****

**IPRC NGOMA**

**Integrated Polytechnic Regional College**

**P.O. Box35 KIBUNGO - RWANDA**

**Tel: +250 785883746**

**Email: info@iprcngoma.rp.ac.rw**

**Website: www.iprcngoma.rp.ac.rw**

**MODULE CODE:** ITLDO801

**MODULE TITLE:** Development Operations

****

***MEMBERS:***

1. ***23RP01381***
2. ***23RP01946***
3. ***23RP00015***

**Group 7**

**Assignment**

**A) Fault tolerance and high availability**

**What is Fault Tolerance?**

Fault tolerance means that a system will almost always maintain uptime — and users will not notice any differences during a primary system outage.

Fault tolerance in cloud computing means creating a blueprint for ongoing work whenever some parts are down or unavailable. It helps enterprises evaluate their infrastructure needs and requirements and provides services in case the respective device becomes unavailable for some reason.

It does not mean that the alternative system can provide 100% of the entire service. Still, the concept is to keep the system usable and, most importantly, at a reasonable level in operational mode. It is important if enterprises continue growing in a continuous mode and increase their productivity levels.

**Main Concepts behind Fault Tolerance in Cloud Computing System**

* **Replication**: Fault-tolerant systems work on running multiple replicas for each service. Thus, if one part of the system goes wrong, other instances can be used to keep it running instead. For example, take a database cluster that has 3 servers with the same information on each. All the actions like data entry, update, and deletion are written on each. Redundant servers will remain idle until a fault tolerance system demands their availability.
* **Redundancy**: When a system part fails or goes downstate, it is important to have a backup type system. The server works with emergency databases that include many redundant services. For example, a website program with MS SQL as its database may fail midway due to some hardware fault. Then the redundancy concept has to take advantage of a new database when the original is in offline mode.

**Techniques for Fault Tolerance in Cloud Computing**

* Priority should be given to all services while designing a fault tolerance system. Special preference should be given to the database as it powers many other entities.
* After setting the priorities, the Enterprise has to work on mock tests. For example, Enterprise has a forums website that enables users to log in and post comments. When authentication services fail due to a problem, users will not be able to log in.

Then, the forum becomes read-only and does not serve the purpose. But with fault-tolerant systems, healing will be ensured, and the user can search for information with minimal impact.

Major Attributes of Fault Tolerance in Cloud Computing

* **None Point of Failure**: The concepts of redundancy and replication define that fault tolerance can occur but with some minor effects. If there is no single point of failure, then the system is not fault-tolerant.
* **Accept the fault isolation concept**: the fault occurrence is handled separately from other systems. It helps to isolate the Enterprise from an existing system failure.

Existence of Fault Tolerance in Cloud Computing

* **System Failure**: This can either be a software or hardware issue. A software failure results in a system crash or hangs, which may be due to Stack Overflow or other reasons. Any improper maintenance of physical hardware machines will result in hardware system failure.
* **Incidents of Security Breach**: There are many reasons why fault tolerance may arise due to security failures. The hacking of the server hurts the server and results in a data breach. Other reasons for requiring fault tolerance in the **form of security breaches include ransomware, phishing, virus attacks, etc.**

**What is High Availability?**

**High availability on AWS**

Most providers of real-time communications align with service levels that provide availability from 99.9% to 99.999%. Depending on the degree of high availability (HA) that you want, you must take increasingly sophisticated measures along the full lifecycle of the application.

AWS recommends following these guidelines to achieve a robust degree of high availability:

* Design the system to have no single point of failure. Use automated monitoring, failure detection, and failover mechanisms for both stateless and stateful components
  + **Single points of failure (SPOF**) are commonly eliminated with an N+1 or 2N redundancy configuration, where N+1 is achieved via load balancing among *active–active* nodes, and 2N is achieved by a pair of nodes in *active–standby* configuration.
  + AWS has several methods for achieving HA through both approaches, such as through a scalable, load balanced cluster or assuming an *active–standby* pair.
* Correctly instrument and test system availability.
* Prepare operating procedures for manual mechanisms to respond to, mitigate, and recover from the failure.

This section focuses on how to achieve no single point of failure using capabilities available on AWS. Specifically, this section describes a subset of core AWS capabilities and design patterns that allow you to build highly available real-time communication applications.

High availability means a system will almost always maintain uptime, albeit sometimes in a degraded state. About AWS, a system has high availability when it has 99.999% uptime, also known as "five nines." To put that in perspective, the system would be down for a mere five minutes and fifteen seconds a year. And yes, that is possible — and even routine — for AWS.

**B) Backup and restore strategies**

What is Cloud Backup and How does it Work?

A copy of a physical or virtual file or database is sent to a secondary, off-site location via cloud backup, often referred to as online backup or remote backup, as a precaution against equipment failure, site disasters, or human error. A third-party cloud or SaaS provider typically hosts the backup server and data storage systems. The backup client is charged a recurrent price depending on the amount of storage space or capacity utilized, data transmission bandwidth, number of users, number of servers, or number of times data is retrieved.

In an organization's data center, a backup application copies data and stores it on different media or another storage system for easy access in the event of a recovery situation. If a business maintains its own private cloud service, it may own the off-site server. However, if the business utilizes a service provider to handle the cloud backup environment and gets a monthly fee for backup storage and services, the chargeback procedure would be identical.

Cloud backup may be done in many different ways, and there are services that are readily integrated into an organization's current data security procedures.

The following are some examples of cloud backup variations:

* **Immediately storing backups on the public cloud:** Using public cloud resource duplication as a storage solution is one method for organizing workloads. Writing data directly to cloud service providers like AWS, Google Cloud, or Microsoft Azure is the approach used here. The company makes a duplicate of the data to transfer to the cloud storage provider using its own backup software.
* **Using cloud backup services available online:** Hardware options are also available that make data backup to a cloud backup provider easier. These appliances are all-in-one backup devices that come with a backup server, backup software, and disc space. A number of companies, including Quantum, Unitrends, Arcserve, Rubrik, Cohesity, Dell EMC, StorageCraft, and Asigra, are involved in the backup appliance market and provide cloud interfaces.

**Types of backups**

There are three primary kinds vary from one another.

1. **Full Backups:** Each time a full backup is started, the whole data set is copied. They thus provide the best degree of protection. However, since complete backups may be time-consuming and need a lot of data storage space, most organizations are unable to do them often.
2. **Incremental Backups:** Only the data that has been updated or altered since the previous backup increment is backed up by incremental backups; the last complete backup is not backed up. Although this strategy saves time and storage space, it may make a full restoration more challenging since it won't be feasible to do a full restore if any backup increment is lost or destroyed. Because incremental cloud backup often uses less resources, it is a popular option.
3. **Differential backups:** Because differential backups only store changed data, they are comparable to incremental backups. Differential backups, on the other hand, only restore data that has changed since the previous complete backup as opposed to the whole backup. The challenge of challenging restorations that might occur with incremental backups is resolved by this technique.

Here are the steps for creating backups in AWS:

1. **Identify the Resources to Back Up**: Determine which AWS resources (e.g., EC2 instances, RDS databases, S3 buckets) need to be backed up.
2. **Create Amazon S3 Buckets**: Set up S3 buckets to store backup data. Ensure the buckets have proper access permissions and encryption enabled.
3. **Configure AWS Backup Service**: Open the AWS Backup console. Create a backup plan specifying the frequency, retention period, and backup vault.
4. **Assign Resources to the Backup Plan**: Use tags or resource IDs to assign AWS resources to the backup plan. AWS Backup will automatically back up these resources according to the defined schedule.
5. **Manual Backups (if needed)**: For ad-hoc backups, manually initiate backups for resources using the AWS Management Console, CLI, or SDKs.
6. **Enable Cross-Region Backup (optional)**: Configure cross-region replication to store backups in multiple AWS regions for enhanced disaster recovery.
7. **Monitor Backup Jobs**: Use AWS Backup console or CloudWatch to monitor the status of backup jobs and ensure they complete successfully.
8. **Verify Backups**: Regularly verify the integrity and restorability of backups by performing test restores.
9. **Set Up Automated Backup Policies**: Define and apply backup policies to ensure compliance with organizational backup requirements.
10. **Review and Optimize**: Periodically review backup settings, schedules, and retention policies. Optimize for cost and performance based on changing requirements and resource usage.

**C) AWS Disaster Recovery services**

**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](https://aws.amazon.com/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.

**AWS DataSync**

[AWS DataSync](https://aws.amazon.com/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](https://aws.amazon.com/snowcone), [Amazon Simple Storage Service](https://aws.amazon.com/s3) (Amazon S3) buckets, [Amazon Elastic File System](https://aws.amazon.com/efs/) (Amazon EFS), and [Amazon FSx for Windows File Server](https://aws.amazon.com/fsx/windows/) file systems.

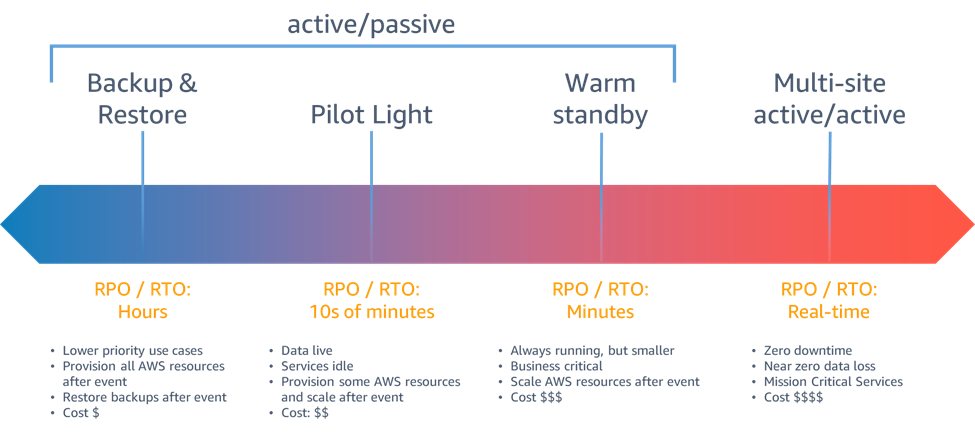
**Amazon Route 53 Application Recovery Controller**

[Amazon Route 53 Application Recovery Controller](https://aws.amazon.com/route53/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.

Backup with point-in-time recovery is available through the following services and resources.

* Amazon Elastic Block Store (Amazon EBS) snapshot
* Amazon DynamoDB backup
* Amazon RDS snapshot
* Amazon Aurora DB snapshot
* Amazon EFS backup (when using AWS Backup)
* Amazon Redshift snapshot
* Amazon Neptune snapshot
* Amazon DocumentDB
* Amazon FSx for Windows File Server, Amazon FSx for Lustre, Amazon FSx for NetApp ONTAP, and Amazon FSx for OpenZFS

**Disaster recovery strategies available to you within AWS can be broadly categorized into four approaches**



The general process are:

1. Install the AWS Replication Agent on the source server.
2. Wait until initial sync is finished.
3. Launch drill instances. Perform acceptance drills on the servers
4. Initiate a failover by redirecting traffic.
5. Confirm that the Recovery instance was launched successfully.
6. To recover your data, initiate a failback.
7. Complete the failback
8. Return to normal operations.

**Reference:**

<https://www.javatpoint.com/fault-tolerance-in-cloud-computing>

<https://docs.aws.amazon.com/whitepapers/latest/real-time-communication-on-aws/high-availability-and-scalability-on-aws.html>

<https://docs.aws.amazon.com/whitepapers/latest/disaster-recovery-of-on-premises-applications-to-aws/aws-services-related-to-disaster-recovery.html>

<https://www.javatpoint.com/what-is-cloud-backup-and-how-does-it-work>