

# Module 5: Restoring Data from Backups

## 5.1 Restoration Scenarios

### Understanding Restoration Needs

Restoration is critical in scenarios such as:

- 1. Website Recovery:**
  - After accidental deletion or malware infection.
  - Restoring site files and configurations.
- 2. Database Recovery:**
  - When a database is corrupted, accidentally deleted, or rolled back to a previous state.
- 3. Email Recovery:**
  - Recovering lost email accounts or individual emails.
- 4. Full Server Restoration:**
  - In cases of catastrophic server failure or when migrating to a new server.

### Steps for Different Restoration Scenarios

- 1. Website Restoration:**
  - Locate the backup file (tar.gz, zip, etc.).
  - Extract files to the web directory (`/var/www/html` or equivalent).
  - Verify file permissions and ownership.
- 2. Database Restoration:**
  - Identify the appropriate backup file (e.g., `mysql_backup.sql`).
  - Use database-specific tools to restore:
    - MySQL: `mysql -u user -p database_name < backup.sql`
    - PostgreSQL: `psql -U user -d database_name -f backup.sql`
- 3. Email Restoration:**

- Restore email directories to the server's email storage location.
- For example, for cPanel this is **/home/username/mail** directory.

#### 4. **Server-Wide Restoration:**

- Boot the server into recovery mode.
  - Restore files and databases from backups.
  - Reconfigure critical services like web servers, databases, and DNS.
  - See guide below on how to get database backups in rescue or recovery mode
  - <https://truehost.com/support/knowledge-base/how-to-activate-a-nd-use-rescue-mode-on-a-vps/>
  -
- 

## 5.2 Tools and Procedures

### Restoring via cPanel

#### 1. **File Restoration:**

- Access **File Manager** in cPanel.
- Upload and extract backup files to the public\_html directory.
- Set correct file permissions via cPanel's interface or command line.

#### 2. **Database Restoration:**

- Go to **phpMyAdmin** in cPanel.
- Use the **Import** option to upload **.sql** files.
- Ensure the database user has appropriate privileges.

#### 3. **Email Restoration:**

- Upload the mail directory archive to the file manager in home/cpusername/mail directory
- Extract your files
- Verify mailbox integrity after restoration.

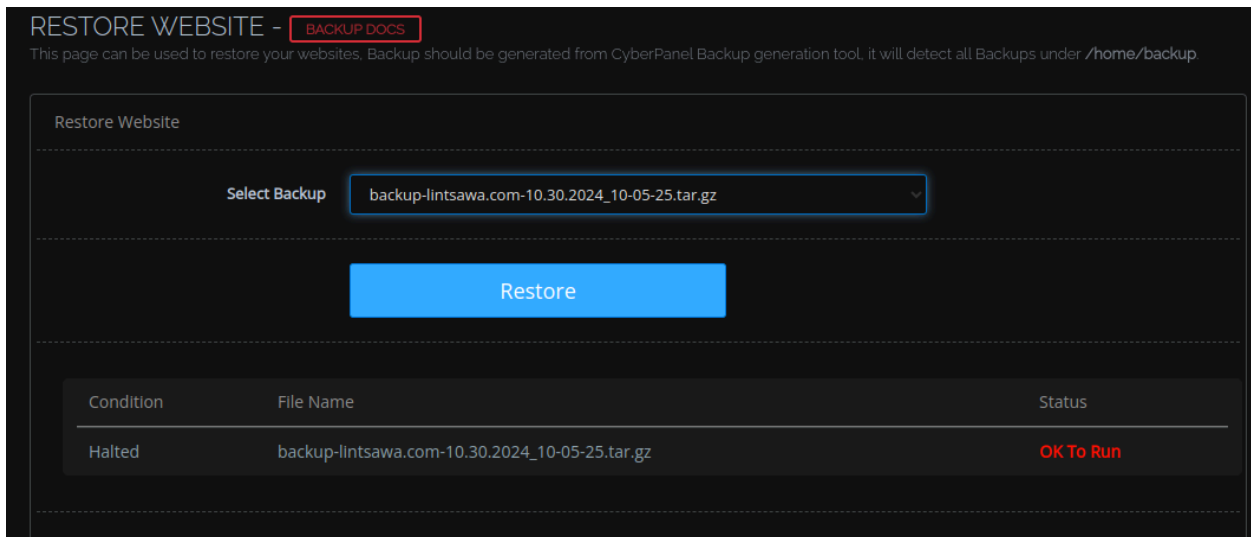
#### 4. **Restore Script**

- Use cPanel restore script via SSH to restore full cPanel Accounts.

## Restoring via CyberPanel

### 1. File Restoration:

- Navigate to the CyberPanel **Backup** module.
- Click on Restore Button
- Select the backup file and restore website files.



### 2. Database Restoration:

- Access the **Databases** section in CyberPanel.
- Import the backup **.sql** file using the CyberPanel Phpmyadmin or SSH.

### 3. Email Restoration:

- Restore using the Step 1 above or use IMAPSYNC if coming from a different control panel or platform.

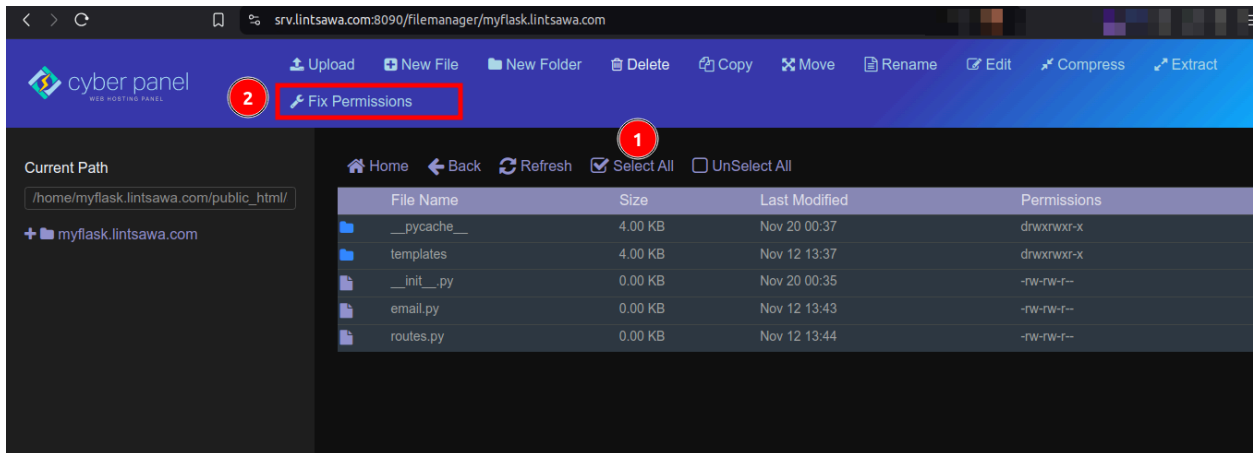
## Restoring via FTP

### 1. Connect to the Server:

- Use an FTP client like FileZilla to upload backup files.
- Place files in the appropriate directories (**//public\_html**).

### 2. Verify File Ownership:

- After uploading, ensure the correct ownership - Click on Fix Permissions on cyberpanel Filemanager).



## Command-Line Tools for Database Restoration

### 1. MySQL:

- Restore databases directly:

```
mysql -u root -p database_name < backup.sql
```

- For compressed backups:

```
gunzip < backup.sql.gz | mysql -u root -p database_name
```

### 2. PostgreSQL:

- Restore using **psql**:

```
psql -U user -d database_name -f backup.sql
```

- For compressed backups:

```
gunzip < backup.sql.gz | psql -U user -d database_name
```

---

## 5.3 Common Restoration Challenges and Solutions

### Challenge 1: Corrupted Backup Files

- **Problem:** Backup files are incomplete or corrupted during storage or transfer.
- **Solutions:**

Use checksum tools like **md5sum** to verify file integrity:

```
md5sum backup.tar.gz
```

- Attempt partial recovery with **tar or zip:**

```
tar -xvf backup.tar.gz --ignore-zeros
```

Restore data from secondary or incremental backups.

### Challenge 2: Restoring to a Different Server or Environment

- **Problem:** The new server has different configurations or software versions.
- **Solutions:**
  1. Ensure software compatibility:
    - Install the same versions of PHP, MySQL/PostgreSQL, and web server.
    - Migrate necessary extensions and dependencies.
  2. Adjust file paths and permissions:
    - Update configurations to match the new environment.
- Use tools like **sed** to modify file paths in bulk:

```
sed -i 's/old_path/new_path/g' configuration_file
```

3. Test restored services in a staging environment before going live.

### Challenge 3: Missing Dependencies or Services

- **Problem:** Required dependencies or services are missing after restoration.

- **Solutions:**
  1. Use package managers like **apt, yum, or dnf** to install missing software.
- Refer to logs for missing services:

**tail -f /var/log/syslog**

2.

- Restart services to reload configurations:

**systemctl restart service\_name**

3.

---

### **Tips for Successful Restoration**

1. **Always Test Backups:**
  - Regularly test restoring backups in a staging environment.
2. **Document Restoration Procedures:**
  - Maintain clear documentation to reduce downtime during recovery.
3. **Enable Incremental Backups:**
  - Use incremental backups to reduce storage usage and speed up recovery.
-