

Assignment Cloud Architect

Table of Contents

Objective	1
Architecture Requirements.....	1
Assumptions.....	1
Architecture Overview	2
Builds and Deployments	2
DNS Management	2
Content Delivery Network	2
Application and Load Balancing.....	2
Auto Scaling.....	3
Api Gateway	3
Database.....	3
Security.....	3
Archiving and Restoring	3
Sessions and Caching	3
AWS Identity and Access Management (IAM).....	3
Static Content and Hosting	4
Cloud Watch & Cloud Trails	4
Cloud Formation.....	4
Architecture Design Considerations	4

Objective

Design a manageable, secure, scalable, high performance, efficient, elastic, highly available, fault tolerant and recoverable architecture that allows the startup to organically grow.

Architecture Requirements

The main business requirements for the proposal are as following:

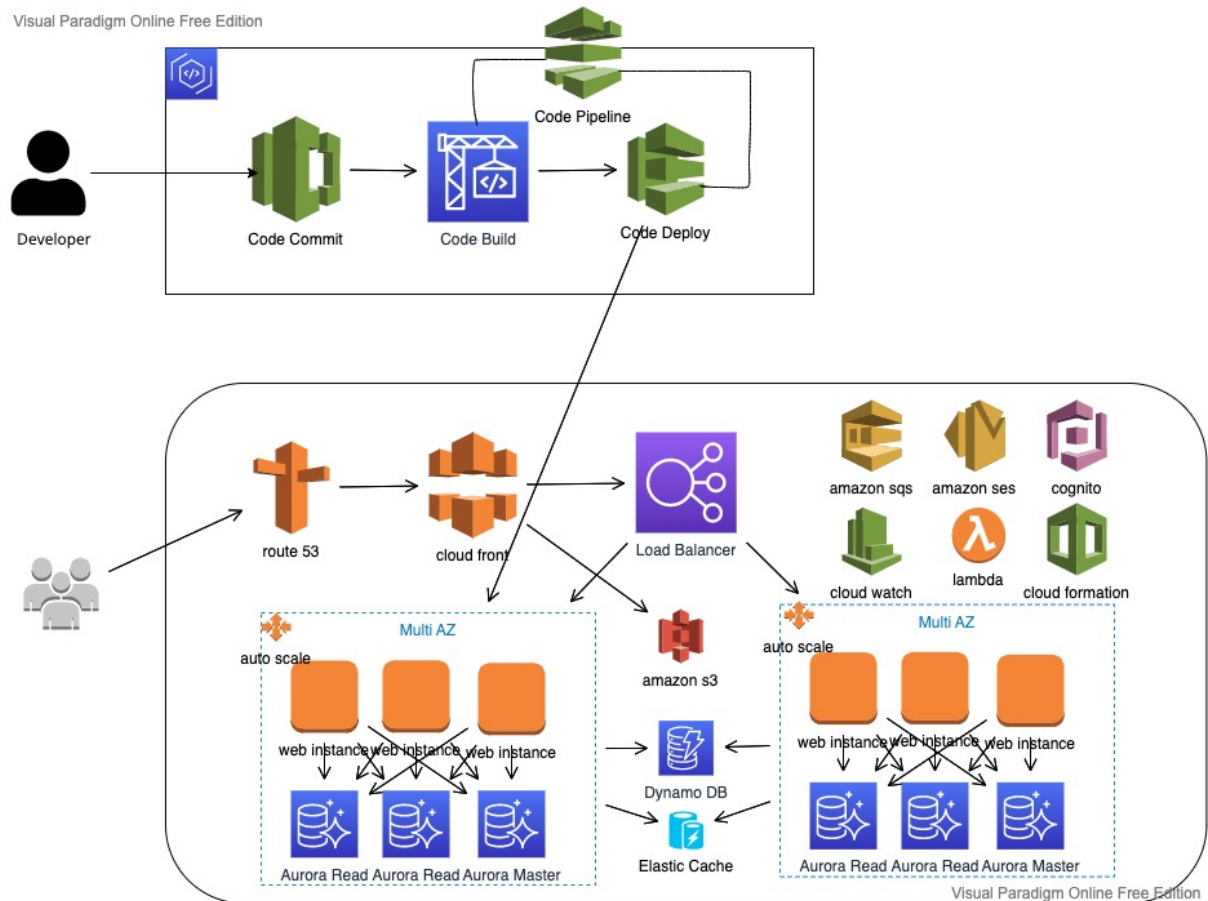
1. Development of Application
 - a) Code Repository
 - b) How the Developers commit the code
 - c) Build and Deploy
 - d) Create Environments Dev, Testing, Prod
2. API Services Layer
 - a) Creating
 - b) Publishing
 - c) Maintaining
 - d) Securing
 - e) Scaling
3. Distribute the Load
4. High performance database with read and write throughput
5. Auto Scaling and Global Users
6. Security of Data

Assumptions

Like many enterprise applications, one of the things as an Architect I design to decouple the services it offers using SOA (Service Oriented Architecture). It helps the application perform faster as the services of amazon which I mentioned below are auto scalable, high performant. I just assumed and mentioned few services here, which are very frequent to most of the applications.

- Email Service – Amazon SES
- Queuing – Amazon SQS
- Notifications – Amazon SNS
- Caching – Amazon Elastic Cache using Redis
- Cognito Services – AWS Cognito for user registrations, authentication.
- Monitoring – AWS Cloudwatch

Architecture Overview



Builds and Deployments

After analyzing the requirements, the first thing I would like to propose in Technical Architecture is to maintain the source code, automate the code builds and deployment.

I would like to propose

- AWS Code Commit for source code management. It is free for 5 active users.
- AWS CodeBuild is a fully managed continuous integration service that compiles source code, runs tests, and produces software packages that are ready to deploy. 100 Build Minutes Free with AWS Free Tier.
- AWS Code Deploy to automate the Deployment Process.
- AWS Code Pipeline fully managed Continuous Delivery Service. Its pay as you use service. No commitment. 1 Free Pipeline.

DNS Management

One of the crucial things to manage for any application is the Domains. Amazon Route 53 is a highly available and scalable Domain Name System (DNS) web service. Route 53 connects user requests to internet applications running on AWS or on-premises. Pricing – Pay as you use.

Content Delivery Network

Amazon CloudFront is a global content delivery network (CDN) that makes it easy to deliver websites, videos, apps, and APIs securely at high speeds with low latency. Built with developers in mind, CloudFront makes it easy to customize your delivery to find the perfect balance of speed, security, and cost suited to your organization's needs.

Application and Load Balancing

Elastic Beanstalk is used to host the application. The Elastic Beanstalk is configured to use Multi AZ, which makes the application to be Fault Tolerant. Elastic Load Balancer is configured to distribute the network traffic to the Availability zones thus making it high performant. Beanstalk provides option to create multiple environment with RDS, Elastic load balancer, Autoscaling etc. for dev, test and production.

Auto Scaling

The architecture is designed in such a way to Horizontally and Vertically Scale the Application EC2 Instances using Auto Scaling. It is the one of the services of Amazon by configuring which we can Scale in and Scale out the instances depending on the metrix. Such as traffic, CPU performance etc..

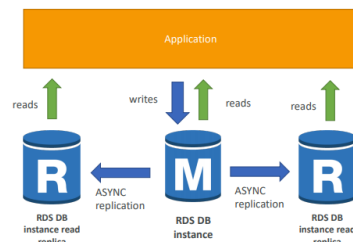
Api Gateway

Amazon API Gateway is a fully managed service that makes it easy for developers to create, publish, maintain, monitor, and secure APIs at any scale. APIs act as the "front door" for applications to access data, business logic, or functionality from your backend services. Using API Gateway, you can create RESTful APIs and WebSocket APIs that enable real-time two-way communication applications. API Gateway supports containerized and serverless workloads, as well as web applications. 1 million API calls received free per month for 12 months with the AWS Free Tier.

Database

One of the important aspects of the application is managing the data. How the data is stored and retrieved. For this application, Amazon Aurora is the best database as it is relational and high performant. The Database is configured in multi az environment to make sure it is fault tolerant. Also, it is configured to Auto Scaling option, which is made sure to act in performance bottlenecks.

To perform high, the Database will be configured with Read and Write Replicas.



Reference: <https://bikramat.medium.com/rds-read-replicas-vs-multi-az-601cd1edb283>

Data is very important. In order to Secure and encrypt the data, Encryption of data at Rest is used. Which can be configured by using **KMS**.

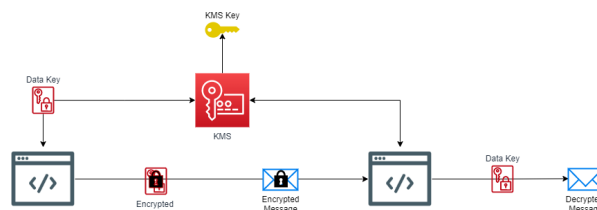
Database Backup is provided default by Amazon RDS which is by default 7 days. The Backup can be stored on S3 to make sure of disaster.

A full database snapshot is taken every day by default.

Security

Security is one important concept as now a days protecting data is very important. Amazon provides easy way to secure data by providing encryption of data. There are different ways, but for this application, we will use Encryption at Rest.

KMS is one of the services which offers this Security.



Reference: <https://awswith.net/2022/01/01/encrypting-data-in-net-apps-using-aws-key-management-service/>

Archiving and Restoring

The Amazon S3 Glacier storage classes are purpose-built for data archiving, providing you with the highest performance, most retrieval flexibility, and the lowest cost archive storage in the cloud. All S3 Glacier storage classes provide virtually unlimited scalability and are designed for 99.999999999% (11 nines) of data durability. The S3 Glacier storage classes deliver options for the fastest access to your archive data and the lowest-cost archive storage in the cloud.

Sessions and Caching

As the user base grows one of the things to make the application perform high is to cache the frequent database queries. It can be done by using AWS Elastic Cache Redis.

Also, AWS Dynamo DB is used to store the sessions related to application users which makes the user experience wow, by storing user preferences. DynamoDB is fast, scalable, easy to set up, and handles replication of your data automatically.

AWS Identity and Access Management (IAM)

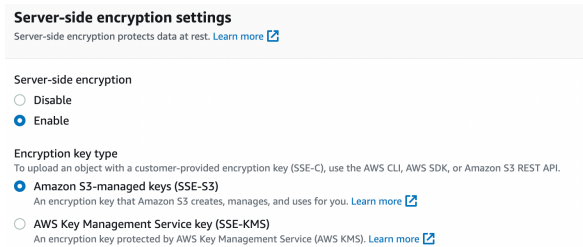
AWS web service that helps you securely control access to AWS resources. You use IAM to control who is authenticated (signed in) and authorized (has permissions) to use resources.

Static Content and Hosting

The Amazon S3 is used to store the static content

- Images
- Files, Javascript and CSS
- Database Backups and Snapshots.

Amazon S3 is a global service. It is designed fault tolerant. Object versioning is enabled. Server Side Encryption is enabled for security.



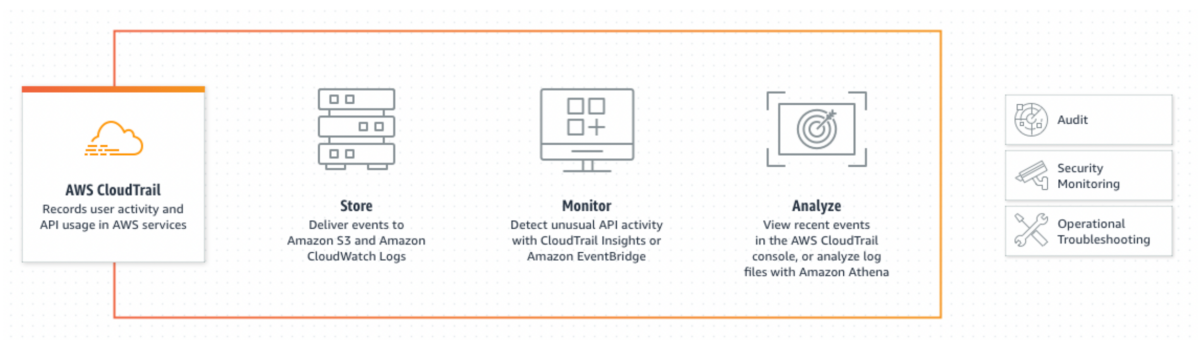
Amazon S3 is also used to host static websites. So, I designed to host Mobile Application on S3. Which makes it scalable and high performant.

Cloud Watch & Cloud Trails

One of the most important considerations while designing a system is to constantly monitor the system, record the logs, events and metrics.

Amazon CloudWatch is a monitoring and observability service built for DevOps engineers, developers, site reliability engineers (SREs), IT managers, and product owners. CloudWatch provides you with data and actionable insights to monitor your applications, respond to system-wide performance changes, and optimize resource utilization. CloudWatch collects monitoring and operational data in the form of logs, metrics, and events and can be stored in S3.

AWS CloudTrail monitors and records account activity across your AWS infrastructure, giving you control over storage, analysis, and remediation actions.



Reference: <https://aws.amazon.com/cloudtrail/>

Cloud Formation

AWS CloudFormation is a service that helps model and set up AWS resources so that you can spend less time managing those resources and more time focusing on your applications that run in AWS.

You create a template that describes all the AWS resources that you want (like Amazon EC2 instances or Amazon RDS DB instances), and CloudFormation takes care of provisioning and configuring those resources for you. You don't need to individually create and configure AWS resources and figure out what's dependent on what; CloudFormation handles that.

Architecture Design Considerations

When designing the architecture, there are many considerations and standards based on which I have designed this architecture.

- The Technical Architecture diagram is an illustrate of all the services which can be used in the application development.
- Security, all the data are encrypted by using the latest protocol tls.
- Reliability, I made sure to design a fault tolerant architecture.
- High performance and serverless architecture.
- Operational Excellence.
- Used the latest services offered by AWS, to ensure the sustainability (Latest Hardware).