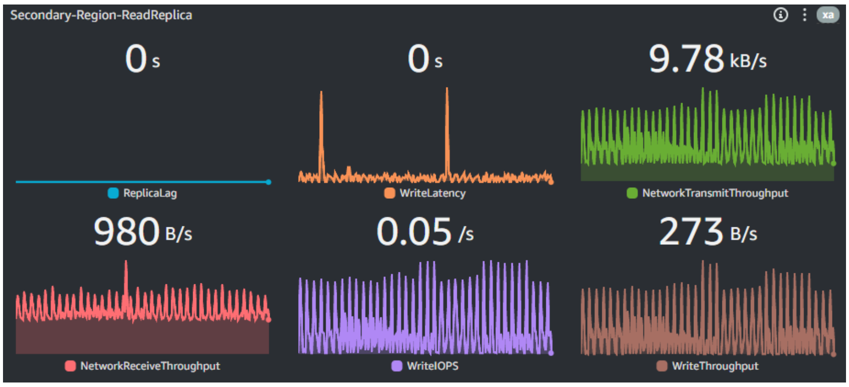


Figure 12: RPO for Secondary Region

Conditions: While stating the RPO as 35 seconds and ReplicaLag as 0 seconds we always have to keep in mind that it is not always straight 35 or 0 seconds mentioned by AWS because it involves few factors in between which is Replication Lag which is the input lag to sync the database with master. Second is Network latency between the two regions which plays a major factor in deciding the RPO.

[4] https://pages.awscloud.com/rs/112-TZM-766/images/2022_0408-DAT_Slide-Deck.pdf

6.3 Discussion

For the evaluation purpose the author performed two evaluation tests which are standard in any disaster recovery plan testing i.e. calculating recovery time objective (RTO) and calculating recovery point objective (RPO) for both the primary and secondary region infrastructure. In the bar chart Figure 13 author have included the results of RTO for both regions and discussed RPO results as well.

- The discussion goes into the outcomes of two evaluations, providing a detailed assessment of the performance of the disaster recovery infrastructure. Particularly, both region's Recovery Time Objective (RTO) showed results below the desired 1-hour constraint, confirming the infrastructure's dependability. On the other hand due to lot of mission critical application it is important to have automation into the DRP plan so that manual interaction with the DRP is minimum which will decrease the RTO more.

- Recovery Point Objective (RPO) results were excellent but the research demonstrates that the impact of factors such as ReplicaLag and Network latency on the accuracy of RPO. One of the alternatives that can be used instead of AWS RDS is AWS Arora which can have 15 total ReadReplicas in all overall availability zones and also each of the Replicas can be promoted to master database without even restarting the database instance, which makes the RTO exactly 0.

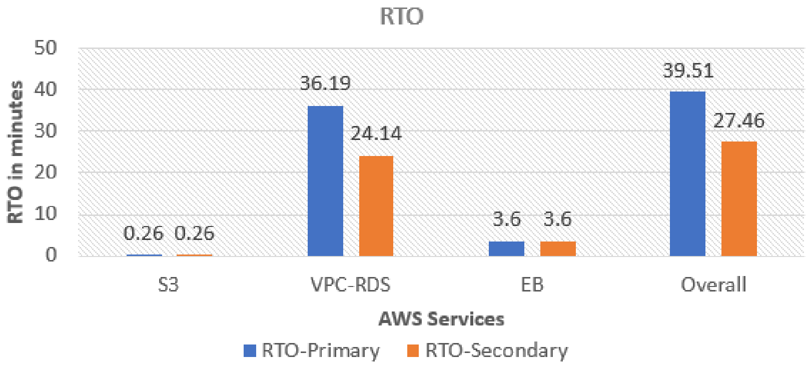


Figure 13: Bar Chart showing RTO for both regions.

6.3.1 Improvements

1. Automation: Priority should be given to improvements in the automation of the disaster recovery process. By utilizing modern automation techniques such as AWS EventBridge along with Lambda during service outages it can automatically failover event ingestion to a backup region without requiring user intervention [5].

2. Expansion of Services: The study might be extended to include a larger variety of cloud services. Incorporating and evaluating other services during disaster recovery can give a more thorough knowledge of the hybrid approach's usefulness.

[5] https://shorturl.at/yCGMN

3. Stress Testing the Database with HammerDB: Stress testing the database with tools like HammerDB may recreate real-world scenarios to assess the system's performance under harsh conditions to further confirm the system's resilience. This would help to provide a more thorough and accurate assessment of disaster recovery ability.

4. Vendor Lock-in: One of the important things to note is that the code should be platform neutral which means that it should not be made for just one cloud platform. Another factor that comes into play is that AWS might decide to discontinue some of the service so there should be alternative arrangements accordingly Weldemicheal (2023).

7 Conclusion and Future Work

Every disaster recovery plan has room for adjustment and improvement. In the hybrid pilot light and active-active strategy, the author presented acceptable results well within the targeted recovery time thresholds. Both primary and secondary regions gave RTO of approximately 40 minutes and 27 minutes respectively which is 20 minutes and 33 minutes well within the targeted 1 hour threshold. RPO is estimated to be 35 seconds due to cross-region ReadReplica implementation which is 35 seconds in order to promote the ReadReplica to master database instance [6].

As this DRP performs up to expectations, it would be enlightening to test this DRP for some business critical application while using multiple services with proper automation and pipeline triggers in place. Secondly, as discussed before AWS Arora can be a better alternative to achieve real-time recovery point objectives. Third is that the DRP should be applicable to all the cloud service providers so we can take advantage of technologies such as terraform and ansible to achieve such neutral DRP and scripts. Fourth, with more time in hand, the author would like to look into multi cloud DR Gupta and Mahto (2021). Lastly, it would be beneficial to work on placement strategy which will improve the system network latency issue and help with proper placement of services across regions.

References

Abualkishik, A. Z., Alwan, A. A. and Gulzar, Y. (2020). Disaster recovery in cloud computing systems: An overview, *International Journal of Advanced Computer Science and Applications* **11**(9).

Alhazmi, O. H. and Malaiya, Y. K. (2012). Assessing disaster recovery alternatives: On-site, colocation or cloud, *2012 IEEE 23rd International Symposium on Software Reliability Engineering Workshops*, pp. 19–20.

Baginda, Y. P., Affandi, A. and Pratomo, I. (2018). Analysis of rto and rpo of a service stored on amazon web service (aws) and google cloud engine (gce), *2018 10th International Conference on Information Technology and Electrical Engineering (ICITEE)*, IEEE, pp. 418–422.

[6] https://pages.awscloud.com/rs/112-TZM-766/images/2022_0408-DAT_Slide-Deck.pdf

Chen, W. and Shang, Y. T. (2017). Disaster recovery of online system based on cloud computing, *Applied Mechanics and Materials* **865**: 636–641.

Emejeamara, U. (2020). Effective method for managing automation and monitoring in multi-cloud computing: Panacea for multi-cloud security snags, *International Journal of Network Security & Its Applications (IJNSA) Vol* **12**.

Gupta, V. and Mahto, A. (2021). Optimal solution for a disaster recovery (dr) site across multiple cloud service providers, *Proceedings of the 2nd International Conference on ICT for Digital, Smart, and Sustainable Development, ICIDSSD 2020, 27-28 February 2020, Jamia Hamdard, New Delhi, India*.

Kartheeyayini, V., Madhumitha, S., Lalitha, G., Jackulin, C. and Subramanian, K. (2022). Aws cloud computing platforms deployment of landing zone-infrastructure as a code, *AIP Conference Proceedings*, Vol. 2393, AIP Publishing.

Mendonça, J., Lima, R., Matos, R., Ferreira, J. and Andrade, E. (2018). Availability analysis of a disaster recovery solution through stochastic models and fault injection experiments, *2018 IEEE 32nd International Conference on Advanced Information Networking and Applications (AINA)*, IEEE, pp. 135–142.

Mendonça, J., Lima, R., Queiroz, E., Andrade, E. and Kim, D. S. (2019). Evaluation of a backup-as-a-service environment for disaster recovery, *2019 IEEE Symposium on Computers and Communications (ISCC)*, IEEE, pp. 1–6.

Patel, J. and Keerthana (2019). Disaster recovery in business continuity management, *International Journal of Trend in Scientific Research and Development* **Volume-3**: 319–322.

Paul, J. J. (2023). Disaster recovery architectures, *Distributed Serverless Architectures on AWS: Design and Implement Serverless Architectures*, Springer, pp. 49–73.

Prabantoro, R. and Aji, R. F. (2021). Cloud computing implementation to support a disaster recovery plan: A case study of institut agama islam negeri manado, *2021 International Conference on Converging Technology in Electrical and Information Engineering (ICCTEIE)*, IEEE, pp. 112–117.

Sabbaghi, F., Mahboubi, A. and Othman, S. H. (2017). Hybrid service for business contingency plan and recovery service as a disaster recovery framework for cloud computing, *Journal of Soft Computing and Decision Support Systems* **4**(4): 1–10.

Shahzadi, S., Ubakanma, G., Iqbal, M. and Dagiuklas, T. (2018). Autonomous, seamless and resilience carrier cloud brokerage solution for business contingencies during disaster recovery, *2018 IEEE 20th International Conference on High Performance Computing and Communications; IEEE 16th International Conference on Smart City; IEEE 4th International Conference on Data Science and Systems (HPCC/SmartCity/DSS)*, IEEE, pp. 1048–1053.

Solis, R., Shashidhar, N. and Varol, C. (2021). A novel risk mitigation & cloud-based disaster recovery framework for small to medium size businesses, *2021 9th International Symposium on Digital Forensics and Security (ISDFS)*, IEEE, pp. 1–5.

Stamenkov, G. (2022). Layered business continuity and disaster recovery model, *Continuity & Resilience Review* **4**(3): 267–279.

Tamimi, A. A., Dawood, R. and Sadaqa, L. (2019). Disaster recovery techniques in cloud computing, *2019 IEEE Jordan International Joint Conference on Electrical Engineering and Information Technology (JEEIT)*, IEEE, pp. 845–850.

Tomás, L., Kokkinos, P., Anagnostopoulos, V., Feder, O., Kyriazis, D., Meth, K., Varvarigos, E. and Varvarigou, T. (2017). Disaster recovery layer for distributed openstack deployments, *IEEE Transactions on Cloud Computing* **8**(1): 112–123.

Tomás, L., Kokkinos, P., Anagnostopoulos, V., Feder, O., Kyriazis, D., Meth, K., Varvarigos, E. and Varvarigou, T. (2020). Disaster recovery layer for distributed openstack deployments, *IEEE Transactions on Cloud Computing* **8**(1): 112–123.

Trovato, F., Sharp, A. and Siman, T. (2019). Cloud, co-location, on-premises and hybrid disaster recovery solutions: Pros, cons, and a cost comparison, *Journal of Business Continuity & Emergency Planning* **13**(2): 120–135.

Tsubaki, T., Ishibashi, R., Kuwahara, T. and Okazaki, Y. (2020). Effective disaster recovery for edge computing against large-scale natural disasters, *2020 IEEE 17th Annual Consumer Communications & Networking Conference (CCNC)*, IEEE, pp. 1–2.

Weldemicheal, T. (2023). Vendor lock-in and its impact on cloud computing migration.

Yu, X., Wang, D., Sun, X., Zheng, B. and Du, Y. (2022). Design and implementation of a software disaster recovery service for cloud computing-based aerospace ground systems, *2022 11th International Conference on Communications, Circuits and Systems (ICCCAS)*, IEEE, pp. 220–225.