■ LAMINAR INSTRUMENTS ■

CQDAM Free Edition Developer Guide

Overview

CQDAM (Congruent Quantum Data Architecture Method) Free Edition is a high-performance key-value database server designed to deliver exceptional speed and reliability. This free edition provides up to 2.5 million operations per second while maintaining full compatibility with standard database protocols.

Created by Darreck Lamar Bender II

Powered by Congruent Quantum Data Architecture Method

System Requirements

• Operating System: Linux, macOS, or Windows

• Memory: Minimum 512MB RAM (2GB recommended)

• Storage: 100MB available disk space

• Network: TCP/IP connectivity

Quick Start Guide

Installation

1. Extract the CQDAM Free Edition package

2. Run the installation script:

```
sudo ./install.sh
```

3. Start the CQDAM server:

```
./cqdam_free --port 6379
```

Basic Usage

Connect to CQDAM using any compatible client:

```
cqdam-cli -h localhost -p 6379
```

Basic operations:

```
SET mykey "Hello World"
GET mykey
DEL mykey
```

Configuration

CQDAM can be configured using the provided `cqdam.conf` file or command-line arguments.

Configuration File Options

• `port`: Server port (default: 6379)

• `bind`: IP address to bind (default: 127.0.0.1)

• `maxmemory`: Maximum memory usage

• `timeout`: Client connection timeout

• `logfile`: Path to log file

Command Line Options

• `--port <port>`: Set server port

• `--bind <ip>`: Set bind address

• `--config <file>`: Use configuration file

• `--version`: Display version information

• `--help`: Show help message

Performance Features

High-Speed Operations

• Up to 2.5 million operations per second

• Optimized memory management

• Efficient data structures

• Low-latency networking

Reliability

• Data persistence options

• Automatic error recovery

• Connection pooling support

• Health monitoring

Client Integration

Python Integration

Install compatible client library:

```
pip install cqdam-client
```

Basic Python example:

```
import cqdam_client as cqdam

client = cqdam.Client(host="localhost", port=6379)

# Set a value
client.set("greeting", "Hello CQDAM")

# Get a value
value = client.get("greeting")
print(value)  # Output: Hello CQDAM

# Close connection
client.close()
```

Node.js Integration

Install compatible client library:

```
npm install cqdam-client
```

Basic Node.js example:

```
const cqdam = require("cqdam-client");

const client = cqdam.createClient({
    host: "localhost",
    port: 6379
});

client.on("connect", () => {
    console.log("Connected to CQDAM");

    // Set a value
    client.set("counter", "1", (err, result) => {
        if (!err) {
            console.log("Value set successfully");
        }
    });
    // Get a value
```

```
    client.get("counter", (err, value) => {
        console.log("Counter value:", value);
    });
});
```

Java Integration

Add dependency to your project:

```
<dependency>
    <groupId>cqdam.clients</groupId>
    <artifactId>cqdam-java</artifactId>
    <version>1.0.0</version>
</dependency>
```

Basic Java example:

```
import cqdam.clients.java.CQDAM;

public class CQDAMExample {
    public static void main(String[] args) {
        CQDAM client = new CQDAM("localhost", 6379);

        // Set a value
        client.set("message", "Hello from Java");

        // Get a value
        String value = client.get("message");
        System.out.println("Message: " + value);

        client.close();
    }
}
```

Production Deployment

System Service Setup

Create a systemd service file:

```
[Unit]
Description=CQDAM Free Edition
After=network.target

[Service]
Type=simple
User=cqdam
ExecStart=/usr/local/bin/cqdam_free --config /etc/cqdam/cqdam.conf
Restart=always
RestartSec=10
[Install]
```

```
WantedBy=multi-user.target
```

Enable and start the service:

```
sudo systemctl enable cqdam
sudo systemctl start cqdam
```

Load Balancing

For high-availability deployments, use a load balancer like HAProxy:

```
backend cqdam_servers
    balance roundrobin
    server cqdam1 192.168.1.10:6379 check
    server cqdam2 192.168.1.11:6379 check
    server cqdam3 192.168.1.12:6379 check
```

Monitoring

Monitor CQDAM using standard tools:

• System metrics: CPU, memory, disk usage

• Network metrics: connections, throughput

• Application metrics: operations per second, response time

Basic monitoring script:

```
#!/bin/bash
while true; do
    echo "CQDAM Status: $(cqdam-cli ping)"
    echo "Memory usage: $(ps -o pid,vsz,rss,comm -p $(pgrep cqdam_free))"
    sleep 30
done
```

Data Types and Operations

Supported Data Types

CQDAM supports the following data types:

• Strings: Text values up to 512MB

• Numbers: Integer and floating-point values

• Lists: Ordered collections of strings

• Sets: Unordered collections of unique strings

• Hashes: Key-value pairs within a single key

Core Operations

String Operations

• `SET key value`: Set a string value

• `GET key`: Get a string value

• `DEL key`: Delete a key

• `EXISTS key`: Check if key exists

• `TTL key`: Get time-to-live

• `EXPIRE key seconds`: Set expiration

List Operations

• `LPUSH key value`: Add to list head

• `RPUSH key value`: Add to list tail

• `LPOP key`: Remove from list head

• `RPOP key`: Remove from list tail

• `LLEN key`: Get list length

Set Operations

• `SADD key member`: Add to set

• `SREM key member`: Remove from set

• `SMEMBERS key`: Get all members

• `SCARD key`: Get set size

Hash Operations

• `HSET key field value`: Set hash field

• `HGET key field`: Get hash field

• `HDEL key field`: Delete hash field

• `HKEYS key`: Get all hash keys

Performance Optimization

Connection Management

• Use connection pooling for multiple clients

• Reuse connections when possible

• Set appropriate timeouts

• Monitor connection counts

Data Management

• Use appropriate data types for your use case

• Set expiration times for temporary data

• Monitor memory usage

• Implement data archiving strategies

Query Optimization

• Batch multiple operations when possible

• Use pipelining for bulk operations

• Avoid long-running operations

• Monitor query performance

Troubleshooting

Common Issues

Connection Refused

• Check if CQDAM server is running

• Verify port and bind address

• Check firewall settings

• Review network connectivity

High Memory Usage

• Monitor key expiration settings

• Check for memory leaks in client code

• Review data size and structure

• Consider data archiving

Performance Issues

• Monitor system resources

• Check network latency

• Review client connection patterns

• Analyze operation types and frequency

Diagnostic Commands

Check server status:

```
cqdam-cli ping
```

Get server information:

```
cqdam-cli info
```

Monitor real-time commands:

```
cqdam-cli monitor
```

Log Analysis

CQDAM logs important events and errors. Common log patterns:

• Connection events: Client connections and disconnections

• Error conditions: Memory issues, network problems

• Performance metrics: Operation counts, response times

Security Considerations

Network Security

• Use firewalls to restrict access

• Consider VPN for remote connections

• Monitor connection sources

• Implement rate limiting

Data Security

• Enable authentication if required

• Use TLS encryption for sensitive data

• Regular security updates

• Access logging and monitoring

Backup and Recovery

• Regular data backups

• Test recovery procedures

• Monitor backup integrity

• Document recovery processes

Upgrading to Enterprise

CQDAM Enterprise Edition offers enhanced features:

Enhanced Performance

• Up to 7+ million operations per second

• Advanced caching algorithms

• Optimized for high-concurrency workloads

• Enhanced memory management

Advanced Features

• Multi-node clustering

• Automatic failover

• Advanced monitoring and metrics

• Professional support

Enterprise Support

• 24/7 technical support

• Custom integration assistance

• Performance optimization consulting

• Priority bug fixes and updates

For Enterprise Edition information:

• Email: darreck@laminarinstruments.com

• Website: https://laminarinstruments.com/enterprise

Support and Resources

Community Support

• Documentation: Complete user guides and API references

• Examples: Sample code and integration patterns

• Best practices: Performance and deployment guidelines

Professional Support

For production deployments and enterprise needs:

• Email: darreck@laminarinstruments.com

• Website: https://laminarinstruments.com

• Response time: Within 24 hours for support inquiries

Additional Resources

• Configuration examples

• Performance benchmarking tools

• Integration templates

• Monitoring scripts

CQDAM Free Edition v1.0

Created by Darreck Lamar Bender II

Powered by Congruent Quantum Data Architecture Method

For Enterprise solutions with enhanced performance and features:

darreck@laminarinstruments.com