

# JST-CREST

## 研究領域

「実用化を目指した組込みシステム用ディペンダブル・オペレーティングシステム」

## DEOS プロジェクト



## D-RE コマンド仕様書

Version 1.02

2013/09/01

DEOS 研究開発センター

## 目次

---

1. dre-app.....	3
2. dre-sys.....	7
3. dre-cgroup.....	12

## 変更履歴

---

Version	変更内容	変更者	日付
1.01	誤表記を修正	Dependable Embedded OS R&D Center	2013.Jul.15
1.02	誤表記、記載漏れを修正	Dependable Embedded OS R&D Center	2013.Sep.1

# 1. dre-app

---

## NAME

dre-app - アプリケーションコンテナの操作を行う

## SYNOPSIS

dre-app command [command options]

## DESCRIPTION

アプリケーションコンテナの操作を行います。

## COMMANDS

```
create -n <name> [-b|--base <base_image>] [-veth [-ip|--ipv4 <address>]] [-p|--port <port>] [-c|--cggroup <cggroup>] [-e|--exclude <path> [<mount_point>]] [-e|--exclude <path> [<mount_point>]] ...
```

アプリケーションコンテナ <name> を生成します。別途、以下のオプションを指定可能です。

*-b|--base <base\_image>*

アプリケーションコンテナのベースとなるイメージ(以下ベースイメージ) <base\_image> を指定します。省略した場合は、デフォルトのものを使用します。

*-veth|--veth*

ホスト OS とアプリケーションコンテナの通信に使用する、仮想ネットワークインタフェースを生成します。このオプションを使用することで、アプリケーションコンテナにホスト OS と異なる IP アドレスを設定可能になります。

*-ip|--ip|--ipv4 <address>*

アプリケーションコンテナの IP アドレスを <address> に設定します。省略した場合は、アプリケーションコンテナの IP アドレスは動的に割り当てられます。-veth オプションと併用する必要があります。

*-p|--port <port>*

アプリケーションコンテナ上で動作する ssh が使用するポート番号を <port> に設定します。省略した場合は、デフォルト値(2200)が設定されます。-veth オプションとの併用はできません。

*-c|--cggroup <cggroup>*

アプリケーションコンテナ <name> 上に、コントロールグループ <cggroup> を生成します。

*-e|--exclude <path> [<mount\_point>]*

コンテナ上の <path> をホスト OS 上の <mount\_point> にマウントします。このオプションで指定したパスは、snapshot 機能で保存・復元されません。

<mount\_point> を省略した場合、<path> は OS 上の同一パスを参照します。

!!! 注意 !!!

ファイルが破損する恐れがあるため、ホスト OS 上の同一パスを複数のコンテナから同時に参照しないで下さい。

---

*start -n <name>*

アプリケーションコンテナ <name> を起動します。

---

*console -n <name>*

アプリケーションコンテナ <name> にシリアル接続します。このオプションを使用する場合、アプリケーションコンテナは事前に起動しておく必要があります。キーボードから <Ctrl+a q> の順に入力すると、切断されます。

---

*stop -n <name>*

動作中のアプリケーションコンテナ <name> を停止します。

---

*suspend -n <name>*

動作中のアプリケーションコンテナ <name> を一時停止します。

---

*resume -n <name>*

一時停止中のアプリケーションコンテナ <name> の動作を再開します。

---

*checkpoint -n <name> -t <tag>*

アプリケーションコンテナ <name> のチェックポイント <tag> を作成します(未実装)。

---

*restart -n <name> -t <tag>*

アプリケーションコンテナ <name> を、チェックポイント <tag> から再開します(未実装)。

---

*save -n <name> -t <tag>*

アプリケーションコンテナ <name> のスナップショット <tag> を作成します。

---

*load -n <name> -t <tag>*

アプリケーションコンテナ <name> を、スナップショット <tag> で復元します。

---

*remove -b|--base <base>*

ベースイメージ <base> を削除します。

---

*remove -n <name> --snapshot -t <tag>*

アプリケーションコンテナ <name> のスナップショット <tag> を削除します。

---

*commit -n <name> -t <tag> -b <base\_image>*

アプリケーションコンテナ <name> のスナップショット <tag> を使用して、新たなベースイメージ <base> を作成します。

---

***status -n <name>***

---

アプリケーションコンテナ <name> の状態を取得します。

---

***getpid -n <name>***

---

アプリケーションコンテナ <name> のプロセス ID を取得します。

---

***register -n <name> {-pid|-vpid} <pid>***

---

アプリケーションコンテナ <name> 上で動作するプロセスを、/proc/daware に登録します。<pid> は、「OS から見た場合」、または「アプリケーションコンテナから見た場合」のいずれかで指定します。<pid> の指定は、それぞれ以下のオプションを使用します。

- pid OS から見た場合のプロセス ID
- vpid アプリケーションコンテナから見た場合のプロセス ID

---

***unregister -n <name> {-pid|-vpid} <pid>***

---

アプリケーションコンテナ <name> 上で動作するプロセスを、/proc/daware から削除します。<pid> は、「OS から見た場合」、または「アプリケーションコンテナから見た場合」のいずれかで指定します。<pid> の指定は、それぞれ以下のオプションを使用します。

- pid OS から見た場合のプロセス ID
- vpid アプリケーションコンテナから見た場合のプロセス ID

---

***pidlist -n <name>***

---

/proc/daware に登録されたプロセスのうち、-n で指定したアプリケーションコンテナ上のプロセスの PID の一覧を取得します。

---

***destroy -n <name>***

---

アプリケーションコンテナ <name> を破棄します。破棄する前に、アプリケーションコンテナを停止する必要があります。別途、以下のオプションを指定可能です。

- c|-g|--cgroup <cgroup>  
アプリケーションコンテナ <name> 上のコントロールグループ <cgroup> を破棄します。

---

***search -n <name>***

---

アプリケーションコンテナ <name> が存在するか確認します。

---

***list -b|--base***

---

システム上に存在するベースイメージの一覧を取得します。

---

***list [--checkpoint|--snapshot] -n <name>***

---

システム上に存在するアプリケーションコンテナの一覧を取得します。別途、以下のオプションを指定可能です。

`{--checkpoint|--snapshot} -n <name>`

<name>上に存在するチェックポイントまたはスナップショットの一覧を取得します。(チェックポイントは未実装。)

---

`bind -n <name> -g <cgroup> -pid|--pid <pid>`

アプリケーションコンテナ <name> 上のプロセス <pid> を、コントロールグループ <cgroup> にバインドします。

---

`unbind -n <name> -pid|--pid <pid>`

bind で <cgroup> にバインドされたプロセスをアンバインドします。

---

`usage [args]`

dre-cgroup usage [args] をコールします。詳細は、dre-cgroup の項目を参照して下さい。

---

`limit [args]`

dre-cgroup limit [args] をコールします。詳細は、dre-cgroup の項目を参照して下さい。

---

`unlimit [args]`

dre-cgroup unlimit [args] をコールします。詳細は、dre-cgroup の項目を参照して下さい。

---

`cgroup -n <name> [args]`

lxc-cgroup -n <name> [args] をコールします。詳細は、"man lxc-cgroup"を参照して下さい。

---

`info -n <name>`

アプリケーションコンテナ <name> の設定情報を表示します。

---

`help`

ヘルプを表示します。

## 2. dre-sys

---

### NAME

dre-sys – システムコンテナの操作を行う

### SYNOPSIS

```
dre-sys command [command options]
dre-sys command [command options] [-a <app_container>]
```

### DESCRIPTION

システムコンテナの操作を行います。-a <app\_container> オプションを指定した場合は、システムコンテナ上で dre-app command [command options] -n <app\_container> をコールします。(詳細は、dre-app の項目を参照して下さい。)

### COMMANDS

```
create -n <name> [-b|--base <base_image>] [-f|--rootfs <rootfs>] [-ip|--ipv4 <ip_address>]
[-mask|--netmask <mask>] [-gw|--gateway <address>] [-dns|--dns <address>] [-m|--ram
<memory>] [-if|--if <interface>] [-c|--cgroupp <cgroupp>]
```

システムコンテナ <name> を生成します。ゲスト OS のホスト名は、システムコンテナと同名になります。このコマンドで作成されたゲスト OS は、初期状態ではユーザ名 "dre" / パスワード "dre" でログイン可能です。別途、以下のオプションを指定可能です。

- b|--base <base\_image>**  
システムコンテナのベースとなる kvm イメージ(以下ベースイメージ)を指定します。省略した場合は、デフォルトのものを使用します。
- f|--rootfs <rootfs>**  
kvm イメージの代わりに、別パーティションにインストールされている OS をゲスト OS として使用します。  
-b オプションとの併用は出来ません。
- ip|--ipv4 <address>**  
システムコンテナの IP アドレスを指定します。省略した場合は、システムコンテナの IP アドレスは動的に割り当てられます。
- mask|--netmask <mask>**  
システムコンテナが使用するサブネットマスクを指定します。このオプションは、-ip オプションと同時に使用する必要があります。-ip オプションを使用し、このオプションを省略した場合は、デフォルト値 (255.255.255.0) が使用されます。
- gw|--gateway <address>**  
システムコンテナが使用するデフォルトゲートウェイのアドレスを指定します。このオプションは、-ip オプションと同時に使用する必要があります。-ip オプションを使用し、このオプションを省略した場合は、デフォルト値(192.168.1.1)が使用されます。
- dns|--dns <address>**  
システムコンテナが使用する DNS サーバのアドレスを指定します。このオプションは、-ip オプションと同時に使用する必要があります。-ip オプションを使用し、このオプションを省略した場合は、デフォルト値(192.168.1.1)が使用されます。

`-m|--ram <memory>`

システムコンテナに割り当てるメモリの容量を MB 単位で指定します。このオプションは、`-ip` オプションと同時に使用する必要があります。省略した場合は、デフォルト値(512)が使用されます。

`-if|--if <interface>`

システムコンテナが使用するブリッジインタフェースを指定します。省略した場合は、デフォルト値(br0)が使用されます。

`-c|--cgroup <cgroup>`

システムコンテナ `<name>` 上に、コントロールグループ `<cgroup>` を生成します。

---

### `start -n <name>`

システムコンテナ `<name>` を起動します。

---

### `stop -n <name>`

動作中のシステムコンテナ `<name>` を停止します。このオプションを実行すると、ゲスト OS がシャットダウンされます。

---

### `suspend -n <name>`

動作中のシステムコンテナ `<name>` を一時停止します。このオプションを使用するためには、対象のシステムコンテナが動作中である必要があります。

---

### `resume -n <name>`

一時停止中のシステムコンテナ `<name>` の動作を再開します。このオプションを使用するためには、対象のシステムコンテナが一時停止中である必要があります。

---

### `checkpoint -n <name> -t <tag>`

システムコンテナ `<name>` のチェックポイント `<tag>` を作成します。

---

### `restart -n <name> -t <tag>`

システムコンテナ `<name>` を、チェックポイント `<tag>` から再開します。

---

### `save -n <name> -t <tag>`

システムコンテナ `<name>` のスナップショット `<tag>` を作成します。

---

### `load -n <name> -t <tag>`

システムコンテナ `<name>` を、スナップショット `<tag>` で復元します。

---

### `remove {checkpoint|snapshot} -n <name> -t <tag>`

システムコンテナ<name> のチェックポイント <tag> またはスナップショット <tag> を削除します。

---

### *commit -n <name> -t <tag> -b <base\_image>*

システムコンテナ <name> のスナップショット <tag> を使用して、新たなベースイメージ <base\_image> を作成します。

---

### *status -n <name>*

システムコンテナ <name> の状態を取得します。

---

### *getpid -n <name>*

システムコンテナ <name> のプロセス ID を取得します。

---

### *share -n <name>*

システムコンテナ <name> を、NFS サーバと共有します。<name> が NFS サーバ上にはない場合は、ローカルのシステムコンテナを NFS サーバにアップロードします。<name> が既に NFS サーバ上に存在する場合は、NFS サーバ上にあるシステムコンテナをローカルで使用可能にします。

---

### *unshare -n <name>*

システムコンテナ <name> の NFS サーバとの共有を解除します。

---

### *immigrate -n <name> -p <port>*

システムコンテナ <name> のマイグレーションを行うため、マイグレーション先となる仮想マシンの動作を開始します。マイグレーションを行うためには、<name> は予め NFS サーバと共有されている必要があります ("dre-sys share" の項目を参照)。マイグレーションを行うと、マイグレーション元の <name> と同一の環境が、当該仮想マシン上に生成されます。マイグレーション完了後、マイグレーション元の <name> は破棄されず。マイグレーション時の通信に、ポート番号 <port> を使用します。

(注: マイグレーションを行う場合、マイグレーション元とマイグレーション先で同一のベースイメージを使用している必要があります。)

---

### *migrate -n <name> -d <target> -p <port>*

システムコンテナ <name> のマイグレーションを行います。マイグレーションを行うためには、<name> は予め NFS サーバと共有されている必要があります ("dre-sys share" の項目を参照)。マイグレーションを行うと、マイグレーション元の <name> と同一の環境が、ホスト <target> で動作している仮想マシン上に生成されます (immigrate の項目を参照)。マイグレーション完了後、マイグレーション元の <name> は破棄されます。マイグレーション時の通信に、ポート番号 <port> を使用します。

(注: マイグレーションを行う場合、マイグレーション元とマイグレーション先で同一のベースイメージを使用している必要があります。)

---

### *kill -n <name>*

システムコンテナ <name> を強制終了します。

---

### *destroy -n <name>*

システムコンテナ *<name>* を破棄します。破棄する前に、対象のシステムコンテナを停止する必要があります。別途、以下のオプションを指定可能です。

*-c|-g|--cgroup <cgroup>*  
システムコンテナ *<name>* 上のコントロールグループ *<cgroup>* を破棄します。

---

### *search -n <name>*

システムコンテナ *<name>* が存在するか確認します。

---

### *list [--checkpoint|--snapshot] -n <name>*

システム上に存在するシステムコンテナの一覧を取得します。別途、以下のオプションを指定可能です。

*{--checkpoint|--snapshot} -n <name>*  
*<name>* 上に存在するチェックポイントまたはスナップショットの一覧を取得します。

---

### *bind -n <name> -g <cgroup> -pid|--pid <pid>*

システムコンテナ *<name>* 上のプロセス *<pid>* を、コントロールグループ *<cgroup>* にバインドします。

---

### *unbind -n <name> -pid|--pid <pid> -g <cgroup>*

*bind* で *<cgroup>* にバインドされたプロセスをアンバインドします。

---

### *usage [args]*

*dre-cgroup usage [args]* をコールします。詳細は、*dre-cgroup.txt* を参照して下さい。

---

### *limit [args]*

*dre-cgroup limit [args]* をコールします。詳細は、*dre-cgroup.txt* を参照して下さい。

---

### *unlimit [args]*

*dre-cgroup unlimit [args]* をコールします。詳細は、*dre-cgroup.txt* を参照して下さい。

---

### *info -n <name>*

システムコンテナ *<name>* の設定情報を表示します。

---

### *help*

ヘルプを表示します。



## 3. dre-cgroup

---

### NAME

dre-cgroup - cgroup によるリソース制御を行う

### SYNOPSIS

dre-cgroup command [command options]

### DESCRIPTION

任意のプロセスをグループ化し、グループ単位でリソースの制御を行います。

### COMMANDS

---

#### *create -n <cgroup>*

コントロールグループ <cgroup> を生成します。

---

#### *destroy -n <cgroup>*

コントロールグループ <cgroup> を破棄します。

---

#### *search -n <cgroup>*

コントロールグループ <cgroup> が存在するか確認します。

---

#### *list*

コントロールグループの一覧を取得します。

---

#### *cgroup -n <cgroup> [args]*

lxc-cgroup -n <cgroup> [args] をコールします。詳細は、"man lxc-cgroup"を参照して下さい。

---

#### *bind -n <cgroup> -p <pid>*

プロセス ID が <pid> のプロセスを、コントロールグループ <cgroup> に追加します。

---

#### *usage --cpu|--core|--mem|--memsw [-n <cgroup>] [-pid|--pid <pid>]*

リソースの使用量を取得します。リソースは、以下を指定可能です。

--cpu CPU 使用率(%)  
--core CPU コア  
--mem メモリ使用量(bytes)  
--memsw メモリとスワップ領域の使用量(bytes)

別途、以下のオプションを指定可能です。

-n <cgroup>  
コントロールグループ<cgroup>のリソース使用量を取得する。

`-pid/--pid <pid>`  
PID が<pid>のプロセスのリソース使用量を取得する。

### `limit --cpu/--core/--mem/--memsw -n <cgroup> <limitation>`

---

コントロールグループ <cgroup> が使用するリソースの使用量を <limitation> に制限します。リソースは、以下を指定可能です。

`--cpu` CPU 使用率(%)。<limitation>は整数値を指定する。  
`--core` CPU コア。<limitation>の指定方法は、cgroup における cpuset.cpus の書式に準ずる。  
`--mem` メモリ使用量(bytes)。<limitation>は整数値を指定する。  
`--memsw` メモリとスワップ領域の使用量(bytes)。<limitation>は整数値を指定する。

### `unlimit --cpu/--core/--mem/--memsw -n <cgroup>`

---

コントロールグループ <cgroup> が使用するリソースの使用量の制限を解除します。リソースは、以下を指定可能です。

`--cpu` CPU 使用率(%)  
`--core` CPU コア  
`--mem` メモリ使用量(bytes)  
`--memsw` メモリとスワップ領域の使用量(bytes)

### `help`

---

ヘルプを表示します。