A business continuity plan (BCP) helps to ensure that critical business operations continue running after a major disruption. Based on identified threats such as database crash, network failure, DDoS attacks, untested code, unauthorized access or insider threats. I investigate the BCP components and process steps aligned with ISO/IEC 27031.

* Risk Assessment- I identified threats and documented mitigation strategies for potential pitfalls.
* Roles and Responsibilities – assign people to handle key tasks (e.g., Martin backend, Julia fallback UI).
* Understanding requirements, stakeholders and dependencies – LLM, MySQL, Flask API.
* Business Impact Analysis – determined critical LLM feature and MySQL crash.

Disaster recovery planning (DRP) is the process of identifying and restoring system after system interruption.

* Backup and recovery – automate backups, use mysqldump daily and test restore.
* Disaster Recovery Sites – backup server on different cloud.
* Disaster Recovery Test – MySQL database crash.

Scenario: Crash the MySQL database to validate the recovery process.

Expected RTO: < 30 mins

1. Verify MySQL crash detection.

2. Ensure Flask, react.js handles error.

3. Test data backup restore process.

4. Measure RTO (Recovery Time Objective) and RPO (Recovery Point Objective).

To measure Recovery Time Objective (RTO) and Recovery Point Objective (RPO), we should enhance incident logs with timestamps marking the failure and recovery points. We utilized browser developer tools to observe when routes became responsive. We could further support this by logging request sizes, database activity, and leveraging Flask middleware for detailed request tracking.

We used browser developer tools to monitor API endpoint responses during failure and recovery scenarios. By observing the status codes and timing of requests such as /api/data, we established a clear timeline of service availability. (Kloda, 2025) This enabled manual tracking of the Recovery Time Objective (RTO). Next time I would define the Maximum Tolerable Downtime (MTD) using datetime for example, 60 minutes for the MySQL.

By combining these two practices with detailed measurements and testing strategies, our project can recover. We could ensure continuous improvement and structured planning by using ISO/ IEC 27031.

# References

Kloda (2025). [online] github. Available at: https://github.com/Jkloda/LOJuliaKloda/blob/main/Assets/DevTools.png [Accessed 14 May 2025].