Chapter 15: Storing files with S3

When you store files in Amazon S3, there are options for chunking, compression, and encryption, but each process may or may not be applied automatically based on your configuration and requirements. Here's how each can work:

1. Chunking (Multipart Upload)

- For large files (100 MB or larger), S3 supports **multipart upload** to optimize the upload process and improve resiliency. This means the file is split into smaller parts (chunks), each part is uploaded independently, and once all parts are uploaded, S3 assembles them into a complete file.
- The client typically handles this chunking; AWS SDKs often provide built-in support for multipart uploads, where the file is divided into parts, uploaded in parallel, and then reassembled.
- This is especially useful for managing upload failures since individual chunks can be retried if they fail, rather than re-uploading the entire file.

2. Compression

- S3 doesn't automatically compress files on upload, so any compression has to be performed by the client before the upload.
- You can compress the files (e.g., using gzip or zip) locally and then upload the compressed version to S3.
- For retrieval, the client would need to decompress the file after downloading it. If you're working with large datasets (e.g., logs or backups), pre-compression before uploading can save storage space and reduce data transfer costs.

3. Encryption

- S3 supports several **encryption options** for data at rest and in transit, and encryption can be automatically handled by S3:
    - **Server-Side Encryption (SSE)**:

        - **SSE-S3**: S3 manages the encryption keys and encrypts each object using AES-256 encryption. This is automatic if enabled.
        - **SSE-KMS**: Uses AWS Key Management Service (KMS) for encryption, allowing you to control encryption keys and apply more detailed access control policies.
        - **SSE-C**: Allows you to supply your own encryption keys for S3 to use in the encryption process.
    - **Client-Side Encryption**: The client encrypts data before uploading it to S3, ensuring that it remains encrypted end-to-end. The client must manage the encryption keys in this case.
- Encryption is applied to the full object (file) and isn't typically used at the block or chunk level unless client-side encryption divides the data into chunks and encrypts each independently.

In summary:

- **Chunking** is usually controlled by the client application using multipart upload.
- **Compression** must be handled by the client before uploading the file to S3.
- **Encryption** can be managed by S3 on your behalf, with several options to choose from depending on your security requirements.