GUIDE
Backup and Restore Using Duplicati CLI

v1.0.1

Author:

Eldon Gabriel

March 17, 2026

TABLE OF CONTENTS

Disclaimer: This report documents my personal work completing an MCSI lab exercise. It reflects my independent understanding and configuration of Duplicati backup and restore operations using the Duplicati command-line interface in a controlled environment. No MCSI video content, lab materials, or proprietary instructions have been shared or distributed. All information presented follows MCSI's disclosure and academic integrity policies.

REVISION HISTORY

Version	Date	Author	Description of Changes
v1.0.0	02/04/2026	Eldon G.	Initial draft.
v1.0.1	03/17/2026	Eldon G.	Finalized formatting for portfolio publication

1.0 DUPLICATI BACKUP AND RESTORE WORKFLOW

1.1 Project Description

This project demonstrates how to use **Duplicati**, a free backup tool, to safely back up and restore files using the command line. The goal was to learn how to make backups, check data, simulate file loss, and restore them in a safe environment. The process focuses on using the command line, troubleshooting problems, and ensuring that data can be reliably recovered.

1.2 Technical Tasks

1.2.1 Prepare Backup Source Data

- A local folder containing generic test files was created to serve as a backup source.
- The files were ensured to be safe and did not include any sensitive or proprietary information.

1.2.2 Install and Configure Duplicati

- **Duplicati 2** was installed on a Windows system.
- The availability of the (CLInd-line interface CLI (; Duplicati.CommandLine.exe) was verified, and backup destination folders were created.

1.2.3 Execute Backup Using Duplicati CLI

- The backup command in the **Duplicati CLI** was used to back the source folder to the designated backup destination.
- The *--no-encryption* option was applied for a novice-level backup demonstration.
- The command output confirmed that the remote backup files were successfully created.

1.2.4 Verify Backup Integrity

- The find command was used to confirm that the backup had been recorded.
- A test command was executed to verify the integrity of the backup files.
- The logs confirmed that the operation was successful, with no errors reported.

1.2.5 Simulate Data Loss

- The source files were deleted to simulate accidental losses.
- Safe test data were maintained without revealing sensitive exercise content.

1.2.6 Restore Backup Using Duplicati CLI

- The restore command is used to recover the deleted files to a separate restore directory.
- The restored files maintained the folder structure and content integrity.

1.2.7 Confirm Restored Data

- The restore directory was navigated to confirm that all expected files were recovered successfully.
- The workflow was validated, and repeatable and reliable restorations were confirmed.

1.3 Next Steps

- Practice using additional **Duplicati CLI** options, such as filters, retention policies, and scheduling automated backups.
- Explore encrypted backups once foundational workflows have been mastered.
- Document troubleshooting scenarios for common command-line backup and restore errors

1.4 Optional Deep Dive

- Understanding **Duplicati backup metadata**, including how incremental changes and versions are tracked.
- Using CLI logs to troubleshoot missing or corrupted backup files

1.5 Additional or Supporting Work

- The backup and restore commands were tested in temporary directories to ensure repeatable results.
- PowerShell commands were used to automate folder creation and cleanup for testing purposes.

2.0 CONCLUSION

2.1 Key Takeaways

- Demonstrated practical command-line interface-based backup and restored skills using **Duplicati**.
- Gained experience in backup, verification, deletion, and restoration in controlled environments.
- Developed troubleshooting awareness for common issues, including metadata inconsistencies, encryption, and file path errors.
- A repeatable workflow was established to ensure data integrity and recoverability.

2.2 Security Implications and Recommendations

- **Risks:** Accidental file deletion, corrupted backups, backup metadata conflicts, and misconfigured encryption.
- **Remediation:** Verify backup integrity, use isolated restore locations, and maintain clean backup folders.
- **Technical Recommendations:** Use command-line options such as --restore-path and --no-encryption carefully and maintain separate backup directories for each workflow run.
- **Procedural Recommendations:** Document backup procedures and workflows, and regularly check backup destinations for consistency.
- **Compliance Mapping:** Following a backup and restoring workflow ensures secure data management and operational readiness.
