

```

public Status process() throws EventDeliveryException {
    Status status = Status.BACKOFF;
    Transaction tx = null;
    final ByteArrayOutputStream data
        = new ByteArrayOutputStream(bufferSize);
    try {
        tx = getChannel().getTransaction();
        tx.begin();
        int i = 0;
        for (; i < batchSize; i++) {
            Event e = getChannel().take();
            if (e == null) {
                break;
            }
            byte[] body = e.getBody();
            data.write(
                ByteBuffer.allocate(Integer.SIZE / 8).putInt(body.length).array());
            data.write(body);
        }
        if (i != 0) {
            connection.putObject(bucket,
                objPrefix + suffix.incrementAndGet(),
                new ByteArrayInputStream(data.toByteArray()),
                new ObjectMetadata());
            status = Status.READY;
        }
        tx.commit();
    } catch (Exception e) {
        if (tx != null) {
            tx.rollback();
        }
        throw new EventDeliveryException("Error while processing " +
            "data", e);
    } finally {
        if (tx != null) {
            tx.close();
        }
    }
    return status;
}

```

```

@Override
public void configure(Context context) {
    awsAccessKeyId = context.getString("awsAccessKeyId");
    Preconditions.checkArgument(!Strings.isNullOrEmpty(awsAccessKeyId),

```