Operations Runbook

Version: 3.1.0
Purpose: Standard operating procedures for Whale Hunter system

---

1. Start/Stop Components

Start Full System

```
cd /home/ubuntu/whale_hunter
source venv/bin/activate

# Start in order:
python scripts/bootstrap.py           # Initialize/verify
python scripts/harvester.py --daemon &  # Background harvester
python scripts/god_view.py --watch &    # Background graph analysis
python src/main.py                      # Main trading loop
```

Stop Full System

```
# Graceful shutdown
kill -SIGTERM $(pgrep -f "whale_hunter")

# Or force stop
kill -9 $(pgrep -f "whale_hunter")
```

Start Individual Components

```
# Harvester only
python scripts/harvester.py

# Paper trading logger only
python scripts/phase0_logger.py

# God View analysis only
python scripts/god_view.py

# Backtest only
python scripts/backtest.py --start 2024-01-01 --end 2024-03-01
```

Systemd Service (Production)

```
# Start
sudo systemctl start whale-hunter

# Stop
sudo systemctl stop whale-hunter

# Restart
sudo systemctl restart whale-hunter

# Status
sudo systemctl status whale-hunter

# Logs
journalctl -u whale-hunter -f
```

---

2. Rotate API Keys

Procedure

```
# 1. Generate new keys from provider dashboards
# - Helius: https://dashboard.helius.dev/
# - BirdEye: https://birdeye.so/developers
# - Solscan: https://pro.solscan.io/

# 2. Update .env file
cp config/.env config/.env.backup
nano config/.env

# 3. Update values:
# HELIUS_API_KEY=new_key_here
# BIRDEYE_API_KEY=new_key_here
# SOLSCAN_API_KEY=new_key_here

# 4. Restart services
sudo systemctl restart whale-hunter

# 5. Verify connectivity
python -c "from src.api_clients import create_helius_client; import asyncio;
asyncio.run(create_helius_client())"

# 6. Delete old keys from provider dashboards after 24h verification
```

Rotation Schedule

- Monthly: All API keys
- Immediately: If compromise suspected
- Before major phase transitions

---

3. Rotate Execution Wallets

Procedure

```
# 1. Generate new wallets
solana-keygen new -o ~/.config/solana/whale_exec_new.json

# 2. Fund new wallet (small amount for testing)
solana transfer NEW_WALLET_ADDRESS 0.1 --from OLD_WALLET

# 3. Update config
nano config/settings.yaml
# Update execution_wallets list

# 4. Test with paper trade
python scripts/phase0_logger.py
# Execute test signal

# 5. Transfer remaining funds from old wallet
solana transfer NEW_WALLET_ADDRESS ALL --from OLD_WALLET

# 6. Archive old wallet key
mv ~/.config/solana/whale_exec_old.json ~/backups/archived_wallets/

# 7. Update .env with new wallet paths
nano config/.env
```

Rotation Triggers

- Monthly rotation
- If wallet detected being followed
- After significant losses
- Before phase transitions

---

4. Trigger Kill Switches Manually

Graph Kill Switch

```
# Activate
python -c "
from src.database import WalletGraphDB
db = WalletGraphDB('data/wallet_graph.db')
db.store_system_state('mode', 'GRAPH_KILL_SWITCH')
db.log_kill_switch_event('MANUAL', {'reason': 'Operator triggered'})
print('Graph Kill Switch ACTIVATED')
"

# Verify
python -c "
from src.database import WalletGraphDB
db = WalletGraphDB('data/wallet_graph.db')
print(f'Current mode: {db.get_system_state(\"mode\")}')
"
```

Full Kill Switch

```
# Activate
python -c "
from src.database import WalletGraphDB
db = WalletGraphDB('data/wallet_graph.db')
db.store_system_state('mode', 'KILL_SWITCH')
db.log_kill_switch_event('MANUAL_FULL', {'reason': 'Operator triggered full stop'})
print('Full Kill Switch ACTIVATED - ALL trading halted')
"

# Stop all processes
sudo systemctl stop whale-hunter
```

Capital Preservation Mode

```
# Activate
python -c "
from src.database import WalletGraphDB
db = WalletGraphDB('data/wallet_graph.db')
db.store_system_state('mode', 'CAPITAL_PRESERVATION')
print('Capital Preservation Mode ACTIVATED')
"
```

---

5. Resume After Observation Mode

Checklist Before Resume

```
# 1. Review logs
tail -500 logs/whale_hunter.log | grep -E "ERROR|WARNING|KILL"

# 2. Check current state
python -c "
from src.database import WalletGraphDB
db = WalletGraphDB('data/wallet_graph.db')
print(f'Mode: {db.get_system_state(\"mode\")}')
print(f'Capital: {db.get_system_state(\"capital_sol\")} SOL')
print(f'Trade Count: {db.get_system_state(\"trade_count\")}')
"

# 3. Verify API connectivity
python scripts/bootstrap.py --verify-only

# 4. Review recent signals
python scripts/phase0_logger.py --stats

# 5. If all checks pass, resume
python -c "
from src.database import WalletGraphDB
db = WalletGraphDB('data/wallet_graph.db')
db.store_system_state('mode', 'NORMAL')
print('Resumed NORMAL mode')
"

# 6. Restart services
sudo systemctl start whale-hunter
```

---

6. Resume After Capital Preservation Mode

Required Steps

```
# 1. Analyze what caused the drawdown
python scripts/backtest.py --last-30-days --verbose

# 2. Review losing trades
sqlite3 data/wallet_graph.db "
SELECT timestamp, token_address, pnl_sol, exit_reason
FROM trades
WHERE pnl_sol < 0
ORDER BY timestamp DESC
LIMIT 20;
"

# 3. Check current capital vs peak
python -c "
from src.database import WalletGraphDB
db = WalletGraphDB('data/wallet_graph.db')
current = float(db.get_system_state('capital_sol'))
peak = float(db.get_system_state('peak_capital_sol'))
drawdown = (peak - current) / peak * 100
print(f'Current: {current} SOL')
print(f'Peak: {peak} SOL')
print(f'Drawdown: {drawdown:.1f}%')
"

# 4. Document findings and decision
echo "$(date): Resuming from CP mode. Reason: [your analysis]" >> logs/decisions.log

# 5. Manual confirmation required
read -p "Type RESUME to confirm exit from Capital Preservation: " confirm
if [ "$confirm" = "RESUME" ]; then
    python -c "
from src.database import WalletGraphDB
db = WalletGraphDB('data/wallet_graph.db')
db.store_system_state('mode', 'NORMAL')
print('Capital Preservation Mode DEACTIVATED')
"
fi
```

7. Daily Operations Checklist

```
□ Check system status
  - systemctl status whale-hunter
  - Check logs for errors

□ Review overnight signals
  - python scripts/phase0_logger.py --stats
  - Note any unusual patterns

□ Verify capital balance
  - solana balance (execution wallet)
  - Compare to recorded state

□ Check API rate limit usage
  - Review API dashboard metrics
  - Ensure <80% of limits used

□ Monitor drawdown
  - Current vs peak capital
  - Alert if >10%

□ Quick database health check
  - sqlite3 data/wallet_graph.db "PRAGMA integrity_check;"

□ Review any pending manual approvals (First 50 trades)
```

---

8. Weekly Maintenance Checklist

```
□ Database backup
  - cp data/wallet_graph.db ~/backups/wallet_graph_$(date +%Y%m%d).db
  - Verify backup: sqlite3 ~/backups/wallet_graph_*.db "PRAGMA integrity_check;"

□ Log rotation
  - Archive logs older than 7 days
  - Compress: gzip logs/whale_hunter.log.1

□ Performance review
  - python scripts/backtest.py --last-7-days
  - Win rate, PnL, false positive rate

□ CEX blacklist update
  - Check for new CEX addresses
  - Update config/cex_blacklist.yaml if needed

□ System updates (monthly)
  - pip list --outdated
  - Update non-critical packages

□ Kill switch test (monthly)
  - Test manual activation/deactivation
  - Document test results

□ Review Go/No-Go metrics
  - Are we ready for next phase?
  - python scripts/backtest.py --go-no-go
```

9. Emergency Procedures

E1: Unexpected Large Loss

```
# 1. Immediate stop
sudo systemctl stop whale-hunter

# 2. Activate kill switch
python -c "from src.database import WalletGraphDB; db = WalletGraphDB('data/wal-
let_graph.db'); db.store_system_state('mode', 'KILL_SWITCH')"

# 3. Document the incident
echo "$(date): EMERGENCY STOP - Large loss detected" >> logs/incidents.log

# 4. Analyze
sqlite3 data/wallet_graph.db "SELECT * FROM trades ORDER BY timestamp DESC LIMIT 5;"

# 5. Do NOT resume without full investigation
```

E2: API Keys Compromised

```
# 1. Revoke all keys immediately (from provider dashboards)
# 2. Stop all services
sudo systemctl stop whale-hunter

# 3. Generate new keys
# 4. Update .env
# 5. Audit for unauthorized activity
# 6. Resume only after security review
```

E3: Database Corruption

```
# 1. Stop services
sudo systemctl stop whale-hunter

# 2. Attempt recovery
sqlite3 data/wallet_graph.db ".recover" | sqlite3 data/wallet_graph_recovered.db

# 3. If recovery fails, restore from backup
cp ~/backups/wallet_graph_LATEST.db data/wallet_graph.db

# 4. Verify
sqlite3 data/wallet_graph.db "PRAGMA integrity_check;"

# 5. Resume
sudo systemctl start whale-hunter
```

E4: Server Unresponsive

```
# 1. SSH from backup machine
ssh user@server_ip

# 2. Check processes
ps aux | grep whale

# 3. Check resources
free -h
df -h
top

# 4. Kill stuck processes
kill -9 $(pgrep -f whale_hunter)

# 5. Restart
sudo systemctl restart whale-hunter
```

Contact & Escalation

Severity	Response Time	Action
Low	24 hours	Log and review
Medium	4 hours	Investigate, may pause
High	1 hour	Stop trading, investigate
Critical	Immediate	Full stop, all hands

Log all incidents in: `logs/incidents.log`