

## Technology Stack

- C#
- Windows Server
- PHP
- Amazon Web Services (AWS)
- Route 53
- Elastic Load Balancing (ELB)
- Elastic Compute Cloud (EC2)
- Amazon RDS
- Amazon S3
- ElastiCache
- CloudWatch

## Abbreviations

- EC2 Elastic Compute Cloud
- AWS Amazon Web Services
- S3 Simple Storage Service
- MFA Multi Factor Authentication
- EBS Elastic Block Storage
- IAM Identity and Access Management
- SQS Simple Queue Service
- AZ Availability zones
- SNS Simple Notification Service
- VPN Virtual Private Network
- ELB Elastic Load Balancer
- SPOF Single Point Of Failure

## Overview of TimeTec Cloud

TimeTec Cloud is a powerful web-based solution for automated time attendance for various kinds of businesses. Built with Microsoft.NET framework and MySQL Relational Database Management, TimeTec is basically an advanced web-based version of the FingerTec TCMS V2 and this application has vast potential for use in SMEs or even large multinational companies (MNC).

TimeTec centralizes all FingerTec terminals to the server where you can control, manage and download the transaction logs all from one place. On top of that, we also have the TimeTec Mobile application for iPhone/ iPad/ iPod Touch and Android device users which allows the user to check attendance records and to clock-in their attendance from a GPS-tagged location on-the-go, as long as an Internet connection is available. FingerTec is an attendance management solutions provider.

FingerTec has decided to leverage AWS cloud services to create a multi-tenant version of TimeTec system to reduce operational cost and to effectively utilize the infrastructure. In addition, FingerTec would like to utilize the cloud to ensure consistent performance; automatic scaling of the infrastructure based on load and high availability of its TimeTec services.

The front end web application is accessed via another PHP web front end which takes care of authentication and acts as the product information portal for TimeTec. <http://www.timeteccloud.com/>

## Introduction

TimeTec Cloud Service consists of the following 3 main cores:

- System Study and Architecture
- Cloud Engineering
- Infrastructure Design

## System Study And Architecture

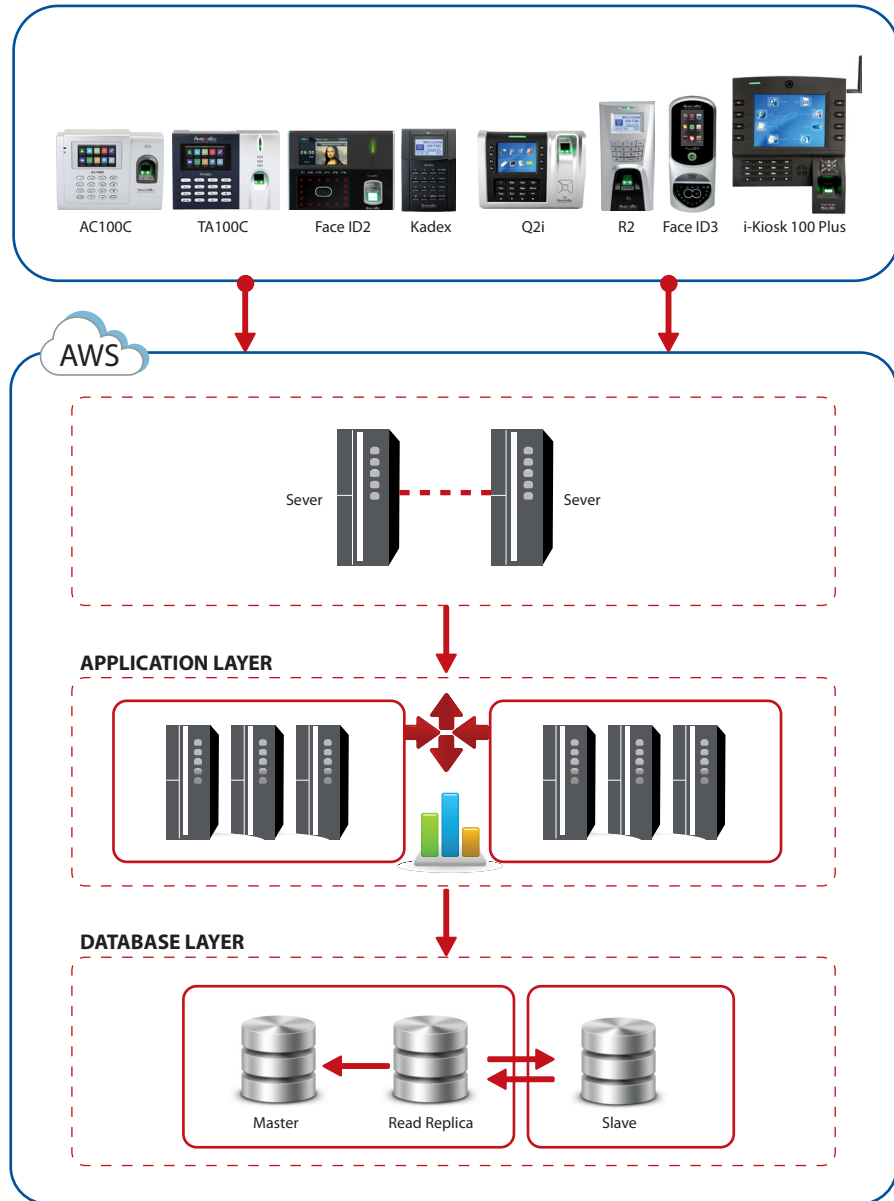
An in-depth technical study of TimeTec Cloud application as well as the Amazon Web Services Platform has been performed which covered all the major layers of the application such as the front-end web layer, database, file storage, external integration points and user/system generated files. FingerTec has designed the infrastructure architecture based on the input gathered during the system study.

The recommended architecture will follow best practices for setting up scalable and highly available infrastructure on AWS (Examples: Loose coupling, Design for failure). Some objectives for this exercise are:

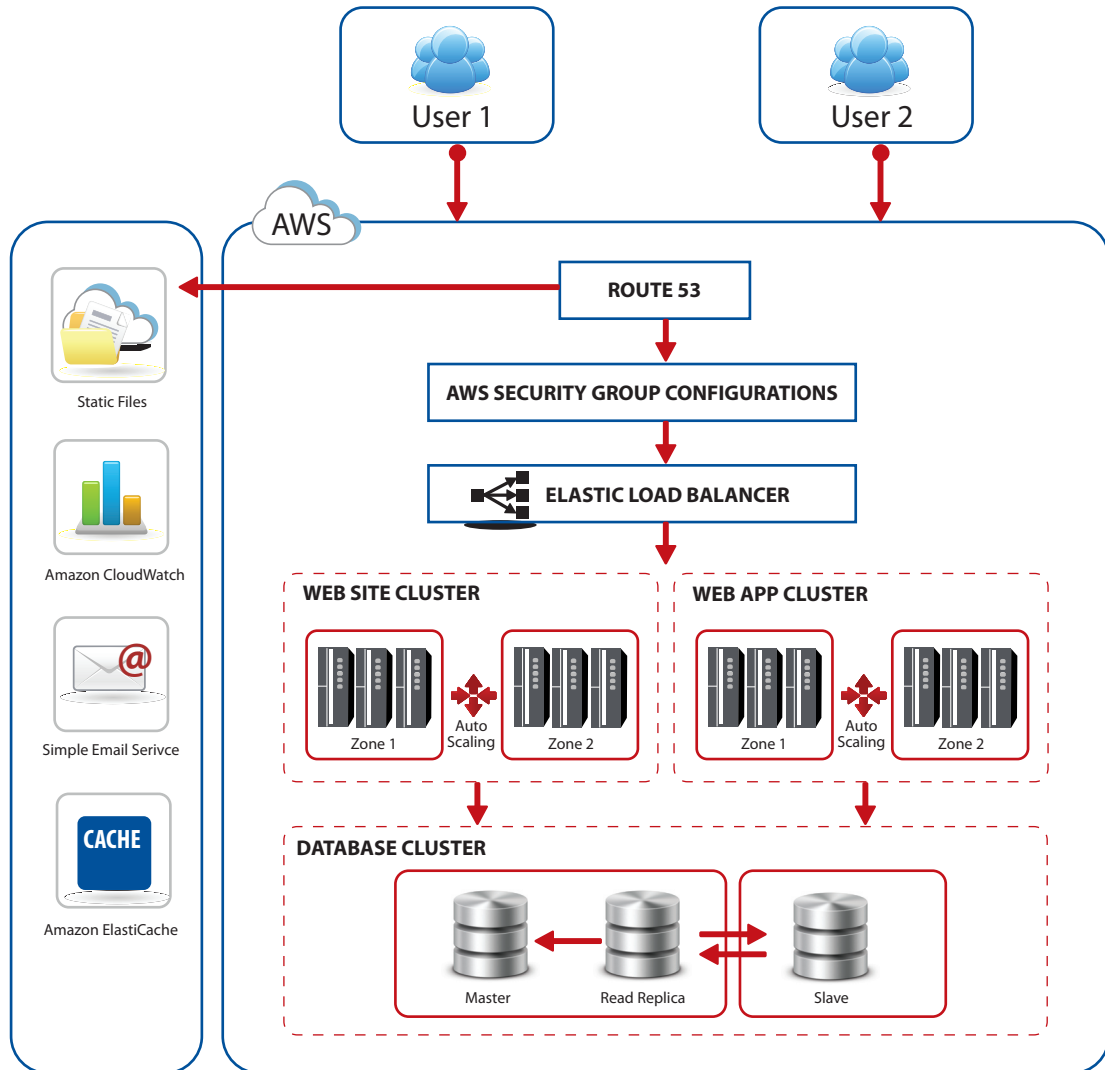
- ✓ High Availability using multiple AWS Availability Zones within a region
- ✓ Failover of Amazon EC2 instances (Web/App) using Amazon Auto Scaling
- ✓ Scalability of Amazon EC2 instances (Web/App) using Amazon Auto Scaling
- ✓ Load Balancing using Amazon Elastic Load Balancing
- ✓ Amazon RDS (MySQL) Database in High Availability Mode with X Read Replica Slave
- ✓ Monitoring of infrastructure using Amazon Cloud Watch metrics
- ✓ Infrastructure related alerts via Email using Amazon SNS
- ✓ Backup configuration of Amazon RDS database and Web/App layer
- ✓ Amazon S3 and Amazon EBS for Storage

Based on the study, TimeTec Cloud will contain 2 types of architecture setup as follows.

Device To Cloud



Web Frontend



## AWS Advantage

- Elasticity**  
 New EC2 instances can be automatically added during Peak loads and reduced down during valleys
- High Availability**  
 Web, Application and Database instances can be run on Multiple Availability Zones inside a Region for High Availability
- Fault Tolerance**  
 Inherently Fault tolerant building blocks like S3, EBS, CloudWatch, SQS, SNS, SES can be used for Storage, Monitoring and Messaging
- Security**  
 Amazon Security groups, IAM policies, Secure Access through Keys can be leveraged for security
- Flexibility**  
 Multiple Instances Types/Capacities, Full permissions and Mix Match option

The architecture involves the following AWS Infrastructure components like:



### Amazon EC2

Web, Application and Database servers will be hosted as Amazon EC2 server instances.



### Amazon S3

AMI's, Logs, Snapshots, Backups and static assets can be maintained in Amazon S3.



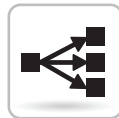
### AWS Security groups & Firewall

Security groups will be configured in Amazon firewall to allow port/IP range access to/from internet.



### Amazon Regions/AZ

Web, Application servers, Database and Load Balancers will be setup on Multiple Availability zones inside an Amazon Region.



### Amazon Elastic Load balancers

Software Load balancers HAProxy/Nginx/Amazon ELB will be used for load balancing.



### Amazon Auto Scaling

Amazon Auto Scaling and Custom Scaling Scripts will be used to enable Auto Scaling of the Web and Application Servers.



### Amazon EBS

Elastic Block storage will be used for persistent file storage. EBS will be attached/detached with relevant server instances.



### Amazon Elastic IP

Elastic IP is a public IP that will be attached with relevant EC2 instances.



### Deployment Automation

Automated Deployment tools / scripts will be configured for application and system patches.



### Amazon CloudWatch

Components of Amazon Web Services like EC2, EBS etc. will be monitored using AWS CloudWatch.



### Amazon SNS

Notifications, Alerts triggered from CloudWatch will be published to the System administrators using SNS.



### IAM

Identity Access Management enables you to securely control access to AWS services and resources for your users.

## SAAS - Multi Tenancy

Multi Tenancy is an essential component of any SAAS implementation and it helps in reducing the overall cost of the infrastructure used by all customers. MultiTenancy within TimeTec can be enabled by introducing a simple abstract layer that offers to identify the tenant and guides the application to ensure that the data sent by devices can be saved in the right data store for the given customer. The introduction of the abstract layer would involve most probably a single change within the application layer where the database connection string is identified to read/write into the data store.

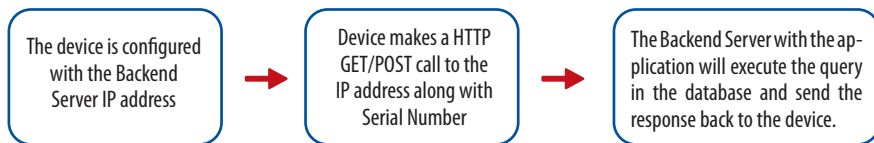
## Cloud Engineering

TimeTec Cloud involves a great amount of coding. Cloud Engineering is a process of getting our internal cloud developers to implement cloud related portions of the architecture. Our cloud developers are well versed with the AWS cloud services API and involved in the development of scalable/highly available architecture which will be implemented in TimeTec Cloud.

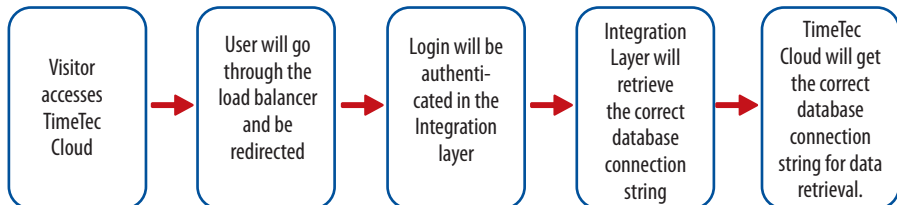
During the Cloud Engineering stage, our cloud developers has developed an integration layer for the front end application to communicate with the backend process. It is to automated the whole TimeTec Cloud into fully Do-It-Yourself concept from the moment you start your purchase until you start using the TimeTec Cloud service.

The overall flow the application data is represented below:-

### From Device to Database



### Web Frontend



With multi Tenancy in place, the application software can be scaled based on load and performance needs, thereby reducing the overall operational cost of the entire infrastructure. A simple comparison table between Single/Multi Tenancy is given below:-

METRIC	SINGLE TENANT	MULTI TENANCY
<b>Infrastructure</b>	Fixed. Provisioned during the initial setup and requires constant updates when new devices are added. Device -> Application	Flexible. Provisioned partially during initial setup and the rest is auto scaled. Device -> Bridge -> Application Tier
<b>Scaling</b>	No Scaling	Auto Scaled based on load
<b>Cost</b>	Fixed and High	Variable and Low
<b>Utilization</b>	<b>Low</b> Since the device is mapped to the application, if the customer is on a lower plan, he might not be utilizing the servers fully.	<b>High</b> Since any customer can use any application tier server, the utilization is maximum keeping the overall cost low.

## Infrastructure Design

FingerTec set up the infrastructure in line with the architecture design illustrated in the earlier section.

The approximate Scope of Work is presented below:

### **Capacity and EC2 instance provisioning:**

- Provisioned the required infrastructure capacity in terms of the minimum number of Amazon EC2 instances required are Instance Types, Launch Types, Regions And Availability Zones.
- Only single region configuration / setup were considered in this scope.
- Provisioned and configured EC2 instances for Web/App layer, RDS DB and ElastiCache layers.
- Provisioned and configured custom EC2 Instances as required to suit FingerTec's application architecture.
- Custom load balancers will be implemented in this EC2 Instance launched for FingerTec's functional operations.

### **DNS Setup:**

- Configured the relevant domain entries in Amazon Route 53 (only in R53 scope) according to the architecture recommendation.

### **Security and IAM:**

- Configured the required firewall / security group rules for / between the ELB, Web/App layer instances, storage layer and database layer.
- Provisioned and setup the required IAM (Identity and Access Management) users and policies.

### **Load Balancing Layer:**

- Provisioned Amazon Elastic Load Balancing for the web setup.
- Configured Web/App layers with Amazon Auto Scaling.
- Configure the load balancer to point to multiple Amazon EC2 servers to enable no-single-point-of-failure of Web/App Layer.
- Configured Port forwarding, health check frequencies, Load balancing algorithms.
- Configured ELB with Amazon Auto Scaling wherever identified.
- Provision and configured the custom load balancer for the bridging server setup.

### **Storage and CDN: Configure and Provision required.**

- Number of S3 buckets and access policies for Snapshots, logs, archives, CDN etc.
- EBS and Ephemeral disks for application storage.
- Required backup policy and implement the snapshot backup solution for the Amazon EC2 servers.

**AWS ElastiCache Layer:**

- Configured the identified number( 2 node) of AWS ElastiCache nodes in the identified AZ/Region.
- Passed the ElastiCache configurations/credentials to FingerTec's team for corresponding changes.
- FingerTec's team used these ElastiCache properties in their configuration sections accordingly.
- Configured AWS CloudWatch Monitoring on the ElastiCache Nodes.

**Database Layer:**

- Configured and Provisioned RDS MySQL instance for Primary Database
- Configured Standby RDS instance for High Availability in Multi-AZ Mode
- Configured the DB security groups to allow DB access from specific machines
- Configured the DB Backup Snapshots, Periodic Dumps , Backup retention period
- Configured the CloudWatch Monitoring for the RDS Master
- Configured 1-2 Read Replica's in the cases needed

**Monitoring:**

- Configured Amazon CloudWatch to monitor important metrics
- Enabled Detailed CloudWatch monitoring for the Web/App layer, Elasticache and Database layer instances
- SNS Integration for alerts on infrastructure parameters such as CPU Utilization, Disk Usage, etc. (parameters available and monitored by CloudWatch)
- Identify and configure the 3rd party Website Health monitoring tool (if required)
- Backup:  
Created custom scripts for backing up Application Deployment Files in Web/App Layer  
Enable RDS snapshot based backups and Periodic Dumps for RDS database layer  
Error logs and System logs generated in Web/App Layer are rotated and uploaded to S3 using custom scripts

**System Patches:** Updated the Operating System, Application server and other server patches at regular intervals and manage the AMI versions inside AWS infrastructure

**Application Patches:** Deployed the application executable to the AWS production environment. FingerTec Development team will pass the detailed deployment instructions steps/guides to FingerTec Deployment services team.

**Security Configuration:** Made modifications in AWS Security groups to open new ports IP ranges /close existing ports and IP ranges and manage the IAM on the AWS end.

**AMI Bundling:** Created base AMI (Amazon Machine Image) once the application has been setup and is implemented for Web/App Layer. This AMI will be used for AutoScaling.

**Support:** Registered for the required AWS premium support after providing the details about various support plans available at AWS. AWS Premium Support cost is additionally payable to AWS.