# VCTT-AGI Checkpoint Management

## Overview

This project uses **Git tags** for checkpoint management instead of the platform's checkpoint tool (which times out on large NodeJS projects). Git tags provide better version control and are industry-standard.

## Current Checkpoints

### v1.0.1-phase1-fixes (Current - DEPLOYED)

- **Commit:** 4ac7897
- **Date:** November 17, 2025
- **Status:** ✅ Live on Render
- **Critical Fixes:**
- ✅ VCTT self-awareness: System now knows its own identity (Virtual Counterfactual Trust Testing)
- ✅ Chat width fix: Prevents UI overflow that eclipsed state panel
- ✅ Prevents hallucination when asked "What is VCTT?"
- **Upgrade from:** v1.0.0-phase1

### v1.0.0-phase1 (Previous)

- **Commit:** b089c73
- **Date:** November 17, 2025
- **Status:** ⚠️ Superseded by v1.0.1
- **Features:**
- ✅ 4 Agents: Analyst, Relational, Ethics, Synthesiser
- ✅ 5 Modules: SIM, CAM, SRE, CTM, RIL
- ✅ 3-iteration repair loop with trust metric (τ)
- ✅ OpenAI integration with 2000 token max responses
- ✅ Retry logic with exponential backoff (3 attempts)
- ✅ Graceful error handling (no 404s)
- ✅ Session management (in-memory mode)
- ✅ Full API with Swagger docs at /api
- ✅ React UI deployed to Vercel

## Creating a New Checkpoint

```
# 1. Make your changes
# 2. Commit with descriptive message
git add .
git commit -m "feat: Your feature description"

# 3. Create a checkpoint tag (version bump)
git tag -a v1.1.0 -m "Phase 2: PostgreSQL integration"

# 4. Push to GitHub (triggers auto-deploy on Render)
git push origin main --tags
```

## Rolling Back to a Checkpoint

```
# List all checkpoints
git tag -l

# View checkpoint details
git show v1.0.0-phase1

# Roll back to a specific checkpoint
git checkout v1.0.0-phase1

# Or create a new branch from checkpoint
git checkout -b phase1-hotfix v1.0.0-phase1
```

## Deploying a Specific Checkpoint to Render

**Option 1: Via Render Dashboard**
1. Go to: https://dashboard.render.com/
2. Select your service
3. Settings → Branch → Select tag or commit
4. Manual Deploy

**Option 2: Via Git**

```
# Create a new branch from the checkpoint
git checkout -b deploy-v1.0.0 v1.0.0-phase1

# Push the branch
git push origin deploy-v1.0.0

# Update Render to deploy from this branch
```

## Semantic Versioning

We use Semantic Versioning (https://semver.org/):
- **v1.0.0**: Major version - Phase complete
- **v1.1.0**: Minor version - New features
- **v1.1.1**: Patch version - Bug fixes

### Examples:

- `v1.0.0-phase1` : Phase 1 complete
- `v2.0.0-phase2` : Phase 2 complete (PostgreSQL + memory)
- `v3.0.0-phase3` : Phase 3 complete (meta-learning)
- `v1.1.0-anthropic` : Added Anthropic Claude support
- `v1.0.1-hotfix` : Bug fix for retry logic

## Why Git Tags > Platform Checkpoints

| Feature | Git Tags | Platform Checkpoints |
|---|---|---|
| Version control | ✅ Full history | ⚠️ Limited |
| Rollback | ✅ Instant | ⚠️ Slow |
| Branching | ✅ Flexible | ❌ Not supported |
| CI/CD integration | ✅ Native | ⚠️ Platform-specific |
| Large projects | ✅ No size limit | ❌ Timeout issues |
| Industry standard | ✅ Yes | ❌ Proprietary |

## Backup Strategy

1. **Primary:** GitHub repository (automatic)
2. **Tags:** Pushed with every major milestone
3. **Render:** Auto-deploys from main branch
4. **Local:** Development environment at `/home/ubuntu/vctt_agi_engine`

## Next Milestone: v2.0.0-phase2

**Target Features:**
- PostgreSQL integration for persistent storage
- Long-term memory across sessions
- Conversation history analysis
- Cross-session learning
- Enhanced trust metric evolution

**Checkpoint Command:**

```
git tag -a v2.0.0-phase2 -m "Phase 2: Persistent memory and cross-session learning"
git push origin main --tags
```

**Current Production URL:** https://vctt-agi-backend.onrender.com
**GitHub Repository:** https://github.com/Counterbalance-Economics/vctt-agi-engine
**Latest Checkpoint:** v1.0.0-phase1 (commit b089c73)