Chief Keeper

Keeper that lifts the hat and streamlines executive actions

Introduction

Thechief-keeper monitors and interacts withDSChief and DSSSpells, which is the executive voting contract and a type of proposal object of theMaker Protocol .

Its purpose is to lift thehat in DSChief as well as streamline executive actions.

Tolift a spell, that spell must have more approvals than the currenthat . The approvals of this spell can fluctuate and be surpassed by other spells, some of which could be malicious. This keeper "guards" thehat by ensuring the spell with the most approval is always thehat . Thechief-keeper does this in order to maximize the barrier of entry (approval) tolift a spell to the hat, thus acting as a "guard" against malicious governance actions.

While in operation, thechief-keeper :

- Monitors each new block for a change in the state of executive votes
- lift
- s the hat for the spell (yay
-) most favored (approvals[yay]
-)
- Schedules spells in the GSM by callingDSSSpell.schedule()
- Executes spells after theireta
- has elapsed in the GSM by callingDSSSpell.cast()
-

Review

The following section assumes familiarity with theDSChief , DSSSpells, andDSPause (Governance Security Module), as well as the processes withinMakerDAO Governance .

Architecture

?

chief-keeper interacts directly with theDS-Chief andDSSSpell s.

Operation

This keeper is run continuously, and saves a local database ofyays (spell addresses) and anyay:eta dictionary to reduce chain state reads. If you'd like to create your own database from scratch, first deletesrc/database/db_mainnet.json before runningbin/chief-keeper ; the initial query could take up to 15 minutes.

Installation

Prerequisites:

- Python v3.6.6
- virtualenv
-
    - This project requiresvirtualenv
-
    - to be installed if you want to use Maker's python tools. This helps with making sure that you are running the right version of python and checks that all of the pip packages that are installed in theinstall.sh
-
    - are in the right place and have the right versions.
- *
-

In order to clone the project and install required third-party packages please execute:

```

Copy git clone https://github.com/makerdao/chief-keeper.git cd chief-keeper git submodule update --init --recursive ./install.sh

```

If tinydb isn't visible/installed through ./install.sh , simply run pip3 install tinydb after the commands above.

For some known Ubuntu and macOS issues see the pymaker README.

Sample Startup Script

Make a run-chief-keeper.sh to easily spin up the chief-keeper.

```

Copy

# !/bin/bash

/full/path/to/chief-keeper/bin/chief-keeper \ --rpc-host 'sample.ParityNode.com' \ --network 'kovan' \ --eth-from '0xABCAddress' \ --eth-key 'key_file=/full/path/to/keystoreFile.json,pass_file=/full/path/to/passphrase/file.txt' \ --chief-deployment-block 14374534

```

Testing

- Download docker and docker-compose
-

This project uses pytest for unit testing. Testing of Multi-collateral Dai is performed on a Dockerized local testchain included in tests\config .

In order to be able to run tests, please install development dependencies first by executing:

```

Copy pip3 install -r requirements-dev.txt

```

You can then run all tests with:

```

Copy ./test.sh

```

Roadmap

- Dynamic gas pricing strategy
-

License

See COPYING file.

Support

If you have questions regarding Cage Keepers, please reach out to us on the #keeper channel on chat.makerdao.com .

Previous Simple Arbitrage Keeper Next Seth Last updated 4 years ago On this page * Introduction * Architecture * Operation * Testing * Roadmap * License * Support

Export as PDF