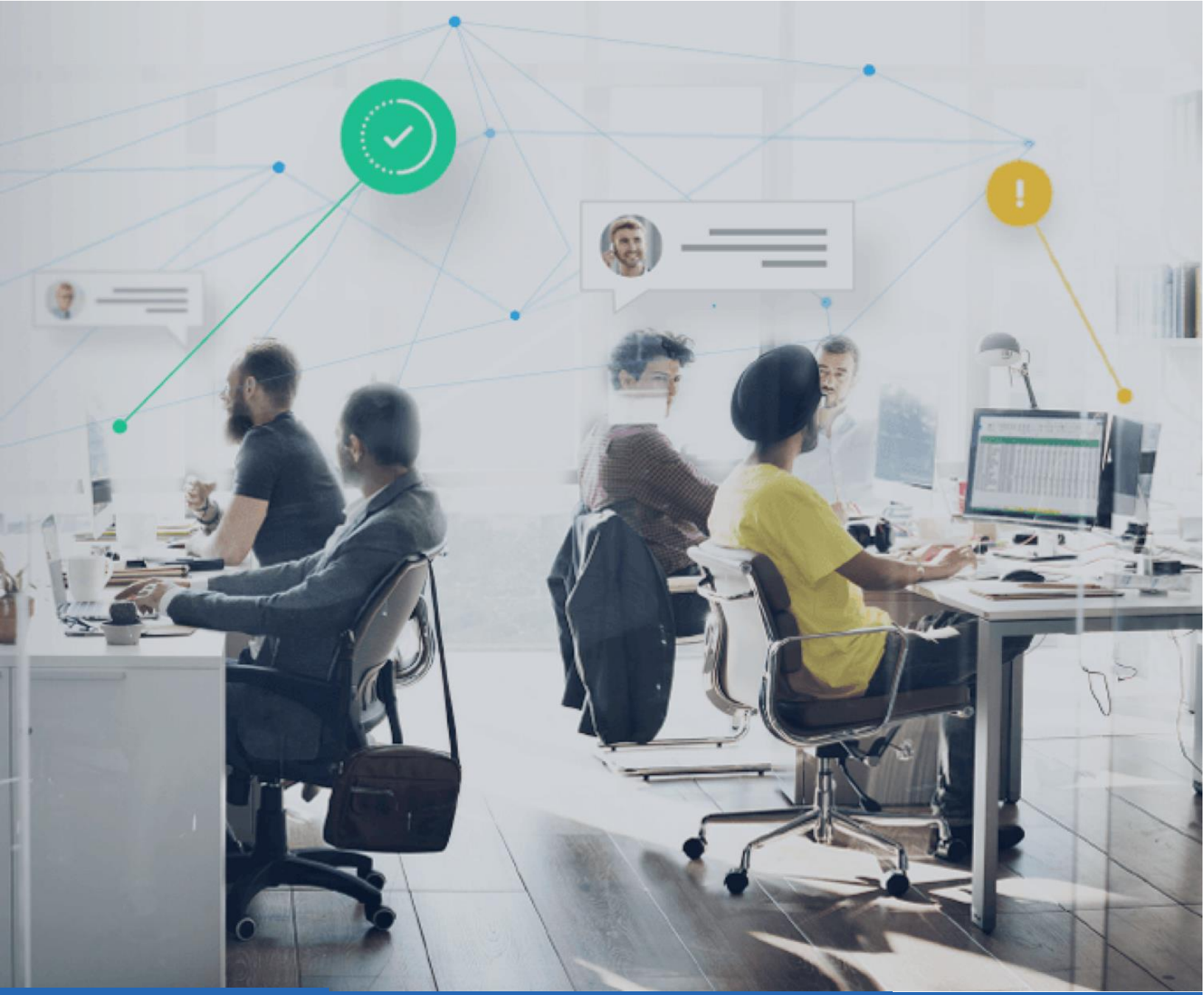

DataRails

Business Security Whitepaper

Contents

1. Under The Hood

The DataRails Application is hosted on Microsoft Azure platform, hence leveraging the extended security capabilities provided by the platform.

The DataRails applications includes several application servers (some for serving web-requests and some for batch processing) set behind a load balancer (Application Gateway). The data storage is stored in a polyglot architecture including an SQL DB (postgres-sql) for storing user and application information and a NoSQL (MongoDB) database that is used for cell-based excel data , logs and events storage. Both databases runs on dedicated VM's where the NoSQL DB is deployed using 3-servers replica set.

2. Application Security

The DataRails service can be utilized and accessed through a number of interfaces. Each has security settings and features that process and protect user data while ensuring ease of access.

a. Web. This interface can be accessed through any modern web browser. It allows admin users to upload, download, view, and share their files. The web interface also allows users to open existing local versions of files through their computer's default application.

b. Desktop. The DataRails sync application is a powerful client that identify changes in Excel spreadsheets and sync it to DataRails cloud DB.
The application runs under the local windows user credentials.

c. Share folder server. The DataRails sync service is a powerful service that identify all Excel related changes in selected folders within the organization shared folder.

d. Mobile. DataRails can be accessed via mobile web browsers, allowing users to access all their files on the go including the responsive dashboard.

e. Passwords and Identity Storage
   - Only the hashed signature of a user password is stored in the database, PBKDF2 together and SHA-256 algorithms are used to generate the hash.
   - Valid Email address are used as user names, these address are used for account activation and password reset
   -

f. Web Security
   - Application access can be done only via HTTPS protocol on port 443
   - Transport using TLS 1.2 with AES_128_GCM authentication mechanism and ECDHE_ECDSA as key exchange mechanism
   - SSL certificate is provided by Comodo RSA CA
   - Application uses CloudFlare for both Web Application Firewall and advanced DDOS protection.

- Cross Site Request Forgery (CSRF) mechanism is applied using HTTP headers , restricting API call to application only
- Host header validation is applied
- DB is accessed by application using ORM, therefore providing complete protection against SQL hijacking
- All API inputs are validated on application-entry and escaped on output to provide protection against XSS attacks.

Admin access is not exposed to internet, and allowed only using a VPN (restricted to DataRails R&D team only).

3. Data Recovery

In order to ensure maximum data recovery, all of DataRails data, including the end user uploaded files are being stored in Azure blobs Geo-redundant storage.
Each file is stored in 6 geo-separated locations (according to Azure GRS policy) and can be recovered anytime.
a snapshot backups of DataRails SQL databases is being created on a daily basis and stored on the GRS azure storage.
Our NoSQL database is using 3-nodes replica sets and can be recovered anytime using the uploaded documents and the SQL database
- SLA can be found here https://azure.microsoft.com/en-us/support/legal/sla/storage/v1_1/

4. Encryption

a. Data in transit

To protect data in transit between DataRails apps and our servers, DataRails uses Secure Sockets Layer (SSL)/Transport Layer Security (TLS) for data transfer, creating a secure tunnel protected by 128-bit or higher Advanced Encryption Standard (AES) encryption. File data in transit between a DataRails client (currently desktop, mobile, API, or web) and the hosted service is DataRails Business security always encrypted via SSL/TLS. For end points we control (desktop and mobile) and modern browsers, we use strong ciphers and support perfect forward secrecy and certificate pinning.

Note: DataRails uses TLS exclusively and has deprecated the use of SSLv3 due to known vulnerabilities. However, TLS is frequently referred to as “SSL/TLS,” so we use that designation here.

To prevent man-in-the-middle attacks, authentication of DataRails front-end servers is performed through public certificates held by the client. An encrypted connection is negotiated before the transfer of any files and ensures secure delivery of files to DataRails front-end servers.

b. Data at rest

All the spreadsheets data and other statistical information gathered from the DataRails client are stored within DataRails's database servers secured by a dedicated firewall. Every DataRails customer is assigned a unique user ID with access control mechanisms embedded in the application and in the database that prevents unauthorized access. Although data may be stored on shared database servers, the data is strictly protected and segregated in a way that ensures only authorized entities can have access to it.

- Documents are encrypted using AES-256 (server side encryption)
- Access to document are allowed only to the application (with dedicated application identity) and is done over HTTPS (encryption while in transit)
- Application file permission mechanism restricts access to file only to the document owner.

c. Key management

The DataRails key management infrastructure is designed with operational, technical, and procedural security controls with very limited direct access to keys. Encryption key generation, exchange, and storage is distributed for decentralized processing.

- File encryption keys. By design, DataRails uses Microsoft Azure blob-storage AES-128 encryption for file encryption. All keys are stored and managed by Microsoft.
- Internal SSH keys. Access to production server is restricted with unique SSH key pairs. Security policies and procedures require protection of SSH keys. An internal system manages the secure public key exchange process, and private keys are stored securely.

5. Data Restriction

Employee access to the DataRails production environment is maintained by a central directory and authenticated using a combination of strong passwords, and requires the use of VPN protected with two-factor authentication. Any special access is reviewed and vetted by the security team. Access to corporate and production networks is strictly limited based on defined policies. For example, production network access is SSH key-based and restricted to engineering teams requiring access as part of their duties. Firewall configuration is tightly controlled and limited to a small number of administrators.

6. Uptime and Business Continuity

DataRails's architecture was designed with the appropriate mindset to manage large scale traffic and CPS (connections per second) to provide high quality speed, availability, and service for our customers. The massive usage by users submitting and downloading files is being done against Microsoft Azure, which has a service commitment of a Monthly Uptime Percentage of at least 99.9%.

7. Vulnerability Management

DataRails's application and network are tested for security vulnerabilities by independent penetration scenarios regardless to the one that is being taken independently by Microsoft Azure who tests its services frequently.
(Penetration tests documents can be delivered upon request).
In addition, DataRails's security team makes sure to ensure that all aspects of security adhere to the highest standards.
On top of Microsoft Azure default security platform, DataRails has chosen CloudFlare as its Web Application Firewall and as a protection form of DDOS attacks.

8. Disaster Recovery and BCP DataRails

Our Disaster Recovery Plan (DRP) addresses both durability and availability disasters which resides on Microsoft Azure disaster recovery plans ensures that our customers experience almost no interruption of service in the event of a loss of data occurring at Azure data centers

9. Summary

DataRails strives to provide the highest security possible for its customers by utilizing top tier technology and products to provide the best user and customer experience possible. By emphasizing security and responsible planning for expected future growth with our partners and users, DataRails has built the building blocks to provide the best step-by-step guidance tool on the internet to help millions of people perform complicated tasks with ease around the world.

The following configuration is suitable for an installation supporting up to **100 users**, each uploading approximately **50MB** of excel data (total file size) per day and include in total **6** servers.

Technical Requirements

The DataRails backend solution (including the web servers, application servers and databases) is deployed using Docker containers, therefore all that is required is to allocate servers with Docker Engine installed (any underlying OS will work) according to the following specification:

SMTP Mail server access

An SMTP mail server user & password are required in order to enable automatic mails to be sent from DataRails system.

Storage Requirements

A network storage space should be allocated for storing the original uploaded files, 500 GB of storage allocation is recommended.

Network Requirements

The web server should be assigned with an IP address accessible from all client computers (including access from home via VPN etc) on port 443 (HTTPS).
It's recommended to provide also a DNS name for the server IP.

Remote Access
As part of 3rd tier support, all Docker containers should be accessible to DataRails 3rd level support team via Docker-machine (either directly or through a VPN)

Network access should be granted for the server to communicate with the databases instances (in case the customer choose to use it's own instances)

Security

An X.509 certificate should be provided by the customer, and be installed on the web servers.
The certificate should be configured with either the IP address/FQDN of the server.