caso Documentation

Spanish National Research Council (CSIC)

Feb 05, 2021

Contents

1 cASO Release Notes 3

2 Installation 5

3 Configuration 7

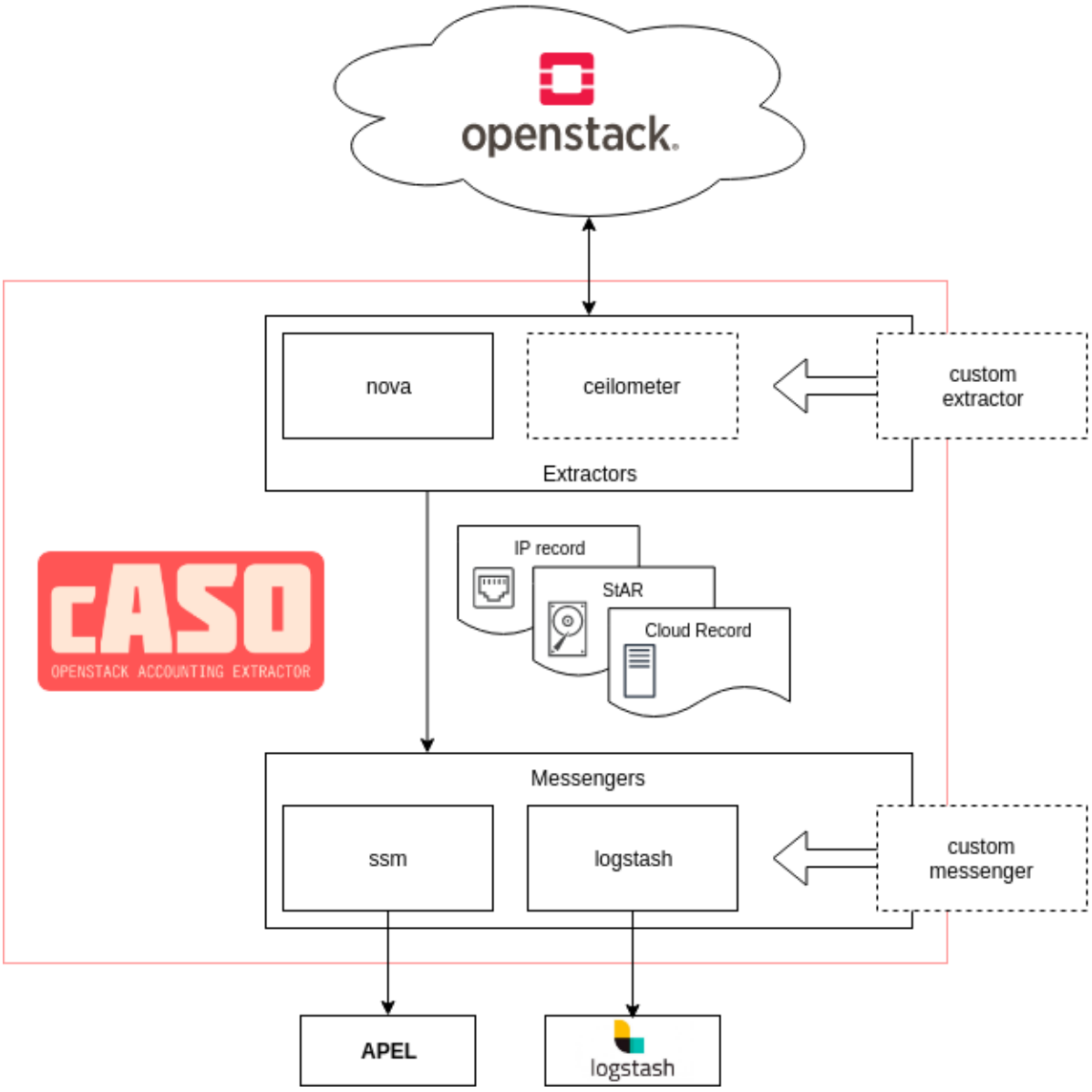
4 cASO multi-region support 17

5 Usage 19

6 Troubleshooting 21

Index 23

cASO is an accounting reporter (currently supports Cloud Accounting Usage Records) for OpenStack deployments. cASO gets usage information from OpenStack public APIs (no access to DB is required) and can generate valid output for Apel SSM or logstash.



Contents:

CHAPTER 1

cASO Release Notes

1.1 Current Release Notes

1.1.1 2.1.0

Prelude

This version includes a refactoring of the base extractors, dropping support for the ceilometer extractor that was unmaintained for a long period of time.

Upgrade Notes

- Ceilometer extractor is no longer supported.

1.1.2 2.0.0

Prelude

Starting with this version cASO release notes are published within the documentation. This version is a major release that implements IP accounting record, as well as several bugfixes. There are no upgrade notes to take into account.

New Features

- Add multi-region support in order to extract information from several regions through different configuration files.
- New IP accounting record is implemented. Now cASO is able to extract IP accounting and publish it using its JSON rendering. No new configuration needs to be done, but the cASO user needs to have access to the Neutron endpoints.

- cASO now allows to specity the projects to extract records from as project IDs, rather than names. When dealing with different identity domains this is troublesome, therefore we need to allow users to specify project IDs rather than names.

Bug Fixes

- Define the correct entrypoints for the V2 and V4 messengers.
- Generate LOG warnings when mappings cannot be found.

CHAPTER 2

Installation

2.1 Pre-requisites

If you are planning to use cASO for generating accounting records for EGI, you will need a valid APEL/SSM configuration. Follow the documentation available at the EGI.eu Federated Cloud documentation in order to set it up.

2.2 Installation

The best way to install cASO and have the most up to date version is using the repositories and packages provided in the EGI AppDB:

https://appdb.egi.eu/store/software/caso

2.2.1 Manual installation

Even the reccomended method is to use a package from the EGI AppDB, it is possible to install it from the Python Pacakge Index as follows:

```
$ pip install caso
```

Or you can install it on a virtualenv:

```
$ virtualenv --python python3 caso
$ source caso/bin/activate
(caso) $ pip install caso
```

CHAPTER 3

Configuration

3.1 OpenStack Configuration

Apart from configuring cASO, several actions need to be performed in your OpenStack installation in order to be able to extract accounting records.

3.1.1 User credentials (required)

In the next section you will configure an OpenStack Keystone credentials in order to extract the records. The cASO user has to be a member of each of the projects (another option is to convert that user in an administrator, but the former option is a safer approach) for which it is extracting the accounting. Otherwise, `cASO` will not be able to get the usages and will fail.

In order to do so, we are going to setup a new role `accounting` a new user `accounting`, adding it to each of the projects with that role:

```
openstack role create accounting
openstack user create --password <password> accounting
# For each of the projects, add the user with the accounting role
openstack role add --user accounting --project <project> accounting
```

Moreover, this user needs access to Keystone so as to extract the users information. In this case, we can can grant the user just the rights for listing the users adding the appropriate rules in your `/etc/keystone/policy.json` as follows. Replace the line:

```
"identity:list_users": "rule:admin_required",
```

with:

```
"identity:list_users": "rule:admin_required or role:accounting",
```

Recent Keystone versions leverage a `/etc/keystone/policy-yaml` file, if this is your case, substitute the line:

```
"identity:list_users": "rule:admin_required"
```

with:

```
"identity:list_users": "rule:admin_required or role:accounting"
```

3.1.2 Publishing benchmark information for OpenStack flavors (optional)

Starting with the V0.4 of the accounting record it is possible to publish benchmark information. In order to do so, you need to add this information to the flavor properties and configure caso to retrieve this information. There are two different values that need to be added to the flavor:

- The benchmark name, indicated with the `accounting:benchmark_name` flavor property.
- The benchmark value, indicated with the `accounting:benchmark_value` flavor property.

For example, if you are using HEPSPEC06 and the benchmark value is `99` for the flavor `m1.foo`, the benchmark information is configured as follows:

```
openstack flavor set --property accounting:benchmark_name="HEPSPEC06" --property␣
↪accounting:benchmark_value=99 m1.foo
```

Using different keys to specify bechmark information

If you do not want to use cASO's default flavor properties `accounting:benchmark_name` and `accounting:benchmark_value` (for example because you are using different benchmark types and values) you can specify which properties `cASO` should look for by using the `benchmark_name_key` `benchkark_value_key` in the configuration file.

Important: Please note that there is an OpenStack scheduler filter that removes hosts based on flavor properties. In order to not interfere with the behaviour of this filter you must prefix the property with a `scope:` so that cASO's properties are not taken into account for this filtering. When adding these properties in cASO's configuration file, please include the complete name (i.e. `scope:property`).

3.2 cASO configuration

`cASO` uses a config file (default at `/etc/caso/caso.conf`) with several sections. A sample file is available at `etc/caso/caso.conf.sample`.

3.2.1 `[DEFAULT]` section

The `[DEFAULT]` section configures the basic behavior of `cASO`. The sample config file (`/etc/caso/caso.conf.sample`) includes a description of every option. You should check at least the following options:

- `extractor` (default value: `nova`), specifies which extractor to use for getting the data. The following APIs are supported: `ceilometer` and `nova`. Both should generate equivalent information.
- `site_name` (default value: `<None>`). Name of the site as defined in GOCDB.
- `service_name` (default value: `$site_name`). Name of the service within a site. This is used if you have several endpoints within your site.

- `projects` (list value, default empty). List of the projects to extract records from. You can use either the project ID or the project name. We recommend that you use the project ID, especially if you are using domain-based authentication, as otherwise gathering the information might fail.
- `messengers` (list, default: `noop`). List of the messengers to publish data to. Records will be pushed to all these messengers, in order. Valid messengers shipped with cASO are:
  - `ssm` for publishing APEL V0.2 records (deprecated).
  - `ssmv2` for publishing APEL V0.2 records (deprecated).
  - `ssmv4` for publishing APEL V0.3 records (current).
  - `logstash` for publishing to Logstash.
  - `noop` do nothing at all.

  Note that there might be other messengers available in the system if they are registered into the `caso.messenger` entry point namespace.
- `mapping_file` (default: `/etc/caso/voms.json`). File containing the mapping from VOs to local projects as configured in Keystone-VOMS, in the following form:

```
{
    "VO": {
        "projects": ["foo", "bar"],
    }
}
```

  Note that you have to use either the project ID or project name for the mapping, as configured in the `projects` configuration variable.
- `benchmark_name_key` and `benchmark_value_key`. These two configuration options are used by `cASO` to retrieve the benchmark information form the OpenStack flavors.

3.2.2 `[keystone_auth]` section

This section is used to specify the authentication credentials to be used to connect to the OpenStack APIs. cASO leverages the OpenStack keystoneauth library for authentication, so that it is possible to use any authentication plugin that is available there (so starting on version 1.0 of cASO it is possible to use the Keystone V3 API).

Important: You need to specify the `auth_type` that you want to use (normally `v3password` is a good choice.

For an exhaustive list of available plugins please refer to the keystoneauth documentation.

3.2.3 `[ssm]` section

Options defined here configure the SSM messenger. There is only one option at the moment:

- `output_path` (default: `/var/spool/apel/outgoing/openstack`), directory to put the generated SSM records. APEL/SSM should be configured to take records from that directory.

3.2.4 `[logstash]` section

Options defined here configure the logstash messenger. Available options:

- `host` (default: `localhost`), host of Logstash server.
- `port` (default: `5000`), Logstash server port.

3.2.5 Other cASO configuration options

For an exhaustive list of the defined options, please check the following page:

cASO configuration file

oslo.log: DEFAULT

`debug`

Type boolean

Default `False`

Mutable This option can be changed without restarting.

If set to true, the logging level will be set to DEBUG instead of the default INFO level.

`log_config_append`

Type string

Default `<None>`

Mutable This option can be changed without restarting.

The name of a logging configuration file. This file is appended to any existing logging configuration files. For details about logging configuration files, see the Python logging module documentation. Note that when logging configuration files are used then all logging configuration is set in the configuration file and other logging configuration options are ignored (for example, log-date-format).

Table 1: Deprecated Variations

Group	Name
DEFAULT	log-config
DEFAULT	log_config

`log_date_format`

Type string

Default `%Y-%m-%d %H:%M:%S`

Defines the format string for %(asctime)s in log records. Default: the value above . This option is ignored if log_config_append is set.

`log_file`

Type string

Default `<None>`

(Optional) Name of log file to send logging output to. If no default is set, logging will go to stderr as defined by use_stderr. This option is ignored if log_config_append is set.

Table 2: Deprecated Variations

Group	Name
DEFAULT	logfile

log_dir

Type string

Default <None>

(Optional) The base directory used for relative log_file paths. This option is ignored if log_config_append is set.

Table 3: Deprecated Variations

Group	Name
DEFAULT	logdir

watch_log_file

Type boolean

Default False

Uses logging handler designed to watch file system. When log file is moved or removed this handler will open a new log file with specified path instantaneously. It makes sense only if log_file option is specified and Linux platform is used. This option is ignored if log_config_append is set.

use_syslog

Type boolean

Default False

Use syslog for logging. Existing syslog format is DEPRECATED and will be changed later to honor RFC5424. This option is ignored if log_config_append is set.

use_journal

Type boolean

Default False

Enable journald for logging. If running in a systemd environment you may wish to enable journal support. Doing so will use the journal native protocol which includes structured metadata in addition to log messages.This option is ignored if log_config_append is set.

syslog_log_facility

Type string

Default LOG_USER

Syslog facility to receive log lines. This option is ignored if log_config_append is set.

use_json

Type boolean

Default False

Use JSON formatting for logging. This option is ignored if log_config_append is set.

use_stderr

Type boolean

Default `False`

Log output to standard error. This option is ignored if log_config_append is set.

use_eventlog

Type boolean

Default `False`

Log output to Windows Event Log.

log_rotate_interval

Type integer

Default `1`

The amount of time before the log files are rotated. This option is ignored unless log_rotation_type is setto "interval".

log_rotate_interval_type

Type string

Default `days`

Valid Values Seconds, Minutes, Hours, Days, Weekday, Midnight

Rotation interval type. The time of the last file change (or the time when the service was started) is used when scheduling the next rotation.

max_logfile_count

Type integer

Default `30`

Maximum number of rotated log files.

max_logfile_size_mb

Type integer

Default `200`

Log file maximum size in MB. This option is ignored if "log_rotation_type" is not set to "size".

log_rotation_type

Type string

Default `none`

Valid Values interval, size, none

Log rotation type.

Possible values

interval Rotate logs at predefined time intervals.

size Rotate logs once they reach a predefined size.

none Do not rotate log files.

logging_context_format_string

Type string

Default `%(asctime)s.%(msecs)03d %(process)d %(levelname)s %(name)s [%(request_id)s %(user_identity)s] %(instance)s%(message)s`

Format string to use for log messages with context. Used by oslo_log.formatters.ContextFormatter

logging_default_format_string

Type string

Default `%(asctime)s.%(msecs)03d %(process)d %(levelname)s %(name)s [-] %(instance)s%(message)s`

Format string to use for log messages when context is undefined. Used by oslo_log.formatters.ContextFormatter

logging_debug_format_suffix

Type string

Default `%(funcName)s %(pathname)s:%(lineno)d`

Additional data to append to log message when logging level for the message is DEBUG. Used by oslo_log.formatters.ContextFormatter

logging_exception_prefix

Type string

Default `%(asctime)s.%(msecs)03d %(process)d ERROR %(name)s %(instance)s`

Prefix each line of exception output with this format. Used by oslo_log.formatters.ContextFormatter

logging_user_identity_format

Type string

Default `%(user)s %(tenant)s %(domain)s %(user_domain)s %(project_domain)s`

Defines the format string for %(user_identity)s that is used in logging_context_format_string. Used by oslo_log.formatters.ContextFormatter

default_log_levels

Type list

Default `['amqp=WARN', 'amqplib=WARN', 'boto=WARN', 'qpid=WARN', 'sqlalchemy=WARN', 'suds=INFO', 'oslo.messaging=INFO', 'oslo_messaging=INFO', 'iso8601=WARN', 'requests.packages.urllib3.connectionpool=WARN', 'urllib3.connectionpool=WARN', 'websocket=WARN', 'requests.packages.urllib3.util.retry=WARN', 'urllib3.util.retry=WARN', 'keystonemiddleware=WARN', 'routes.middleware=WARN', 'stevedore=WARN', 'taskflow=WARN', 'keystoneauth=WARN', 'oslo.cache=INFO', 'oslo_policy=INFO', 'dogpile.core.dogpile=INFO']`

List of package logging levels in logger=LEVEL pairs. This option is ignored if log_config_append is set.

publish_errors

Type boolean

Default `False`

Enables or disables publication of error events.

instance_format

Type string

Default `"[instance: %(uuid)s] "`

The format for an instance that is passed with the log message.

instance_uuid_format

Type string

Default `"[instance: %(uuid)s] "`

The format for an instance UUID that is passed with the log message.

rate_limit_interval

Type integer

Default `0`

Interval, number of seconds, of log rate limiting.

rate_limit_burst

Type integer

Default `0`

Maximum number of logged messages per rate_limit_interval.

rate_limit_except_level

Type string

Default `CRITICAL`

Log level name used by rate limiting: CRITICAL, ERROR, INFO, WARNING, DEBUG or empty string. Logs with level greater or equal to rate_limit_except_level are not filtered. An empty string means that all levels are filtered.

fatal_deprecations

Type boolean

Default `False`

Enables or disables fatal status of deprecations.

oslo.config: DEFAULT

config_file

Type list of filenames

Default `['~/.project/project.conf', '~/project.conf', '/etc/project/project.conf', '/etc/project.conf']`

Path to a config file to use. Multiple config files can be specified, with values in later files taking precedence. Defaults to the value above. This option must be set from the command-line.

config_dir

Type list of directory names

Default `['~/.project/project.conf.d/', '~/project.conf.d/', '/etc/project/project.conf.d/', '/etc/project.conf.d/']`

Path to a config directory to pull **.conf* files from. This file set is sorted, so as to provide a predictable parse order if individual options are over-ridden. The set is parsed after the file(s) specified via previous –config-file, arguments hence over-ridden options in the directory take precedence. This option must be set from the command-line.

config_source

Type list

Default `[]`

Lists configuration groups that provide more details for accessing configuration settings from locations other than local files.

driver

Type string

Default `remote_file`

This option has a sample default set, which means that its actual default value may vary from the one documented above.

The name of the driver that can load this configuration source.

uri

Type URI

Default `https://example.com/my-configuration.ini`

This option has a sample default set, which means that its actual default value may vary from the one documented above.

Required option with the URI of the extra configuration file's location.

ca_path

Type string

Default `/etc/ca-certificates`

This option has a sample default set, which means that its actual default value may vary from the one documented above.

The path to a CA_BUNDLE file or directory with certificates of trusted CAs.

client_cert

Type string

Default `/etc/ca-certificates/service-client-keystore`

This option has a sample default set, which means that its actual default value may vary from the one documented above.

Client side certificate, as a single file path containing either the certificate only or the private key and the certificate.

client_key

Type string

Default `<None>`

Client side private key, in case client_cert is specified but does not includes the private key.

CHAPTER 4

cASO multi-region support

- In case the monitored projects rely on a specific region, define the following variable in the /etc/caso/caso.conf

```
[DEFAULT]
region_name = <REGION>
```

- In case the monitored Project(s) rely on different regions, prepare different files /etc/caso/caso-<REGION>.conf

```
[DEFAULT]
region_name = <REGION>
```

- List the Project(s) in the /etc/caso/voms.json as from the documentation

```
{
  "Project1": {
    "projects": ["Project1"]
  },
  "Project2": {
    "projects": ["Project2"]
  }
}
```

- Execute caso-extract for each Project (and related REGION) to be monitored (Project1-REGION1, Project2-REGION2)

```
/usr/bin/caso-extract --projects "Project1" --config-file /etc/caso/caso-<REGION1>.
↪conf
/usr/bin/caso-extract --projects "Project2" --config-file /etc/caso/caso-<REGION2>.
↪conf
```

CHAPTER 5

Usage

5.1 command line

cASO provides the `caso-extract` command to generate new records from your OpenStack deployment. `caso-extract -h` will show a complete list of available arguments.

Use the `--extract-from` argument to specify the date from when the records should be extracted. If no value is set, then cASO will extract the records from the last run. If equal to "None", then extract records from the beggining of time. If not time zone is specified, UTC will be used.

Important: If you are running an OpenStack Nova version lower than Kilo there is a bug in its API, making impossible to paginate over deleted results.

Since nova is limiting the results to 1000 by default, if you are expecting more than 1000 results you will get just the last 1000. This is important if you are publishing data for the first time, or if you are republishing all your accounting). If this is your case, adjust the `osapi_max_limit` to a larger value in `/etc/nova/nova.conf`.

5.1.1 Available options

Apart from other options, the following ones are the ones that specify how to extract accountig records:

`--config-dir` `DIR`
: Path to a config directory to pull **.conf* files from. This file set is sorted, so as to provide a predictable parse order if individual options are over-ridden. The set is parsed after the file(s) specified via previous –config-file, arguments hence over-ridden options in the directory take precedence. This option must be set from the command-line.

`--config-file` `PATH`
: Path to a config file to use. Multiple config files can be specified, with values in later files taking precedence. Defaults to None. This option must be set from the command-line.

`--debug, -d`

If set to true, the logging level will be set to DEBUG instead of the default INFO level.

--dry-run, --dry_run
Extract records but do not push records to SSM. This will not update the last run date.

--extract-from EXTRACT_FROM, **--extract_from** EXTRACT_FROM
Extract records that have changed after this date. This means that if a record has started before this date, and it has changed after this date (i.e. it is still running or it has ended) it will be reported. If it is not set, extract records from last run. If it is set to None and last run file is not present, it will extract records from the beginning of time. If no time zone is specified, UTC will be used.

--extract-to EXTRACT_TO, **--extract_to** EXTRACT_TO
Extract record changes until this date. If it is not set, we use now. If a server has ended after this date, it will be included, but the consuption reported will end on this date. If no time zone is specified, UTC will be used.

--extractor EXTRACTOR
Which extractor to use for getting the data. If you do not specify anything, nova will be used. Allowed values: nova

--projects PROJECTS, **--tenants** PROJECTS
List of projects to extract accounting records from.

5.2 Running as a cron job

The best way of running cASO is via a cron job like the following:

```
10 * * * * caso-extract
```

CHAPTER 6

Troubleshooting

6.1 Cannot-find-VM-in-API

Danger: There is not a single recipe to fix this issue, and this involves touching and modifying the DB directly. We reccomend that you ignore these messages, unless you know what you are doing.

In the logs you can see the following warnings (caso version < 1.4.4):

```
WARNING caso.extract.nova [-] Cannot get server '072e77c0-4295-4a83-9bdf-6afde796a00d
↪' from the Nova API, probably because it is an old VM that whose metadata is wrong␣
↪in the DB. There will be no record generated for this VM. : NotFound: Instance␣
↪072e77c0-4295-4a83-9bdf-6afde796a00d could not be found. (HTTP 404) (Request-ID:␣
↪req-8eabf5d8-b722-4ee4-b211-aec36fc0499e)
```

Or the following one (caso version >= 1.4.4):

```
WARNING caso.extract.nova [-] Cannot get server '072e77c0-4295-4a83-9bdf-6afde796a00d
↪' from the Nova API, probably because it is an error in the DB. Please refer to the␣
↪following page for more details: https://caso.readthedocs.io/en/stable/
↪troubleshooting.html#Cannot-find-VM-in-API
```

These errors are caused by a VM that is in a bad state on the DB. The `os-simple-tenant-usage` API is returning instances that cannot be obtained from the API.

This may be caused by any of the following cases:

1. VMs that have changed their status on a date that enters into the extrating period.
2. VMs that are terminated and deleted, but their status is incorrect (i.e. no value for `terminated_at`). This can be fixed by setting a `terminated_at` value that is correct, directly in the DB.

Index

Symbols

-config-dir DIR
    command line option, 19
-config-file PATH
    command line option, 19
-debug, -d
    command line option, 19
-dry-run, -dry_run
    command line option, 20
-extract-from EXTRACT_FROM,
       -extract_from EXTRACT_FROM
    command line option, 20
-extract-to EXTRACT_TO, -extract_to
       EXTRACT_TO
    command line option, 20
-extractor EXTRACTOR
    command line option, 20
-projects PROJECTS, -tenants PROJECTS
    command line option, 20

C

command line option
    -config-dir DIR, 19
    -config-file PATH, 19
    -debug, -d, 19
    -dry-run, -dry_run, 20
    -extract-from EXTRACT_FROM,
       -extract_from EXTRACT_FROM, 20
    -extract-to EXTRACT_TO,
       -extract_to EXTRACT_TO, 20
    -extractor EXTRACTOR, 20
    -projects PROJECTS, -tenants
       PROJECTS, 20