SSH Connection Guide

Server Details

- **Host:** 24.123.87.42
- **SSH Port:** 224 (NOT the default 22 or 2222)
- **Username:** ubuntu
- **Password:** 6809233DjD$$$

Working SSH Command

The following SSH command has been tested and verified to work reliably:

```
sshpass -p '6809233DjD$$$' ssh -p 224 -o StrictHostKeyChecking=no -o ServerAliveInterval=30 -o ServerAliveCountMax=3 ubuntu@24.123.87.42
```

Connection Options Explained

- `-p 224` : Specifies the custom SSH port (224)
- `StrictHostKeyChecking=no` : Automatically accepts the host key (useful for automated deployments)
- `ServerAliveInterval=30` : Sends a keepalive packet every 30 seconds to maintain the connection
- `ServerAliveCountMax=3` : Allows up to 3 failed keepalive attempts before disconnecting

Tested Methods

✅ Method 1: Standard SSH with Keep-Alive (WORKING)

```
sshpass -p '6809233DjD$$$' ssh -p 224 -o StrictHostKeyChecking=no -o ServerAliveInterval=30 -o ServerAliveCountMax=3 ubuntu@24.123.87.42
```

Status: ✓ SUCCESS - This method works reliably and is recommended for all deployments.

Method 2: SSH with Compression

```
sshpass -p '6809233DjD$$$' ssh -p 224 -C -o StrictHostKeyChecking=no -o ServerAliveInterval=30 ubuntu@24.123.87.42
```

Status: Not tested (Method 1 succeeded first)

Method 3: SSH with Specific Cipher

```
sshpass -p '6809233DjD$$$' ssh -p 224 -c aes128-ctr -o StrictHostKeyChecking=no -o ServerAliveInterval=30 ubuntu@24.123.87.42
```

Status: Not tested (Method 1 succeeded first)

Method 4: SSH with Longer Timeout

```
sshpass -p '6809233DjD$$$' ssh -p 224 -o ConnectTimeout=60 -o
StrictHostKeyChecking=no -o ServerAliveInterval=30 ubuntu@24.123.87.42
```

Status: Not tested (Method 1 succeeded first)

Common Issues and Solutions

Issue: Connection Timeout

Solution: Verify that port 224 is open on the firewall and that you're using the correct port number.

Issue: "Connection refused"

Solution: Ensure the SSH service is running on the remote server and listening on port 224.

Issue: Authentication Failed

Solution: Double-check the password. Note that it contains special characters: `6809233DjD$$$`

Deployment Usage

For automated deployments, use the working SSH command in scripts:

```
# Example: Pull latest code
sshpass -p '6809233DjD$$$' ssh -p 224 \
    -o StrictHostKeyChecking=no \
    -o ServerAliveInterval=30 \
    -o ServerAliveCountMax=3 \
    ubuntu@24.123.87.42 \
    "cd ~/Sports-Bar-TV-Controller && git pull origin main"
```

Security Notes

- This configuration uses password authentication for convenience in a controlled environment
- For production environments, consider using SSH key-based authentication instead
- The `StrictHostKeyChecking=no` option should be used cautiously as it bypasses host verification

Last Updated

October 20, 2025 - Verified working during PR #220 deployment

n8n Automation Platform

Access Details

- **URL:** http://24.123.87.42:5678
- **Email:** sgtfulton@gmail.com
- **Password:** 6809233DjD$$$

Purpose

n8n is used for automation workflows, including:
- Prisma to Drizzle migration scanning
- Automated code analysis and reporting
- CI/CD integration tasks

Usage

1. Navigate to http://24.123.87.42:5678
2. Log in with the credentials above
3. Access workflows from the dashboard

Security Notes

- n8n instance is running on the same server as the application
- Credentials should be rotated periodically
- Consider implementing OAuth or SSO for production environments