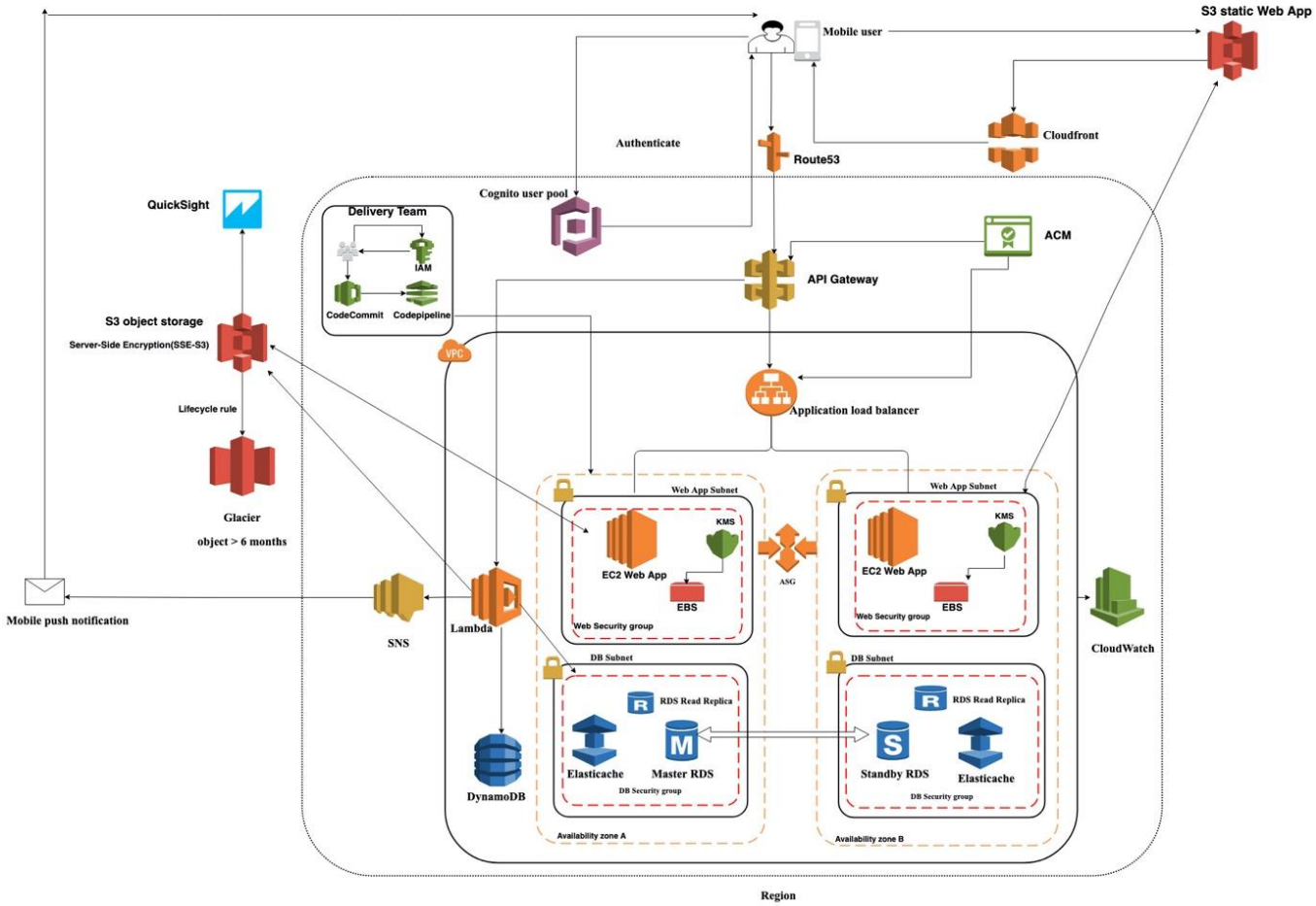
AWS Solution Architecture Design – Startup use case



Mobile application that allows customers and service providers to interact real time

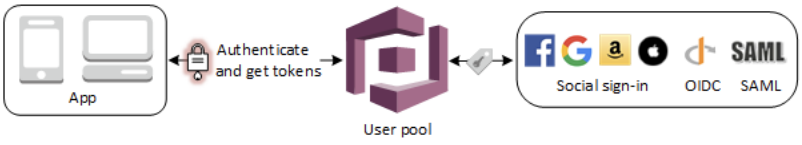
Solution approach

Proposed above is a high level architecture reference for start-up company planning to launch a new mobile application that allows customers and service providers to interact real time. The above architecture contains existing web-based architecture with proposed mobile-based solution.

Solution workflow

User Authentication

- Customers and service providers will sign-in their mobile app to interact real time through Cognito. Cognito will also enable sync user specific data across multiple devices.
- Once the mobile user has received an identity, the user is granted access to application.
- Mobile application uses combination of web identity and federation with AWS Security token service(STS).
- AWS Cognito can provide user authentication and authorization with Cognito user pools, social identity providers



AWS Solution Architecture Design – Startup use case

- Mobile app will communicate to backend application using Amazon Route53 → API gateway and Application load balancer.

Presentation tier

- Presentation tier consists of Android and IOS application that includes the user interface and presentation logic of the application.
- Application static files and media contents are hosted in AWS S3 bucket and distributed via CloudFront for faster access.
- User generated static media files are also stored in AWS S3 and distributed using CloudFront for low latency faster access.
- Presentation tier interacts with Logic tier using API gateway. Mobile users send requests to API gateway to access application logic and dynamic data.
- API gateway act as an entry point to application logic running in Lambda function.

Logic tier

- Logic tier contains the business logic of the application which is handled by AWS Lambda functions.
- Making use of Lambda functions provide highly scalable backend infrastructure that is need to support the mobile user request. Lambda code runs in response to requests and automatically manages and scales the underlying resources. And also eliminates the need to mange servers.
- API gateway endpoint invokes the Lambda functions and Lambda functions automatically scales up and down to match the request traffic.
- Lambda functions internally communicate with the Data tier and dependency to execute the desired logic. Lambda function provides endpoint for users to store and retrieve app data on DynamoDB.
- Lambda function integrates with AWS SNS to send real time push notifications. Lambda function formats communication requests and send push notification to specific users via SNS.
- Lambda provides an asynchronous endpoint for users to communicate each other.

Data tier

- **DynamoDB : -** Fully managed, scalable and high available NoSql data store for storing and querying unstructured app data. DynamoDB provides automatic high availability by replicating data across multiple availability zones with in that region.
- DynamoDB provides least privilege to Lambda functions to query specific data.
- **S3:-** Highly available, durable and scalable object storage for storing media files, data which can be processed by Lambda functions for executing business logic. S3 events can invoke Lambda function based on events like object creation or deletion.
- S3 can be used to store static contents, which can be delivered by CloudFront. And also to store application specific objects or user specific objects retrieved by Lambda functions.
- **Glacier :-** Data stored in S3 can be archived to Glacier by applying lifecycle policy to S3 bucket.

Push Notification

- **AWS SNS(Simple notification service) :-** SNS is fast , flexible, cost effective solution to send push notifications to mobile app, which appears as a notification alert on mobile app. Lambda integrates with SNS to send the notification after processing the business logic.

Security

- Security at rest is implemented by enabling encryption with KMS for EBS . For S3 server side encryption is enabled and support secure data transit with SSL.
- ACM is used to store and access certificate for API gateway and ALB.
- Access to AWS resources is managed by IAM and security groups used in EC2 and RDS for controlled access.
- RDS DB instances are encrypted for additional security.

Analytics: Mobile App data / user data stored in S3 bucket can be analysed and visualized using AWS Quicksight . Quicksight contains dashboard to provide insight to S3 analytics data. Also, Quicksight allows to query data.

AWS Solution Architecture Design – Startup use case

Disaster recovery

- AWS services used in the architecture offer built in fault tolerance and high availability by using multiple Availability zones in each regions and to help protect against individual server or data centre failure.
- Design Lambda for high availability by selecting multiple subnets in different Availability Zone(AZ).
- We need to make sure sufficient IP address are allocated to each subnet to handle concurrent Lambda request.
- Enable versioning and cross region replication in S3.
- The above architecture can be converted to CloudFormation template and these templates can be easily deployed in standby regions in case of disaster. And route traffic using Route53 fail over routing policy.
- Frequently taking AMI, EBS snapshots and RDS snapshots.

AWS Services used for Mobile App architecture

AWS Cognito:- Cognito allows Mobile users to authenticate and authorise to app. Cognito works with external identity provider that support SAML or OpenID connect and Social identity providers.

AWS S3:- Company can use S3 to store static contents and also storing app specific objects and mobile user data files.

AWS Glacier:- Data stored in S3 can be archived to Glacier by lifecycle policy.

AWS QuickSight: Integrates with S3 bucket to perform analytics and visualise data collected.

AWS CloudFront:- Static contents stored in S3 can be delivered via CloudFront. CloudFront read and cache all static contents in global edge locations and can be accessed with low latency.

AWS Route53: For creating custom domain name that can be provided to the API users. Fail over routing policy can be used in case of disaster recovery

AWS API gateway: For creating HTTPS API that can be used by clients to access the logic tier.

AWS ACM: Associate ACM SSL certificate with ALB for HTTPS. Also associate ACM certificate with API gateway for configuring custom domain.

AWS Lambda:- Lambda is used for backend code execution without managing servers. Lambda function executes the core business logic. Using Lambda start-up can create an event driven architecture that is cost effective and highly scalable.

AWS DynamoDB:- DynamoDB can be used to store app data by the Lambda function. DynamoDB is fully managed NoSQL database that is highly scalable and cost effective and able to handle high request.

AWS SNS: Simple notification service is used to send push notification to mobile users. Lambda integrates with SNS to send the real time push notification.

AWS CloudWatch:- CloudWatch can be used to monitor the AWS resources like Lambda, DynamoDB, ALB API gateway . Start-up can collect real usage data of the resources.

AWS CloudFormation: CloudFormation can be used to create the architecture resources so that it is easy to track changes and can managed via code repositories. So that the entire application can be deployed in another region in case of disaster.