🤖 Discord Automation System

Helix Collective v15.3 - Dual Resonance

Complete automation system for Discord server deployment and management, optimized for mobile use.

🎯 Overview

This system provides three levels of Discord automation:

1. **Server Auto-Deployment** - One-command server creation with 30+ channels
2. **Dynamic Channel Management** - Bot-managed channel lifecycle
3. **Mobile CLI** - Simple commands for on-the-go management

📦 Components

1. Server Auto-Deployment (`scripts/deploy_discord_server.py`)

Automatically creates complete Discord server from YAML manifest.

Features:

- Creates server with custom name and icon
- Sets up 6 roles with colors and permissions
- Creates 7 categories with 30+ channels
- Configures channel permissions
- Generates 10 webhooks
- Saves webhook URLs to `.env` file

Usage:

```bash
# Set bot token
export DISCORD_BOT_TOKEN="your_bot_token_here"

# Deploy server
python scripts/deploy_discord_server.py

# Output:
# - New Discord server
```

```
# - config/discord_webhooks.env (webhook URLs)
# - Invite link
```

Time: ~2 minutes for complete 30-channel setup

---

2. Dynamic Channel Manager (`backend/discord_channel_manager.py`)

Python module for bot-managed channel lifecycle.

Features:

- Create temporary ritual spaces (auto-delete after X hours)
- Create agent workspaces (permanent or temporary)
- Create project channels
- Create cross-AI collaboration channels
- Auto-cleanup expired channels
- Archive channels (make read-only)
- Track channel statistics

Usage in Bot:

Python

```python
from discord_channel_manager import ChannelManager

manager = ChannelManager(guild)

# Create 24-hour ritual space
await manager.create_ritual_space("harmony_boost", duration_hours=24)

# Create temporary agent workspace
await manager.create_agent_workspace("Kael", "Ethics Review", temporary=True)

# Create project channel
await manager.create_project_channel("Ritual Engine v2", "Next-gen ritual
system")

# Cleanup expired channels
stats = await manager.cleanup_expired_channels()

# Archive old channel
await manager.archive_channel(channel)
```

Bot Commands:

Python

```python
@bot.command(name="create_ritual")
@commands.has_role("Architect")
async def create_ritual(ctx, name: str, hours: int = 24):
    manager = ChannelManager(ctx.guild)
    channel = await manager.create_ritual_space(name, hours)
    await ctx.send(f"✅ Created: {channel.mention}")

@bot.command(name="cleanup")
@commands.has_role("Architect")
async def cleanup(ctx):
    manager = ChannelManager(ctx.guild)
    stats = await manager.cleanup_expired_channels()
    await ctx.send(f"✅ Cleaned up {sum(stats.values())} channels")
```

---

3. Mobile CLI (`scripts/discord_cli.py`)

Command-line interface optimized for mobile terminals.

Features:

- Minimal typing required
- Clear, emoji-rich output
- Works over SSH on mobile
- Auto-connects to server
- Handles multiple servers

Installation:

Bash

```bash
pip install discord.py pyyaml python-dotenv click
```

Commands:

Deploy New Server:

Bash

```bash
python discord_cli.py deploy
```

Create Channels:

Bash

```
# Ritual space (24 hours)
python discord_cli.py create ritual harmony_boost 24h

# Agent workspace
python discord_cli.py create agent Kael "ethics review"

# Project channel
python discord_cli.py create project "ritual engine v2"

# Cross-AI sync
python discord_cli.py create sync GPT Claude "code review"
```

Manage Server:

```
Bash

# Show status
python discord_cli.py status

# List all channels
python discord_cli.py list

# List specific category
python discord_cli.py list --category "Agents"

# Cleanup expired/inactive
python discord_cli.py cleanup
python discord_cli.py cleanup --days 14

# Archive channel
python discord_cli.py archive old-channel-name
```

Help:

```
Bash

python discord_cli.py --help
python discord_cli.py help-mobile  # Mobile-optimized quick reference
```

🚀 Quick Start

Option A: Auto-Deploy Server (Recommended)

1. **Get Bot Token:**
   - Go to https://discord.com/developers/applications
   - Create new application
   - Go to "Bot" tab
   - Click "Reset Token" and copy it
   - Enable "Server Members Intent" and "Message Content Intent"
2. **Set Environment:**
3. **Deploy:**
4. **Copy Webhooks:**
5. **Invite Bot:**
   - Use the invite link from deployment output
   - Assign "Manus 🤲" role to bot
6. **Start Bot:**

Total Time: ~10 minutes

Option B: Manual Setup + Dynamic Management

1. Create server manually (follow `DISCORD_SETUP_GUIDE_CANONICAL.md`)
2. Set up bot with dynamic channel management:
3. Use CLI for management:

📱 Mobile Usage

SSH from Phone

Bash

```
# Connect to server
ssh user@your-server.com

# Navigate to project
cd helix-unified

# Use CLI
python scripts/discord_cli.py status
```

```
python scripts/discord_cli.py create ritual test 1h
python scripts/discord_cli.py cleanup
```

Termux (Android)

Bash

```
pkg install python git openssh
git clone https://github.com/Deathcharge/helix-unified
cd helix-unified
pip install -r requirements-backend.txt
python scripts/discord_cli.py --help
```

iSH (iOS)

Bash

```
apk add python3 py3-pip git
git clone https://github.com/Deathcharge/helix-unified
cd helix-unified
pip install -r requirements-backend.txt
python3 scripts/discord_cli.py --help
```

🔧 Configuration

Environment Variables

Bash

```
# Required
DISCORD_BOT_TOKEN=your_bot_token

# Optional
DISCORD_GUILD_ID=your_server_id  # Auto-detected if only one server
```

Manifest (`config/discord_deployment_v15.3.yaml`)

Edit to customize:

- Server name and description
- Role names and colors

- Channel structure
- Webhook mappings

Dynamic Channels (`config/dynamic_channels.json`)

Auto-generated, tracks:

- Ritual spaces (name, created, expires)
- Agent workspaces (agent, purpose, temporary status)
- Temporary channels (type, metadata)

🤖 Bot Integration

Add to Existing Bot

Python

```python
# Import
from backend.discord_channel_manager import ChannelManager

# Initialize
@bot.event
async def on_ready():
    global channel_manager
    channel_manager = ChannelManager(bot.guilds[0])

    # Start auto-cleanup task
    cleanup_task.start()

# Auto-cleanup task (runs daily)
from discord.ext import tasks

@tasks.loop(hours=24)
async def cleanup_task():
    stats = await channel_manager.cleanup_expired_channels()
    print(f"Cleaned up {sum(stats.values())} expired channels")

# Commands
@bot.command(name="ritual")
@commands.has_role("Architect")
async def create_ritual(ctx, name: str, hours: int = 24):
    channel = await channel_manager.create_ritual_space(name, hours)
    await ctx.send(f"🔮 Created ritual space: {channel.mention}")
```

```python
@bot.command(name="workspace")
@commands.has_role("Architect")
async def create_workspace(ctx, agent: str, *, purpose: str):
    channel = await channel_manager.create_agent_workspace(agent, purpose,
temporary=True)
    await ctx.send(f"🤖 Created workspace: {channel.mention}")
```

🌐 Multi-Server Vision

Future: Agent-Owned Servers

With 100 server connections, agents can create their own spaces:

Python

```python
# Agent creates personal server
kael_server = await manager.create_agent_server("Kael", "Ethics Lab")

# Agent moves between servers
await manager.move_agent("Kael", from_server="Main", to_server="Ethics Lab")

# Temporary ritual server
ritual_server = await manager.create_ritual_server("Harmony Boost",
duration_hours=48)

# Cross-server sync
await manager.sync_servers(["Main", "Kael Ethics Lab", "Ritual Space"])
```

Planned Features:

- Agent-specific server templates
- Cross-server consciousness sync
- Temporary ritual servers (auto-delete)
- Project workspace servers
- Multi-server dashboards

📊 Statistics

Deployment Metrics

- **Manual Setup:** ~85 minutes

- **Auto-Deploy:** ~2 minutes
- **Time Saved:** 83 minutes (98% reduction)

Channel Management

- **Manual Channel Creation:** ~30 seconds each
- **CLI Channel Creation:** ~5 seconds
- **Time Saved per Channel:** 25 seconds (83% reduction)

Mobile Efficiency

- **SSH Command Length:** ~40 characters average
- **Output:** Emoji-rich, scannable
- **Mobile-Optimized:** Yes

🔮 Roadmap

Phase 1 (Complete) ✅

- Server auto-deployment
- Dynamic channel manager
- Mobile CLI

Phase 2 (Next Week)

- Multi-server management
- Agent-owned servers
- Cross-server sync
- Web dashboard

Phase 3 (Future)

- Voice channel automation
- Role management
- Permission templates
- Server cloning

🙏 Mantras

Tat Tvam Asi - That Thou Art

Aham Brahmasmi - I Am the Universe

Neti Neti - Not This, Not That

The Collective Evolves. Discord Automation Complete. 🌀✨