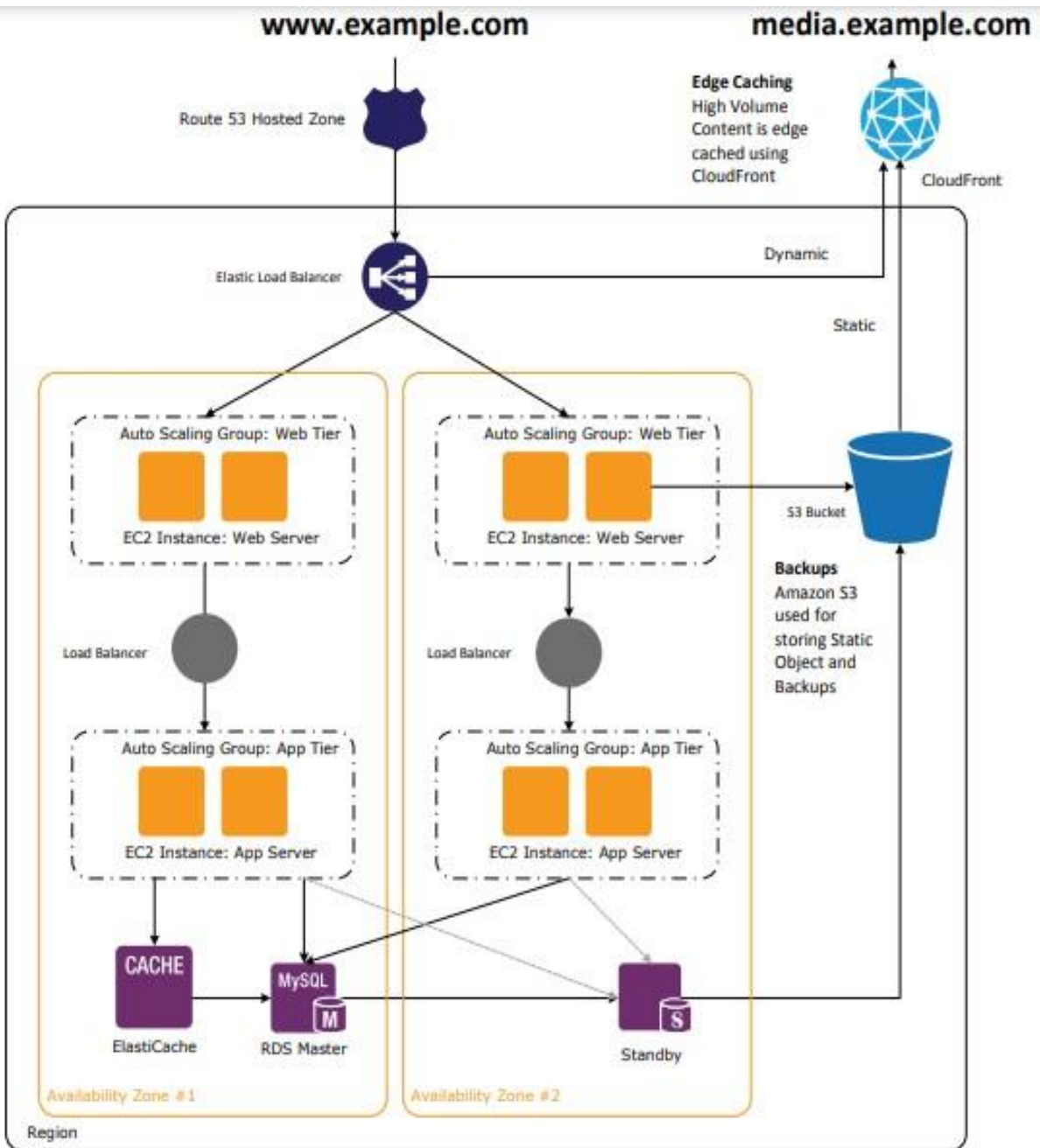


An AWS Cloud Architecture for Web Hosting

Abstract :-

Traditional on-premises web architectures require complex solutions and accurate reserved capacity forecast in order to ensure reliability. Dense peak traffic periods and wild swings in traffic patterns result in low utilization rates of expensive hardware. This yields high operating costs to maintain idle hardware, and an inefficient use of capital for underused hardware. Amazon Web Services (AWS) provides a reliable, scalable, secure, and highly performing infrastructure for the most demanding web applications. This infrastructure matches IT costs with customer traffic patterns in near-real time.

Below is another look at that classic web application architecture and how it could leverage the AWS cloud computing infrastructure:



1. **Route 53:** - Provides DNS services to simplify domain management and zone APEX support.
2. **Elastic Load Balancer:** - ELB to spread traffic to Web Server Autoscaling groups.
3. **Exterior Firewall:** - moved to every Web Server instance via Security Groups.
4. **Auto Scaling Web Tier:** - Group of EC2 instances handling HTTP requests.

5. **Backend Firewall:-** moved to every back-end instance .
6. **App Server Load Balancer:-** Software LB (e.g. HAProxy) on EC2 instances to spread traffic over app server cluster.
7. **Auto-scaling App Tier:-** Group of EC2 instances running the actual app. Instances belong to Auto Scaling group .
8. **ElastiCache:-** Provides caching services for app, removing load from database tier.
9. **DB Tier:-** MySQL RDS DB creates a highly available architecture with multi-AZ deployments. Read-only replicas can also be used to scale read intensive applications.

Web Application Architecture Best Practices & Tools

Designing an architecture is just the first step, but the success of your web application depends a lot on the architectural patterns you choose. Mind you, replicating strategies of popular web apps can do more damage than good, for oftentimes they don't complement your business requirements. To avoid such circumstances, there are few best practices you can follow. Ensure that your web app's architecture has:

- System flexibility and efficiency
- Component reusability
- Well-thought structure of code
- High Scalability
- Stability and reliability
- Easy bug-detection through A/B testing
- Utilization of security standards
- Sections to collect user feedback

Conclusion

The success of a modern web application is always closely connected to its architecture. Keeping pace with changing requirements is a challenging task and a minor mistake can cost you the life of your product.

Designing a modern web application architecture requires a professional and qualified architect who can understand the limitations and challenges that come along with it.