Redis Connection Fix - Upstash TCP Endpoint for BullMQ

Date: November 15, 2025
Issue: ParserError: Protocol error, got 'H' as reply type byte
Status: ✅ FIXED

🔴 Problem Summary

The application was crashing with the error:

```
ParserError: Protocol error, got 'H' as reply type byte
```

Root Cause

The code was attempting to use Upstash's **REST API endpoint** (`UPSTASH_REDIS_REST_URL`) as a **TCP Redis connection**.

- **REST API endpoint**: HTTP-based (e.g., `https://your-redis.upstash.io`)
- **TCP endpoint**: Redis protocol (e.g., `rediss://default:password@your-redis.upstash.io:6380`)

BullMQ (our queue system) requires a **TCP Redis connection** via ioredis, not REST API. When ioredis tried to connect to the HTTP endpoint, it received HTML response bytes (starting with 'H'), causing the protocol error.

✅ Solution Implemented

Changes Made

1. **Updated `lib/queue/queue.ts`** :
   - Changed to use `UPSTASH_REDIS_URL` (TCP endpoint) instead of parsing REST URL
   - Added IORedis import and proper connection configuration
   - Implemented graceful degradation with clear error messages
   - Added helpful warnings when REST credentials are present but TCP endpoint is missing

2. **Updated `.env.example`** :
   - Added `UPSTASH_REDIS_URL` with clear documentation
   - Explained the difference between TCP and REST endpoints
   - Marked TCP endpoint as REQUIRED for BullMQ

3. **Updated `SERVICES_SETUP.md`** :
   - Added comprehensive instructions for getting TCP endpoint from Upstash Console
   - Clarified that BullMQ requires TCP, not REST
   - Updated quick start guides with correct environment variables

4. **Installed** `@upstash/redis`:
   - Available for future REST API features (direct cache operations)
   - Currently, BullMQ uses ioredis with TCP endpoint

🚀 How to Fix Your Environment

Step 1: Get Upstash TCP Endpoint

1. Go to Upstash Console (https://console.upstash.com/)
2. Select your Redis database
3. Click **"Redis Connect"** tab
4. Select **"ioredis"** option
5. Copy the connection string (format: `rediss://default:<password>@<host>:<port>`)

Step 2: Update Your `.env` File

Add the TCP endpoint to your `.env` file:

```
# ✅ REQUIRED FOR BULLMQ: TCP Endpoint
UPSTASH_REDIS_URL="rediss://default:YOUR_PASSWORD@your-redis.upstash.io:6380"

# Optional: REST API (kept for future features)
UPSTASH_REDIS_REST_URL="https://your-redis-instance.upstash.io"
UPSTASH_REDIS_REST_TOKEN="your-upstash-rest-token"
```

Step 3: Restart Your Application

```
npm run dev
```

You should see:

```
✅ Connected to Upstash Redis via TCP endpoint for BullMQ
```

🔍 What Changed Under the Hood

Before (Broken)

```
// ❌ INCORRECT: Trying to parse REST URL as TCP endpoint
const url = new URL(upstashUrl); // upstashUrl = "https://..."
const config: ConnectionOptions = {
  host: url.hostname,
  port: url.port ? parseInt(url.port) : 443,
  password: upstashToken,
  tls: url.protocol === 'https:' ? {} : undefined,
};
```

Problem: This treats HTTPS REST API URL as a TCP connection, causing protocol errors.

After (Fixed)

```
// ✅ CORRECT: Using TCP endpoint directly
const upstashUrl = process.env.UPSTASH_REDIS_URL; // "rediss://..."
const connection = new IORedis(upstashUrl, {
  maxRetriesPerRequest: null, // Required for BullMQ
  enableReadyCheck: false,
  family: 0,
});
```

Solution: Uses proper TCP endpoint with Redis protocol (rediss://).

🎯 Understanding Upstash Endpoints

Upstash provides **TWO types of endpoints** for the same database:

Type	Protocol	Format	Use Case
TCP	Redis Protocol	`rediss://default:pwd@host:port`	BullMQ, ioredis, traditional Redis clients
REST	HTTP/HTTPS	`https://host` + Token	Direct HTTP calls, serverless edge functions

When to Use Each

- **TCP Endpoint** (`UPSTASH_REDIS_URL`):
- ✅ BullMQ queue system
- ✅ Traditional Redis clients (ioredis, node-redis)
- ✅ Long-running processes
- ✅ Background workers
- **REST Endpoint** (`UPSTASH_REDIS_REST_URL/TOKEN`):
- ✅ Edge functions (Vercel, Cloudflare Workers)
- ✅ Direct HTTP API calls
- ✅ Environments without TCP support
- ❌ NOT compatible with BullMQ

🔧 Graceful Degradation

The queue system now handles missing configuration gracefully:

Scenario 1: TCP Endpoint Configured ✅

```
UPSTASH_REDIS_URL="rediss://..."
```

Result: BullMQ queue system fully functional, background jobs processed asynchronously

Scenario 2: Only REST Credentials ⚠️

```
UPSTASH_REDIS_REST_URL="https://..."
UPSTASH_REDIS_REST_TOKEN="token..."
```

Result:
- Queue system DISABLED with helpful warning
- Application continues to run (no crash)
- Jobs execute immediately (synchronous)
- Clear instructions provided in logs

Scenario 3: No Redis Configuration ⚠️

```
# No Redis variables set
```

Result:
- Queue system DISABLED
- Application continues to run (no crash)
- Jobs execute immediately (synchronous)
- Warning logged on startup

📊 Impact on Application

With Queue System (Redis TCP Configured)

- ✅ Background job processing
- ✅ Email/SMS queued and sent asynchronously
- ✅ Retry mechanism for failed operations
- ✅ Rate limiting
- ✅ Better performance (non-blocking)

Without Queue System (Graceful Degradation)

- ⚠️ Jobs execute immediately (synchronous)
- ⚠️ No retry mechanism
- ⚠️ Slower response times
- ✅ Application still functional
- ✅ No crashes

🧪 Testing the Fix

Test 1: Verify Connection

```
# Start the application
npm run dev

# Look for this log message:
# ✅ Connected to Upstash Redis via TCP endpoint for BullMQ
```

Test 2: Send Test Email

```
import { emailService } from '@/lib/email/emailService';

await emailService.send({
  to: 'test@example.com',
  subject: 'Test Email',
  html: '<p>This is a test</p>',
});
```

Test 3: Check Queue Status

```
# In your application logs, you should see:
# [Queue] Adding job to queue: email
# [Queue] Job completed: email
```

🚨 Troubleshooting

Error: “ECONNREFUSED” or “Connection timeout”

Cause: Invalid TCP endpoint or network issues
Solution:
1. Verify TCP endpoint format: `rediss://default:password@host:port`
2. Check Upstash Console for correct endpoint
3. Ensure no firewall blocking outbound connections

Error: “Authentication failed”

Cause: Incorrect password in TCP URL
Solution: Copy the complete TCP URL from Upstash Console (includes correct password)

Warning: “Queue system is DISABLED”

Cause: Missing `UPSTASH_REDIS_URL`
Solution: Add TCP endpoint to `.env` file (see Step 1 above)

📚 Additional Resources

- Upstash Redis Documentation (https://docs.upstash.com/redis)
- BullMQ Documentation (https://docs.bullmq.io/)

- IORedis Documentation (https://github.com/redis/ioredis)
- Upstash Console (https://console.upstash.com/)

🎉 Summary

Before: Application crashed with protocol error when trying to use REST API endpoint for BullMQ

After:
- ✅ Uses proper TCP endpoint for BullMQ
- ✅ Graceful degradation if Redis not configured
- ✅ Clear documentation and error messages
- ✅ No more crashes due to Redis misconfiguration

Action Required: Update your `.env` file with `UPSTASH_REDIS_URL` (TCP endpoint)

Last Updated: November 15, 2025
Fixed By: DeepAgent AI Assistant