Common Bucket Operations Flow

Participants: Application | Bucket | S3 Service | Object Storage (MinIO/S3)

Initialization
- Application → Bucket: init(auto_connect=True, debug=False)
- Bucket: _auto_connect()
- Bucket: connect()
- Bucket → S3 Service: boto3.resource("s3", credentials)
- S3 Service → Object Storage: Establish connection
- Object Storage → S3 Service: Connection OK
- S3 Service → Bucket: resource object
- Bucket → S3 Service: list_buckets() [health check]
- S3 Service → Object Storage: List buckets
- Object Storage → S3 Service: Bucket list
- S3 Service → Bucket: Success
- Bucket → Application: Bucket instance (connected)

Create Bucket
- Application → Bucket: create_bucket("my-bucket")
- Bucket: _ensure_connected()
- Bucket → S3 Service: create_bucket(Bucket="my-bucket")
- S3 Service → Object Storage: Create bucket
- Object Storage → S3 Service: Bucket created
- S3 Service → Bucket: Success
- Bucket → Application: success code (0)

Upload File
- Application → Bucket: upload_file("my-bucket", "/tmp/file.txt", "data/file.txt")
- Bucket: _ensure_connected()
- Bucket → S3 Service: Bucket("my-bucket")
- S3 Service → Bucket: S3BucketLike
- Bucket → S3 Service: upload_file("/tmp/file.txt", "data/file.txt")
- S3 Service → Object Storage: PUT object: data/file.txt
- Object Storage → S3 Service: Object stored
- S3 Service → Bucket: Success
- Bucket → Application: success code (0)

List Files
- Application → Bucket: get_bucket_files("my-bucket")
- Bucket: _ensure_connected()
- Bucket → S3 Service: Bucket("my-bucket")
- S3 Service → Bucket: S3BucketLike
- Bucket → S3 Service: objects.all()
- S3 Service → Object Storage: List objects
- Object Storage → S3 Service: Object list
- S3 Service → Bucket: Iterator[S3ObjectLike]
- Bucket: [obj.key for obj in objects]
- Bucket → Application: ["data/file.txt", "data/file2.txt"]

Get File Metadata
- Application → Bucket: get_bucket_file("my-bucket", "data/file.txt")
- Bucket: _ensure_connected()
- Bucket → S3 Service: Bucket("my-bucket")
- S3 Service → Bucket: S3BucketLike
- Bucket → S3 Service: Object("data/file.txt")
- S3 Service → Object Storage: HEAD object: data/file.txt
- Object Storage → S3 Service: Object metadata
- S3 Service → Bucket: S3ObjectLike
- Bucket: Extract key and content_length
- Bucket → Application: {"file_path": "data/file.txt", "file_size": 1024}

Download File
- Application → Bucket: download_file("my-bucket", "data/file.txt", "/tmp/downloaded.txt")
- Bucket: _ensure_connected()
- Bucket → S3 Service: Bucket("my-bucket")
- S3 Service → Bucket: S3BucketLike
- Bucket → S3 Service: download_file("data/file.txt", "/tmp/downloaded.txt")
- S3 Service → Object Storage: GET object: data/file.txt
- Object Storage → S3 Service: Object data stream
- S3 Service: Write to /tmp/downloaded.txt
- S3 Service → Bucket: Success
- Bucket → Application: success code (0)

Delete File
- Application → Bucket: delete_file("my-bucket", "data/file.txt")
- Bucket: _ensure_connected()
- Bucket → S3 Service: Bucket("my-bucket")
- S3 Service → Bucket: S3BucketLike
- Bucket → S3 Service: Object("data/file.txt").delete()
- S3 Service → Object Storage: DELETE object: data/file.txt
- Object Storage → S3 Service: Object deleted
- S3 Service → Bucket: Success
- Bucket → Application: success code (0)

List Buckets
- Application → Bucket: get_bucket_names()
- Bucket: _ensure_connected()
- Bucket → S3 Service: buckets.all()
- S3 Service → Object Storage: List buckets
- Object Storage → S3 Service: Bucket list
- S3 Service → Bucket: Iterator[BucketLike]
- Bucket: [bucket.name for bucket in buckets]
- Bucket → Application: ["my-bucket", "another-bucket"]

Check Connection
- Application → Bucket: is_connected()
- Bucket → S3 Service: list_buckets() [health check]
- S3 Service → Object Storage: List buckets
- Object Storage → S3 Service: Bucket list
- S3 Service → Bucket: Success
- Bucket → Application: True

Cleanup
- Application → Bucket: disconnect()
- Bucket: connection = None
- Bucket → Application: success code (0)
- Application → Bucket: __del__() [garbage collection]
- Bucket: disconnect()
- Bucket: connection = None

Notes:
- All operations automatically ensure connection is established via _ensure_connected() before proceeding
- Return codes: 0 for success, 84 for error (configurable in constructor)
- List operations return List[str] on success or error code on failure
- boto3 handles:
  - Connection pooling
  - Automatic retries
  - Multipart uploads (large files)
  - Streaming downloads
  - Request signing