

CLUSTERPRO

MC ProcessSaver 2.4 for Linux

ユーザーズガイド

(コンテナ監視機能)

© 2019(Apr) NEC Corporation

- 本機能の概要について
- インストールおよびアンインストールについて
- 操作・設定について
- メッセージ
- 障害解析情報の採取
- 注意・制限事項
- リファレンス

改版履歴

版数	改版	内容
1.0	2018.04	新規作成
2.0	2018.06	商標の記載の修正
3.0	2019.04	MC 2.4 に対応

はしがき

本書は、CLUSTERPRO MC ProcessSaver 2.4 for Linux (以後 ProcessSaver と記載します) のコンテナ監視機能について記載したものです。

- (1) 本書は以下のオペレーティングシステムに対応します。
サポート対象ハードウェアは、x86_64 搭載マシンです。
 - ・ Red Hat Enterprise Linux 7.0~7.6
 - ・ Oracle Linux 7.0~7.6

- (2) 概要
本機能は、Docker の実行環境がインストールされたシステムにおいて動作します。
コンテナリストを元に、指定したコンテナの起動状況および指定したイメージで起動するコンテナ数の監視を行います。
想定外のコンテナの起動・停止に起因する業務への影響に、迅速に対応が可能になります。

- (3) 本リリースの強化点について
ProcessSaver 2.4 (2019.4 月出荷版)では、下記の機能を強化しています。
 - ・ 対応 OS の拡大
新たに対応する OS を追加しました。

- (4) 商標および登録商標
 - ✓ Linux は、米国およびその他の国における Linus Torvalds の登録商標です。
 - ✓ Red Hat、Red Hat Enterprise Linux は、米国およびその他の国における Red Hat, Inc.およびその子会社の商標または登録商標です。
 - ✓ Oracle は、Oracle Corporation 及びその子会社、関連会社の米国及びその他の国における登録商標です。文中の社名、商品名等は各社の商標または登録商標である場合があります。
 - ✓ Docker は、米国およびその他の国における Docker, Inc.の登録商標または商標です。
 - ✓ CLUSTERPRO、ProcessSaver は、日本電気株式会社の登録商標です。
 - ✓ その他記載の製品名および会社名は、すべて各社の商標または登録商標です。
 - ✓ なお、本書では®、TM マークを明記しておりません。

目次

1	本機能の概要について	1
1.1	本機能の提供する主な機能について	1
1.2	本機能を導入する前に	2
2	インストールおよびアンインストールについて	3
2.1	インストール手順	3
2.2	アンインストール手順	4
3	操作・設定について	5
3.1	コンテナ監視	5
3.1.1	導入手順	5
3.1.2	cfile について	7
3.1.3	再起動スクリプトについて	13
3.1.4	起動、終了ファイルの導入について	16
3.2	操作・運用手順	17
3.2.1	コンテナの状態監視について	17
3.2.2	コンテナの運用管理について	22
3.2.3	異常検知時の確認手順	25
3.2.4	デバッグ支援機能	26
3.3	高度な運用手順について	27
3.3.1	グループ監視の導入手順	27
3.3.2	イメージ名監視 (コンテナ起動数監視) の導入手順	29
3.3.3	監視間隔を短くした運用	30
3.3.4	サイレントモードでの運用手順	31
3.3.5	ccheck 終了時に子プロセスを回収する手順	32
3.3.6	ccheck 起動時の自動待ち合わせ時間を変更する手順	33
3.3.7	ccheck 起動時のサマリー情報を syslog 出力する手順	35
3.3.8	監視対象選択時のコンテナ情報取得量を変更する手順	37
3.3.9	監視対象選択時のリトライ回数を変更する手順	39
3.3.10	コンテナ情報を収集するコマンドの最大待ち時間を変更する手順	40
4	メッセージ	41
4.1	syslog メッセージ	41
4.1.1	フォーマットについて	41
4.1.2	監視対象となる syslog メッセージ	41
4.1.3	運用メッセージ	42
4.1.4	cadmin コマンドのメッセージ	44
4.1.5	ccheck コマンドのメッセージ	46
4.1.6	停止監視の異常メッセージ	48
4.1.7	cfile 関連のメッセージ	55
5	障害解析情報の採取	59
5.1	障害解析情報	59
6	注意・制限事項	61

6.1	cadmin	61
6.2	ccheck.....	62
6.3	システム構成における注意事項.....	64
7	リファレンス	65
7.1	cadmin	65
7.2	ccheck.....	71

1 本機能の概要について

1.1 本機能の提供する主な機能について

本機能は、コンテナの状態監視、障害時の自動再開を行うことでシステムの高可用性を実現します。

- ・ **コンテナの起動監視と再開機能**
システムで稼働しているコンテナの起動状態を監視し、監視中のコンテナが停止した場合や起動コンテナ数の異常を検出した場合に、コンテナの自動再開を行うフレームワークを提供します。
- ・ **グループ監視機能**
特定の依存関係をもつコンテナ群をグループとして定義することで、グループ単位でのコンテナの監視、再開を行うことができます。
起動、終了時に依存関係を持つコンテナを監視する場合に有効です。
- ・ **起動コンテナ数監視機能**
指定したイメージで起動するコンテナの数を監視し、想定外のコンテナの起動・停止を検出した場合に、コンテナの自動再開を行うフレームワークを提供します。
- ・ **管理コマンドによるメンテナンス**
監視対象ごとの状態表示や動的な監視停止、再開をコマンドインターフェースで操作可能です。

1.2 本機能を導入する前に

(1) 機能の構成について

下記のコマンドにより構成されます。

- ・ccheck コンテナ監視コマンド
- ・cadmin 運用管理コマンド

下記のディレクトリを使用します。

- ・/opt/HA/CCHECK 実行形式ファイルなどの格納ディレクトリ
- ・/var/opt/HA/CCHECK 設定ファイル、トレースファイルなどの格納ディレクトリ

(注) 上記の実行環境ディレクトリはデフォルトであり、運用環境に合わせて任意のディレクトリを使用することができます。

(2) マニュアルの参照について

/var/opt/HA/CCHECK/doc/man1 配下にテキストファイルがありますので、そちらを参照してください。

(注) 上記のテキストファイルが文字化けする場合は、端末の画面上で日本語環境を設定してください。

- ・ sh の場合
 - # LANG=ja_JP.utf8
 - # export LANG

2 インストールおよびアンインストールについて

2.1 インストール手順

(1) 本機能を含む CD-R 媒体を CD-ROM(DVD-ROM) ドライブに挿入します。

(2) mount(8) コマンドを使用して、CD-R 媒体をマウントします。

※ /dev/cdrom は CD-ROM(DVD-ROM) ドライブのデバイスファイル名です。

```
# mount /dev/cdrom /media
```

(3) rpm(8) コマンドを使用して、本機能のパッケージをインストールします。

※ w,x,y-z には、バージョン番号が入ります。

```
# rpm -ih /media/Util/ccheck/Linux/rpm/clusterpro-mc-pscm-w.x.y-z.x86_64.rpm
```

(4) rpm(8) コマンドを使用して、本機能のパッケージが正しくインストールされたことを確認します。

※ w,x,y-z には、バージョン番号が入ります。

※ ***** 部分は OS のバージョンおよびインストールしたパッケージの名称に依存します。

```
# rpm -qa | grep clusterpro-mc-pscm  
clusterpro-mc-pscm-w.x.y-z.*****
```

(5) マウントした媒体を umount(8) コマンドを使用してアンマウントします。

```
# umount /media
```

(6) 媒体を CD-ROM(DVD-ROM) ドライブから取り出します。

以上でコンテナ監視機能のインストールは終了です。

2.2 アンインストール手順

- (1) rpm(8) コマンドを使用して、本機能がインストールされていることを確認します。
 - ※ w,x,y-z には、バージョン番号が入ります。
 - ※ ***** 部分は OS のバージョンおよびインストールしたパッケージの名称に依存します。

```
# rpm -qa | grep clusterpro-mc-pscm  
clusterpro-mc-pscm-w.x.y-z.*****
```

- (2) rpm(8) コマンドを実行して、アンインストールを行います。
 - ※ w,x,y-z には、バージョン番号が入ります。

```
# rpm -e clusterpro-mc-pscm-w.x.y-z
```

- (3) rpm(8) コマンドを使用して、本機能が正しくアンインストールされたことを確認します。

```
# rpm -qa | grep clusterpro-mc-pscm
```

正常にアンインストールされていれば、何も表示されません。

以上でコンテナ監視機能のアンインストールは終了です。

3 操作・設定について

3.1 コンテナ監視

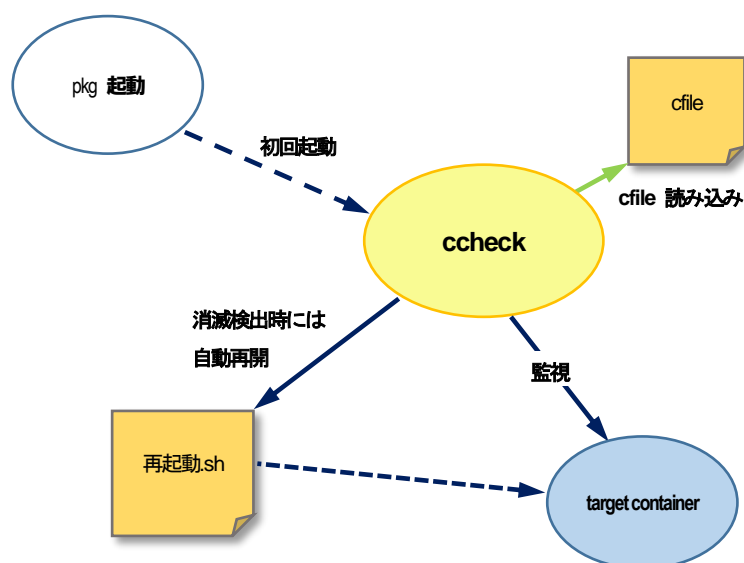
3.1.1 導入手順

導入手順は以下となります。

- (1) インストール
本機能をインストールします。
インストール方法については、「2.1 インストール手順」を参照してください。
- (2) cfile (コンテナ監視定義ファイル) の設定
cfile には以下の設定が必要です。
 - 全体の監視ルールを定義する共通部情報の設定 (必須パラメーター)
 - コンテナ/イメージ単位の停止監視を定義する個別部情報の設定 (必須パラメーター)
 - コンテナ/イメージ単位の任意の監視ルールを定義するブロック情報の設定 (オプションパラメーター)cfile の詳細については、「3.1.2 cfile について」を参照してください。
- (3) 再起動スクリプトの作成
コンテナ停止時の再起動手順を再起動スクリプトに記述します。
再起動スクリプトは通常のシェルスクリプトです。
再起動スクリプトの詳細については、「3.1.3 再起動スクリプトについて」を参照してください。

これらのファイルは、テンプレートディレクトリにサンプルがありますので、実行環境にコピーしてからカスタマイズしてください。

なお、ひとつの cfile で複数のコンテナ/イメージを監視できますが、再起動スクリプトは対象コンテナ/イメージごとに定義してください。



(4) 起動

ccheck (コンテナ監視コマンド) を起動します。

ccheck の起動、終了手順は運用環境によって、選択することができます。

① systemd から起動、終了

systemd に ccheck コマンドを登録することで、OS の起動、終了と連動して ccheck の起動・終了を行うことができます。

導入手順の詳細については「3.1.4 起動、終了ファイルの導入について」を参照してください。

② コマンドラインからの起動、終了

コマンドラインからの起動、終了手順は以下のとおりです。

- ccheck 起動

```
#!/opt/HA/CCHECK/bin/ccheck -f <cfile> &
```

- ccheck 終了

```
#!/opt/HA/CCHECK/bin/cadmin -f <cfile> -c shutdown
```

(注) 上記コマンドの cfile 名は絶対パスを指定してください。

以上でコンテナ監視の導入手順は終了です。

3.1.2 cfile について

(1) cfile の構成について

cfile の構成は ccheck 全体の動作を定義する共通部情報と、コンテナ/イメージ単位の管理情報を定義する個別部情報で構成されます。

cfile は ccheck 単位の定義するファイルです。

複数の ccheck を起動する場合は ccheck ごとに cfile を用意してください。

本機能をインストールすると、自動的にディレクトリが作成され、サンプルは /var/opt/HA/CCHECK/conf/src 配下に置かれます。

```
# cd /var/opt/HA/CCHECK/conf/src/SG_sample
```

サンプルの cfile の名前を変更してコピーします。

```
# cp <cfile> /var/opt/HA/CCHECK/conf/bin/cfile_copy
```

※ 使用する cfile はどこかのディレクトリ配下に置いても使用できますが、通常は /var/opt/HA/CCHECK/conf/bin 配下に置いて使用します。

コピーした cfile を vi で開き、以下の値を変更してください。

cfile </var/opt/HA/CCHECK/conf/bin/cfile_copy >

```
##### PARAM #####
IPCKEY_CCHECK          0x1f000001 →①
MSG_CHECK_INTERVAL    5
MONITOR_INTERVAL      10
SHM_DUMP_FILE          /var/opt/HA/CCHECK/log/ccheck_dump_copy →②

##### CENT #####
container_name:shell_path:grace_time:retry_num:retry_over_action
  ↓           ↓           ↓           ↓           ↓
  ③           ④           ⑤           ⑥           ⑦
```

① IPCKEY_CCHECK (共有メモリのkey) = システム上で一意の値に設定します。

② SHM_DUMP_FILE (ダンプファイル名) = システム上で一意の値に設定します。

※ ①②については、「(2) 共通部情報 (PARAM) の設定」で詳しく説明します。

③ 監視対象のコンテナ/イメージ名を指定します。

④ ③の停止を検出した場合に実行される再起動スクリプト名を指定します。

⑤ GRACE 値 (リトライ回数に達するまでの監視時間) を設定します。

⑥ リトライ回数 (GRACE 値内でリトライすることができる上限回数) を設定します。

⑦ リトライオーバーアクション (リトライ回数を超えた場合の処理) を指定します。

※ ③④⑤⑥⑦については、「(3) 個別部情報 (CENT) の設定」で詳しく説明します。

(2) 共通部情報 (PARAM) の設定

ccheck 全体の動作を規定する共通部情報を指定する部分です。フォーマットは以下のとおりです。

アイテム名	設定値
IPCKEY_CCHECK	<i>ipckey</i>
MSG_CHECK_INTERVAL	<i>msg_check_interval</i>
MONITOR_INTERVAL	<i>monitor_interval</i>
SHM_DUMP_FILE	<i>shm_dump_file</i>

おのこの設定値について以下に説明します。

ipckey

共有メモリのkey を指定します。

OS 内で一意となる 1 ~ 0x7ffffff の数値を指定してください。

本機能では、cfile 情報を共有メモリに展開して監視状態の管理などを行っています。

ccheck コマンドを複数起動する設定を行っている場合、***ipckey*** を OS 内で一意にするために ipcs(1) コマンドを実行し、key に cfile 指定の ***ipckey*** が存在しないことを確認してください。

また、OS や他製品が使用する共有メモリのkey と競合しないように比較的大きな値 (一般的には0x1f000000 以降) を指定することを推奨しています。

msg_check_interval

内部イベントをスケジュールするタイマー値を設定します。

指定値は 1 秒 ~ 60*60*24 秒 (24 時間) の範囲です。

monitor_interval より小さい値を設定してください。

デフォルト値は 5 秒です。デフォルト値を使用することを推奨します。

cadmind から実行されるコンテナ監視停止、再開といったメッセージを受け付ける間隔に使用されるため、cadmind コマンドからのオペレーションは、最大でこのタイマー値だけ遅延する可能性があります。

monitor_interval

コンテナ監視を行うタイマー値を設定します。

指定値は 1 秒 ~ 60*60*24 秒 (24 時間) の範囲です。

msg_check_interval より大きい値で、***msg_check_interval*** の正の整数倍の値を設定してください。

デフォルト値は 10 秒です。デフォルト値を使用することを推奨します。

shm_dump_file

コンテナ監視の内部トレースを出力するファイル名を絶対パスで設定します。

デフォルトのファイル名は、***/var/opt/HA/CCHECK/log/ccheck_dump*** です。

ファイル名は 256 文字未満で設定してください。

複数の ccheck を起動する場合、重複しないように設定する必要があります。

なお、トレースファイルは約 3MB 程度のサイクリックログとなり、ccheck の終了時、cadmind の reload コマンド実行時、リトライオーバーを契機として cfile の SHM_DUMP_FILE に指定した場所へ出力します。また、ファイルは 2 世代までバックアップされます。

トレースファイルは開発元で障害時などの情報解析に使用します。

通常運用中は利用者が特に意識する必要はありません。

- (3) 個別部情報 (CENT) の設定
監視対象単位の管理情報を指定する部分です。フォーマットは以下のとおりです。

container_name:shell_path:grace_time:retry_num:retry_over_action

または

container_name:shell_path:grace_time:retry_num:retry_over_action:option

または

image_name:shell_path:grace_time:retry_num:retry_over_action:option

ひとつの cfile において CENT は最大 256 まで記述できます。
おのおの設定値について以下に説明します。

container_name / image_name

監視対象名を設定します。

コンテナを監視するときは docker ps コマンドの実行で NAMES に出力される文字列を指定、
イメージを監視するときは docker ps コマンドの実行で IMAGE に出力される文字列を指定して
ください。

登録できる文字列の長さは引数を含めて 1024 文字までとなります。

この文字列を使用して監視対象のマッチング処理を行いますので、正確に設定してください。

shell_path

監視コンテナの再開時に実行するシェルスクリプト (再起動スクリプト) のファイル名を設定します。
ファイル名は絶対パスで 256 文字未満を指定してください。

ファイル名には引数付のシェルスクリプトを指定することが可能です。

また、複数のシェルスクリプトを列記する場合は、;(セミコロン) 区切りで指定してください。

; (セミコロン) 区切りでスクリプトを指定した場合は、/bin/sh -c の引数として指定されたスクリプト
を実行します。

そのため、/bin/sh -c の引数として指定することのできないスクリプトおよびコマンドについては
指定することができません。

コンテナの再開を実行しない場合、本パラメーターは省略可能です。

再起動スクリプトを指定しない場合は 0 (ゼロ) または - (ハイフン) を指定してください。

grace_time

システムが安定稼働しているとみなし、これまでのリトライ回数をリセットするまでの時間を設定
します。

指定値は 0 秒 ~ 31104000 秒 (60*60*24*360 秒 = 約 1 年) の範囲です。

一般的な例では、86400 秒 (60*60*24 秒 = 1 日) を指定します。

この値に設定した時間内に、最大リトライ回数に達した場合はリトライオーバーアクションの設定
にしたがって処理を行います。最大リトライ回数に達しなかった場合はこの値を超えた後のコ
ンテナ停止検出時を 1 回目のリトライとして処理を行います。

この際に grace_time のカウントはリセットされます。

また、この値のカウントは最初のコンテナ停止検出時から開始されます。

なお、再起動が不要な場合は、0 (ゼロ) を指定してください。

本パラメーターを省略した場合、0 (ゼロ) が設定されます。

retry_num

コンテナ停止検出時に行う最大リトライ回数を設定します。
指定値は 0 回～ 1024 回の範囲です。
一般的な例では、3 回を指定します。
なお、再起動が不要な場合は、0 (ゼロ) を指定してください。

再起動の無限リトライを指定する場合、**grace_time** に 0 (ゼロ) を指定し、**retry_num** に 0 以外の値 (たとえば 1) を指定してください。

本パラメーターを省略した場合、0 (ゼロ) が設定されます。

retry_over_action

最大リトライ回数を超えてコンテナ停止を検出した場合の動作を設定します。
指定値は、以下のいずれかを設定してください。

- | | |
|-----------------|---|
| continue | 最大リトライ回数に達したコンテナを監視対象からはずし、他のコンテナの監視は継続して行います。 |
| shutdown | 監視対象コンテナの再起動に失敗し、最大リトライ回数に達した場合に、ccheck を終了します。(終了コード 0)
たとえば ccheck に親子関係を持たせる場合に子の ccheck にはこのパラメーターを指定します。 |
| exit | 監視対象コンテナの再起動に失敗し、最大リトライ回数に達した場合に、ccheck を終了します。(終了コード -1)
CLUSTERPRO などのクラスターウェア製品と連携し、フェールオーバーを行う場合はこの値を設定してください。 |

<リトライオーバーアクションの shutdown と exit の違いについて>

本機能では終了コード以外の処理の違いや動作の差分はありません。

そのため ccheck の終了コードを意識しないクラスターウェア製品と連携する場合は、どちらを使用しても問題ありませんが、今後はクラスターウェア製品への対応のため exit の処理を強化する可能性がありますので、基本的にクラスターウェア製品と連携する場合は exit を指定してください。

option

監視条件を限定するためのキーを指定します。
本パラメーターは省略可能です。

オプション情報には、以下のパラメーターのいずれかを設定してください。
複数のオプションを指定する場合には、オプション情報を "," で区切ることで指定できます。

オプションは最大で 6 個まで同時に指定できます。
また、指定できる文字数はオプション情報全体で最大 255 文字以内です。

image_name_check=enable または
image_name_check=disable

enable を指定した場合は、監視対象名をイメージ名として解釈し、そのイメージで起動するコンテナ数の監視を行います。

disable を指定した場合や本オプションを指定しない場合は、監視対象名をコンテナ名として解釈し、そのコンテナの起動 / 停止の監視を行います。
(注) enable、disable 以外を指定することはできません。

min_count=xxx xxx には同一イメージの起動コンテナ数を監視する場合に最小限起動しておきたいコンテナ数を指定します。
指定値は 1 ~ 64 (個) の範囲です。
指定された数未満となった場合に最小起動数エラーを検出します。

max_count=xxx xxx には同一イメージの起動コンテナ数を監視する場合に最大限起動しておきたいコンテナ数を指定します。
指定値は 1 ~ 64 (個) の範囲です。
指定された数を超えた場合に最大起動数エラーを検出します。

grouptag=xxx xxx にはグループ監視を行う場合のグループ名を指定します。
グループ名は 63 文字までとし、英数字で指定します。
また数字のみの grouptag 名は指定することができません。
英字のみまたは英数字を組み合わせた方式で指定してください。
(例) 指定可: grouptag=oracle_group grouptag= group1
指定不可: grouptag=1 grouptag=8

コンテナの停止を検出した場合に、同一のグループ名が指定されている監視対象をすべて再起動します。

restart_timeout=xxx

xxx には再起動スクリプトの実行タイムアウトを設定します。
指定値は 1 秒 ~ 60*60*24 秒 (24 時間) の範囲です。
指定された時間を経過しても再起動スクリプトが終了しない場合に再起動スクリプトを強制終了します。再起動を取りやめて処理を継続させたい場合に有効です。

(注) **monitor_interval** の正の整数倍の値を設定してください。
正の整数倍でない値を指定した場合、この値は **monitor_interval** の設定値をもとに切り上げられる場合があります。

(4) 一般的な cfile のサンプル

コンテナを監視する cfile の事例

```
# ProcessSaver configuration file
##### PARAM #####
IPCKEY_CCHECK      0x1f000301
MSG_CHECK_INTERVAL 5
MONITOR_INTERVAL   10
SHM_DUMP_FILE      /var/opt/HA/CCHECK/log/cchk_1f000301_dump

##### CENT (Multiple containers) #####
happy_hamilton:/bin/cm/restart_happy_hamilton.sh:86400:3:continue
sick_colden:-:0:0:continue
small_mayer:-:0:0:continue
modest_newton:-:0:0:continue
berserk_roentgen:-:0:0:continue
determined_hamilton:-:0:0:continue
```

監視対象に option を指定する cfile の事例

```
# ProcessSaver configuration file
##### PARAM #####
IPCKEY_CCHECK      0x1f000302
MSG_CHECK_INTERVAL 5
MONITOR_INTERVAL   10
SHM_DUMP_FILE      /var/opt/HA/CCHECK/log/cchk_1f000302_dump

##### CENT (check the group containers) #####
stoic_almeida:/bin/cm/restart_grp_almeida.sh:86400:3:continue:groupstag=almeida
tiny_almeida:/bin/cm/restart_grp_almeida.sh:86400:3:continue:groupstag=almeida
elegant_brown:/bin/cm/restart_grp_brown.sh:86400:3:continue:groupstag=brown,restart_timeout=600
awesome_brown:/bin/cm/restart_grp_brown.sh:86400:3:continue:groupstag=brown,restart_timeout=600

##### CENT (check the number of starting containers) #####
rhel7-5:/bin/cm/restart_image_rhel7_5.sh:86400:3:continue:image_name_check=enable,min_count=2,max_count=5
```

3.1.3 再起動スクリプトについて

(1) 再起動スクリプトとは

ccheck がコンテナの停止を検出した場合に、コンテナを再起動するための手続きを記述した実行ファイルを再起動スクリプトと呼びます。

再起動スクリプトには、対象コンテナが停止した場合に必要な解放処理と、再起動のために必要な処理を記述してください。フォーマットは通常のシェルスクリプトです。

(2) 再起動スクリプト作成時の注意事項

再起動スクリプトは、対象コンテナの振る舞いに影響を受けますので、十分な注意のもとで作成してください。

- ・ 対象コンテナが起動中に使用していた資源を、再起動スクリプトで解放してからコンテナを再開するようにしてください。
- ・ 対象コンテナを起動した後に `/bin/sleep` などによって一定時間の猶予をとってください。対象コンテナの起動に時間がかかると、`ccheck` の監視が先行する可能性があります。
- ・ `ccheck` は、コンテナ再開時に再起動スクリプトの終了を待ち合わせてからコンテナの監視を再開します。再起動スクリプト内で長時間 `sleep` により待ち合わせを行った場合や、処理に時間がかかる場合、コンテナ監視が効果的に作用しない場合があります。
- ・ 再起動スクリプトで実行するコマンドは絶対パスで呼び出してください。コマンドへのパスが張られていないとコマンドの実行に失敗する場合があります。
- ・ 再起動スクリプトが異常終了のステータス (0 以外) を返却すると、`ccheck` はコンテナの再起動に失敗したとみなします。
- ・ 再起動スクリプトから起動するコンテナが環境変数に依存している場合は、その環境変数を設定してから起動してください。再起動に失敗した場合は、環境変数 `PATH` に `/usr`、`/usr/bin`、`/bin`などを追加することで回避できる場合があります。
- ・ 再起動スクリプトを ;(セミicolon) 区切りで複数指定した場合、`/bin/sh -c` の引数として指定されたスクリプトを実行します。そのため、`/bin/sh -c` の引数として指定することのできないスクリプトやコマンドについては指定することができません。また、記載された構文のまま実行されますので、シェルの構文に従った方式で指定する必要があります。
- ・ 再起動スクリプトを ;(セミicolon) 区切りで複数指定した場合には、最後に指定されたスクリプトの終了ステータスでコンテナの再起動に成功したかどうかを判定します。
- ・ 再起動スクリプトを ;(セミicolon) 区切りで複数指定した場合には、`restart_timeout` オプションは使用できません。

(3) 再起動スクリプトのサンプル

サンプルは /var/opt/HA/CCHECK/conf/src 配下に置かれます。

```
# cd /var/opt/HA/CCHECK/conf/src/SG_sample
```

サンプルのスクリプトの名前を変更してコピーします。

```
# cp sample_restart_carson_grp.sh /var/opt/HA/CCHECK/conf/bin/restart_container.sh
```

コピーした再起動ファイルを vi で開き、以下の値を編集してください。

再起動スクリプト < /var/opt/HA/CCHECK/conf/bin/restart_container.sh >

```
#!/bin/sh
docker stop container_name →①
/bin/sleep 5

docker start container_name →②
/bin/sleep 5

exit 0 →③
```

- ① まずコンテナの停止を行います。
また、念のため sleep で待ち合わせます。
- ② コンテナを起動します。停止同様、sleep で待ち合わせます。
- ③ exit 0 を記述します。
(ProcessSaver は再起動スクリプトの戻り値によって、正常に再起動できたかどうか判断をしていますので、再起動スクリプトは、必ず exit 0 で終了する必要があります)

ファイルを保存して編集を終了します。

chmod コマンドにより編集した再起動スクリプトに実行権をつけます。

```
# chmod +x /var/opt/HA/CCHECK/conf/bin/restart_container.sh
```

※ 実行権がない場合、再起動スクリプトが実行されないため、必ず実行権をつけます。

◆ 依存関係のある複数コンテナのサンプル

コンテナ `adoring_mayer` が停止した場合に、依存関係のある他のコンテナを強制停止させてから、再起動する記述例です。

サンプルのスクリプトを名前を変更してコピーします。

```
# cp sample_restart_img_group56.sh /var/opt/HA/CCHECK/conf/bin/ restart_depends_container.sh
```

コピーした再起動ファイルを `vi` で開き、以下の値を編集してください。

再起動スクリプト < `/var/opt/HA/CCHECK/conf/bin/restart_depends_container.sh` >

```
#!/bin/sh
for i in `docker ps --filter ancestor=rhel7 --format {{.Names}}` →①
do
  docker stop $i;
done

docker stop adoring_mayer →②
/bin/sleep 5

docker start adoring_mayer →③
/bin/sleep 5

for i in `docker ps -a --filter ancestor=rhel7 --filter status=exited --format {{.Names}}` →④
do
  docker start $i;
done

exit 0 →⑤
```

- ① 依存関係にあるコンテナをいったんすべて停止しておきます。
- ② 目的のコンテナの停止を行います。
また、念のため `sleep` で待ち合わせます。
- ③ 目的のコンテナを起動します。停止同様、`sleep` で待ち合わせます。
- ④ ①で停止しておいたコンテナを起動します。
- ⑤ `exit 0` を記述します。
(ProcessSaver は再起動スクリプトの戻り値によって、正常に再起動できたかどうか判断をしていますので、再起動スクリプトは、必ず `exit 0` で終了する必要があります)

ファイルを保存して編集を終了します。

`chmod` コマンドにより編集した再起動スクリプトに実行権をつけます。

```
# chmod +x /var/opt/HA/CCHECK/conf/bin/restart_depends_container.sh
```

※ 実行権がない場合、再起動スクリプトが実行されないため、必ず実行権をつけます。

3.1.4 起動、終了ファイルの導入について

起動、終了ファイルの導入手順について説明します。

Red Hat Enterprise Linux 7.x、Oracle Linux 7.x の場合

OS の起動、終了と同期をとって ccheck を起動、終了する場合は、systemd への設定が必要です。サンプルの Unit ファイルを必要に応じてカスタマイズし、systemd 管理ディレクトリにコピーします。ファイル名は任意の名前に変更可能です。

```
# cp /var/opt/HA/CCHECK/conf/src/systemd_sample/psaver.service /etc/systemd/system/psaver.service
```

次に、以下のコマンドを実行して systemd へ登録してください。

```
# systemctl enable psaver.service
```

登録後、以下のコマンドを実行して登録内容を反映し、ccheck を起動してください。

```
# systemctl daemon-reload  
# systemctl start psaver.service
```

また、上記を設定した場合、ccheck が監視するプロセスも systemd への設定が必要です。設定の際は、ccheck との依存関係を正しく設定してください。依存関係の設定方法の詳細については、systemctl のマニュアルを参照してください。

3.2 操作・運用手順

3.2.1 コンテナの状態監視について

コンテナの監視状態は `/opt/HA/CCHECK/bin/cadmin` コマンドで確認できます。

(1) コンテナ監視コマンドの一覧を表示する

```
# /opt/HA/CCHECK/bin/cadmin -l
ipckey      = 0x1f000301
mypid       = 19141
myname      = ccheck
cfile       = /var/opt/HA/CCHECK/conf/cchk812_1_3_1
message     = start

ipckey      = 0x1f000302
mypid       = 19142
myname      = ccheck
cfile       = /var/opt/HA/CCHECK/conf/cchk812_1_3_2
message     = stop
```

(2) コンテナの監視ルールを表示する

```
# /opt/HA/CCHECK/bin/cadmin -f <cfile> -c show param
MSG_CHECK_INTERVAL          = 5
MONITOR_INTERVAL            = 10
MONITOR_TRY_COUNT           = 2
SHM_DUMP_FILE                = /var/opt/HA/CCHECK/log/ccheck_dump_6_3_1
CFILE                        = ccheck_6_3_1
MESSAGE_BOX                  = start
MONITOR_STOP_COUNT          = 0
FAIL_CONTAINER_COUNT        = 0
ALL_CONTAINER_COUNT         = 100
UP_MESSAGE_REDUCE_MODE      = enable
```

MESSAGE_BOX には、処理中のメッセージ (イベント) が表示されますが、以下の種類があります。

メッセージ	説明
start	コンテナ監視の実行
resume	コンテナ監視の実行 (start と同じ)
stop	コンテナ監視の停止
suspend	コンテナ監視の停止 (stop と同じ)
shutdown	コンテナ監視の終了
change	一時的な cfile 値の変更
dump	共有メモリ情報のファイル出力

MONITOR_STOP_COUNT には、監視停止中のコンテナ数が表示されます。
FAIL_CONTAINER_COUNT には、リトライオーバーとなっているコンテナ数が表示されます。
ALL_CONTAINER_COUNT には、監視中、停止中にかかわらず、すべての監視コンテナ数が表示されます。

(3) コンテナの監視状態を表示する

```
# /opt/HA/CCHECK/bin/cadmin -f <cfile> -c show cent
```

```
monitor_type      = image_name
name              = rhel7
retry_count       = 0
restart_count     = 0
sts              = AVAIL
retry_over_act    = continue
rerun_time        = -----
container_count   = 50
cent_id          = 1
monitor_sts      = on
```

コンテナ名監視が指定されたときは monitor_type は container_name となり、name に監視対象コンテナ名が表示されます。

イメージ名監視のときは monitor_type は image_name となり、name に監視対象イメージ名が表示されます。

sts には、コンテナの監視状態が表示されますが、以下の種類があります。

メッセージ	説明
INIT	初期状態
RESTARTED	コンテナ再開成功
AVAIL	コンテナ正常動作中 (通常状態)
RESTART_FAIL	コンテナ再開失敗
UNAVAIL	コンテナ異常検出
RETRY_OVER	コンテナ再開リトライオーバー
RESTARTING	コンテナ再開開始
UNKNOWN	状態不明

cent_id は、cfile の cent の通番が、ccheck 起動時に cent の昇順に付与されます。ID は、1 ~ 256 (cent の上限値)とし、cfile 単位で一意となります。

monitor_sts には、以下のように cent 単位の監視状態が表示されます。

cent 監視状態	説明
On	監視状態
Off	監視停止状態
Unknown	状態不明

(4) グループの監視状態を表示する (cfile でグループ指定した場合)

```
# /opt/HA/CCHECK/bin/cadmin -f <cfile> -c show cent
```

```
monitor_type      = image_name
name              = image05
retry_count       = 0
restart_count     = 0
sts               = AVAIL
retry_over_act    = continue
rerun_time        = -----
container_count   = 2
group_name        = img_group56
group_sts         = AVAIL
cent_id           = 1
monitor_sts      = on
```

```
# /opt/HA/CCHECK/bin/cadmin -f <cfile> -c show group
```

```
group_name        = img_group56
group_sts         = AVAIL
restart_time      = -----
monitor_sts      = on
```

cent_id は、cfile の cent の通番が、ccheck 起動時に cent の昇順に付与されます。ID は、1 ~ 256 (cent の上限値) とし、cfile 単位で一意となります。

group_sts には、コンテナの監視状態が表示されますが、以下の種類があります。

グループ監視状態	説明
INIT	初期状態
RESTARTED	コンテナ再開成功
AVAIL	コンテナ正常動作中 (通常状態)
RESTART_FAIL	コンテナ再開失敗
UNAVAIL	コンテナ異常検出
RETRY_OVER	コンテナ再開リトライオーバー
RESTARTING	コンテナ再開開始
UNKNOWN	状態不明

monitor_sts には、以下のように cent 単位の監視状態が表示されます。

cent 監視状態	説明
On	監視状態
Off	監視停止状態
Unknown	状態不明

※ show group で表示される monitor_sts は、grouptag 指定されたグループの中で、監視状態のコンテナが一つでも存在する場合 on となります。

(5) コンテナの再起動が発生すると

監視対象の停止を検出し再起動が発生すると、retry_count、restart_count の値が加算されますが、次回の再起動契機まで元に戻ることはありません。運用には支障はありませんが、reload を実行することで、初期値に戻すことができます。

retry_count は、コンテナを再起動した回数です。
GRACE 値の範囲で再起動できれば、次回停止を検出した時点でリセットされます。

restart_count は、コンテナを再起動した累計です。
ccheck が終了するまでリセットされることはありません。

```
# /opt/HA/CCHECK/bin/cadmin -f <file> -c show cent
```

```
monitor_type      = image_name
name              = image04
retry_count       = 1
restart_count     = 1
sts               = RESTARTED
retry_over_act    = continue
rerun_time        = Wed Jan 10 14:44:23 2018
container_count  = 4
min_count         = 3
max_count         = 3
cent_id           = 1
monitor_sts      = on
```

reload でファイルを再読み込みすることで retry_count、restart_count の値をクリアできます。

```
# /opt/HA/CCHECK/bin/cadmin -f <file> -c reload
```

3.2.2 コンテナの運用管理について

- (1) コンテナ監視の停止

```
# /opt/HA/CCHECK/bin/cadmin -f <cfile> -c stop
```

- (2) 停止中のコンテナ監視の再開

```
# /opt/HA/CCHECK/bin/cadmin -f <cfile> -c start
```

- (3) コンテナ監視の終了

```
# /opt/HA/CCHECK/bin/cadmin -f <cfile> -c shutdown
```

- (4) タイマー値の一時的な変更
監視タイマーの値を一時的に変更できます。

```
# /opt/HA/CCHECK/bin/cadmin -f <cfile> -c change XXX val
```

- (5) cfile の再読み込み
reload 機能を使うと、ccheck を終了せずに変更した cfile の情報を反映できます。

```
# /opt/HA/CCHECK/bin/cadmin -f <cfile> -c reload
```

(注) cfile の ipckey を変更した場合は、reload 機能で情報を反映することはできません。
ccheck を一度終了してから再起動を行ってください。

- (6) リトライオーバー時におけるコンテナ監視の再開
リトライオーバーアクションに continue を指定した環境で、対象コンテナのリカバリに失敗すると、ccheck プロセスは常駐したまま当該コンテナを監視対象から外します。
監視対象コンテナ復旧後に、reload 機能を使うと ccheck を終了することなく監視の再開を行うことができます。

```
# /opt/HA/CCHECK/bin/cadmin -f <cfile> -c reload
```

- (7) コンテナ監視中におけるメンテナンス作業手順
ccheck を終了することなく対象コンテナのメンテナンス、再起動を行う場合には、stop、reload、start 機能が有効です。

コンテナ監視の一時停止

```
# /opt/HA/CCHECK/bin/cadmin -f <cfile> -c stop
```

<実行形式の変更によるコンテナの再起動などのメンテナンス作業実施>

コンテナ監視の再開

```
# /opt/HA/CCHECK/bin/cadmin -f <cfile> -c start
```

※ メンテナンスで cfile を変更した場合には、監視再開の前に reload を使用して cfile の再読み込みが必要となります。

```
cfile の再読み込みでコンテナ監視を初期化  
# /opt/HA/CCHECK/bin/cadmin -f <cfile> -c reload
```

```
コンテナ監視の再開  
# /opt/HA/CCHECK/bin/cadmin -f <cfile> -c start
```

- (8) ccheck 実行時の環境変数を確認
ccheck 起動時に使用されている環境変数を確認することが可能です

```
# /opt/HA/CCHECK/bin/cadmin -f <cfile> -c show env  
HACM_CHILDPROC_KILL           = on  
HACM_SILENT_MODE               = off  
HACM_PENDING_TIME              = 60  
HACM_GETCNTNR_COUNT            = 1000  
HACM_GETCNTNR_RETRY_COUNT      = 6  
HACM_DCKCMD_TIMEOUT            = 10
```

おのおのの表示内容について以下に説明します。

HACM_CHILDPROC_KILL、HACM_SILENT_MODE については、おのおのの環境変数に 1 が設定されている場合は on、未設定時および 1 以外が設定されている場合は off が表示されます。

```
- HACM_CHILDPROC_KILL = off   子プロセス終了モード off (デフォルト)  
- HACM_CHILDPROC_KILL = on   子プロセス終了モード on  
  
- HACM_SILENT_MODE     = off   サイレントモード off (デフォルト)  
- HACM_SILENT_MODE     = on   サイレントモード on
```

HACM_PENDING_TIME は、環境変数に設定された値が表示されます。
また環境変数未設定時は、デフォルトの 60 が表示されます。

```
- HACM_PENDING_TIME     = xx    ccheck 自動待ち合わせ時間 (デフォルト=60)
```

HACM_GETCNTNR_COUNT は、環境変数に設定された値が表示されます。
また環境変数未設定時は、デフォルトの 1000 が表示されます。

```
- HACM_GETCNTNR_COUNT = xx    監視対象選択時のコンテナ情報取得量  
                          (デフォルト=1000)
```

HACM_GETCNTNR_RETRY_COUNT は、環境変数に設定された値が表示されます。
また未設定時は、デフォルトの 6 が表示されます。

```
- HACM_GETCNTNR_RETRY_COUNT = xx  監視対象選択時のリトライ回数  
                                  (デフォルト=6)
```

HACM_DCKCMD_TIMEOUT は、環境変数に設定された値が表示されます。
また未設定時は、デフォルトの 10 が表示されます。

- HACM_DCKCMD_TIMEOUT = xx コンテナ情報を収集するコマンドを発行してから
結果を取得するまでの最大待ち時間
(デフォルト=10)

これらの環境変数の設定手順および詳細については、「3.3.4 サイレントモードでの運用手順」および、「3.3.5 ccheck 終了時に子プロセスを回収する手順」、「3.3.6 ccheck 起動時の自動待ち合わせ時間を変更する手順」、「3.3.8 監視対象選択時のコンテナ情報取得量を変更する手順」、「3.3.9 監視対象選択時のリトライ回数を変更する手順」、「3.3.10 コンテナ情報を収集するコマンドの最大待ち時間を変更する手順」を参照してください。

なお、通常はデフォルト値の使用で問題ありません。

3.2.3 異常検知時の確認手順

- (1) コンテナ監視における異常通知について

ccheck は、コンテナ監視中に致命的な異常を検出するとシステムログファイル (/var/log/messages) にエラーメッセージを出力します。
システムログメッセージの説明と対処方法については、「4.1 syslog メッセージ」を参照してください。

- (2) 共有メモリトレースファイルについて

ccheck は、共有メモリ上にコンテナ監視の履歴を常時取得していますが、コンテナ監視で異常を検出した場合の解析手段として有効です。
このトレースファイルは約 3Mbyte 程度のサイクリックログですが、ccheck の終了時、reload コマンド実行時、リトライオーバー時を契機として cfile の SHM_DUMP_FILE に指定した場所へ出力します。
このファイルは、2 世代までバックアップされます。
なお、以下のコマンドで強制的に共有メモリトレースをファイルへ出力できます。

```
# /opt/HA/CCHECK/bin/cadmin -f <cfile> -c dump filename
```

- (3) 共有メモリトレースファイルの参照手順について

本ファイルは cadmin で参照できます。

```
# /opt/HA/CCHECK/bin/cadmin -f <cfile> -c unload filename
```

また、strings コマンドでもメッセージのみ参照できます。

```
# strings filename
```

3.2.4 デバッグ支援機能

cfile や再起動スクリプトの妥当性を確認するために、デバッグ支援機能があります。

(1) デバッグ手順について

ccheck をトレースオプション付き (-t) で起動すると、標準出力にコンテナ監視の実行履歴が表示されます。このとき、端末画面を占有しますので注意が必要です。

他の端末画面から、cadmin コマンドでの操作や、stop コマンドなどで対象コンテナを停止することで、ccheck の動作が確認できます。

本機能は、cfile をデバッグするには非常に有効です。

トレースモードでコンテナ監視を起動 (バックグラウンド起動である & は付与しません)

```
# /opt/HA/CCHECK/bin/ccheck -f <cfile> -t
```

(2) コンテナ監視の一時停止機能について

特定のファイルを生成することで、ccheck 起動直後からコンテナ監視の一時停止が可能です。

対象コンテナが正常に動作しない場合でも ccheck を起動することができます。

ccheck 起動前に、一時ファイルを作成します。

```
# touch /opt/HA/CCHECK/bin/ccheck.ignore
```

ccheck を起動すると、システムログに監視停止中のメッセージが表示されます。

```
:  
Jan 30 09:12:56 localhost ccheck: ccheck wait... by /opt/HA/CCHECK/bin/ccheck.ignore
```

一時ファイルを削除すると、監視を再開します。

```
# rm /opt/HA/CCHECK/bin/ccheck.ignore
```

ccheck が監視を開始したメッセージがシステムログに表示されます。

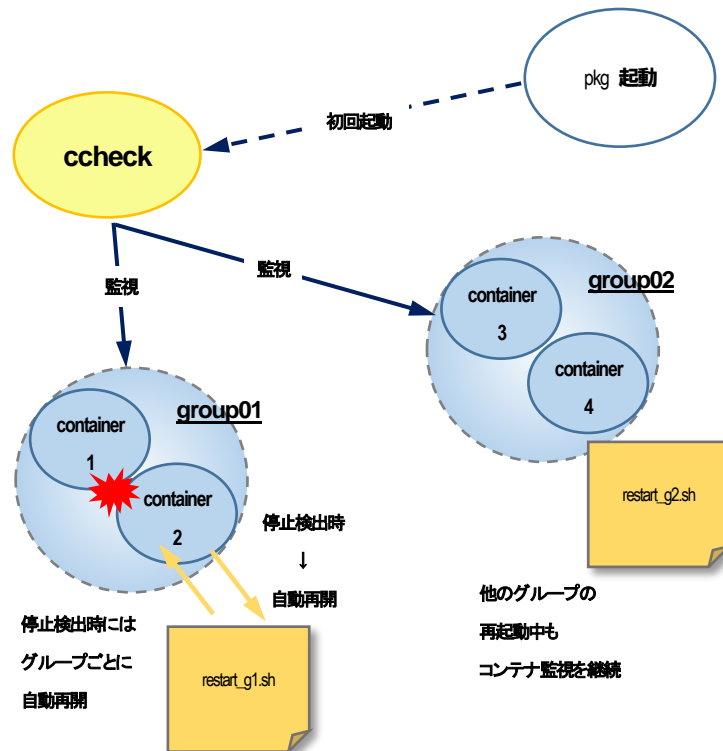
```
:  
Jan 30 09:12:56 localhost ccheck: ccheck wait... by /opt/HA/CCHECK/bin/ccheck.ignore  
Jan 30 09:17:56 localhost ccheck: ccheck start by /opt/HA/CCHECK/bin/ccheck.ignore
```

(注) ccheck 起動後にコンテナ監視の一時停止を行うには、cadmin で stop を指定してください。

3.3 高度な運用手順について

3.3.1 グループ監視の導入手順

本機能を導入するには、cfile にグループ名を指定する手続きが必要となります。
ここでは、group01、group02 の 2 種類のグループを監視する ccheck を想定して説明します。
cfile を cfile_group、group01 を再起動するスクリプトを restart_g1.sh、
group02 を再起動するスクリプトを restart_g2.sh とします。



① ccheck

ccheck は指定されたグループごとにコンテナの停止監視および再開を行います。

cfile で定義する再起動スクリプトには、グループごとに監視するすべてのコンテナを停止させた後に再起動するような処理を必要に応じて作成してください。

このとき、以下の点に注意してください。

- ・ cfile で定義する再起動スクリプトは、同一グループ内で同じ再起動スクリプトを使用してください。
- ・ cfile で定義する retry_over_action 、 grace 値、retry_count などの各設定値は同一グループ内で同じ値を指定してください。

グループ監視における ccheck のサンプル (cfile_group)

```
##### PARAM #####
IPCKEY_CCHECK          0x1f000001
MSG_CHECK_INTERVAL    5
MONITOR_INTERVAL      10
SHM_DUMP_FILE         /var/opt/HA/CCHECK/log/ccount_group_dump

##### CENT (group01) #####
container1:/bin/cm/restart_g1.sh:86400:3:continue:grouptag=group01
container2:/bin/cm/restart_g1.sh:86400:3:continue:grouptag=group01

##### CENT (group02) #####
container3:/bin/cm/restart_g2.sh:86400:3:continue:grouptag=group02
container4:/bin/cm/restart_g2.sh:86400:3:continue:grouptag=group02
```

② 再起動スクリプト

グループ監視における再起動スクリプトのサンプル (restart_g1.sh)

```
#!/bin/sh

# グループのコンテナをすべて停止する。
docker stop container1
docker stop container2

# グループのコンテナをすべて起動する。
docker start container1
docker start container2
/bin/sleep 10
```

グループ監視における再起動スクリプトのサンプル (restart_g2.sh)

```
#!/bin/sh

# グループのコンテナをすべて停止する。
docker stop container3
docker stop container4

# グループのコンテナをすべて起動する。
docker start container3
docker start container4
/bin/sleep 10
```

3.3.2 イメージ名監視 (コンテナ起動数監視) の導入手順

同一イメージで複数のコンテナが同時に起動する環境で、起動するコンテナ数の監視を行います。

(1) 起動するコンテナ数で下限監視する方式

cfile のコンテナ監視エントリに `image_name_check=enable` とともに `min_count` オプションを指定することで、監視対象イメージの起動コンテナ数が指定した数を下回ったときに異常を検出します。

以下の例では、イメージ "rhel7" で起動するコンテナが 2 個未満になると異常を検出し、再起動スクリプト "restart_img_rhel7.sh" を実行します。

```
##### CENT #####
## image_name:restart shell:grace:retry_count_max:retry_over_action:option
rhel7:/bin/cm/restart_img_rhel7.sh:86400:3:continue:image_name_check=enable,min_count=2
```

(2) 起動するコンテナ数で上限監視する方式

cfile のコンテナ監視エントリに `image_name_check=enable` とともに `max_count` オプションを指定することで、監視対象イメージの起動コンテナ数が指定した数を上回ったときに異常を検出します。

以下の例では、イメージ "rhel7" で起動するコンテナが 5 個を超えると異常を検出し、再起動スクリプト "restart_img_rhel7.sh" を実行します。

```
##### CENT #####
## image_name:restart shell:grace:retry_count_max:retry_over_action:option
rhel7:/bin/cm/restart_img_rhel7.sh:86400:3:continue:image_name_check=enable,max_count=5
```

(3) 起動するコンテナ数の範囲で監視する方式

cfile のコンテナ監視エントリに `image_name_check=enable` とともに `min_count` オプションと `max_count` オプションを同時に指定することで、監視対象イメージの起動コンテナ数が指定した範囲外になると異常を検出します。

`max_count` には、`min_count` 以上の値を設定する必要があります。

以下の例では、イメージ "rhel7" で起動するコンテナが 2 ~ 5 以外の個数になると異常を検出し、再起動スクリプト "restart_img_rhel7.sh" を実行します。

```
##### CENT #####
## image_name:restart shell:grace:retry_count_max:retry_over_action:option
rhel7:/bin/cm/restart_img_rhel7.sh:86400:3:continue:image_name_check=enable,min_count=2,max_count=5
```

3.3.3 監視間隔を短くした運用

cfile の共通部情報 (PARAM) に設定する MSG_CHECK_INTERVAL (内部イベントをスケジュールするタイマー値)、MONITOR_INTERVAL (コンテナ監視間隔) を短く設定することによって、コンテナ停止の検出とリカバリを早期に実現できるようにすることが可能となります。

次の cfile のサンプルでは、MSG_CHECK_INTERVAL=1 (秒)、MONITOR_INTERVAL=1 (秒) 間隔でコンテナ監視を行います。

```
##### PARAM #####
IPCKEY_CCHECK           0x1f000001
MSG_CHECK_INTERVAL     1
MONITOR_INTERVAL 1
SHM_DUMP_FILE          /var/opt/HA/CCHECK/log/ccheck_dump

##### CENT #####
container1:/bin/cm/restart_container1.sh:86400:3:continue
```

ただし、上記のような運用事例もありますが、システムによっては負荷が高くなり業務運用に影響する場合がありますので、事前に性能的に問題がないか動作確認を行ってください。

3.3.4 サイレントモードでの運用手順

ログメッセージを抑制して運用する手順について説明します。

ccheckは、運用中システムログにメッセージを出力します。
システムログにメッセージを出力させたくない場合、メッセージを抑制したサイレントモードで実行することができます。

(1) 導入手順

- ① 環境変数 HACM_SILENT_MODE に 1 を設定します。

- ・ sh の場合

```
# HACM_SILENT_MODE=1  
# export HACM_SILENT_MODE
```

- ・ csh の場合

```
% setenv HACM_SILENT_MODE 1
```

※ 上記のように環境変数を設定した状態で起動した ccheck については、システムログメッセージを抑制した状態で実行されます。

- ② ccheck を起動し、コンテナ監視を開始します。

```
#/opt/HA/CCHECK/bin/ccheck -f <cfile 名> &
```

(2) 確認手順

動作中の ccheck のモードについては、cadmin コマンドで確認することができます。

```
#/opt/HA/CCHECK/bin/cadmin -f <cfile 名> -c show env
```

HACM_SILENT_MODE = on ★

サイレントモードで ccheck を起動している場合は "on" 、
サイレントモードでない場合は、"off" が表示されます。

(3) 運用上の注意・制限事項

- ・ サイレントモードで運用した場合は、ccheck のメッセージは一切出力されません。
そのため障害発生時の解析などは困難となりますので、あらかじめご了承ください。
一般的には、通常モードでの運用を推奨します。
- ・ すでに起動中の ccheck を、サイレントモードに変更することはできません。
一度 ccheck を cadmin -c shutdown で終了させてから、環境変数を設定し ccheck を再起動してください。

3.3.5 ccheck 終了時に子プロセスを回収する手順

ccheck コマンドは、コンテナ情報収集時および再起動スクリプト実行時に内部で fork(2) を使用しているため ccheck 終了時に生成された子プロセスなどが残ってしまう場合があります。子プロセスが残ってしまうことを避けるために、ccheck 終了時に関連のプロセスを終了する「子プロセス終了モード」で運用することが可能です。子プロセス終了モードの設定手順は以下のとおりです。

(1) 導入手順

- ① 環境変数 HACM_CHILDPROC_KILL に 1 を設定します。

- ・ sh の場合

```
# HACM_CHILDPROC_KILL=1  
# export HACM_CHILDPROC_KILL
```

- ・ csh の場合

```
% setenv HACM_CHILDPROC_KILL 1
```

※ 上記のように環境変数を設定した状態で起動した ccheck については、終了時に子プロセスなどの関連プロセスを終了します。

- ② ccheck を起動し、コンテナ監視を開始します。

```
#/opt/HA/CCHECK/bin/ccheck -f <cfile 名> &
```

(2) 確認手順

動作中の ccheck のモードについては、cadmin コマンドで確認することができます。

```
#/opt/HA/CCHECK/bin/cadmin -f <cfile 名> -c show env
```

HACM_CHILDPROC_KILL = on ★

子プロセス終了モードで起動している ccheck の場合は "on" 、
子プロセス終了モードでない場合は、"off" が表示されます。

(3) 運用上の注意・制限事項

- ・ すでに起動中の ccheck を、子プロセス終了モードに変更することはできません。一度 ccheck を cadmin -c shutdown で終了させてから、環境変数を設定し ccheck を再起動してください。
- ・ 子プロセス終了モードで実行しない場合、ccheck 終了後に ccheck の子プロセスが残留する可能性があります。基本的な運用には特に影響はございません。

3.3.6 ccheck 起動時の自動待ち合わせ時間を変更する手順

ccheck コマンドは、起動時および reload 実行時に監視対象のコンテナが存在しない場合に、監視対象コンテナが起動するまで一定時間自動的に待ち合わせを行います。自動待ち合わせ時間内に監視対象のコンテナが起動した場合は、その時点から監視を開始します。

自動待ち合わせ時間はデフォルトで 60 秒となっていますが、環境変数 HACM_PENDING_TIME を設定することで待ち合わせ時間を変更することができます。

ccheck 起動時にすぐに停止を検出したい場合や監視対象コンテナの起動に時間がかかる場合などに自動待ち合わせ時間を変更してください。

通常は変更する必要はありません。

設定は秒単位で指定し、設定可能な範囲は 0 ~ 86400 (=60*60*24 (1 日)) です。

自動待ち合わせ時間を変更する手順は以下のとおりです。

(1) ccheck 起動後すぐに監視対象コンテナの停止を検出したい (自動待ち合わせを行わない) 場合の手順

① 環境変数 HACM_PENDING_TIME に 0 を設定します。

・ sh の場合

```
# HACM_PENDING_TIME=0
# export HACM_PENDING_TIME
```

・ csh の場合

```
% setenv HACM_PENDING_TIME 0
```

※ 上記のように環境変数を設定した状態で起動した ccheck については、ccheck 起動時に監視対象コンテナが起動していない場合、すぐに監視対象コンテナの停止を検出します。

② ccheck を起動し、コンテナ監視を開始します。

```
#/opt/HA/CCHECK/bin/ccheck -f <cfile 名> &
```

③ 確認手順

動作中の ccheck の自動待ち合わせ時間は、cadmin コマンドで確認することができます。

```
#/opt/HA/CCHECK/bin/cadmin -f <cfile 名> -c show env
```

HACM_PENDING_TIME = 0 ★

設定した自動待ち合わせ時間が表示されます。

環境変数 HACM_PENDING_TIME を設定していない場合は、デフォルトの 60 が表示されます。

(2) 自動待ち合わせ時間を 120 秒に変更する場合の手順

① 環境変数 HACM_PENDING_TIME に 120 を設定します。

・ sh の場合

```
# HACM_PENDING_TIME=120  
# export HACM_PENDING_TIME
```

・ csh の場合

```
% setenv HACM_PENDING_TIME 120
```

※ 上記のように環境変数を設定した状態で起動した ccheck については、ccheck 起動時に監視対象コンテナが起動していない場合、120 秒間待ち合わせを行います。

ccheck 起動後、120 秒経過しても監視対象コンテナが起動していない場合にすぐに監視対象コンテナの停止を検出します。

② ccheck を起動し、コンテナ監視を開始します。

```
#/opt/HA/CCHECK/bin/ccheck -f <cfile 名> &
```

③ 確認手順

動作中の ccheck の自動待ち合わせ時間は、cadmin コマンドで確認することができます。

```
#/opt/HA/CCHECK/bin/cadmin -f <cfile 名> -c show env
```

HACM_PENDING_TIME = 120 ★

設定した自動待ち合わせ時間が表示されます。

環境変数 HACM_PENDING_TIME を設定していない場合は、デフォルトの 60 が表示されます。

(3) 運用上の注意・制限事項

- ・ すでに起動中の ccheck の自動待ち合わせ時間を変更することはできません。一度 ccheck を cadmin -c shutdown で終了させてから、環境変数を設定し ccheck を再起動してください。指定した時間内に監視対象のコンテナが起動した場合は、その時点で監視を開始します。

3.3.7 ccheck 起動時のサマリー情報を syslog 出力する手順

ccheck 起動時に、設定した監視対象の数と実際に監視を開始できた数を syslog に出力します。サマリーメッセージを出力するように設定することで、大量の監視対象を有するシステムにおいても、おのおのが設定された監視条件下で、問題なく監視され始めたか否かの判断が容易になります。

(1) cfile の記述方法

cfile の共通部情報 (PARAM) に、UP_MESSAGE_REDUCE_MODE を指定することで、ccheck 起動時の監視コンテナ数を syslog に出力することが可能となります。

共通部情報 (PARAM) の設定

共通部情報のフォーマットは以下のとおりです。

```
##### PARAM #####
IPCKEY_CCHECK                ipckey
MSG_CHECK_INTERVAL          msg_check_interval
MONITOR_INTERVAL            monitor_interval
SHM_DUMP_FILE               shm_dump_file
UP_MESSAGE_REDUCE_MODE      up_message_reduce_mode ←★追加
```

設定値について以下に説明します。

up_message_reduce_mode

enable または disable のみ設定可能です。

enable を設定した場合、ccheck 起動時および reload 実行時に監視コンテナ数と、cent に記載されたコンテナ数を syslog に出力します。

disable またはパラメーター未設定の場合は、従来のメッセージのみ出力し、ccheck 起動時の監視コンテナ数を示すメッセージは出力されません。

cfile の記載例は以下となります。

```
##### PARAM #####
IPCKEY_CCHECK                0x1f000001
MSG_CHECK_INTERVAL          5
MONITOR_INTERVAL            10
SHM_DUMP_FILE               /var/opt/HA/CCHECK/log/ccheck_dump
UP_MESSAGE_REDUCE_MODE      enable

##### CENT #####
container1:/bin/cm/restart_container1.sh:86400:3:continue
```


(2) syslog メッセージ

syslog の facility と level は以下のとおりです。

facility: LOG_USER
level: LOG_INFO

ccheck 起動時の監視開始コンテナ数出力機能で出力されるメッセージは以下です。

**ccheck start. (up_num=<監視を開始した対象の数>,
total_cent_num=<CENTの記載数>) cfile=<cfile名>**

syslog メッセージの出力例は以下となります。

(例) ccheck 起動時、cent に記載された 6 個の監視対象のうち 5 個の起動を確認した場合

```
Jan 22 10:04:20 localhost ccheck[53113]: ccheck start. (up_num=5, total_cent_num=6) cfile=extra_4_1_1
```

(3) 確認手順

動作中の ccheck の設定については、cadmin コマンドで確認することができます。

```
#/opt/HA/CCHECK/bin/cadmin -f <cfile 名> -c show param  
MSG_CHECK_INTERVAL          = 5  
MONITOR_INTERVAL            = 10  
MONITOR_TRY_COUNT           = 2  
SHM_DUMP_FILE               = /opt/HA/CCHECK/log/ccheck_dump  
CFILE                        = cfile  
MESSAGE_BOX                  = start  
MONITOR_STOP_COUNT          = 0  
FAIL_CONTAINER_COUNT        = 0  
ALL_CONTAINER_COUNT         = 5  
UP_MESSAGE_REDUCE_MODE     = enable ★
```

★ UP_MESSAGE_REDUCE_MODE enable 設定時に表示されます。
UP_MESSAGE_REDUCE_MODE disable 設定時または、
UP_MESSAGE_REDUCE_MODE 未指定時は表示されません。

3.3.8 監視対象選択時のコンテナ情報取得量を変更する手順

ccheck は、起動時および cadmin の start コマンド実行時に OS で起動している全コンテナの情報を検索して、監視対象の特定を行います。以後は監視タイミングごとに全コンテナ情報を検索します。

デフォルトではこの情報取得処理はコンテナ数で 1000 個を最大として行いますが、環境変数 HACM_GETCNTNR_COUNT の設定で情報取得量 (コンテナ数の最大値) を変更することができます。

OS で起動するコンテナ数が 1000 を越えるような大規模環境の場合に、必要に応じて起動コンテナ数よりも大きなコンテナ情報取得量を設定してください。

通常は変更する必要はありません。

設定可能な範囲は 1 ~ 10000 です。

コンテナ情報取得量を変更する手順は以下のとおりです。

- (1) ccheck 起動時に 1500 コンテナ分の情報を取得する場合
(OS で起動するコンテナ数が 1400 程度の場合など)

- ① 環境変数 HACM_GETCNTNR_COUNT に 1500 を設定します。

- ・ sh の場合

```
# HACM_GETCNTNR_COUNT=1500  
# export HACM_GETCNTNR_COUNT
```

- ・ csh の場合

```
% setenv HACM_GETCNTNR_COUNT 1500
```

※ 上記のように環境変数を設定した状態で起動した ccheck については、ccheck 起動時に最大 1500 コンテナ分の情報を取得します。

- ② ccheck を起動し、コンテナ監視を開始します。

```
#/opt/HA/CCHECK/bin/ccheck -f <cfile 名> &
```

- ③ 確認手順

動作中の ccheck のコンテナ情報取得量は、cadmin コマンドで確認することができます。

```
#/opt/HA/CCHECK/bin/cadmind -f <cfile 名> -c show env
```

HACM_GETCNTNR_COUNT = 1500 ★

設定したコンテナ情報取得量が表示されます。

環境変数 HACM_GETCNTNR_COUNT を設定していない場合は、デフォルトの 1000 が表示されます。

(2) 運用上の注意・制限事項

- ・すでに起動中の ccheck のコンテナ情報取得量を変更することはできません。
一度 ccheck を `cadmin -c shutdown` で終了させてから、環境変数を設定し ccheck を再起動してください。
- ・情報取得量を多くするほど、ccheck 内部で一時的に確保するメモリの量が増えることになります。
たとえばデフォルトの 1000 コンテナ指定時は約 30 MB 程度ですが 1500 コンテナ指定時は約 42 MB 程度を ccheck プロセスで使用します。

3.3.9 監視対象選択時のリトライ回数を変更する手順

ccheck は、起動時および cadmin の start コマンド実行時に OS で起動するコンテナの情報取得を行います。

このとき、デフォルトでは最大 6 回の情報取得のリトライを行いますが、環境変数 HACM_GETCNTNR_RETRY_COUNT を設定することで変更することができます。通常は変更する必要はありません。

設定可能な範囲は 1 ~ 100 です。

コンテナ情報取得時のリトライ回数を変更する手順は以下のとおりです。

(1) リトライを最大 10 回行う場合

- ① 環境変数 HACM_GETCNTNR_RETRY_COUNT に 10 を設定します。

・ sh の場合

```
# HACM_GETCNTNR_RETRY_COUNT=10
# export HACM_GETCNTNR_RETRY_COUNT
```

・ csh の場合

```
% setenv HACM_GETCNTNR_RETRY_COUNT 10
```

※ 上記のように環境変数を設定した状態で起動した ccheck については、最大 10 回のコンテナ情報取得のリトライを行います。

- ② ccheck を起動し、コンテナ監視を開始します。

```
#/opt/HA/CCHECK/bin/ccheck -f <cfile 名> &
```

- ③ 確認手順

動作中の ccheck のコンテナ情報取得のリトライ回数は、cadmin コマンドで確認することができます。

```
#/opt/HA/CCHECK/bin/cadmin -f <cfile 名> -c show env
```

HACM_GETCNTNR_RETRY_COUNT = 10 ★

設定したコンテナ情報取得のリトライ回数が表示されます。

環境変数 HACM_GETCNTNR_RETRY_COUNT を設定していない場合は、デフォルトの 6 が表示されます。

(2) 運用上の注意・制限事項

- ・ すでに起動中の ccheck のコンテナ情報取得のリトライ回数を変更することはできません。一度 ccheck を cadmin -c shutdown で終了させてから、環境変数を設定し ccheck を再起動してください。

3.3.10 コンテナ情報を収集するコマンドの最大待ち時間を変更する手順

ccheck は、コンテナ情報を取得するために docker コマンドを実行しています。
このとき、デフォルトでは最大 10 秒のコマンド実行結果待ちを行います。
環境変数 HACM_DCKCMD_TIMEOUT を設定することで最大待ち時間を変更することができます。
通常は変更する必要はありません。

設定可能な範囲は 1 ~ 86400 (=60*60*24 (1 日)) です。

docker コマンドの実行待ち時間を変更する手順は以下のとおりです。

(1) 最大 20 秒の docker コマンド実行待ちを行う場合

① 環境変数 HACM_DCKCMD_TIMEOUT に 20 を設定します。

・ sh の場合

```
# HACM_DCKCMD_TIMEOUT=20  
# export HACM_DCKCMD_TIMEOUT
```

・ csh の場合

```
% setenv HACM_DCKCMD_TIMEOUT 20
```

※ 上記のように環境変数を設定した状態で起動した ccheck については、
最大 20 秒 docker コマンド実行待ちを行います。

② ccheck を起動し、コンテナ監視を開始します。

```
#/opt/HA/CCHECK/bin/ccheck -f <cfile 名> &
```

③ 確認手順

動作中の ccheck の docker コマンド実行待ち時間は、cadmin コマンドで
確認することができます。

```
#/opt/HA/CCHECK/bin/cadmin -f <cfile 名> -c show env
```

HACM_DCKCMD_TIMEOUT = 20 ★

設定した docker コマンド実行待ち時間が表示されます。
環境変数 HACM_DCKCMD_TIMEOUT を設定していない場合は、
デフォルトの 10 が表示されます。

(2) 運用上の注意・制限事項

- ・ すでに起動中の ccheck の docker コマンド実行待ち時間を変更することはできません。
一度 ccheck を cadmin -c shutdown で終了させてから、環境変数を設定し ccheck を
再起動してください。

4 メッセージ

4.1 syslog メッセージ

4.1.1 フォーマットについて

syslog に出力するフォーマットは以下のとおりです。

```
Jan 24 09:35:26 hostname xxxx[yyyy]: msg
- xxxx   : コマンド名 (ccheck, cadmin)
- yyyy   : pid
- msg    : メッセージ
```

syslog の facility と level は以下のとおりです。

```
facility : LOG_USER
level   : LOG_ERR または LOG_WARNING または LOG_INFO
```

4.1.2 監視対象となる syslog メッセージ

ccheck が出力する syslog メッセージには、コンテナ監視で復旧不可能なエラーを報告する致命的なものがあります。これらのメッセージは運用管理ソフト等により監視することをおすすめします。

対象となる syslog メッセージは下記の通りです。

なお、下記以外のメッセージの監視については、特に必須ではありません。

停止監視によるリトライオーバーのメッセージ (3 種類)

```
RETRY_OVER:CONTINUE
RETRY_OVER:SHUTDOWN
RETRY_OVER:EXIT
```

処置

対象コンテナの停止を検出し、一定回数リトライしても再起動できない状態です。
対象コンテナが起動不可の原因を調査してください。

4.1.3 運用メッセージ

以下に ccheck の運用メッセージの説明を記載します。

(1) LOG_ERR

SIGINT Catch (xxx)!!

説明: シグナル SIGINT を受信しました。(シグナル xxx)

SIGQUIT Catch (xxx)!!

説明: シグナル SIGQUIT を受信しました。(シグナル xxx)

SIGTERM Catch (xxx)!!

説明: シグナル SIGTERM を受信しました。(シグナル xxx)

Stop xxxx!!

説明: シグナル SIGTERM を受信したため、ccheck を終了しました。(シグナル xxx)

Undefine signal Catch (xxx)!!

説明: その他のシグナルを受信しました。(シグナル xxx)

(2) LOG_WARNING

Try to restart (xxx)

説明: 監視対象 xxx の再起動を行います。(監視対象 xxx)

Restart fail (xxx)

説明: 監視対象 xxx の再起動に失敗しました。(監視対象 xxx)

(image_name=xxx) Down

説明: イメージ名監視において監視対象の停止を検出しました。(イメージ名 xxx)

(cname=xxx) Down

説明: コンテナ名監視において監視対象の停止を検出しました。(コンテナ名 xxx)

(image_name=xxx) Up

説明: イメージ名監視において監視対象の起動を検出しました。(イメージ名 xxx)

(cname=xxx) Up

説明: コンテナ名監視において監視対象の起動を検出しました。(コンテナ名 xxx)

Shutdown ccheck

説明: コンテナ監視を終了しました。

ccheck wait... by /opt/HA/CCHECK/bin/ccheck.ignore

説明: 監視停止用の一時ファイルが存在するため、コンテナ監視を待ち合わせています。

ccheck start by /opt/HA/CCHECK/bin/ccheck.ignore

説明: 監視停止用の一時ファイルが削除されたため、コンテナ監視を開始します。

(3) LOG_INFO

Container information collection timeout. cent_id=xxx

説明: コンテナ情報の収集がタイムアウトしました。(CENT id xxx)

Container information collection error. cent_id=xxx

説明: コンテナ情報の収集が異常終了しました。(CENT id xxx)

ccheck start. (up_cntr_num=xxx, total_cent_num=yyy) cfile=zzz

説明: コンテナ監視を開始します。(起動コンテナ数 xxx, CENT数 yyy, cfile zzz)

Operation timeout. [command] cfile=zzz (arg="xxxxxxx")

説明: cadmin コマンド [command] (引数="xxxxxxx") はタイムアウトしました。(cfile zzz)

Command execute. [message] cfile=zzz (arg="xxxxxxx")

説明: cadmin コマンド [command] (引数="xxxxxxx") を実行しました。(cfile zzz)

4.1.4 cadmin コマンドのメッセージ

以下に cadmin コマンドのメッセージの説明、および行うべき処置方法を記載します。

(1) LOG_ERR

shmget(2) fail. errno = xxx.

説明: 共有メモリの確保に失敗しました。(エラー番号 xxx)

処置: システムの状態を確認して、システムの再起動を行ってください。
システムの再起動を行っても異常が改善されない場合は、
障害解析情報を採取し、サポートセンターに連絡してください。

shmat(2) fail. errno = xxx.

説明: 共有メモリの確保に失敗しました。(エラー番号 xxx)

処置: システムの状態を確認して、システムの再起動を行ってください。
システムの再起動を行っても異常が改善されない場合は、
障害解析情報を採取し、サポートセンターに連絡してください。

lockfile open(2) fail. lockfile = xxx.

説明: ロックファイルのオープンに失敗しました。(ファイル xxx)

処置: システムファイルの上限値を上げるか、オープンしているファイルを
クローズしてから、システムの再起動を行ってください。
システムの再起動を行っても異常が改善されない場合は、
障害解析情報を採取し、サポートセンターに連絡してください。

lockf(F_LOCK) fail. errno = xxx.

説明: ファイルのロックに失敗しました。(エラー番号 xxx)

処置: システムの再起動を行ってください。
システムの再起動を行っても異常が改善されない場合は、
障害解析情報を採取し、サポートセンターに連絡してください。

lockf(F_ULOCK) fail. errno = xxx.

説明: ファイルのアンロックに失敗しました。(エラー番号 xxx)

処置: システムの再起動を行ってください。
システムの再起動を行っても異常が改善されない場合は、
障害解析情報を採取し、サポートセンターに連絡してください。

(2) LOG_WARNING

illegal parameter

説明: change オプションに指定されたパラメーターが正しくありません。

処置: change オプションに正しいパラメーターを指定してください。

monitor_interval illegal value

説明: change オプションに指定された monitor_interval の値が正しくありません。

処置: change オプションの monitor_interval の値を正しく指定してください。

msg_check_interval illegal value

説明: change オプションに指定された msg_check_interval の値が monitor_interval で指定された値を超えています。

処置: change オプションのmsg_check_interval の値は、monitor_interval より小さな値を指定してください。

msg_check_interval must be less than monitor_interval

説明: msg_check_interval と monitor_interval の関係は monitor_interval の値が大きくなければなりません。

処置: change オプションのmsg_check_interval の値は、monitor_interval より小さな値を指定してください。

(3) LOG_INFO

Operation timeout. [command] cfile=zzz (arg="xxxxxxx")

説明: cadmin コマンド [command] (引数 xxxxxxx) はタイムアウトしました。(cfile zzz)

処置: システムの再起動を行ってください。
システムの再起動を行っても異常が改善されない場合は、
障害解析情報を採取し、サポートセンターに連絡してください。

Command execute. [command] cfile=zzz (arg="xxxxxxx")

説明: cadmin コマンド [command] (引数 xxxxxxx) を実行しました。(cfile zzz)

処置: 動作に支障ないため、特に処置の必要はありません。

4.1.5 ccheck コマンドのメッセージ

以下に ccheck コマンドのメッセージの説明、および行うべき処置方法を記載します。

(1) LOG_ERR

illegal option

説明: 無効なオプションが指定されました。

処置: 同時に出力される Usage にしたがって正しいコマンドを再投入してください。

```
Usage:  ccheck -f filename
        ccheck -f filename -w waittime(second)
        ccheck -f filename -t
```

lockfile open(2) fail. lockfile = xxx.

説明: ロックファイルのオープンに失敗しました。(ファイル xxx)

処置: システムファイルの上限値を上げるか、オープンしているファイルをクローズしてから、システムの再起動を行ってください。

システムの再起動を行っても異常が改善されない場合は、
障害解析情報を採取し、サポートセンターに連絡してください。

lockf(F_LOCK) fail. errno = xxx.

説明: ファイルのロックに失敗しました。(エラー番号 xxx)

処置: システムの再起動を行ってください。

システムの再起動を行っても異常が改善されない場合は、
障害解析情報を採取し、サポートセンターに連絡してください。

lockf(F_ULOCK) fail. errno = xxx.

説明: ファイルのアンロックに失敗しました。(エラー番号 xxx)

処置: システムの再起動を行ってください。

システムの再起動を行っても異常が改善されない場合は、
障害解析情報を採取し、サポートセンターに連絡してください。

shmat(2) fail. errno = xxx.

説明: 共有メモリの確保に失敗しました。(エラー番号 xxx)

処置: システムの状態を確認して、システムの再起動を行ってください。

システムの再起動を行っても異常が改善されない場合は、
障害解析情報を採取し、サポートセンターに連絡してください。

shmget(2) fail. errno = xxx.

説明: 共有メモリの確保に失敗しました。(エラー番号 xxx)

処置: システムの状態を確認して、システムの再起動を行ってください。

システムの再起動を行っても異常が改善されない場合は、
障害解析情報を採取し、サポートセンターに連絡してください。

shmdt(2) fail. errno = xxx.

説明: 共有メモリの開放に失敗しました。(エラー番号 xxx)

処置: システムの状態を確認し、手動で共有メモリを回収して、システムの再起動を行ってください。

システムの再起動を行っても異常が改善されない場合は、
障害解析情報を採取し、サポートセンターに連絡してください。

shmctl(2) fail. errno = xxx.

説明: 共有メモリの開放に失敗しました。(エラー番号 xxx)

処置: システムの状態を確認し、手動で共有メモリを回収して、システムの再起動を行ってください。
システムの再起動を行っても異常が改善されない場合は、障害解析情報を採取し、サポートセンターに連絡してください。

(xxx) RETRY_OVER:CONTINUE

説明: 監視対象 xxx の再起動回数がリトライ回数に達したため監視を終了します。
retry_over_action の記述にしたがって、他の動作を継続します。

処置: 監視対象 xxx が停止した原因を調査してください。

(xxx) RETRY_OVER:SHUTDOWN

説明: 監視対象 xxx の再起動回数がリトライ回数に達したため監視を終了します。
retry_over_action の記述にしたがって、ccheck を終了します。

処置: 監視対象 xxx が停止した原因を調査してください。

(xxx) RETRY_OVER:EXIT

説明: 監視対象 xxx の再起動回数がリトライ回数に達したため監視を終了します。
retry_over_action の記述にしたがって、ccheck を終了します。

処置: 監視対象 xxx が停止した原因を調査してください。

(2) LOG_WARNING

SHUTDOWN caused by (xxx)

説明: 監視対象 xxx のリトライオーバーにより、ccheck を終了します。

処置: 監視対象 xxx が停止した原因を調査してください。

(xxx) FATAL

説明: 監視対象 xxx に異常が発生したため、ccheck を終了します。

処置: 監視対象 xxx が停止した原因を調査してください。

Something wrong with Sharedmemory

説明: 共有メモリが不正です。

処置: 共有メモリの空き容量を増やすか、システムの再起動を行ってください。
システムの再起動を行っても異常が改善されない場合は、障害解析情報を採取し、サポートセンターに連絡してください。

can not remove xxxx (errno=xxx)

説明: ロックファイル xxxx の削除に失敗しました。(エラー番号 xxx)

処置: 動作に支障ないため、特に処置の必要はありません。

4.1.6 停止監視の異常メッセージ

以下に監視時の異常メッセージの説明、および行うべき処置方法を記載します。

(1) LOG_ERR

Can't get xxxx from shm.

説明: 共有メモリの情報 xxxx が取得できません。

処置: 内部エラーです。ccheck の再起動を行ってください。
ccheck の再起動を行っても異常が改善されない場合は、
障害解析情報を採取し、サポートセンターに連絡してください。

Set container fail (xxx).

説明: ccheck の起動時に監視対象 xxx が起動していないことを検出しました。

処置: 監視対象 xxx が起動されていることを確認してください。
起動されていれば、cfile の監視対象名が正しいことを確認してください。

-> fork(2) fail. errno=xxx

説明: コンテナ情報取得時、子プロセス生成に失敗しました。(エラー番号 xxx)

処置: ccheck の再起動を行ってください。
ccheck の再起動を行っても異常が改善されない場合は、
障害解析情報を採取し、サポートセンターに連絡してください。

-> fork(2) fail (xxx). Suspend ccheck

説明: 再起動に失敗しました。(エラー番号 xxx) fork に失敗したため監視を中断します。

処置: 再起動スクリプトが単体で実行できるか確認してください。
単体で実行できる場合、以下を確認してください。

- ・再起動スクリプトに実行権が付与されていること
- ・再起動スクリプト中のコマンドへのPATH が通っていること
- ・再起動される監視対象の起動完了までに時間がかかるため
ccheck が先に監視を再開していないこと

The limit on the container_cnt has been reached.

説明: コンテナの起動個数がコンテナ情報取得量の上限値に達しました。

処置: 「3.3.8 監視対象選択時のコンテナ情報取得量を変更する手順」を参照して、
コンテナ情報取得量を増やしてください。
ccheck の再起動を行ってください。
ccheck の再起動を行っても異常が改善されない場合は、障害解析情報を採取し、
サポートセンターに連絡してください。

The image does not exist or the container of the image is not running. (xxx)

説明: イメージ xxx の起動コンテナはありません。

処置: イメージ xxx が存在するか調査してください。
イメージ xxx が存在するときは、起動コンテナがない原因を調査してください。

The container does not exist or the container is not running. (xxx)

説明: コンテナ xxx は起動していません。

処置: コンテナ xxx が起動しない原因を調査してください。

Container count check error. min_cnt_count. (xxx)

The number of containers is insufficient. (xxx)

説明: イメージ xxx のコンテナの起動数が
min_count で指定された数値を下回ったことを検出しました。

処置: 起動コンテナ数が減少した原因を調査してください。

Container count check error. max_cntr_count. (xxx)

Too many starting containers. (xxx)

説明: イメージ xxx のコンテナの起動数が
max_count で指定された数値を上回ったことを検出しました。

処置: 起動コンテナ数が増加した原因を調査してください。

The container does not exist or the container is not running. (xxx)

説明: コンテナの停止を検出しました。(コンテナxxx)

処置: コンテナが停止した原因を調査してください。

Container information collection or Image name empty. (xxx)

説明: 起動コンテナ数を取得できません。(監視対象名 xxx)

処置: 内部エラーです。ccheck の再起動を行ってください。
ccheck の再起動を行っても異常が改善されない場合は、
障害解析情報を採取し、サポートセンターに連絡してください。

Unable to get container name from docker command execution result.

説明: コンテナ名を取得できません。

処置: 内部エラーです。ccheck の再起動を行ってください。
ccheck の再起動を行っても異常が改善されない場合は、
障害解析情報を採取し、サポートセンターに連絡してください。

Unable to get image name from docker command execution result.

説明: イメージ名を取得できません。

処置: 内部エラーです。ccheck の再起動を行ってください。
ccheck の再起動を行っても異常が改善されない場合は、
障害解析情報を採取し、サポートセンターに連絡してください。

Command execution was unsuccessful.

説明: コンテナ情報取得コマンドの実行に失敗しました。

処置: 内部エラーです。ccheck の再起動を行ってください。
ccheck の再起動を行っても異常が改善されない場合は、
障害解析情報を採取し、サポートセンターに連絡してください。

cent->is_image_name illegal value. (xxx)

説明: コンテナ個別部情報の監視対象フラグの値が異常です。(監視対象名 xxx)

処置: 内部エラーです。ccheck の再起動を行ってください。
ccheck の再起動を行っても異常が改善されない場合は、
障害解析情報を採取し、サポートセンターに連絡してください。

lck_shm_get_cntnr_cnt_by_image fail. (xxx)

説明: コンテナ情報のイメージ xxx のコンテナ数の取得に失敗しました。
処置: 内部エラーです。ccheck の再起動を行ってください。
ccheck の再起動を行っても異常が改善されない場合は、
障害解析情報を採取し、サポートセンターに連絡してください。

lck_shm_get_cntnr_cnt_by_cname fail. (xxx)

説明: コンテナ情報のコンテナ xxx のコンテナ数の取得に失敗しました。
処置: 内部エラーです。ccheck の再起動を行ってください。
ccheck の再起動を行っても異常が改善されない場合は、
障害解析情報を採取し、サポートセンターに連絡してください。

lck_shm_save_cntnr_cnt_by_image fail.

説明: イメージ名別のコンテナ数の保存に失敗しました。
処置: 内部エラーです。ccheck の再起動を行ってください。
ccheck の再起動を行っても異常が改善されない場合は、
障害解析情報を採取し、サポートセンターに連絡してください。

(2) LOG_WARNING

-> time(2) fail (erno=xxx)

説明: 現在時刻の取得に失敗しました。(エラー番号 xxx)
処置: 内部エラーです。ccheck の再起動を行ってください。
ccheck の再起動を行っても異常が改善されない場合は、
障害解析情報を採取し、サポートセンターに連絡してください。

-> waitpid(2) fail (xxx)

説明: 子プロセスの実行中にエラーが発生しました。(エラー番号 xxx)
処置: 内部エラーです。ccheck の再起動を行ってください。
ccheck の再起動を行っても異常が改善されない場合は、
障害解析情報を採取し、サポートセンターに連絡してください。

-> popen(3) fail. erno = xxx

説明: docker コマンドの実行中にエラーが発生しました。(エラー番号 xxx)
処置: 内部エラーです。ccheck の再起動を行ってください。
ccheck の再起動を行っても異常が改善されない場合は、
障害解析情報を採取し、サポートセンターに連絡してください。

Try to restart (xxx)

説明: 再起動を行います。(監視対象名 xxx)
処置: 動作に支障ないため、特に処置の必要はありません。

Restart fail (xxx)

説明: 再起動スクリプトがエラー終了しました。(監視対象名 xxx)
処置: 再起動スクリプトが単体で実行できるか確認してください。
単体で実行できる場合、以下を確認してください。

- ・再起動スクリプトに実行権が付与されていること
- ・再起動スクリプト中のコマンドへのPATH が通っていること
- ・再起動される監視対象の起動完了までに時間がかかるため
ccheck が先に監視を再開していないこと

Get time fail. errno=xxx

- 説明: 再起動シェルのタイムアウト確認時に時刻取得に失敗しました。(エラー番号 xxx)
- 処置: 内部エラーです。ccheck の再起動を行ってください。
ccheck の再起動を行っても異常が改善されない場合は、
障害解析情報を採取し、サポートセンターに連絡してください。

Restart fail (Timeout) (xxx)

- 説明: 監視対象 xxx の再起動スクリプトがタイムアウトしました。(監視対象名 xxx)
- 処置: 再起動スクリプトが単体で実行できるか確認してください。
単体で実行できる場合、以下を確認してください。
- ・ 再起動スクリプトに実行権が付与されていること
 - ・ 再起動スクリプト中のコマンドへのPATH が通っていること
 - ・ 再起動される監視対象の起動完了までに時間がかかるため
ccheck が先に監視を再開していないこと

Illegal return value from check_restart(xxx)

- 説明: 内部エラーが発生しました。(戻り値 xxx)
- 処置: ccheck の再起動を行ってください。
ccheck の再起動を行っても異常が改善されない場合は、
障害解析情報を採取し、サポートセンターに連絡してください。

(image_name=xxx) Down

- 説明: イメージ名監視において監視対象の停止を検出しました。(イメージ名 xxx)
- 処置: 対象イメージのコンテナが停止した原因を調査してください。

(cname=xxx) Down

- 説明: コンテナ名監視において監視対象の停止を検出しました。(コンテナ名 xxx)
- 処置: 対象コンテナが停止した原因を調査してください。

(image_name=xxx) Up

- 説明: イメージ名監視において監視対象の起動を検出しました。(イメージ名 xxx)
- 処置: 運用に支障ないため、処置は特に必要ありません。

(cname=xxx) Up

- 説明: コンテナ名監視において監視対象の起動を検出しました。(コンテナ名 xxx)
- 処置: 運用に支障ないため、処置は特に必要ありません。

-> CCK_PS_RESTART_FAIL (xxx)

- 説明: 再起動に失敗しました。(監視対象名 xxx)
- 処置: 再起動スクリプトが単体で実行できるか確認してください。
単体で実行できる場合、以下を確認してください。
- ・ 再起動スクリプトに実行権が付与されていること
 - ・ 再起動スクリプト中のコマンドへのPATH が通っていること
 - ・ 再起動される監視対象の起動完了までに時間がかかるため
ccheck が先に監視を再開していないこと

-> restart shell fail (exit status) (xxx)

- 説明: 再起動スクリプトが終了ステータス以外で終了したため、再起動に失敗しました。(終了ステータス xxx)
- 処置: 再起動スクリプトが単体で実行できるか確認してください。単体で実行できる場合、以下を確認してください。
- ・再起動スクリプトに実行権が付与されていること
 - ・再起動スクリプト中のコマンドへのPATH が通っていること
 - ・再起動される監視対象の起動完了までに時間がかかるため ccheck が先に監視を再開していないこと

-> restart shell fail (signal accept) (xxx)

- 説明: 再起動スクリプトがシグナルを受信したため、再起動に失敗しました。(シグナル xxx)
- 処置: 再起動スクリプトが単体で実行できるか確認してください。単体で実行できる場合、以下を確認してください。
- ・再起動スクリプトに実行権が付与されていること
 - ・再起動スクリプト中のコマンドへのPATH が通っていること
 - ・再起動される監視対象の起動完了までに時間がかかるため ccheck が先に監視を再開していないこと

-> restart shell fail (another error) (xxx)

- 説明: 予期せぬエラー原因で、再起動シェルの実行が失敗しました。(終了ステータス xxx)
- 処置: 再起動スクリプトが単体で実行できるか確認してください。単体で実行できる場合、以下を確認してください。
- ・再起動スクリプトに実行権が付与されていること
 - ・再起動スクリプト中のコマンドへのPATH が通っていること
 - ・再起動される監視対象の起動完了までに時間がかかるため ccheck が先に監視を再開していないこと

-> waitpid(2) fail (xxx)

- 説明: 再起動シェルの実行に際して子プロセスが異常終了しました。(エラー番号 xxx)
- 処置: 再起動スクリプトが単体で実行できるか確認してください。単体で実行できる場合、以下を確認してください。
- ・再起動スクリプトに実行権が付与されていること
 - ・再起動スクリプト中のコマンドへのPATH が通っていること
 - ・再起動される監視対象の起動完了までに時間がかかるため ccheck が先に監視を再開していないこと

strdup(3) fail (xxx)

- 説明: 再起動スクリプトのパスのコピーに失敗しました。(エラー番号 xxx)
- 処置: 再起動スクリプトが単体で実行できるか確認してください。単体で実行できる場合、以下を確認してください。
- ・再起動スクリプトに実行権が付与されていること
 - ・再起動スクリプト中のコマンドへのPATH が通っていること
 - ・再起動される監視対象の起動完了までに時間がかかるため ccheck が先に監視を再開していないこと

-> execvp(2) fail (xxx)

- 説明: 再起動スクリプトの実行に失敗しました。(エラー番号 xxx)
処置: 再起動スクリプトが単体で実行できるか確認してください。
単体で実行できる場合、以下を確認してください。
- ・再起動スクリプトに実行権が付与されていること
 - ・再起動スクリプト中のコマンドへのPATH が通っていること
 - ・再起動される監視対象の起動完了までに時間がかかるため ccheck が先に監視を再開していないこと

fopen(3) fail. cinfofile = zzz, errno = xxx.

- 説明: コンテナ情報ファイル zzz のオープンに失敗しました。(エラー番号 xxx)
処置: 運用に支障ないため、処置は特に必要ありません。

illegal message

- 説明: 内部エラーが発生しました。
処置: ccheck の再起動を行ってください。
ccheck の再起動を行っても異常が改善されない場合は、
障害解析情報を採取し、サポートセンターに連絡してください。

dumpfile open(2) fail. dumpfile = zzz, errno = xxx.

- 説明: 共有メモリトレースファイル zzz のオープンに失敗したか、
オープン可能なファイル数を超過しました。(エラー番号 xxx)
処置: オープンされている共有メモリトレースファイルがあれば、
クローズを行ってからシステムの再起動を行ってください。
システムの再起動を行っても異常が改善されない場合は、
障害解析情報を採取し、サポートセンターに連絡してください。

write(2) fail to dumpfile. dumpfile = zzz, errno = xxx.

- 説明: 共有メモリトレースファイル zzz の書き込みに失敗しました。(エラー番号 xxx)
処置: 運用に支障ないため、処置は特に必要ありません。

dumpfile close(2) fail. dumpfile = zzz, errno = xxx.

- 説明: 共有メモリトレースファイル zzz のクローズに失敗しました。(エラー番号 xxx)
処置: ccheck の再起動またはシステムの再起動を行ってください。
ccheck の再起動、システムの再起動を行っても異常が改善されない場合は、
障害解析情報を採取し、サポートセンターに連絡してください。

(3) LOG_INFO

Container information collection timeout. cent_id=xxx

- 説明: コンテナ情報の収集がタイムアウトしました。(CENT id xxx)
処置: コンテナ情報を収集するコマンドの実行待ち時間を増やした後、
ccheck の再起動を行ってください。
変更手順の詳細については、「3.3.10 コンテナ情報を収集するコマンドの
最大待ち時間を変更する手順」を参照してください。

Container information collection error. cent_id=xxx

説明: コンテナ情報の収集が異常終了しました。(CENT id xxx)

処置: docker ps コマンドが単体で実行できるか確認してください。

docker ps コマンドが実行できる場合、ccheck の再起動またはシステムの再起動を行ってください。

ccheck の再起動、システムの再起動を行っても異常が改善されない場合は、障害解析情報を採取し、サポートセンターに連絡してください。

ccheck start. (up_cntr_num=xxx, total_cent_num=yyy) cfile=zzz

説明: コンテナ監視を開始します。(起動検知数 xxx, CENT数 yyy, cfile zzz)

処置: 運用に支障ないため、処置は特に必要ありません。

4.1.7 cfile 関連のメッセージ

以下に cfile 関連の異常に関するメッセージの説明、および行うべき処置方法を記載します。

(1) LOG_ERR

cfile init fail. cfile = zzz.

- 説明: cfile 読み込みに失敗しました。(cfile zzz)
処置: cfile をオープンできるか確認してください。
オープンできない場合、以下を確認してください。
ccheck の再起動を行っても異常が改善されない場合は、
障害解析情報を採取し、サポートセンターに連絡してください。

IPCKEY_CCHECK and container_entry must be specified. cfile = zzz.

- 説明: IPC キー、個別部情報 (CENT) の設定がありません。(cfile zzz)
処置: cfile 中の ipc キー、個別部情報 (CENT) の記載を確認してください。
詳細については、「3.1.2 cfile について」を参照してください。

can not load cfile information. cfile = zzz.

- 説明: cfile 中の個別部情報 (CENT) に不正な行があります。(cfile zzz)
処置: 個別部情報 (CENT) を正しく設定してください。

Invalid value: IPCKEY_CCHECK value = xxx.

- 説明: cfile 中の IPCKEY_CCHECK の値 xxx が不正です。
処置: IPCKEY_CCHECK をシステムで一意となる 0x00000001~0x7fffffff の範囲で
正しく設定してください。
また、OS や他製品が使用する共有メモリの key と競合しないように、
ipcs(8) コマンドを使用して確認してください。

Invalid value: MSG_CHECK_INTERVAL value = xxx.

- 説明: cfile 中の MSG_CHECK_INTERVAL の値 xxx が不正です。
処置: MSG_CHECK_INTERVAL を 1 ~ 60*60*24 秒の範囲で正しく設定してください。
また MONITOR_INTERVAL より小さい値を設定してください。

Invalid value: MONITOR_INTERVAL value = xxx.

- 説明: cfile 中の MONITOR_INTERVAL の値 xxx が不正です。
処置: MONITOR_INTERVAL を 1 ~ 60*60*24 秒の範囲で正しく設定してください。
また MSG_CHECK_INTERVAL より大きい値で、
MSG_CHECK_INTERVAL の整数倍の値を設定してください。

SHM_DUMP_FILE length is too long.

- 説明: cfile 中の SHM_DUMP_FILE が指定できる最大長を超えています。
処置: SHM_DUMP_FILE を 256 文字より少ない文字数で正しく設定してください。

UP_MESSAGE_REDUCE_MODE must be specified. enable or disable.

- 説明: cfile 中の UP_MESSAGE_REDUCE_MODE は enable または
disable を指定してください。
処置: UP_MESSAGE_REDUCE_MODE を正しく設定してください。

Invalid key: key = xxx.

説明: cfile 中の共通部情報 (PARAM) に不正なエントリ xxx があります。
処置: 共通部情報 (PARAM) に不正なエントリがないかを確認してください。

Container count max over.

説明: cfile 中の個別部情報 (CENT) 行のエントリ数が最大数を超過しています。
処置: 個別部情報 (CENT) のエントリは 256 行以内で指定してください。

Monitor target name must be specified.

説明: cfile 中の CENT の監視対象名が正しく指定されていません。
処置: 監視対象名は、1024 文字以内で正しく指定してください。

Restart_cmd must be specified.

説明: cfile 中の CENT の再起動スクリプトが正しく指定されていません。
処置: 再起動スクリプトを正しく指定してください。
なお、再起動スクリプトを省略する場合には、 "-" を指定してください。

Restart_cmd length is too long.

説明: cfile 中の CENT の再起動スクリプトの指定が長すぎます。
処置: 再起動スクリプトは絶対パスで 255 文字以内の値を正しく指定してください。

Grace (xxx) must be digit and greater than 0.

説明: cfile 中の CENT の GRACE 値に 0 より小さい値 xxx が指定されています。
処置: GRACE 値は、0 ~ 60*60*24*360 秒の値を正しく指定してください。

Grace (xxx) is illegal parameter.

説明: cfile 中の CENT の GRACE 値 xxx が不正です。
処置: GRACE 値を正しく指定してください。

retry_count_max (xxx) must be digit and greater than 0.

説明: cfile 中の CENT の retry_count_max に 0 より小さい値 xxx が設定されています。
処置: retry_count_max を 0 ~ 1024 回の範囲で正しく指定してください。

retry_count_max (xxx) is illegal parameter.

説明: cfile 中の CENT の retry_count_max の値 xxx が不正です。
処置: retry_count_max を正しく指定してください。

Retry_over_action must be specified.

説明: cfile 中の CENT の retry_over_action が指定されていません。
処置: retry_over_action に CONTINUE、SHUTDOWN、EXIT のいずれかの値を指定してください。

Retry_over_action (%s) is invalid.

説明: cfile 中の CENT の retry_over_action が不正です。
処置: retry_over_action を正しく指定してください。

option length is too long.

説明: cfile 中の CENT のオプションパラメーターに指定できる文字数を超過しています。
処置: オプション情報を正しく指定してください。

option tag is illegal parameter.

説明: cfile 中の CENT のオプションパラメーターが不正です。

処置: オプション情報のキーに不正パラメーターが指定されていないことを確認してください。

image_name_check must be specified. enable or disable.

説明: cfile 中の CENT の image_name_check オプションに値が指定されていません。

処置: image_name_check オプションは enable または disable を指定してください。

image_name_check is enable or disable.

説明: cfile 中の CENT の image_name_check オプションに指定された値が不正です。

処置: image_name_check オプションは enable または disable を指定してください。

Image name length is too long.

説明: cfile 中の CENT の監視対象名に指定したイメージ名が指定できる最大長を超えています。

処置: イメージ名は 1024 文字以内で指定してください。

Container name length is too long.

説明: cfile 中の CENT の監視対象名に指定したコンテナ名が指定できる最大長を超えています。

処置: コンテナ名は 1024 文字以内で指定してください。

min_count must be specified.

説明: cfile 中の CENT の min_count オプションに値が指定されていません。

処置: min_count オプションを 1 ~ 64 の範囲で正しく指定してください。

min_count (xxx) must be digit and greater than 0.

説明: cfile 中の CENT の min_count オプションに指定された値 xxx が不正です。

処置: min_count オプションを 1 ~ 64 の範囲で正しく指定してください。

max_count must be specified.

説明: cfile 中の CENT の max_count オプションに値が指定されていません。

処置: max_count オプションを 1 ~ 64 の範囲で正しく指定してください。

max_count (xxx) must be digit and greater than 0.

説明: cfile 中の CENT の max_count オプションに指定された値 xxx が不正です。

処置: max_count オプションを 1 ~ 64 の範囲で正しく指定してください。

max_count (xxx) must be beyond min_count(yyy).

説明: cfile 中の CENT の max_count オプションの値 xxx より大きな数が min_count オプションの値 yyy に設定されています。

処置: max_count を超えない値を min_count オプションで指定してください。または min_count 以上の値を max_count オプションで指定してください。

specification of min_count is evaluated only when monitoring image name.

説明: イメージ名監視をしていないにもかかわらず min_count を設定しています。

処置: イメージ名監視であれば image_name_check オプションで enable を指定してください。

specification of max_count is evaluated only when monitoring image name.

- 説明: イメージ名監視をしていないにもかかわらず max_count を設定しています。
- 処置: イメージ名監視であれば image_name_check オプションで enable を指定してください。

restart_timeout must be specified.

- 説明: cfile 中の CENT の restart_timeout オプションに値が指定されていません。
- 処置: restart_timeout オプションを 1 ~ 60*60*24 秒の範囲で正しく設定してください。また MONITOR_INTERVAL の整数倍の値を設定してください。

restart_timeout (xxx) must be digit and greater than 0.

- 説明: cfile 中の CENT の restart_timeout オプションの値 xxx が不正です。
- 処置: restart_timeout オプションを 1 ~ 60*60*24 秒の範囲で正しく設定してください。また MONITOR_INTERVAL の整数倍の値を設定してください。

group tag must be specified.

- 説明: cfile 中の CENT の group tag オプションに値がありません。
- 処置: group tag オプションを正しく設定してください。

group tag length is too long.

- 説明: cfile 中の CENT の group tag オプションが長すぎます。
- 処置: group tag オプションは 63 文字以内で指定してください。

option tag is illegal parameter.

- 説明: cfile 中のオプションパラメーターが不正です。
- 処置: オプション情報のキーに不正パラメーターが指定されていないことを確認してください。

(2) LOG_WARNING

fopen(3) fail. cfile = zzz, errno = xxx.

- 説明: cfile (zzz) のオープンに失敗しました。または cfile が存在しません。(エラー番号 xxx)
- 処置: cfile の妥当性を確認してください。

IPCKEY_CCHECK and container_entry must be specified. cfile = zzz.

- 説明: cfile (zzz) は ipc キーまたは CENT が設定されていません。
- 処置: IPCKEY_CCHECK または個別部情報 (CENT) を正しく設定してください。

Invalid line: [xxxxxxxxxxxxxxxx].

- 説明: cfile 中に、不正な行 xxxxxxxxxxxxxxxxxxx があります。
- 処置: cfile の該当する行を正しく設定してください。

5 障害解析情報の採取

本機能運用中に何らかの障害が発生した場合は、下記の手順にしたがって情報採取を行った後、サポートへ問い合わせをお願いします。

5.1 障害解析情報

docker コマンドなどの一部のコマンドでは、実行結果が途切れてしまう可能性があります。情報採取の際は、ウィンドウ幅を最大にさせていただき、下記例のように、別ファイルにリダイレクトしていただき、ファイルを送付してください。

例) # docker ps -a > docker_ps_a.txt

- ソフトウェア関連
 - rpm -qa | grep clusterpro-mc-pscm の実行結果
- 実行環境関連
 - cat /etc/system-release の実行結果
 - cat /proc/cpuinfo の実行結果
 - uname -a の実行結果
 - docker -v の実行結果
 - service docker status の実行結果
 - docker ps -a の実行結果
 - docker images の実行結果
 - export | grep HACM_ の実行結果
 - export | grep LANG の実行結果
- syslog 関連
 - /var/log/messages から始まるものすべて
 - grep -i -E "ccheck|cadmin" /var/log/messages* の実行結果
- core ファイル関連
 - ls / | grep core の実行結果
 - ls /opt/HA/CCHECK/bin/ | grep core の実行結果
上記コマンドで一致したファイルに
 - file core
コマンドを実行して ccheck | cadmin が出力されるもの
- 自動起動関連
 - /etc/systemd/system/<ProcessSaver の service ファイル名>

- cfile 情報関連 (設定ファイル関連)
 - cfile
 - cfile の共通部情報の SHM_DUMP_FILE に記載された共有メモリダンプファイル
 - cfile の個別部情報の shell_path に記載された再起動スクリプトファイル

- cfile 情報関連 (起動中の ccheck 情報のみ収集)
 - /opt/HA/CCHECK/bin/cadmin -l の実行結果
 - /opt/HA/CCHECK/bin/cadmin -f <cfile 名> -c show cent の実行結果
 - /opt/HA/CCHECK/bin/cadmin -f <cfile 名> -c show param の実行結果
 - /opt/HA/CCHECK/bin/cadmin -f <cfile 名> -c show group の実行結果
 - /opt/HA/CCHECK/bin/cadmin -f <cfile 名> -c show env の実行結果
 - /opt/HA/CCHECK/bin/cadmin -f <cfile 名> -c unload < SHM_DUMP_FILE 名> の実行結果

- その他の情報
 - ls -lt /opt/HA/CCHECK/bin の実行結果
 - ps -eaw -ouser,pid,ppid,%cpu,%mem,vsz,rss,tt,stat,stime,cputime,priority,nice,cmd --sort pid の実行結果
 - ps -eLf の実行結果
 - ipcs -a の実行結果
 - ipcs -m の実行結果
 - ipcs -mt の実行結果
 - ipcs -mp の実行結果
 - ipcs -mc の実行結果
 - ipcs -ml の実行結果

6 注意・制限事項

6.1 cadadmin

- cadadmin コマンドは、ccheck コマンドを起動中でない場合は使用できません。
また、ccheck コマンドを kill コマンドなどで強制終了させた後に cadadmin を実行させた場合、異常終了します。
- cadadmin コマンドからの処理要求は、ccheck で管理する共有メモリ上にメッセージを設定することにより動作しますので、共有メモリ上のメッセージ処理間隔 (MSG_CHECK_INTERVAL) の設定値によって、実行に時間差が発生します。
特に、cadadmin コマンドをシェルスクリプトから連続して呼び出す場合は、一定時間の猶予 (MSG_CHECK_INTERVAL の約 2 倍程度) を sleep で与えてください。
- cadadmin コマンドの実行に失敗した場合、エラーメッセージは共有メモリ上のトレースまたはシステムログファイルに出力され、標準エラー出力に出力されない場合があります。
- cadadmin コマンドを一般ユーザーで使用する場合には、ccheck コマンドも同様に一般ユーザーで起動してください。
ccheck を root ユーザー、cadadmin を一般ユーザーで実行すると失敗する場合があります。
- cadadmin コマンドで監視の停止、再開 (stop、start) を実行した場合、システムログに以下のようなメッセージを出力します。

監視の停止時

```
Jun 14 09:58:59 host1 ccheck[23328]:(['監視対象名', cent_id=xxx]) Monitor stop
```

監視の再開時

```
Jun 14 09:58:59 host1 ccheck[23328]:(['監視対象名', cent_id=xxx]) Monitor start
```

- cadadmin コマンドを start オプションで実行時に該当のコンテナが起動していない場合は、指定されたコンテナの監視は再開されません。
その場合、システムログに以下のようなメッセージを出力します。

```
Jun 14 09:58:59 host1 ccheck[23328]: Set container fail (監視対象名)  
Jun 14 09:58:59 host1 ccheck[23328]:(['監視対象名', cent_id=xxx]) Monitor stop
```

- cadadmin コマンドを start オプションで実行時には、cfile の再読み込みは行いません。
cfile の変更後に cfile の再読み込みを行う場合は、reload オプションを使用してください。
- cadadmin コマンドは、ccheck の動作を予約するコマンドのため、複数同時に実行することはできません。
- ccheck 実行中に IPC キーを変更した場合は、reload によって動的に変更することはできません。
ccheck を一度終了し再起動を行う必要があります。

6.2 ccheck

- ひとつの ccheck コマンドで監視できるコンテナ/イメージの最大値は 256 です。これ以上のコンテナ/イメージを監視する場合は、ccheck を複数起動してください。同時に起動できるプロセス数には、システム設定値の制約があります。
- cfile に指定する MONITOR_INTERVAL (監視間隔) は、以下の制約があります。
 - a) MSG_CHECK_INTERVAL (メッセージ処理間隔) 以上の値であること
 - b) MSG_CHECK_INTERVAL の正の整数倍の値であること

正の整数倍の値が設定されていない場合、ccheck コマンドは、正の整数倍の値に変換して実行します。

- cfile の IPCKEY_CCHECK に指定する IPC キーには、サーバー内で一意の値を指定してください。OS や他の製品の使用する IPC キーと重複すると動作を保証できません。
- イメージ名監視で min_count オプションを指定した場合、指定したイメージ上で起動するコンテナが min_count で指定した数を下回ると異常を検出します。
- イメージ名監視で max_count オプションを指定した場合、指定したイメージ上で起動するコンテナが max_count で指定した数を上回ると異常を検出します。
- イメージ名監視をする場合、監視タイミングごとに OS の起動コンテナ情報を検索して指定イメージ上のコンテナ数を集計するため、OS で稼働するコンテナ数が 1000 を越えるような環境では、高負荷状態となる可能性があります。
稼働するコンテナ数が 1000 を越える場合は、必要に応じてコンテナ情報の取得量を環境にあわせて、大きな値に変更してください。
変更手順の詳細については、「3.3.8 監視対象選択時のコンテナ情報取得量を変更する手順」を参照してください。
- イメージ名監視で min_count オプションと max_count オプションを同時に指定する場合、max_count オプションには、min_count オプションの設定値以上の値を設定してください。
- グループ監視で grouptag オプションを指定した場合、システムログに実際に消滅したコンテナの Down メッセージは出力されますが、同一グループの他のコンテナの再起動に伴って再起動されたコンテナについては、Down のメッセージは出力されません。(再起動完了後の Up メッセージは出力されます) また再起動回数 (retry_count) および再起動した累計 (restart_count) も同様に加算されます。
- グループ監視で grouptag オプションを指定する場合、retry_over_action、restart_count 値、grace 値などの各設定値は同一グループ内で同じ値を指定してください。
- グループ監視で grouptag オプションを指定する場合、数字のみの grouptag 名は指定できません。英字のみまたは英数字を組み合わせた方式で指定してください。

(例) 指定可 : grouptag=oracle_group や grouptag=group01
指定不可 : grouptag=1 や grouptag=8 のように数字のみの grouptag 名
- restart_timeout オプションを指定する場合、monitor_interval の正の整数倍の値を指定してください。正の整数倍でない値を設定した場合、monitor_interval の値に切り上げられます。

- cfile のオプション情報に、各オプションを ";" で区切って複数指定する場合、オプションは最大で 6 個まで同時に指定できます。
また、指定できる文字数はオプション情報全体で最大 255 文字以内です。
また、オプション情報に、"=" を指定することはできません。
- 再起動スクリプトを ;(セミコロン) 区切りで複数指定した場合、/bin/sh -c の引数として指定されたスクリプトを実行します。
そのため、/bin/sh -c の引数として指定することのできないスクリプトやコマンドについては指定することができません。
また、記載された構文のまま実行されますので、シェルの構文に従った方式で指定する必要があります。
- 再起動スクリプトを ;(セミコロン) 区切りで複数指定した場合には、最後に指定されたスクリプトの終了ステータスでコンテナの再起動に成功したかどうかを判定します。

6.3 システム構成における注意事項

- ・ 同一名コンテナあるいはイメージを別々の ccheck から個々に監視を行うことはできません。
- ・ 本製品は、Linux のセキュリティ拡張機能である SELinux には対応しておりません。SELinux の機能が有効になっている OS では本製品を使用することができませんので、必ず SELinux の機能を無効にしてください。

7 リファレンス

7.1 cadmin

名称

cadmin - コンテナ監視運用管理コマンド

構文

```
cadmin -f cfilename -c option  
cadmin -l
```

機能説明

コンテナ監視の停止/再開や動的パラメーターの変更を行います。

-l

実行中の ccheck の一覧を表示します。

-f *cfilename*

コンテナ監視の停止/再開や動的パラメーター変更の際に、対象の *cfile* 名を指定します。
cfile 名は 256 文字未満で指定してください。

-c *option*

option には実行する動作をあわせて 1024 文字未満で指定します。
option に指定する動作の規定は以下のとおりです。

start | restart | resume

コンテナ監視の再開を指定します。

stop | suspend

コンテナ監視の停止を指定します。

shutdown

コンテナ監視の終了を指定します。

reload

cfile の動的変更を指定します。
ccheck を終了させることなく、変更した *cfile* の再読み込みや、
フェールオーバー時のコンテナ監視再開を実現できます。

dump [*dumpfile*]

ccheck が管理する共有メモリ情報のファイル出力を指定します。
dumpfile には、出力ファイル名を指定します。
dumpfile 省略時には、*cfile* の SHM_DUMP_FILE で設定しているファイル
に共有メモリ情報を出力します。

unload dumpfilename

dumpfilename ファイル情報の画面表示を指定します。

dumpfilename には共有メモリ情報を出力したファイル名を指定してください。

change msg_check_interval | monitor_interval l_val

共有メモリのメッセージチェック間隔のタイマー値、または、
コンテナ監視間隔のタイマー値の変更を指定します。

l_val には、変更後のタイマー値を 10 進数で指定します。

msg_check_interval の指定値は 1 秒～60*60*24 秒 (24 時間) の範囲です。

monitor_interval の指定値は 1 秒～60*60*24 秒 (24 時間) の範囲です。

msg_check_interval より大きい値で、msg_check_interval の正の整数倍の値を設定してください。

show param

param を指定した場合、以下のようにコンテナ監視の cfile 設定値を表示します。

MSG_CHECK_INTERVAL = 5

MONITOR_INTERVAL = 10

MONITOR_TRY_COUNT = 2

SHM_DUMP_FILE = /var/opt/HA/CCHECK/log/ccheck_dump

CFILE = cfile

MESSAGE_BOX = start

MONITOR_STOP_COUNT = 0

FAIL_CONTAINER_COUNT = 0

ALL_CONTAINER_COUNT = 5

MESSAGE_BOX には以下のような処理中の共有メモリのメッセージ
が表示されます。

メッセージ	説明
start	コンテナ監視の実行
resume	コンテナ監視の実行(start と同じ)
stop	コンテナ監視の停止
suspend	コンテナ監視の停止 (stop と同じ)
shutdown	コンテナ監視の終了
change	一時的な cfile 値の変更
dump	共有メモリ情報のファイル出力

MONITOR_STOP_COUNT には監視停止中のコンテナ数が表示されます。

FAIL_CONTAINER_COUNT にはリトライオーバーとなっているコンテナ数が
表示されます。

ALL_CONTAINER_COUNT には監視中、停止中にかかわらず、
すべての監視コンテナ数が表示されます。

show cent

cent を指定した場合、以下のように cfile に設定した個別監視情報を表示します。

```
monitor_type      = image_name
name              = image05
retry_count       = 1
restart_count     = 1
sts              = AVAIL
retry_over_act    = continue
rerun_time        = Tue Jan 16 11:30:52 2018
container_count  = 2
min_count         = 2
max_count         = 5
group_name        = img_group56
group_sts         = AVAIL
cent_id           = 1
monitor_sts       = on
```

sts には以下のように監視対象の状態が表示されます。

コンテナ監視状態	説明
INIT	初期状態
RESTARTED	コンテナ再開成功
AVAIL	コンテナ正常動作中 (通常状態)
RESTART_FAIL	コンテナ再開失敗
UNAVAIL	コンテナ異常検出
RETRY_OVER	コンテナ再開リトライオーバー
RESTARTING	コンテナ再開開始
UNKNOWN	状態不明

min_count は cfile のオプション部に min_count を指定している場合のみ、max_count は cfile のオプション部に max_count を指定している場合のみ表示されます。

min_count 、max_count の内容については「3.3.2 イメージ名監視 (コンテナ起動数監視) の導入手順」の項を参照してください。

group_name 、group_sts は cfile のオプション部に grouptag を指定している場合のみ表示されます。

group_sts の表示内容については、後述の show group の項を参照してください。

cent_id には、cfile 中の cent の通番が 1 から順に表示されます。

monitor_sts には、以下のように cent 単位の監視状態が表示されます。

コンテナ監視状態	説明
on	監視状態
off	監視停止状態
unknown	状態不明

show group

group を指定した場合、以下のように cfile に設定したグループ単位の監視情報を表示します。

```
group_name      = img_group56
group_sts       = AVAIL
restart_time    = Tue Jan 16 11:30:52 2018
monitor_sts     = on
```

group_sts には以下のように監視対象の状態が表示されます。

コンテナ監視状態	説明
INIT	初期状態
RESTARTED	コンテナ再開成功
AVAIL	コンテナ正常動作中 (通常状態)
RESTART_FAIL	コンテナ再開失敗
UNAVAIL	コンテナ異常検出
RETRY_OVER	コンテナ再開リトライオーバー
RESTARTING	コンテナ再開開始
UNKNOWN	状態不明

monitor_sts には、以下のようにグループ単位の監視状態が表示されます。

コンテナ監視状態	説明
on	監視状態
off	監視停止状態
unknown	状態不明

※ grouptag 指定されたグループの中で、監視状態のコンテナが一つでも存在する場合、monitor_sts には、on が表示されます。

show env

env を指定した場合、以下のように ccheck 実行時の環境変数の状態を表示します。

```
HACM_CHILDPROC_KILL      = off
HACM_SILENT_MODE         = off
HACM_PENDING_TIME        = 60
HACM_GETCNTNR_COUNT     = 1000
HACM_GETCNTNR_RETRY_COUNT = 6
HACM_DCKCMD_TIMEOUT      = 10
```

HACM_CHILDPROC_KILL が on の場合は、shutdown 実行時に ccheck の子プロセスを kill します。

HACM_SILENT_MODE が on の場合は、サイレントモード on の状態です。

HACM_PENDING_TIME は、ccheck 起動時の自動待ち合わせ時間を表示します。

デフォルトは、60 (秒) で、環境変数 HACM_PENDING_TIME を指定することで 0 ~ 60*60*24 (1 日) の範囲で変更することが可能です。

HACM_GETCNTNR_COUNT は、監視対象選択時のコンテナ情報取得量を表示します。

デフォルトは、1000 (個) で、環境変数 HACM_GETCNTNR_COUNT を指定することで 1 ~ 100000 の範囲で変更することが可能です。

HACM_GETCNTNR_RETRY_COUNT は、監視対象選択時のリトライ回数を表示します。

デフォルトは、6 (回) で、環境変数 HACM_GETCNTNR_RETRY_COUNT を指定することで 1 ~ 100 の範囲で変更することが可能です。

HACM_DCKCMD_TIMEOUT は、コンテナ情報を収集するコマンドを発行してから結果を取得するまでの最大待ち時間を表示します。

デフォルトは、10 (秒) で、環境変数 HACM_DCKCMD_TIMEOUT を指定することで 1 ~ 60*60*24 (1 日) の範囲で変更することが可能です。

環境変数を設定する手順の詳細については、

「3.3.4 サイレントモードでの運用手順」、

「3.3.5 ccheck 終了時に子プロセスを回収する手順」、

「3.3.6 ccheck 起動時の自動待ち合わせ時間を変更する手順」、

「3.3.8 監視対象選択時のコンテナ情報取得量を変更する手順」、

「3.3.9 監視対象選択時のリトライ回数を変更する手順」、

「3.3.10 コンテナ情報を収集するコマンドの最大待ち時間を変更する手順」を参照してください。

-h

Usage の表示を行います。

終了ステータス

成功すると 0 を返し、失敗するとそれ以外を返します。

注意事項

- ・ 本コマンドは、ccheck で設定された共有メモリ領域を参照するため、ccheck コマンド実行中でない場合は、使用できません。
- ・ 本コマンドは、ccheck の動作を予約するコマンドのため複数同時に実行はできません。
- ・ オプションを指定する場合、-c オプションは最後に指定してください。
- ・ 停止したコンテナ監視の再開を行う場合には、-c reload オプションを使用して cfile の再読み込みを行ってください。

使用例

cfile で監視中のコンテナ/イメージについて監視の停止を行います。

```
# /opt/HA/CCHECK/bin/cadmin -f <cfile 名> -c stop
```

cfile で監視中のコンテナ/イメージについて、cent 単位に監視の停止を行います。

※ 以下の例では、cfile で設定している cent について、

grouptag が img_group56 の cent と cent_id が 3 と 4 の cent の監視の停止を行います。

```
# /opt/HA/CCHECK/bin/cadmin -f <cfile 名> -c stop img_group56 3 4
```

cfile の再読み込みを行います。

※ stop オプションなどで停止した cfile の再開を行う場合には、必ず reload を行ってください。

```
# /opt/HA/CCHECK/bin/cadmin -f <cfile 名> -c reload
```

cfile で設定している監視対象について監視の再開を行います。

```
# /opt/HA/CCHECK/bin/cadmin -f <cfile 名> -c start
```

cfile で設定している監視対象について、cent 単位に監視の再開を行います。

※ 以下の例では、cfile で設定している cent について、

grouptag が img_group56 の cent と cent_id が 3 と 4 の cent の監視の再開を行います。

```
# /opt/HA/CCHECK/bin/cadmin -f <cfile 名> -c start img_group56 3 4
```

cfile で監視中のコンテナ/イメージについて設定値を表示します。

```
# /opt/HA/CCHECK/bin/cadmin -f <cfile 名> -c show param
```

関連項目

「7.2 ccheck」

7.2 ccheck

名称

ccheck - コンテナの監視および再開コマンド

構文

```
ccheck -f cfilename
ccheck -f cfilename [-t]
ccheck -f cfilename [-w wait_time]
```

機能説明

入力で指定された *cfile* に基づいて、コンテナの監視、および障害発生時のコンテナの再開処理を行います。

- f *cfilename* コンテナの監視、および障害発生時のコンテナの再開処理を行います。
引数には、コンテナ監視/再開について設定した *cfile* 名を指定します。
cfile 名は 256 文字未満で指定してください。

- w *wait_time* コンテナの監視を始める前に一定の猶予時間を与えます。
引数には、コンテナ監視を開始するまでの待ち合わせ時間を秒単位で指定し、
指定値は 1 秒以上が有効です。
ccheck 起動直後に、対象コンテナが起動されていないケースを考慮し、
コンテナ起動に必要な時間を指定することで無駄な停止検出を防止できます。

- t デバッグモードでの運用を指定します。
標準出力に実行履歴を出力します。

cfile の設定値については、「3.1.2 *cfile* ファイルについて」を参照してください。

終了ステータス

成功すると 0 を返し、失敗するとそれ以外を返します。

注意事項

- ・ 本コマンドを終了する場合、kill(1) コマンドを使用しないでください。
本コマンドを終了する場合には、`cadmin` コマンドを使用してください。

```
# /opt/HA/CCHECK/bin/cadmin -f <cfile 名> -c shutdown
```

- ・ 本コマンドは、コンテナ再開時に、コンテナ再起動スクリプトの終了を待ち合わせます。
そのため、コンテナ再起動スクリプト内で無限ループとなったり、起動コンテナの停止を待ち合わせたり
すると、コンテナ監視が効果的に作用しない場合があります。

- ・ 対象コンテナを再開する場合、依存関係のあるイメージ (同名イメージなど) が存在すると、コンテナの
再開および監視が正常に行われない場合があります。
コンテナ再起動スクリプトにおいて、対象コンテナ再開時に、依存関係のあるコンテナの再設定を行って
ください。

- ・ 本コマンドで監視可能な最大 `cent` 数は、256 です。

使用例

cfile に設定したコンテナ/イメージの監視を行います。

```
#/opt/HA/CCHECK/bin/ccheck -f <cfile 名> &
```

cfile に設定したコンテナ/イメージの監視を、30 秒後に開始します。

```
#/opt/HA/CCHECK/bin/ccheck -f <cfile 名> -w 30 &
```

cfile では tiny_newton というコンテナの監視を行うように以下の設定をします。

```
##### PARAM #####  
IPCKEY_CCHECK      0x1f000001  
MSG_CHECK_INTERVAL 5  
MONITOR_INTERVAL  10  
SHM_DUMP_FILE      /var/opt/HA/CCHECK/log/ccheck_dump  
  
##### CENT(container_name = tiny_newton) #####  
tiny_newton::-0:0:shutdown
```

関連項目

「7.1 cadmin」

CLUSTERPRO
MC ProcessSaver 2.4
ユーザーズガイド
(コンテナ監視機能)

2019年4月第3版
日本電気株式会社
東京都港区芝五丁目7番地1号
TEL (03) 3454-1111(代表)

© NEC Corporation 2019

日本電気株式会社の許可なく複製、改変などを行うことはできません。
本書の内容に関しては将来予告なしに変更することがあります。

保護用紙