Disaster Recovery

With this document, we present an overview of how FeedbackFruits aims to restore services when disaster strikes. The Disaster Recovery Plan shows how our Software-as-a-Service (SaaS) product is structured such that rebuilding (parts of our) architecture can happen quickly, and which procedures we follow in assessing the impact of a disaster and communicating clearly with involved stakeholders.

Terminology

Within SaaS disaster recovery scenarios, **Recovery Time Objective (RTO)** and **Recovery Point Objective (RPO)** are the critical metrics. The RTO communicates the expected time needed to restore access to data, with the RPO indicating how much data is at risk of being lost.

It is also important to make a distinction between efforts for **Disaster Recovery** and **High Availability.** In scenarios where Disaster Recovery is necessary, downtime is implied: services have become unavailable and need to be restored, taking hours/days. The concept of High Availability is about building in resilience through redundancy, allowing for (automatic) failover of components upon failure or disaster striking.

For most failure scenarios, the architecture of our product is such that the RPO and RTO is extremely low: allowing for failing over to other parts of our infrastructure, due to its High Availability setup. This disaster recovery plan therefore focuses on the loss of data centers or our architecture as a whole, not on the failure of individual components, which recover through redundancy/(automatic) failover.

Understanding the SaaS & IaaS context

FeedbackFruits delivers its product as Software as a Service: the tool suite is hosted in the cloud by FeedbackFruits and delivered over the internet. This delivery model means that our customers don't have to host FeedbackFruits themselves, bypassing the need to maintain server hardware and software, apply fixes/patches or do other maintenance.

As a company, our goal is to be focussed on our application and work with specialists for maintaining the infrastructure to host our application on. FeedbackFruits works with Microsoft Azure, which provides us with Platforms as a Service: a cloud platform that has built in redundancy and High Availability, spread

across multiple regions, availability zones and data centers, making its offering resilient and allowing for quick recovery in the event of unforeseen disasters.

Our disaster recovery plans focus on the availability of our software and the associated data and incorporate the practices of our infrastructure provider. Best practices dictate that our customers develop and maintain their own (holistic) disaster recovery plans and procedures.

Types of disaster

We define a disaster as any disruptive event that has (potentially long-term) adverse effects on FeedbackFruits services, requiring immediate addressing at high priority by the company.

These events can be intentional or unintentional and originate from natural circumstances (for example flooding, earthquakes, fires), supply disruptions (for example power outages, fiber cuts) or be human-made/political (for example terrorism, hacks).

Procedure

FeedbackFruits uses external monitoring tools to assess the responsiveness of our services in all regions, with checks ranging from monitoring individual low-level components of our infrastructure to checks monitoring the core functionality of our product.

When disruptions are detected, these tools send real-time alerts to our personnel, which investigate the nature of the alert and potential scope of a disruption. Most alerts are informative of nature and resolve themselves through redundancy/failover.

When a potential disaster is detected, our CEO, COO & CTO get notified and work together on determining next actions, assembling a Disaster Recovery Team and preparing to follow the recovery procedures as specified in this document.

Communication

Relevant operational staff is being alerted/pages automatically upon detection of a potential disruption of disaster. Key personnel have multiple independent ways of establishing communication, mitigating the risk of one of our internal communication tools being unavailable.

When a disaster is declared, we immediately notify impacted customers and business partners, specifying the effects on our service, potential impact to data and expected update frequency.

We use our status portal at status.feedbackfruits.com to provide continuous updates and provide indications of expected recovery time.

Upon completion of recovery and services being available again, we notify relevant parties with steps taken to recovery and expected follow ups. In the case of partial recovery, we send out frequent rolling updates with estimations of full recovery time.

Resilience, Backup & Recovery

FeedbackFruits services are delivered through a multi-tier cloud-based hosting architecture. We've kept our infrastructure as simple as possible, store our configuration as code to be able to quickly rebuild if necessary, and work with different layers. We monitor each level of the infrastructure for failure detection and have automated failovers implemented.

In each region, through Microsoft Azure, we work with a cluster (Kubernetes based), at which traffic arrives first at our **Load Balancers**. These are redundant in an active-active configuration.

Traffic moves to one of our **App Servers**, where depending on the type the request gets routed to the relevant App Server. We have different App Servers for synchronous core application requests, long-running socket connections, asynchronous background jobs, authentication/authorization requests and media handling. Each scale independently based on demand throughout the day, with a minimum number of servers always being available for redundancy and handling sudden peaks.

While App Servers handle the incoming requests, they don't store persistent data: most of that is handled in our **Databases**. These, for example, contain learning activity, student progress and enrollments and are separated per region. We work with managed Azure infrastructure for our primary databases, which are set up for high availability and allow automatic failover within a region (for example, when hardware issues occur). Backups are made continuously, are stored encrypted, geographically redundant (different availability zones within the region) and allow for point-in-time recovery.

For **Content Storage**, like text documents, PDFs and videos, we work with Azure Blob Storage, which has high availability guarantees.

Our Azure infrastructure is set up to use multiple regions and availability zones, which helps with resiliency in many traditionally disruptive failure scenarios, including natural disasters or system failures. For location-dependent privacy and compliance with data sovereignty requirements, such as the EU Data Privacy Directive, data is not replicated between regions. However, in the unlikely event of a disaster that affects a customer's entire region, all services and data can be relocated to numerous active regions.

We keep our infrastructure 'documented as code', allowing for quick recovery from a wide range of scenarios, ranging from complete loss of a hosting region to a failing deployment of a specific application.

Examples of tested disaster scenarios

We continuously test our own infrastructure as part of our development process, by automatically setting up isolated environments from our standard configuration. Next to that, we run simulations of failure scenarios in our staging environment, which closely mimics our various production environments/regions.

Below, we've outlined various example disaster scenarios with their recovery strategies and expected effect on our customers. Scenario forecasting is never exhaustive, projected likelihood is not an expected/guaranteed occurrence.

Complete loss of database: affecting all customers in hosting region
In the unlikely case (projected once every 5+ years) of our Azure high availability database setup completely failing within a region, we restore the latest available backup to a new database instance. The expected RPO is under 1 hour of data, the expected RTO is under 6 hours. While we do simulate this scenario, we have never had to execute on this scenario in production.

Complete loss of hosting region cluster: affecting all customers in hosting region
In the extremely unlikely case (projected once every 25+ years) of a full loss of our hosting cluster in a region, we deploy our architecture to a different region with our cloud vendor, based on stored configuration. The expected RPO is under 6 hours of data, the expected RTO is under 12 hours. While we do simulate this scenario, we have never had to execute on this scenario in production.