Cosmos Chain Operations - Complete Documentation

Overview

`cosmos-ops.sh` is a comprehensive script for managing Cosmos-based validator operations including voting, balance checking, reward withdrawals, and automated governance participation.

Features

- ✅ **Vote on Governance Proposals** - Manual and automated voting
- ✅ **Check Wallet Balances** - View balances, delegations, and rewards
- ✅ **Withdraw Rewards** - Single command for rewards + commission
- ✅ **Consensus State Monitoring** - Real-time consensus information
- ✅ **Auto-voting Mode** - Continuous monitoring and voting
- ✅ **Telegram Notifications** - Real-time alerts for all operations
- ✅ **Smart Gas Estimation** - Automatic retry with adjusted gas
- ✅ **Secure Password Management** - GPG encryption or manual entry
- ✅ **Dry-run Mode** - Test operations without execution

Installation

1. Download the Script

```
chmod +x cosmos-ops.sh
```

2. Set Up Configuration Files

Create your chain config (e.g., `jackal.conf`) with the following structure:

```
# Chain identification
project_name: jackal
daemon_name: canined
chain_id: jackal-1

# Endpoints
rpc_url: http://localhost:13717          # REST API endpoint
rpc_endpoint: http://localhost:26657     # RPC endpoint

# Keyring configuration
keyring_backend: file
```

```
key_name: validator

# Optional: GPG-encrypted password file
password_file: ~/.cosmos-ops/jackal-password.gpg

# Gas settings
max_gas_fee: 0.1ujkl
gas_adjustment: 1.3

# Cosmovisor settings (for upgrade.sh compatibility)
cosmovisor_home: ~/.canine/cosmovisor
# ... other upgrade.sh fields ...
```

3. Set Up Telegram Notifications (Optional)

Create `telegram.conf`:

```
bot_token: 123456789:ABCdefGHIjklMNOpqrsTUVwxyz
chat_id: -1001234567890
enabled: true
```

To get your bot token:

1. Message @BotFather on Telegram
2. Create a new bot: `/newbot`
3. Copy the token

To get your chat ID:

1. Message your bot
2. Visit: `https://api.telegram.org/bot&lt;YOUR_TOKEN&gt;/getUpdates`
3. Find your chat_id in the response

Password Management

Option 1: GPG Encrypted Password (Recommended)

```
# Create encrypted password file
echo "your-keyring-password" | gpg --encrypt --recipient your-email@example.com -o ~/.cos

# Set permissions
chmod 600 ~/.cosmos-ops/jackal-password.gpg

# Configure in jackal.conf
password_file: ~/.cosmos-ops/jackal-password.gpg
```

Option 2: Manual Entry

Leave `password_file` empty or omit it. The script will prompt for password when needed.

Usage

Vote on Proposal

```
# Vote YES
./cosmos-ops.sh --config jackal.conf --vote yes --proposal-id 24

# Vote NO
./cosmos-ops.sh --config jackal.conf --vote no --proposal-id 25

# Vote ABSTAIN
./cosmos-ops.sh --config jackal.conf --vote abstain --proposal-id 26

# Vote NO_WITH_VETO
./cosmos-ops.sh --config jackal.conf --vote no_with_veto --proposal-id 27

# Dry run (test without executing)
./cosmos-ops.sh --config jackal.conf --vote yes --proposal-id 24 --dry-run
```

Output Example:

```
[INFO] ==========================================
[INFO] VOTING ON PROPOSAL #24
[INFO] ==========================================
[INFO] Fetching proposal details...
[INFO] Proposal: v5.0.0: Baobab
[INFO] Vote option: yes
[INFO] Simulating transaction with gas-adjustment=1.3...
[SUCCESS] Transaction submitted successfully
[TX] TxHash: ABC123DEF456...
[SUCCESS] Vote submitted successfully!
```

Check Balance

```
./cosmos-ops.sh --config jackal.conf --balance
```

Shows:

- Available balance
- Delegated tokens
- Pending rewards

Withdraw Rewards + Commission

```
./cosmos-ops.sh --config jackal.conf --withdraw-rewards
```

Single command that:

- Withdraws all delegation rewards
- Withdraws validator commission
- Uses optimized gas estimation

Output Example:

```
[INFO] ==========================================
[INFO] WITHDRAWING REWARDS + COMMISSION
[INFO] ==========================================
[INFO] Validator address: jklvaloper1abc...
[INFO] Withdrawing rewards and commission...
[SUCCESS] Rewards and commission withdrawn successfully!
[TX] TxHash: DEF789GHI012...
```

Check Consensus State

```
./cosmos-ops.sh --config jackal.conf --consensus-state
```

Displays:

- Prevotes bit array
- Precommits bit array
- Current round
- Block height

Output Example:

```
[INFO] CONSENSUS STATE
Prevotes:
BA{30:xxxxx_________________}

Precommits:
BA{30:xxxxxxxxx____________}

Round: 0
Height: 15189345
```

Auto-Vote Mode (Continuous)

```
# Auto-vote YES on all proposals
./cosmos-ops.sh --config jackal.conf --auto-vote yes --interval 360

# Auto-vote with custom interval (10 minutes)
./cosmos-ops.sh --config jackal.conf --auto-vote abstain --interval 600
```

Behavior:

- Checks for new proposals every N seconds (default: 360)
- Automatically votes on any proposal in VOTING_PERIOD
- Sends Telegram notification for each vote
- Tracks voted proposals to avoid duplicates
- Runs continuously until stopped (Ctrl+C)

⚠ **Warning:** This mode votes on ALL proposals including spam. Use at your own risk!

Output Example:

```
[INFO] ==========================================
[INFO] AUTO-VOTE MODE STARTED
[INFO] ==========================================
[INFO] Vote option: yes
[INFO] Check interval: 360 seconds
[WARN] ⚠  Will vote yes on ALL proposals (including spam)!
[INFO] Checking for new proposals...
[INFO] Found new proposal: #24
[INFO] Voting on proposal #24...
[SUCCESS] Voted on proposal #24
[INFO] Waiting 360 seconds until next check...
```

Gas Management

The script automatically handles gas estimation and retries:

1. **Initial Transaction**: Uses `--gas auto --gas-adjustment 1.3`
2. **If Out of Gas**: Automatically retries with `--gas-adjustment 1.5`
3. **Further Failures**: Increases to `2.0`, then `2.5` (maximum)
4. **Fee Limit**: Checks against `max_gas_fee` before broadcasting

Configuration:

```
# In jackal.conf
max_gas_fee: 0.1ujkl        # Maximum acceptable fee
gas_adjustment: 1.3         # Initial multiplier
```

Telegram Notifications

When enabled, you'll receive notifications for:

Successful Vote

```
▯ Jackal Validator
Voted YES on Proposal #24
Status: Success ✅
TxHash: ABC123...
Time: 2025-10-24 16:30:00
```

Successful Withdrawal

```
▯ Jackal Validator
Withdrew rewards + commission ✅
TxHash: DEF456...
Time: 2025-10-24 16:35:00
```

Failed Operation

```
▯ Jackal Validator
Failed to vote on Proposal #24 ❌
Time: 2025-10-24 16:40:00
```

Auto-Vote Started

```
▯ Jackal Auto-Voter
Auto-vote mode started
Default vote: YES
Interval: 360s
```

Logging

All operations are logged to:

```
~/.blockchain_upgrade_logs/cosmos-ops_YYYYMMDD.log
```

Log Format:

```
2025-10-24 16:30:00 [INFO] Fetching proposal 24...
2025-10-24 16:30:01 [SUCCESS] Vote submitted successfully
2025-10-24 16:30:01 [TX] TxHash: ABC123...
```

Log Levels:

- `[INFO]` - General information
- `[SUCCESS]` - Successful operations
- `[WARN]` - Warnings (non-fatal)
- `[ERROR]` - Errors (operation failed)
- `[TX]` - Transaction hashes

Security Best Practices

1. Password File Permissions

```
chmod 600 ~/.cosmos-ops/jackal-password.gpg
chmod 700 ~/.cosmos-ops
```

2. Script Permissions

```
chmod 500 cosmos-ops.sh  # Read + execute for owner only
```

3. Audit Logging

All operations are logged with timestamps and can be audited.

4. Telegram Token Security

Store `telegram.conf` with restricted permissions:

```
chmod 600 telegram.conf
```

Troubleshooting

Password Decryption Fails

Problem: GPG decryption fails

```
[WARN] GPG decryption failed, will prompt for password
```

Solution:

- Check GPG key is available: `gpg --list-keys`
- Verify file exists: `ls -la ~/.cosmos-ops/jackal-password.gpg`
- Re-encrypt password file

Transaction Out of Gas

Problem: Transaction fails with "out of gas"

Solution: Script automatically retries with higher gas adjustment. If it keeps failing:

- Increase `gas_adjustment` in config (e.g., from 1.3 to 1.5)
- Check `max_gas_fee` isn't too low

Telegram Notifications Not Working

Problem: No Telegram messages received

Solution:

1. Verify `telegram.conf` exists in script directory
2. Check `enabled: true` in config
3. Test bot token: Message your bot manually
4. Verify chat_id is correct (with minus sign for groups)

Validator Address Not Found

Problem: "Failed to get validator address"

Solution:

- Verify key exists: `canined keys list --keyring-backend file`
- Check key name matches config
- Ensure key has validator address (not just account)

Running as Systemd Service (Auto-Vote)

Create Service File

```
sudo nano /etc/systemd/system/jackal-autovote.service
```

```
[Unit]
Description=Jackal Auto-Voter
After=network.target

[Service]
Type=simple
User=validator
WorkingDirectory=/home/validator/cosmos-tools
ExecStart=/home/validator/cosmos-tools/cosmos-ops.sh --config jackal.conf --auto-vote yes
```

```
Restart=always
RestartSec=60

[Install]
WantedBy=multi-user.target
```

Enable and Start

```
sudo systemctl daemon-reload
sudo systemctl enable jackal-autovote
sudo systemctl start jackal-autovote

# Check status
sudo systemctl status jackal-autovote

# View logs
sudo journalctl -u jackal-autovote -f
```

Multiple Validators

Use separate config files for each chain:

```
# Jackal
./cosmos-ops.sh --config jackal.conf --balance

# Osmosis
./cosmos-ops.sh --config osmosis.conf --balance

# Cosmos Hub
./cosmos-ops.sh --config cosmoshub.conf --withdraw-rewards
```

Each config has its own:

- Chain ID
- Endpoints
- Keys
- Password file
- Gas settings

Compatibility

- ✅ **Works with all Cosmos SDK chains**
- ✅ **Compatible with existing** upgrade.sh
- ✅ **Supports all Cosmos SDK versions** (v0.42+)
- ✅ **Tested on:** Ubuntu 20.04+, Debian 11+

Support

For issues or questions:

1. Check troubleshooting section
2. Review logs in `~/.blockchain_upgrade_logs/`
3. Test with `--dry-run` first
4. Verify config file syntax

Version History

v1.0 (2025-10-24)

- Initial release
- Vote, balance, withdraw, consensus operations
- Auto-voting mode
- Telegram notifications
- GPG password support
- Smart gas estimation