# CogniFit Technical Security Details

# Table of Contents

# 1. Security

## 1.1 Servers

CogniFit servers run within Amazon Web Services (AWS) in Elastic Compute Cloud (EC2) service. This EC2 from Amazon is a key component in Amazon's Infrastructure as a Service (IaaS), providing resizable computing capacity using server instances in AWS's data centers. Amazon EC2 is designed to make web-scale computing easier by enabling us to obtain and configure capacity with minimal friction.

Security within Amazon EC2 is provided on multiple levels: the operating system (OS) of the host platform, the virtual instance OS or guest OS, a firewall, and signed API calls. Each of these items builds on the capabilities of the others. The goal is to prevent data contained within Amazon EC2 from being intercepted by unauthorized systems or users and to provide Amazon EC2 instances themselves that are as secure as possible without sacrificing the flexibility in configuration.

CogniFit front-end servers are located behind a firewall that allows access only from IP defined in the security groups within EC2.

CogniFit backend servers are deployed using AWS Elastic Beanstalk, a service that uses the infrastructure provided by EC2, with the same characteristics for security.
More information about AWS security: https://aws.amazon.com/security/

## 1.2 Databases

CogniFit databases are deployed within AWS Relational Database Service (RDS). Amazon RDS is a managed service that allows to set up, operate and scale a relational database in the cloud, while automating administration tasks such as hardware provisioning, database setup, patching and backups.

At CogniFit we use two different databases to store user personal data and user cognitive data separately. Therefore, personal data and cognitive data are not directly linked. Our frontend servers which are where the users connect directly, send the data collected to the first database. Only these servers can access the first database and cannot communicate with the second database. All communication between the users and the frontend servers is encrypted. Data is also encrypted at rest using AES-256 encryption algorithm.

On the other hand, the second database is only accessible from the backend servers. Users cannot interact or send data to the backend servers. All communications from the frontend servers to the backend servers is encrypted. Data is encrypted at rest using AES-256 encryption algorithm.

Therefore, since both databases are isolated, the access to them is done from different servers, and we store a different kind of data on each of them, it is very difficult to relate individual personal data with individual cognitive data.

CogniFit databases are not publicly accessible and they are located behind a firewall that allows access only from certain servers within the CogniFit network.

## 1.3 Network configuration

All CogniFit servers are located in an Amazon Virtual Private Cloud (VPC), an isolated section of the AWS Cloud, that allows us to have control over the virtual networking environment. The servers are on the same subnet and they communicate to each other through the internal network.

These servers are located behind a firewall and access to them is controlled by specific security groups. To prevent unauthorized access to servers, the security groups define the IPs that can connect to a server and to which ports they can connect.

In this case, database servers, in RDS, don't have a public IP and can only be accessed from within the CogniFit VPC. They also have assigned a security group that only allows access from certain servers within the VPC. Direct connections to the databases are done through a VPN, and only a few administrators have access to the VPN since it is limited through firewall rules and is password protected.

Public communications to CogniFit frontend servers is limited to allow only HTTPS traffic. This means that the data sent to the frontend servers is encrypted and in consequence, all the communications to the backend servers are done through HTTPS being also encrypted.

All requests made to the frontend servers go through AWS Web Application Firewall. AWS WAF helps us to detect and block malicious web requests. We have rules to filter web traffic based on conditions that include IP addresses, HTTP headers and body, requests rate or custom URLs.

Due to this an additional layer of protection from web attacks that attempt to exploit vulnerabilities is created. We can create or modify this rules any time without affecting the good web requests. We also use AWS Shield that protects our servers from common DDoS (Distributed Denial of Service) attacks.

## 1.4 Access control

Full administrator access to AWS is limited to few administrators and requires two-factor authentication.

Developer access is controlled through AWS Identity and Access Management policies, and developers have only access to limited resources inside AWS. Developers can't ever access to the personal data which is protected by law. They can access CogniFit servers using SSH key-based authentication with a password protected key. Each developer has his own key, passwords are not stored anywhere and each developer has access to a limited number of servers.

## 1.5 User data

Access to CogniFit product is password protected. User passwords are never stored as clear text and they are hashed before being stored in the RDS database inside our VPC which doesn't have public access.

All user data is sent to CogniFit servers encrypted via SSL using HTTPS. As mentioned before all encrypted communication with the backend servers is also done using HTTPS.

CogniFit does not store any financial data from users. Financial data is collected by specialized services as Stripe and PayPal. This typically includes encrypted communication via HTTPS.

# 2. Availability and backups

## 2.1 Availability

Availability of CogniFit service is ensured by distributing the workload between different servers located in separate availability zones, across Amazon Web Services.

All frontend servers are behind a load balancer and distributed between at least two Availability Zones in Amazon Elastic Cloud Computing (EC2). The load balancer provides both balancing load and fail over capability in case a server fails.

For our backend, servers there is a similar configuration, a load balancer distributes traffic between at least two servers located in different Availability Zones.

In regards to data, it is all stored in AWS Relational Database Service (RDS). The databases are deployed using the Multi-AZ option. This allows for a high availability and automated fail-over from the primary database to a synchronously replicated secondary database. In case of failure of the primary database, the system automatically performs an automatic fail over to the standby database.

## 2.2 Backups

Amazon's RDS automated backup feature enables point-in-time recovery for the database instances. This allows us to restore the database instances to any second during the retention period, up to the last five minutes.

Database snapshots are taken daily and stored in Amazon S3. Amazon S3 provides a highly durable storage infrastructure designed for mission-critical and primary data storage. Amazon S3 redundantly stores data in multiple facilities and on multiple devices within each facility. To increase durability, Amazon S3 synchronously stores your data across multiple facilities before confirming that the data has been successfully stored. In addition, Amazon S3 calculates checksums on all network traffic to detect corruption of data packets when storing or retrieving data.

A weekly backup of the databases is encrypted and stored outside AWS, so if there is a fatal failure in AWS, user's data is safe outside AWS infrastructure.

# 3. Monitoring, logging, and testing

## 3.1 Servers monitoring

We use AWS CloudWatch and an internal monitoring tool to measure and analyze the different parameters of the servers. If any of the metrics are out of bonds, an alert email is sent to the IT team. So the team can take the actions required to get the server back into a normal state.

The different parameters we monitor are CPU usage, free memory, drive space usage, number of connections to the databases, among many others. This continuous monitoring allows us to perform scaling activities like changing server capabilities or increasing/decreasing the number of frontend or backend servers behind the load balancers automatically, according to the needs of the infrastructure at any time.

## 3.2 Performance monitoring

We use a mix of third-party services, Newrelic and Pingdom, to measure the performance and availability of the CogniFit website and services. When there is an availability issue, an alert email is sent to the administrators and main developers, so immediate action can be taken to solve the issue.

## 3.3 Access logging

We use AWS CloudTrail to log all configuration changes to our AWS infrastructure and all accesses to AWS. This log information is stored in AWS S3. So we know which user has made which change and the exact date.

When a CogniFit developer or administrator connects to a server through SSH, the date and user who accessed is logged. That allows us to know when and who is connected to each server.

## 3.4 Testing

At CogniFit we develop our code and our services in a pre-production environment. This pre-production environment has its own servers and databases, totally independent from the servers and databases of the production environment.

In the pre-production environment, we test the new code to validate that it is ready for deployment in the production environment. The pre-production environment architecture is similar to the production environment, with the same mapping of servers, load-balancers, and databases. We also try to have a similar number of users created in the preproduction environment as in the production environment. This way our pre-production environment tests are more reliable and can simulate the workload of the production environment.