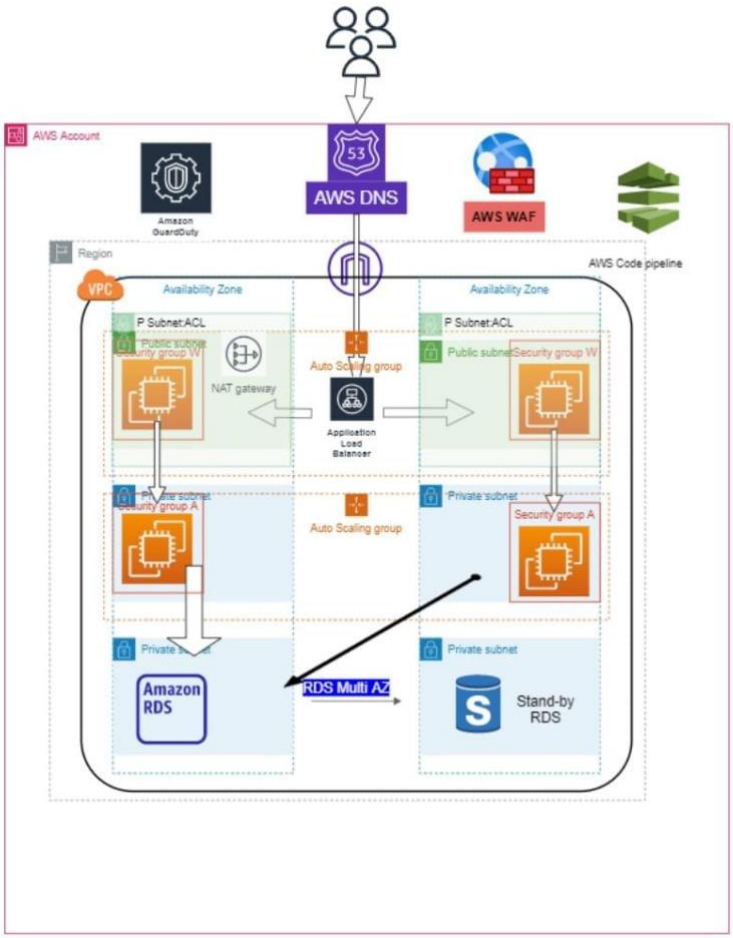
# Cloud Softway

I would like to introduce to you...

# 3-tier e-commerce app design on the AWS cloud.




- We assure you 5 main objectives in our architecture.

1. High availability
2. Scalability
3. Security
4. Cost optimization
5. High performance

## Abstract

3 tier architecture has the web tier in the public subnet, the app tier in the private subnet which will be accessible through the public subnet only and the database tier will be in another private subnet which will be accessible through the app tier. The cloud Softway company provides short-term and long-term recommendations in its designs.

# THE WEB-TIER

We will use 2 EC2 in different AZ with workload 35% max for each one. If one EC2 fails, the other will be up and running with 70% capacity so I can keep performance steady and be fault tolerant.

2 AZ because *"Everything fails, all the time" i*s a famous quote from Amazon's Chief Technology Officer Werner Vogels, anytime there is an AZ down, I will have My app intact.

Auto scaling groups to create more ec2 in a very short time through the usage spikes and terminate them in the off-peak time. App load balancer will distribute the traffic to the EC2s equally, so we can operate cost and keep the performance steady.

# **THE** APP-TIER

We will use 2 EC2 instances, one in each AZ in private subnets which will be accessible only through the public web tier. Auto scaling group and internal app load balancer will do the same work similar to the web-tier.

# **THE DATABASE-TIER**

Amazon RDS with Multi AZ feature and auto scaling enabled will help us achieve high availability, high performance and scalability. We can migrate with zero downtime to Amazon Aurora database "higher performance and bigger database clusters" as well as the app grows in the cloud. Amazon RDS is cheaper than Aurora.

Stand-by RDS in different AZ will be ready anytime there is a fault in the main database.

# SECURITY

There will be layers of security,

### Network layer

Network access control list which will allow or deny the types of traffic to and from the subnets overall.
Security groups on every EC2 which will secure our servers through inbound and outbound rules.
Using VPN between the data center and AWS.

## SERVICE-LEVEL HARDENING

For services like ALB, UDP traffic will be blocked (UDP is the protocol of choice for many DDoS attacks)
For services like AWS route53, all data at rest will be encrypted.

## IDENTITY AND ACCESS CONTROL

(**IAM**): authenticates and controls access for both internal (administrators and developers) and external (end-users, customers or partners) users.

## AUDITING AND LOGGING

**CloudTrail**: detailed information about every API call (both successful and unsuccessful) that was made to your back end.
**CloudWatch Logs**: monitor, store, and access your log files from Amazon EC2 instances, AWS CloudTrail, Route 53, and other sources.

## Security Tools

**AWS WAF:** Web apps must be saved from common attacks such as SQL injection or cross-site scripting (XSS), filter web traffic and Prevent account takeover fraud. For this we can use AWS WAF
**AWS Guard Duty:** Continuously monitor your AWS accounts and instances workloads, users, databases, and storage for potential threats.
**AWS Inspector:** Detect software vulnerabilities and unintended network exposure in near real time in AWS workloads