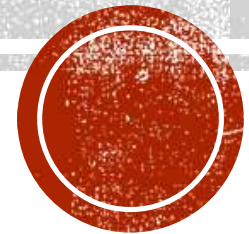


# RELIABILITY AND HIGH AVAILABILITY IN AWS

- Madabhushi Sai Sravani (Z23752172)
- Janamma, Naveen Reddy(Z23746049)
- Gajera, Nihar Yogeshbhai(Z23702475)
- Parsharouthu, Jeethesh Verma (Z23752260)



# INTRODUCTION

## **What is Reliability and High Availability?**

- In the context of AWS, reliability refers to the ability of a system or service to consistently perform its intended function without interruption or failure.
- High availability, on the other hand, refers to the ability of a system or service to remain operational and accessible even in the face of hardware or software failures, natural disasters, or other unforeseen events.

## **Why is it Important?**

- By designing and implementing reliable and highly available architectures, organizations can minimize the impact of failures and disruptions, and provide a seamless experience to their users.



# KEY CONCEPTS

To achieve reliability and high availability in AWS, several key concepts and best practices need to be understood and implemented. These include:

- **Redundancy:** Deploying multiple instances of a service or resource to ensure availability in case of failure.
- **Fault tolerance:** Designing systems that can continue to operate even if individual components fail.
- **Scalability:** The ability to handle increasing workloads by adding resources as needed.
- **Disaster recovery:** Planning and implementing strategies to recover from catastrophic events.



# **AWS INFRASTRUCTURE OVERVIEW**

## **Regions**

- AWS is divided into multiple regions around the world, each consisting of multiple Availability Zones.

## **Availability Zones**

- Availability Zones (AZs) are physically separate data centers within a region



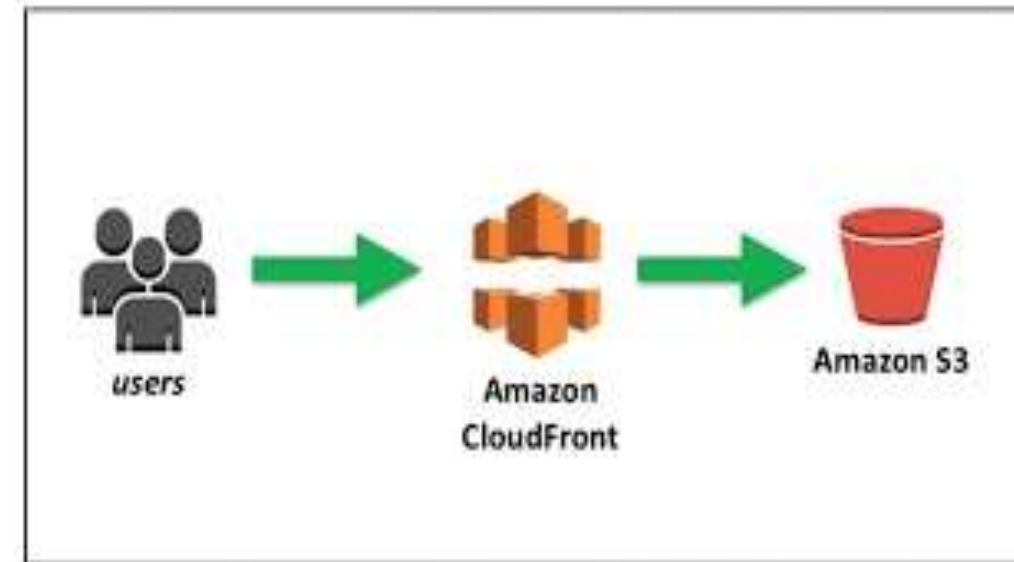
# AMAZON S3

## Features

- Durability and Reliability
- Scalability
- Data Protection:
- Lifecycle Management

## Benefits

- High Availability
- Integration with AWS Services



# AWS LAMBDA

- Automatic Scaling
- Fault Tolerance
- Monitoring and Logging
- Integration with Other AWS Services



# CASE STUDY

## Scenario and Requirements:

- Company seeks highly available architecture for user-file storage and processing.
- Requires resilience, high availability, and seamless scalability.
- Chooses Amazon S3 for storage and AWS Lambda for serverless processing.

## •Architecture Diagram:

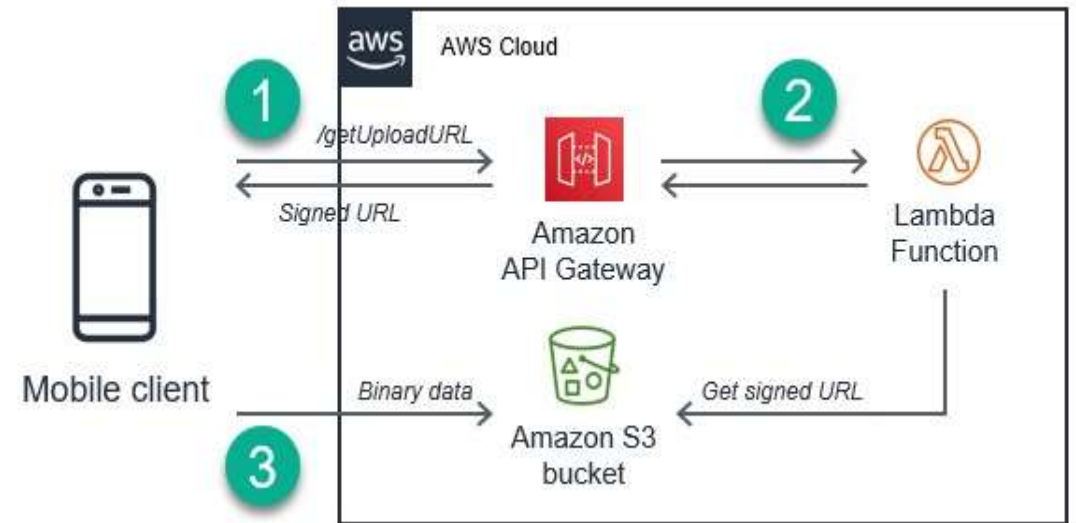
- User uploads file -> Amazon S3 (Storage) | AWS Lambda (Processing)

## •High Availability and Reliability (Amazon S3):

- 99.999999999% durability with data replication across zones.
- Strong consistency ensures immediate file availability post-upload.

## •High Availability and Reliability (AWS Lambda):

- Automatic scaling and fault tolerance in response to requests.
- Functions deployed across multiple zones for redundancy.
- Event-triggered processing ensures immediate action upon file upload.



# IMPLEMENTATION DETAILS

- **1. Amazon S3 Configuration:** Enable versioning, implement lifecycle policies, and enable encryption for data protection. Set up cross-region replication for redundancy and disaster recovery.
- **2. AWS Lambda Configuration:** Configure Lambda functions with event triggers for S3 uploads. Implement error handling and retry mechanisms within Lambda functions.
- **3. Integration:** Configure S3 event notifications to trigger Lambda functions.
- **4. Best Practices:** Deploy resources across multiple Availability Zones. Implement retry strategies and dead letter queues. Monitor resource health and optimize performance.
- **5. Monitoring and Performance:** Monitor key metrics with CloudWatch, enable logging and tracing with CloudWatch Logs and X-Ray. Optimize Lambda function performance and manage costs with AWS Cost Explorer.





# RESULTS AND BENEFITS

## **1. Quantitative and Qualitative Improvements:**

1. Increased uptime and faster processing times.
2. Enhanced data integrity and security measures.

## **2. Reliability, Availability, Scalability, and Cost-Effectiveness:**

1. Achieved 99.999% uptime with S3 and Lambda's scalability.
2. Reduced infrastructure costs with optimized resource utilization.

## **3. Positive Impact on Business Operations and Customer Experience:**

1. Streamlined operations and improved customer satisfaction.
2. Strengthened brand reputation and competitiveness.



# LESSONS LEARNED

- **Challenges & Solutions:**

- Ensure data consistency.
- Address cross-AZ latency.
- Manage costs effectively.
- Simplify configuration with IaC.

- **Key Takeaways:**

- Understand requirements.
- Use Multi-AZ deployment.
- Implement auto-scaling and monitoring.
- Test regularly.



# **BEST PRACTICES FOR DESIGNING HIGH AVAILABILITY ARCHITECTURES**

1. **Multi-AZ Deployment:** Distribute across AZs for redundancy.
2. **Load Balancing:** Balance traffic to prevent bottlenecks.
3. **Auto-scaling:** Adjust capacity based on demand.
4. **Data Replication:** Ensure data availability and durability.
5. **Monitoring:** Detect failures quickly.
6. **Disaster Recovery:** Plan for catastrophic failures.
7. **Decoupling:** Minimize dependencies for fault isolation.
8. **Security:** Protect infrastructure and data.
9. **Testing and Optimization:** Continuously test and optimize.
10. **Documentation:** Maintain thorough documentation.



# CONCLUSION

- Reliability and high availability in AWS needs monitoring, disaster recovery planning and continuous optimization to deliver uninterrupted services and mitigate potential disruptions effectively.



**THANK YOU!**

