1. Dec 2023

High Availability Architecture

# Load balancer

**Description:**

* The load balancer serves as the entry point for incoming traffic, distributing it across multiple web servers. This helps prevent any single server from being overwhelmed and ensures even load distribution.

**Handling Failures:**

* Implement multiple load balancers in an active-passive or active-active configuration. If one load balancer fails, the others can seamlessly take over the traffic distribution.
* Regularly monitor the health of load balancers to detect and replace any failing instances.

# Web Servers

**Description:**

* Multiple web servers host the same application or website, providing redundancy. Each server operates independently and serves identical content.

**Handling Failures:**

* Leverage a load balancer to distribute incoming traffic across multiple web servers. If one server fails, the load balancer redirects traffic to the healthy servers.
* Implement auto-scaling to adjust the number of web servers based on traffic patterns and performance metrics.

# Database servers

**Description:**

* If the web application relies on a database, redundant database servers may be employed. Database replication or clustering is used to maintain data consistency and availability.

**Handling Failures:**

* Implement database replication to create copies of the data on multiple servers. If one database server fails, another can take over with minimal disruption.
* Use automated failover mechanisms to switch to a standby database server in case of a primary server failure.

# Data replication and synchronization

**Description:**

* Data replication ensures that all servers, whether web or database, have identical content and data. This reduces the risk of data loss in case of a failure.

**Handling Failures:**

* Regularly synchronize data between servers to maintain consistency.
* Implement mechanisms for automatic resynchronization in case of temporary network partitions or isolated failures.

# Monitoring System

**Description:**

* A monitoring system continuously checks the health and performance of all components in the architecture.

**Handling Failures:**

* Set up alerts for abnormal conditions, such as increased response times, server failures, or other critical events.
* Integrate automated responses to certain types of issues, such as triggering auto-scaling in response to increased traffic.

# Automated Scaling

**Description:**

* Automated scaling adjusts the number of servers based on predefined criteria, ensuring optimal performance and cost efficiency.

**Handling Failures:**

* Automatically add or remove servers based on demand to handle traffic spikes and ensure consistent performance.
* Define scaling policies to prevent over-provisioning or under-provisioning, optimizing resource usage.

# Handling infrastructure failures

**Automatic Failover:**

* Implement automated failover mechanisms that detect failures in real-time and reroute traffic to healthy components. This reduces downtime and ensures uninterrupted service.

**Regular Backups:**

* Schedule regular backups of data and configurations. Store backups in a separate location to facilitate quick recovery in the event of a catastrophic failure, such as a data center outage.

How to implement HA in AWS – An example

The **AWS (Amazon Web Services) Cloud is unique in building high available business applications** since AWS’ high availability delivers features that let you build these applications with little interaction and minimal upfront investment.

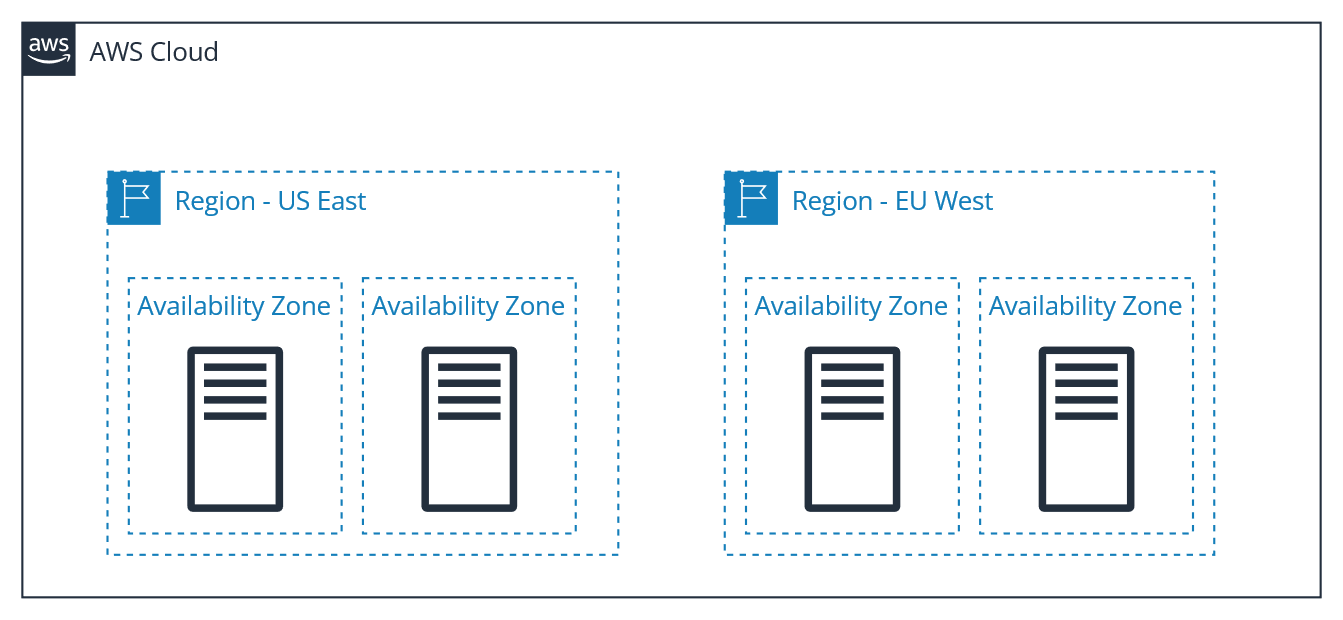
**AWS recommends following these guidelines:**

* **Design the system to have no single point of failure.** This can usually be achieved via load balancing, or by a pair of nodes in an active standby configuration.
* Prepare operating **procedures for manual or automatic mechanisms** to respond to, mitigate, and recover from the failure.

# **AWS high availability with zones**

AZs are **isolated places with servers (data centers) located within regions in which AWS operates**. At this moment there are 25+ regions and the minimum number of AZ in regions is two.

The use of multiple AZs enables customers to replicate workloads across the physical location.



Depending on the workload, it can be difficult to configure deployment across multiple AZs, but almost every AWS service has it covered and you just need to use the right **settings**. We’ll cover this more thoroughly in the next section of this blog post.

# **AWS Compute, Databases, and Storage High Availability**

Most applications will rely on these AWS services, which can be divided depending on their usage in multi-tier applications:

* **Compute**: Compute dimension contains AWS services like Amazon EC2 and AWS Lambda.
* **SQL Databases**: Amazon RDS and other managed SQL databases provide RDS high availability options.
* **Storage Services**: Such as S3, EFS, and EBS.

# **AWS computing high availability**

#### **EC2 high availability**

Amazon Elastic Compute Cloud (Amazon EC2) compute resources are made highly available by using multiple instances at the same time to collectively process incoming requests.

This group of Amazon EC2 instances can be managed with Amazon EC2 Auto Scaling. We can achieve a reasonably good level of high availability by using the following services and features:

**Availability Zones (AZs)** allow you to distribute EC2 instances across many locations. They provide a much greater level of protection against failure, as the data center itself is removed as a single point of failure.

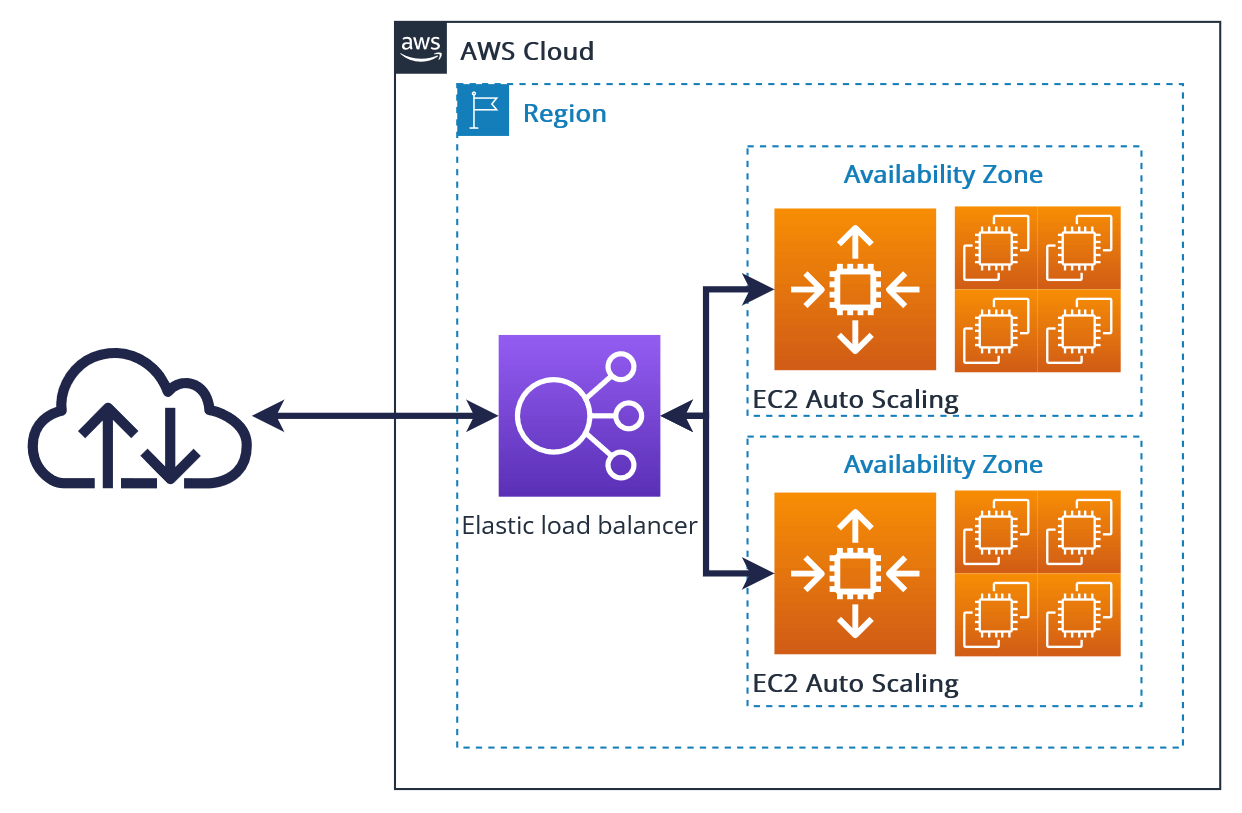
**Elastic Load Balancing (ELB)** allows you to launch several EC2 instances and spread traffic between them. As ELBs exist outside of any particular AZ, they are themselves insulated from AZ failure.

**Auto-scaling** allows for the health of Amazon EC2 instances to be monitored and also self-healed by spinning up new compute instances if an existing one fails, with feature support to help cater to stateful applications.

**Elastic IP addresses,** for example, can reroute a workload from one EC2 instance to another instantaneously.

#### **Best practices for EC2 instance high availability**

* It is recommended that you run EC2 instances in multiple AZs. If these zones become unavailable due to natural circumstances or power outages, Elastic Load Balancer can route traffic to operational AZ.
* If you need more than one instance in every AZ, it is recommended you use Auto Scaling which automatically increases the number of instances to meet your high-traffic demands and saves you money by stopping the instances which are no longer needed.



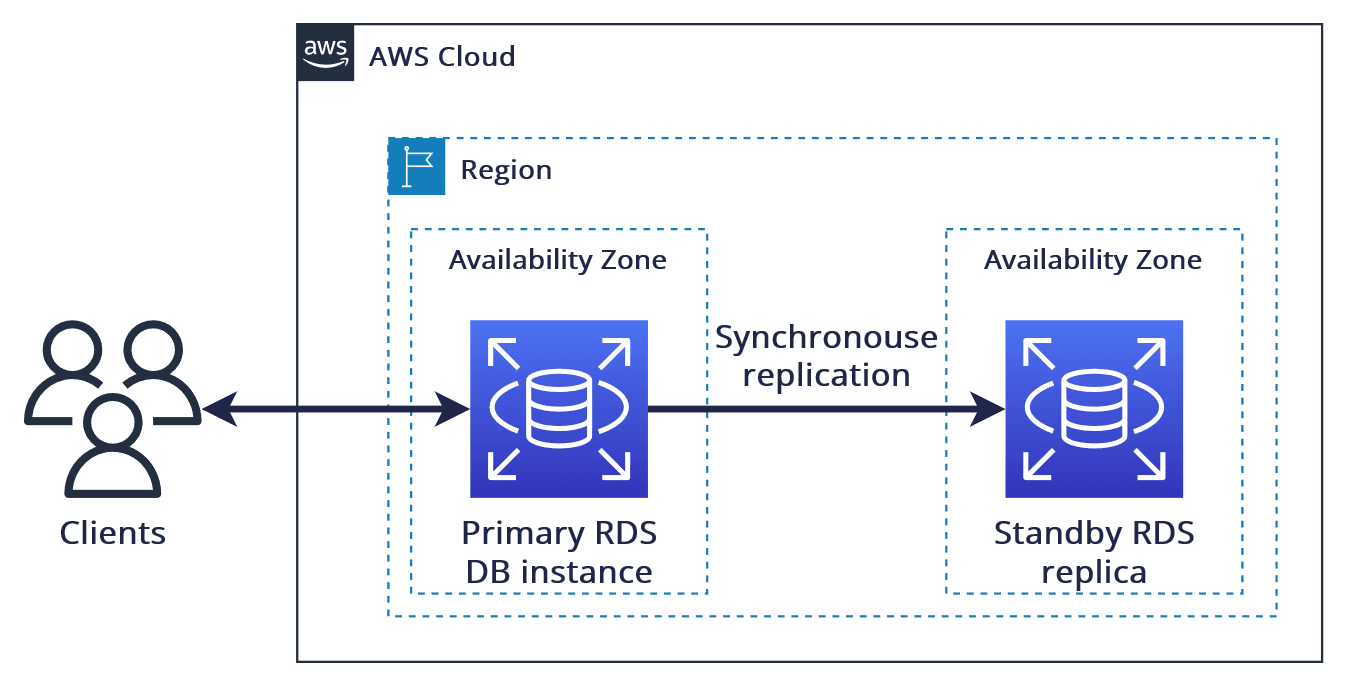
# **AWS high availability for Amazon RDS**

Amazon Relational Database Service (RDS) is similar in some ways to EC2 because RDS instances are EC2 instances managed by AWS. RDS's high availability is achievable by spreading multiple instances with the same data across AZs.

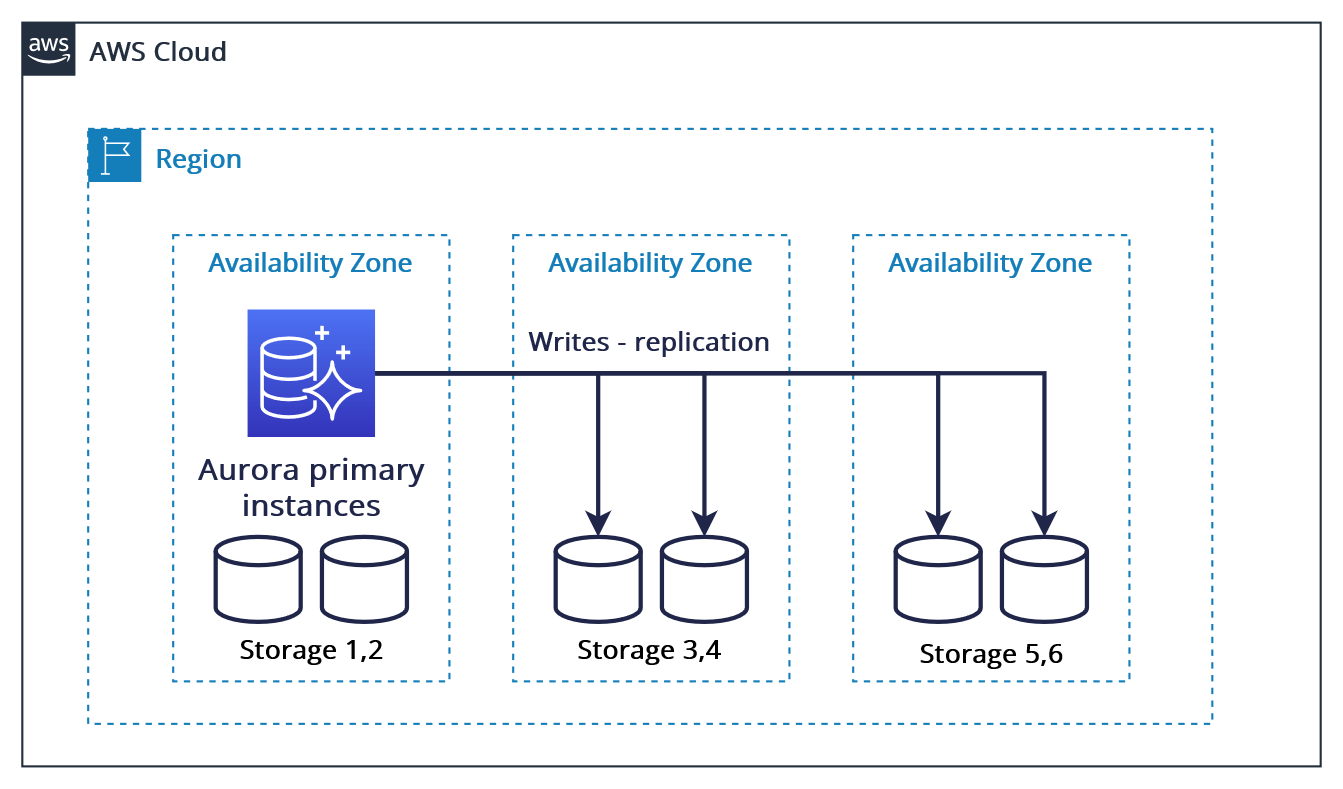
AWS supports two options of RDS for HA:

**Amazon RDS Multi-AZ Deployments**: In Multi-AZ deployments, Amazon RDS automatically generates a primary DB instance and synchronously replicates the data to a standby instance in a separate AZ.

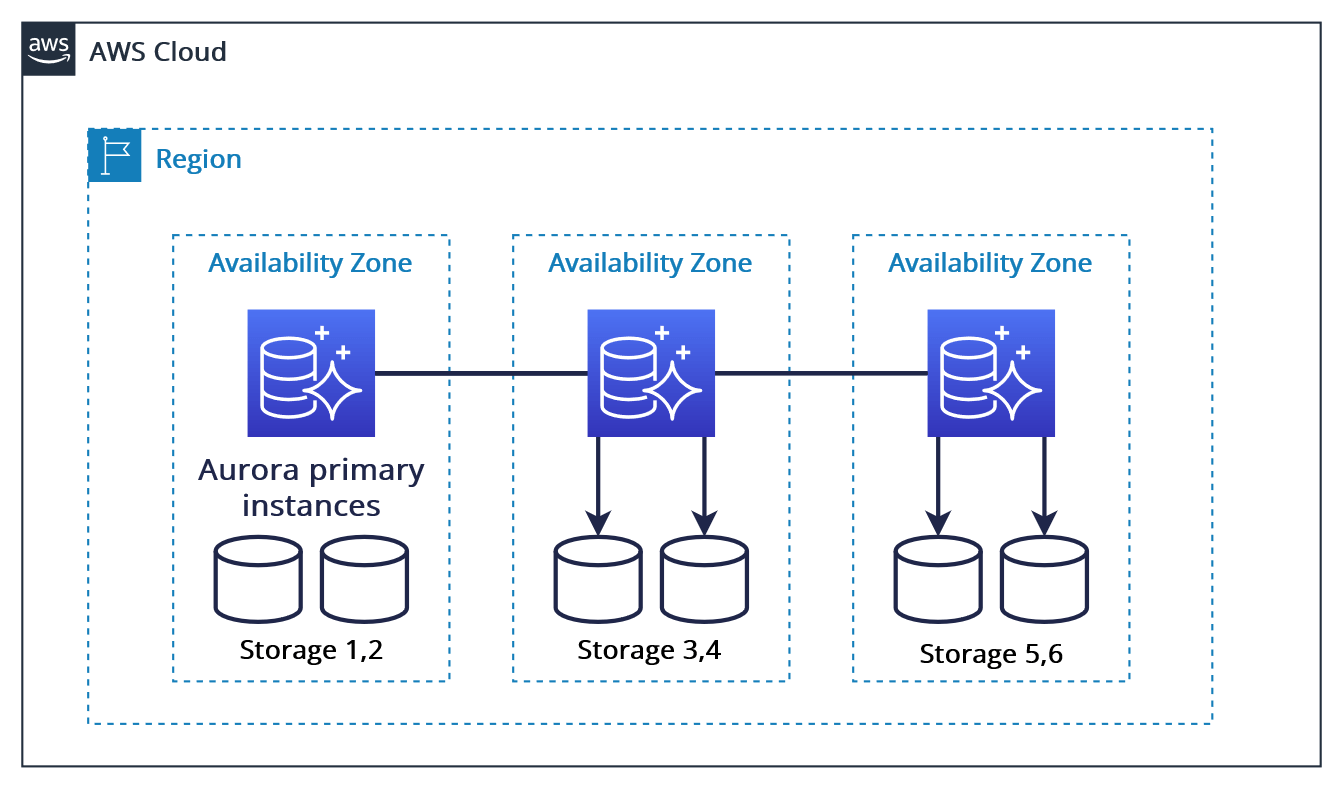
* This can be used for MySQL, MariaDB, PostgreSQL, Oracle, and SQL Server.
* In the event of an infrastructure (AZ) failure, Amazon RDS automatically switches to the backup database instance in the separate AZ.



* Amazon Aurora Multi-AZ achieves high availability by replicating data six times across three AZs. But this is only a storage layer.



We can also choose to run one or more replicas in an Amazon Aurora DB cluster. RDS automatically promotes an existing Aurora Replica to be the new primary replica and changes server endpoints in the event of a primary replica failure in a DB cluster, allowing our application to continue running without human intervention.



If a problem is detected and no replicas are provisioned, RDS will immediately spawn a new replacement DB instance for us, providing AWS RDS with high availability for all our needs.

# **High Availability for AWS storage services**

AWS offers storage for every use case and uniquely achieves high availability.

#### **Amazon Simple Storage Service (S3)**

This service provides object storage for videos, pictures, documents, and more, offering secure, durable, and scalable object storage for a vast majority of projects in AWS.

S3 has high-availability built-in and offers 99.99% availability for object storage with 99.999999999% durability (based on storage type selected).

Depending on which storage tier you choose, data in an Amazon S3 storage bucket may be stored in just one AZ. Data will become unavailable if an S3 storage service disruption affects that AZ.

#### **Amazon Elastic File Storage (EFS)**

This option provides a managed file storage solution for Linux instances and offers an SLA of four-9’s availability (99.99%). Amazon EFS can be used to provide redundancy across AZs, as well as providing the capability to mount the same filesystem concurrently from multiple Amazon EC2 instances.

#### **Amazon Elastic Block Store (EBS)**

This is a block storage service that is designed to work with Amazon EC2 instances and is highly scalable and high-performing. Although Amazon EBS storage is internally replicated to several servers within an AZ, there is no redundancy outside of this zone. To achieve this, you would need to use Amazon EBS snapshots to create copies of the data in Amazon S3.

#### **Amazon FSx**

This is a high-performance file system. As a fully managed service, it handles hardware provisioning, patching, and backups Amazon FSx protects your data against failure by automatically replicating it inside or across AWS AZs.

# **Network high availability**

One of the most important facets of highly available systems is the management of connectivity. In the event of a failure, clients need to be able to find any currently active nodes to re-establish communication and data operations.

#### **Amazon Route 53 DNS**

Provides highly available DNS services that can perform health checks on its targets and perform automatic failover for both active-active or active-passive configurations.

#### **AWS NAT gateway high availability**

NAT Gateway is a zone service. If you have resources communicating through it in multiple AZs, and if the NAT gateway’s AZ is down, resources in the other AZs lose Internet access too.

Depending on your business requirements, make sure to create NAT Gateways in at least two AZs.