

Fujitsu Server
PRIMERGY CDI V1.0
ソフトウェア管理者ガイド
- Composable Disaggregated Infrastructure -



はじめに

本書は Fujitsu PRIMERGY CDI V1.0 で使用される富士通製ソフトウェア Compose Manager for CDI について説明しています。本書を読むにあたっては、Linux に関する基本的な知識が必要になります。
なお、本書は Fujitsu PRIMERGY CDI V1.0 以外では利用できません。

略称／用語

本書で使用する略称と用語を以下に示します。

正式名称	本書での略称
Composable Disaggregated Infrastructure	CDI
PRIMERGY Composable Disaggregated Infrastructure	PRIMERGY CDI
Compose Manager for CDI	CDI 管理ソフトウェア
Command Line Interface	CLI
Fujitsu Server PRIMERGY CDI V1.0 システム構築手順書	システム構築手順書
Fujitsu Server PRIMERGY CDI V1.0 システム運用管理者ガイド	システム運用管理者ガイド
CDI コントローラアプライアンス	Director
PCIe ファブリックスイッチ (48port) for CDI	PCIe スイッチ
PCIe Box (PCIe×8) for CDI	PCIe Box
計算サーバ (PRIMERGY RX2530 M7、PRIMERGY RX2540 M7)	計算サーバ
PCIe SSD-960GB (RI) ×8 for CDI PCIe SSD-800GB (MU) ×8 for CDI	NVMe SSD

関連ドキュメント

関連するドキュメントを以下に示します。必要に応じて参照してください。

ドキュメント	概要	入手方法
システム構築手順書	システムの運用前に実施する構築手順について記載しています。	富士通 CDI 製品情報サイトから入手できます。
システム運用管理者ガイド	システムを運用するために必要な操作について記載しています。	富士通 CDI 製品情報サイトから入手できます。

富士通 CDI 製品情報サイト

<https://www.fujitsu.com/jp/products/computing/servers/primergy/solution/cdi/>

改版履歴

版数	発行月	変更箇所	内容
初版	2023年7月	—	初版発行
1.1版	2023年11月	全体 第6章 2.5.3, 2.5.7	誤記を修正 Director 管理コマンドリファレンスを追加 P2P 関連の説明を追記
1.2版	2023年11月	2.5.7	体裁を修正
1.3版	2024年1月	2.4.1 2.5.5	-E の説明にエラー状態のリソースの定義を追加 電源起動状態でリソース追加・削除時の動作説明を追記
1.4版	2024年1月	3.4.2 3.4.3 7.2	PCIe 10 slot Box に対応
1.5版	2024年3月	2.8.2 2.5.5	誤記を修正(system log) 電源起動状態でリソース追加・削除時の動作説明を追記

目次

第 1 章 ソフトウェアの説明	8
1.1 用語について	8
1.2 CDI 管理ソフトウェアの機能	9
1.3 CDI 管理ソフトウェアの構成	10
第 2 章 CLI コマンド リファレンス	11
2.1 コマンドの形式	11
2.2 コマンドの種類	11
2.3 コマンドの説明内容	12
2.4 リソースの操作	13
2.4.1 resource list.....	13
2.4.2 resource show.....	16
2.4.3 resource sync.....	18
2.4.4 resource set_op_status.....	19
2.5 論理サーバの操作	20
2.5.1 machine list.....	20
2.5.2 machine create.....	22
2.5.3 machine show	24
2.5.4 machine destroy	26
2.5.5 machine modify.....	27
2.5.6 machine power.....	29
2.5.7 machine p2p.....	30
2.6 グループの操作.....	31
2.6.1 group list.....	31
2.6.2 group create	33
2.6.3 group show	35
2.6.4 group destroy.....	37

- 2.6.5 group modify38
- 2.7 イメージの操作..... 39
 - 2.7.1 image list.....39
 - 2.7.2 image add.....40
 - 2.7.3 image show.....41
 - 2.7.4 image remove.....42
 - 2.7.5 resource install43
- 2.8 その他の操作 44
 - 2.8.1 system dump.....44
 - 2.8.2 system log46
 - 2.8.3 resource erase.....47
 - 2.8.4 resource bmc.....48
 - 2.8.5 jbox add49
 - 2.8.6 jboxspecs add50
 - 2.8.7 spec list51
 - 2.8.8 spec add.....53
 - 2.8.9 spec remove54
 - 2.8.10 login.....55
- 2.9 YAML ファイル形式 56
 - 2.9.1 フォーマット56
 - 2.9.2 condition の指定方法57
- 第 3 章 PCIe スイッチ制御コマンドリファレンス..... 58**
 - 3.1 コマンドの形式 58
 - 3.2 コマンドの種類..... 58
 - 3.3 コマンドの説明内容 58
 - 3.4 PCIe スイッチの操作 59
 - 3.4.1 swcfg get.....59
 - 3.4.2 swcfg set60

3.4.3 swcfg check.....	61
第 4 章 Director 制御コマンドリファレンス	62
4.1 コマンドの形式.....	62
4.2 コマンドの種類.....	62
4.3 コマンドの説明内容	62
4.4 Director の操作	63
4.4.1 fabrctl state.....	63
4.4.2 fabrctl shutdown.....	64
4.4.3 fabrctl reset.....	65
4.4.4 fabrctl restart.....	66
4.4.5 fabrctl upload	67
4.4.6 fabrctl upgrade.....	68
4.4.7 fabrctl config.....	69
第 5 章 fpclean コマンドリファレンス.....	70
5.1 コマンドの形式.....	70
5.2 コマンドの説明内容	70
5.3 Director の情報消去	71
5.3.1 fpclean.....	71
第 6 章 Director 管理コマンドリファレンス	72
6.1 コマンドの形式.....	72
6.2 コマンドの種類.....	72
6.3 コマンドの説明内容	72
6.4 Director の管理	73
6.4.1 fabradm shutdown	73
6.4.2 fabradm restart.....	74
6.4.3 fabradm linkloss.....	75
6.4.4 fabradm fpclean.....	76

第7章 メッセージ	77
7.1 CLI コマンドのメッセージ一覧.....	77
7.2 swcfg コマンドのメッセージ一覧.....	81
7.3 fabrctl コマンドのメッセージ一覧.....	85
7.4 fpclean コマンドのメッセージ一覧.....	89
7.5 fabradm コマンドのメッセージ一覧.....	90

第1章 ソフトウェアの説明

本章では、Fujitsu PRIMERGY CDI V1.0 システム（以下システムと表記）で使用する CDI 管理ソフトウェアについて説明します。

1.1 用語について

本書の説明で使用している主な用語について以下に記載します。必要に応じて参照してください。

用語	説明
リソース	計算サーバ, GPU, NVMe SSD, NIC ※システム構築手順書を参照
論理サーバ	リソースを組み合わせる論理空間に作成された論理的なサーバ
グループ	論理サーバ, リソース (未使用のリソースも含む) を要素として持つ
ファブリック	リソースおよびその論理サーバへの割当・解除を一元的に管理するもの
ファブリック フリープール	グループに割り当てられていないリソースのプール
グループ フリープール	グループ内の論理サーバに割り当てられていないリソースのプール
スベックテーブル	リソースごとのスベックを記載したテーブル
JBOX テーブル	リソースの物理位置 (ラック位置) などの情報を記載したテーブル

1.2 CDI 管理ソフトウェアの機能

CDI 管理ソフトウェアは、システム性能の最適化を行うためにシステム内のリソース（GPU、NIC など）の管理を行います。以下に CDI 管理ソフトウェアの機能を記載します。

①リソースの操作

リソース情報の表示など。

②論理サーバの操作

論理サーバの作成・解体や、論理サーバへのリソースの追加・削除など。

③グループの操作

グループの作成・解体や、グループへのリソースの追加・削除など。

④イメージの操作

OS イメージの登録・削除、インストールなど。

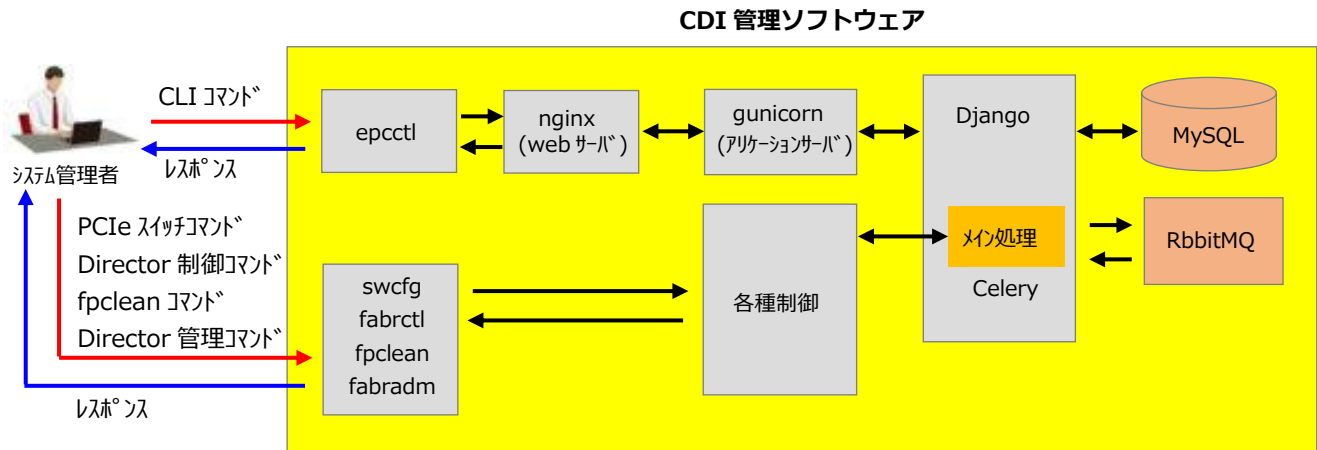
⑤その他の操作

グループ、論理サーバのバックアップや、制御情報の追加、削除など。

1.3 CDI 管理ソフトウェアの構成

CDI 管理ソフトウェアは、CLI コマンド（第 2 章参照）、PCIe スイッチ制御コマンド（第 3 章参照）、Director 制御コマンド（第 4 章参照）、fpclean コマンド（第 5 章参照）、Director 管理コマンド（第 6 章参照）を使用して操作します。

以下に、CDI 管理ソフトウェアの構成を記載します。



- nginx
リバースプロキシやロードバランシング、負荷分散、処理速度向上を行う。
- gunicorn
Web サーバと Web アプリケーションを接続する。WSGI 仕様で転送を行う。
- Django
Web アプリフレームワーク。CDI 管理ソフトウェアのメイン処理部であり、python ライブラリを保有する。
- MySQL
オープンソースの SQL リレーショナルデータベース管理システム。CDI 管理ソフトウェアのデータベースとして使用。
- RabbitMQ
メッセージキューイング処理を行うことができるオープンソースソフトウェア。(AMQP)で broker の役割を担う。CDI 管理ソフトウェアの時間のかかる処理に対して非同期処理を行う。
- Celery
python ライブラリの非同期タスクキュー。RabbitMQ からメッセージを受け取り、タスクを実行する。

第2章 CLI コマンド リファレンス

本章では、CDI 管理ソフトウェアの CLI コマンドについて説明します。なお、文中の Don't Care の箇所については、システム構築の作業(※1)や、運用、保守などの作業(※2)には影響しない項目を表します。

※1 システム構築手順書を参照

※2 システム運用管理者ガイドを参照

2.1 コマンドの形式

CLI コマンドの形式は、以下になります。

```
epcctl -u <URL> subcmd subsubcmd options
```

パラメータ	説明
-u <URL>	コマンドの URL。指定は必須。 ※説明では URL が設定された環境変数 \$URL で表記しています。環境変数の設定についてはシステム構築手順書を参照してください。
subcmd	コマンドの種類。指定は必須。
subsubcmd	コマンドの操作。指定は必須。
options	コマンドのオプション。subsubcmd のオプションを指定。

2.2 コマンドの種類

CLI コマンドには以下の種類があります。詳細については、該当する章を参照してください。

- リソースの操作 ([2.4](#))
- 論理サーバの操作 ([2.5](#))
- グループの操作 ([2.6](#))
- イメージの操作 ([2.7](#))
- その他の操作 ([2.8](#))

2.3 コマンドの説明内容

各 CLI コマンドは、以下の内容について記載しています。

○ [機能説明]

コマンドの機能について説明しています。

○ [形式]

コマンドの形式について説明しています。なお、options の指定については以下のように表記しています。

- []で囲まれた項目 : 指定は任意
- { }で囲まれた項目 : 何れか 1 つを選択
- < >で囲まれた項目 : 必ず指定
- 斜体部分 : 斜体で書かれたそのままの意味に相当する適切な値を入力

○ [補足]

注意事項などの補足事項について説明しています。

○ [使用例]

コマンドの使用例を挙げています。

2.4 リソースの操作

ここでは、リソースの操作に関連した CLI コマンドについて説明します。

2.4.1 resource list

[機能説明]

リソース情報を表示する。

[形式]

```
resource list <{-E, 指定なし}> <{-F, -G, -M, -g group-name, -m machine-name,
-G -g group-name, -M -g group-name, 指定なし}> [-o {JSON,LIST,YAML}]
```

・ options

-E	エラー状態のリソースの情報を表示 ※指定しない場合は、エラーのあるリソースも含めて情報を表示 ※エラー状態のリソースとは res_op_status が 0 以外のリソースです
-F	表示対象はファブリックフリープールのリソース
-G	表示対象はグループフリープールのリソース
-M	表示対象は論理サーバに組み込まれているリソース
-g	表示対象は group-name で指定したグループのリソース
-m	表示対象は machine-name で指定した論理サーバに組み込まれているリソース
-o	表示形式 JSON:JSON 形式(default) LIST:LIST 形式 YAML:YAML 形式

[補足事項]

- 表示されるリソース情報は以下になります。

項目	説明
res_uuid	リソースの UUID
res_type	リソースの種別 compute:計算サーバ, storage:NVMe SSD, gpu:GPU, fpga:FPGA, network:NIC
res_name	リソースの名称
res_status	リソースの状態 1:無効なリソース, 2:ファブリックフリープールのリソース 3:グループフリープールのリソース 4:論理サーバに組み込まれたリソース 5:削除されたリソース
res_op_status	リソースのオプションステータス 0:OK 1:Warning 2:Critical
res_owner	リソースの所有者
rack_name	リソースを設置したラック名
position_in_rack	リソースを設置したラック内の位置
slot_number	リソースを設置したスロット番号

※リソースが複数の場合は、リソースごとに情報を表示します。

[補足事項 (続き)]

- 指定可能なオプションの組み合わせは以下になります。

組み合わせ	説明
オプションなし	全てのリソースについて表示
-F	ファブリックフリープールのリソースについて表示
-G	全てのグループフリープールのリソースについて表示
-G -g	指定したグループのグループフリープールのリソースについて表示
-g	指定したグループのリソースについて表示
-M	全ての論理サーバに組み込まれているリソースについて表示
-M -g	指定したグループの全ての論理サーバに組み込まれているリソースについて表示
-m	指定した論理サーバに組み込まれているリソースについて表示
-E	エラーのある全てのリソースについて表示
-E -F	ファブリックフリープールのエラーのあるリソースについて表示
-E -G	全てのグループフリープールのエラーリソースについて表示
-E -G -g	指定したグループのグループフリープールのエラーリソースについて表示
-E -g	指定したグループのエラーリソースについて表示
-E -M	全ての論理サーバに組み込まれているエラーリソースについて表示
-E -M -g	指定したグループの全ての論理サーバに組み込まれているエラーリソースについて表示
-E -m	指定した論理サーバに組み込まれているエラーリソースについて表示

[使用例]

グループ g0 のリソース情報を JSON 形式で表示

```
# ./epcctl -u $URL resource list -g g0

200
{
  "data": {
    "resources": [
      {
        "res_uuid": " b9c59f7d-0a92-4af2-a28e-3778e5b922c1",
        "res_type": "compute",
        "res_name": "cpu0",
        "res_status": 3,
        "res_op_status": "0",
        "res_owner": "Fujitsu"
        "rack_name": "rack0",
        "position_in_rack": "5U",
        "slot_number": 8
      },
      {
        "res_uuid": "4894c94b-a598-4882-ae2c-dab4c36f5267",
        "res_type": "storage",
        "res_name": "ssd0",
        "res_status": 3,
        "res_op_status": "0",
        "res_owner": "Fujitsu"
        "rack_name": "rack0",
        "position_in_rack": "3U",
        "slot_number": 7
      }
    ]
  }
}
```

[使用例（続き）]

エラーのある全てのリソースを YAML 形式で表示

```
# ./epcctl -u $URL resource list -E -o YAML

200
- position_in_rack: 5U
  rack_name: rack0
  res_name: cpu1
  res_op_status: '2'
  res_owner: Fujitsu
  res_status: 1
  res_type: compute
  slot_number: 8
  res_uuid: b9c59f7d-0a92-4af2-a28e-3778e5b922c1
- position_in_rack: 5U
  rack_name: rack0
  res_name: cpu2
  res_op_status: '2'
  res_owner: Fujitsu
  res_status: 1
  res_type: compute
  res_uuid: 4c69968e-df77-4b37-928a-12bd2d401cc8
  slot_number: 8
```

2.4.2 resource show

[機能説明]

リソースの詳細情報を表示する。

[形式]

```
resource show <-r resource-name> [-o {JSON,LIST,YAML}]
```

・ options

-r	表示するリソースの名称を指定
-o	表示形式 JSON:JSON 形式(default) LIST:LIST 形式 YAML:YAML 形式

[補足事項]

- 表示されるリソース詳細情報は以下になります。

項目	説明
res_uuid	リソースの UUID
res_type	リソースの種別 compute:計算サーバ, storage:NVMe SSD, gpu:GPU, fpga:FPGA, network:NIC
res_name	リソースの名称
res_status	リソースの状態 1:無効なリソース, 2:ファブリックフリープールのリソース 3:グループフリープールのリソース 4:論理サーバに組み込まれたリソース 5:削除されたリソース
res_op_status	リソースのオプションステータス 0:OK 1:Warning 2:Critical
res_owner	リソースの所有者
resspec_uuid	リソーススペックの UUID
resspec_productname	リソーススペックの製品名
resspec_model	リソーススペックのモデル
resspec_vendor	リソーススペックのベンダ
grp_uuid	リソースが使用されているグループの UUID
mach_uuid	リソースが使用されている論理サーバの UUID
bootimg_uuid	リソースにインストールされた OS イメージの UUID
res_id_nonliquid	Don't Care
fabr_gid	Don't Care
resspec_removable	Don't Care
premach_uuid	Don't Care
lanports	Don't Care

[使用例]

SSD1 のリソース詳細情報を JSON 形式で表示

```
# ./epcctl -u $URL resource show -r SSD1

200
{
  "res_uuid": "763e4a90-795e-4178-99df-e1499e7242f4",
  "res_id_nonliquid": null,
  "fabr_gid": "0x201",
  "res_type": "storage",
  "res_name": "SSD1",
  "res_status": 3,
  "res_op_status": "0",
  "res_owner": "Fujitsu",
  "resspecs": [
    {
      "resspec_uuid": "00634259-a957-45ed-bc48-bb5a7950b19e",
      "resspec_productname": "product SSD1",
      "resspec_model": "model SSD1",
      "resspec_vendor": "samsung",
      "resspec_removable": false,
    }
  ],
  "grp_uuid": "90a24506-856c-4e3b-b7a8-a9b45d720b8a",
  "premach_uuid": null,
  "mach_uuid": "4c1a3572-3fc3-4748-92ff-ea18339506f5",
  "bootimg_uuid": "778172f8-7231-4eaa-adab-38d01e25762e",
  "lanports": []
}
```

2.4.3 resource sync

[機能説明]

リソース情報を同期させる。

[形式]

resource sync

- ・ options
なし

[補足事項]

本コマンドの実行により、作成済みのグループや論理サーバの構成情報がクリアされます。CDI 管理ソフトウェアのセットアップ時と保守操作時以外での実行は控えてください。

[使用例]

リソース情報を同期

```
# ./epcctl -u $URL resource sync  
  
200  
success
```

2.4.4 resource set_op_status

[機能説明]

リソースのオプションステータスを更新する。

[形式]

```
resource set_op_status <-r resource-name> <-s {0,1,2}>
```

• options

-r	オプションステータスを更新するリソースの名称を指定
-s	更新する値 0:OK 1:Warning 2:Critical

[補足事項]

対象リソースの res_op_status（オプションステータス）を更新します。

[使用例]

SSD0 のオプションステータスを 1(Warning)に更新

```
# ./epcctl $u resource set_op_status -r ssd0 -s 1
```

```
200  
success
```

2.5 論理サーバの操作

ここでは、論理サーバの操作に関連した CLI コマンドについて説明します。

2.5.1 machine list

[機能説明]

論理サーバ情報を表示する。

[形式]

machine list <-g group-name> [-o {JSON,LIST,YAML}]

・ options

-g	論理サーバ情報 を表示するグループの名称を指定
-o	表示形式 JSON:JSON 形式(default) LIST:LIST 形式 YAML:YAML 形式

[補足事項]

- 表示される論理サーバの情報は以下になります。

項目	説明
mach_uuid	論理サーバの UUID
mach_id	論理サーバの ID
mach_name	論理サーバの名称
mach_owner	論理サーバの所有者
resources	以下は論理サーバを構成しているリソースの情報 ※表示形式が JSON の場合のみ表示されます
res_uuid	リソースの UUID
res_name	リソースの名称
res_type	リソースの種別 compute:計算サーバ, storage:NVMe SSD, gpu:GPU, fpga:FPGA, network:NIC
res_status	リソースの状態 1:無効なリソース, 2:ファブリックフリープールのリソース 3:グループフリープールのリソース 4:論理サーバに組み込まれたリソース 5:削除されたリソース
res_op_status	リソースのオプションステータス 0:OK 1:Warning 2:Critical

[使用例]

グループ g0 の論理サーバ情報を JSON 形式で表示

```
# ./epcctl -u $URL machine list -g g0

200
{
  "data": {
    "machines": [
      {
        "mach_uuid": "7d6210ef-8a6a-403e-a980-5542b7852c30",
        "mach_id": 1 ,
        "mach_name": " F]Machine ",
        "mach_owner": "Fujitsu",
        "resources": [
          {
            "res_uuid": "f59b7fc1-ad50-494b-9f45-2ff027c6cf55",
            "res_name": "pcpu3",
            "res_type": "compute",
            "res_status": 4,
            "res_op_status": "0"
          }
        ]
      }
    ]
  }
}
```

2.5.2 machine create

[機能説明]

論理サーバを作成する。

[形式]

- ① machine create <-g *group-name*> <-m *machine_name*> <-r *resource_enum*>
- ② machine create <-y *yaml_filename*>

• options

-g	対象となるグループの名称を指定
-m	論理サーバの名称を指定
-r	論理サーバを構成するリソースを列挙。カンマ区切りでリソースの名称を指定
-y	形式①の内容を記載した yml ファイル名を指定

[補足事項]

- yml ファイル例

```
groups:
  grp_name: "group1"
  machines:
    - mach_name: " FJMachine "
      resources:
        - res_specs:
            - res_type: "compute"
              res_spec:
                condition:
                  -
                    column: "info_product_name"
                    operator: "eq"
                    value: "product node1"
            - res_type: "storage"
              res_name: "SSD1"
```

[補足事項 (続き)]

項目	説明
grp_name	グループの名称
mach_name	論理サーバの名称
res_type	リソースの種別 compute:計算サーバ, storage:NVMe SSD, gpu:GPU, fpga:FPGA, network:NIC
res_spec	リソースのスペック Column : info_model, info_product_name, info_vendor Operator : eq を設定 Value : 比較する値 ※詳細は 2.9 参照
res_name	リソースの名称 ※resource list コマンドなどで表示される res_name

※ res_spec,res_name は、res_type に対して、何れか 1 つを設定します。

- 指定できるリソースは、グループフリープールのものに限ります。
- 本コマンドは突き放し処理のため正常終了後に machine show コマンドを使用し、論理サーバの詳細ステータスが POWER_OFF になるまで繰り返し確認してください。
- 論理サーバの名称はシステム内で一意となるように指定してください。

[使用例]

グループ g0 のフリープールのリソース pcpu0,gpu0,ssd0 から論理サーバ m00 を作成

```
# ./epcctl -u $URL machine create -g g0 -m m00 -r pcpu0,gpu0,ssd0

200
{'data': {'machines': [{'mach_uuid': '67b1d3d7-32c8-4da8-a758-b3bba5cc2e0e'}]}}
```

2.5.3 machine show

[機能説明]

論理サーバの詳細情報を表示する。

[形式]

machine show <-m *machine_name*> [-o {JSON,LIST,YAML}]

・ options

-m	詳細情報を表示する論理サーバの名称を指定
-o	表示形式 JSON:JSON 形式(default) LIST:LIST 形式 YAML:YAML 形式

[補足事項]

- 表示される詳細情報は以下になります。

項目	説明
mach_uuid	論理サーバの UUID
mach_id	論理サーバの ID
mach_name	論理サーバの名称
mach_status	論理サーバのステータス ※mach_status_detail に値と詳細ステータスを記載しています
mach_op_status	論理サーバのオプションステータス
mach_status_detail	論理サーバの詳細ステータス <ul style="list-style-type: none"> ・ 10. READY TO BUILDING : 論理サーバ作成処理実行待ち。 ・ 11. CREATING : 論理サーバ作成中。完了後に INACTIVE POFF ステータスへ遷移。 ・ 12. BOOTING : 論理サーバ電源オン実行中。完了後に ACTIVE PON ステータスへ遷移。 ・ 13. ACTIVE PON : 論理サーバ電源オン。 ・ 14. POWERING OFF : 論理サーバの電源オフ実行中。完了後に INACTIVE POFF ステータスへ遷移。 ・ 15. INACTIVE POFF : 論理サーバの電源オフ。 ・ 16. DELETING : 論理サーバ解体中。 ・ 17. DELETED : 論理サーバ解体済み。 ・ 18. INSTALLING OS : OS イメージインストール中。完了後に INACTIVE POFF ステータスへ遷移。 ・ 19. ERASING DISK : ディスクデータ削除中。完了後に INACTIVE POFF ステータスへ遷移。 ・ 20. ADDING RESOURCES : 論理サーバにリソース追加中。 ・ 21. REMOVING RESOURCES : 論理サーバからリソース削除中。 ・ 30. READY TO UNBUILDING : 論理サーバの解体処理実行待ち。 ・ 90. ERROR : エラー状態。※この状態では論理サーバは解体しか行えません。 ※10~90 の値は mach_status で表示されます
mach_owner	論理サーバの所有者
grp_uuid	グループの UUID
resources	論理サーバを構成しているリソース
res_uuid	リソースの UUID
res_name	リソースの名称
res_type	リソースの種別 compute:計算サーバ, storage:NVMe SSD, gpu:GPU, fpga:FPGA, network:NIC
res_op_status	リソースのオプションステータス 0:OK 1:Warning 2:Critical
res_status	リソースの状態 1:無効なリソース, 2:ファブリックフリープールのリソース 3:グループフリープールのリソース 4:論理サーバに組み込まれたリソース 5:削除されたリソース

[補足事項 (続き)]

p2p	PCIe バスの Peer to Peer(P2P)機能の設定状態 on:有効, off:無効
mach_id_nonliquid	Don't Care
lanports	Don't Care
boot_ssd	Don't Care

[使用例]

論理サーバ FJMachine の詳細情報を JSON 形式で表示

```
# ./epcctl -u $URL machine show -m FJMachine

200
{
  "data": {
    "machines": [
      {
        "mach_uuid": "b9e25b57-521a-4047-b2cd-6159c452ac26",
        "mach_id_nonliquid": null,
        "mach_id": 1,
        "mach_name": " FJMachine ",
        "mach_status": 15,
        "mach_op_status": "00",
        "mach_status_detail": " INACTIVE POFF ",
        "mach_owner": "Fujitsu",
        "grp_uuid": "7cd208b6-8987-495f-b9bd-5f257ae850b7",
        "p2p": "off",
        "boot_ssd": null,
        "lanports": null,
        "resources": [
          {
            "res_uuid": "f59b7fc1-ad50-494b-9f45-2ff027c6cf55",
            "res_name": "pcpu3",
            "res_type": "compute",
            "res_status": 4,
            "res_op_status": "0"
          }
        ]
      }
    ]
  }
}
```

2.5.4 machine destroy

[機能説明]

論理サーバを解体する。

[形式]

```
machine destroy <-m machine_name>
```

・ options

-m	解体する論理サーバの名称を指定
----	-----------------

[補足事項]

- 本コマンドは突き放し処理のため正常終了後に machine list コマンドを使用し、解体した論理サーバが表示されなくなるまで繰り返し確認してください。
- 論理サーバで使用されていたリソースは、解体後にグループフリープールに移動されます。
- 論理サーバで使用されていたリソースのうち res_type が storage のリソース(NVMe SSD)に格納されているデータは自動的に削除されます。

[使用例]

論理サーバ m0 を解体

```
# ./epcctl -u $URL machine destroy -m m0

200
{'data': {'machines': [{'mach_uuid': 'b9e25b57-521a-4047-b2cd-6159c452ac26'}]}}
```

2.5.5 machine modify

[機能説明]

論理サーバへのリソースの追加または、論理サーバからのリソースの削除を行う。

[形式]

- ① machine modify <-M {add, remove}> <-m machine_name> <-r resource_enum>
- ② machine modify <-M {add, remove}> <-y yaml_filename>

・ options

-M	モード add : 論理サーバにリソースを追加 remove : 論理サーバからリソースを削除
-m	論理サーバの名称を指定
-r	論理サーバを構成するリソースを列挙。カンマ区切りでリソースの名称を指定
-y	形式①のオプション -m -r の内容を記載した yaml ファイル名を指定

[補足事項]

● yaml ファイル例

```
machines:
  mach_name: "FJMachine "
  resources:
    - res_specs:
      - res_type: "compute"
        res_spec:
          condition:
            -
              column: "info_product_name"
              operator: "eq"
              value: "product node1"
      - res_type: "storage"
        res_name: "SSD1"
```

[補足事項 (続き)]

項目	説明
mach_name	論理サーバの名称
res_type	リソースの種別 compute:計算サーバ, storage:NVMe SSD, gpu:GPU, fpga:FPGA, network:NIC
res_spec	リソースのスペック Column : info_model, info_product_name, info_vendor Operator : eq を設定 Value : 比較する値 ※詳細は 2.9 参照
res_name	リソースの名称 ※resource list コマンドなどで表示される res_name

※ res_spec,res_name は、res_type に対して、何れか 1 つを設定します。

- 実行前に詳細ステータスが POWER_OFF であることを確認してください。なお、詳細ステータスが POWER_OFF でなければ、machine power -P off コマンドにより POWER_OFF にしてから実行してください。
- 詳細ステータスが POWER_ON の状態で実行した場合、リソースの追加・削除の前に論理サーバの電源がオフに変更され、また、リソースの追加・削除の後に論理サーバの電源がオンに変更されますが、詳細ステータスが POWER_OFF の状態で実行してください。
- 本コマンドは突き放し処理のため、正常終了後 machine show コマンドを使用して、論理サーバの詳細ステータスが POWER_OFF になるまで繰り返し確認してください。
- res_type が storage のリソースを削除した場合は、リソース(NVMe SSD) に格納されているデータは自動的に削除されます。

[使用例]

論理サーバ m0 にリソース ssd7 を追加

```
# ./epcctl -u $URL machine modify -M add -m m0 -r ssd7

200
{'data': {'machines': [{'mach_uuid': '5b650c7a-a44b-40ec-b4eb-6e6860c0bdde'}]}}
```

2.5.6 machine power

[機能説明]

論理サーバのパワーオンまたは、パワーオフを行う。

[形式]

machine power <-P {on, off}> <-m machine_name>

・ options

-P	モード on:パワーオンを実施,off:パワーオフを実施
-m	対象となる論理サーバの名称を指定

[補足事項]

- 本コマンドは突き放し処理のため正常終了後に machine show コマンドを使用し、論理サーバの詳細ステータスが以下のようにになるまで繰り返し確認してください。
 - P on を指定時 → POWER_ON
 - P off を指定時 → POWER_OFF

[使用例]

論理サーバ m0 をパワーオン

```
# ./epcctl -u $URL machine power -P on -m m0

200
{'data': {'machines': [{'mach_uuid': '3027d7e8-6bd6-422a-8130-a12c809be254'}]}}
```

2.5.7 machine p2p

[機能説明]

PCIe バスの Peer to Peer(P2P)機能の有効または無効設定を行う。

[形式]

machine p2p <-P {on, off}> <-m machine_name>

・ options

-P	モード on:P2P 有効,off:P2P 無効
-m	対象となる論理サーバの名称を指定

[補足事項]

- 本コマンドは対象となる論理サーバが電源オンの状態で実行してください。なお、論理サーバの電源をオンした直後に実行するとエラーになることがあります。"machine is powered off."または"Failed to Enable/disable P2P for a machine."が表示された場合は、一定時間おいてから再度実行してください。
- 論理サーバの電源をオフすると P2P 設定は自動的に無効になります。有効にする場合は、論理サーバの電源をオンした後に本コマンドを使用して有効にしてください。

[使用例]

論理サーバ m0 の P2P 設定を有効化

```
# ./epcctl -u $URL machine p2p -P on -m m0

200
{'mach_uuid': '3027d7e8-6bd6-422a-8130-a12c809be254'}
```

2.6 グループの操作

ここでは、グループの操作に関連した CLI コマンドについて説明します。

2.6.1 group list

[機能説明]

グループ情報を表示する。

[形式]

group list [-o {JSON,LIST,YAML}]

・ options

-o	表示形式 JSON:JSON 形式(default) LIST:LIST 形式 YAML:YAML 形式
----	---

[補足事項]

- 表示されるグループの情報は以下になります。

項目	説明
grp_uuid	グループの UUID
grp_id	グループの ID
grp_name	グループの名称
grp_owner	グループの所有者
resources	グループを構成しているリソースの情報
res_uuid	リソースの UUID
res_name	リソースの名称
res_type	リソースの種別 compute:計算サーバ, storage:NVMe SSD, gpu:GPU, fpga:FPGA, network:NIC
res_status	リソースの状態 1:無効なリソース, 2:ファブリックフリープールのリソース 3:グループフリープールのリソース 4:論理サーバに組み込まれたリソース 5:削除されたリソース
res_option_status	リソースのオプションステータス 0:OK 1:Warning 2:Critical
machines	グループを構成している論理サーバの情報
uuid	論理サーバの UUID
status	論理サーバのステータス
option_status	論理サーバのオプションステータス

※resources,machines は、グループに存在する個々のリソース、論理サーバについて表示します。

[使用例]

グループの情報を JSON 形式で表示

```
# ./epcctl -u $URL group list

200
{
  "data": {
    "groups": [
      {
        "grp_uuid": "7cd208b6-8987-495f-b9bd-5f257ae850b7",
        "grp_id": 1,
        "grp_name": "g0",
        "grp_owner": "Fujitsu",
        "resources": [
          {
            "res_uuid": "3ade12c8-bbd4-4306-8d5f-8c374dd82620",
            "res_name": "gpu0",
            "res_type": "gpu",
            "res_status": 3,
            "res_option_status": "0"
          },
          {
            "res_uuid": "3b9f159b-3a67-466a-9c8c-c2933293a962",
            "res_name": "nic0",
            "res_type": "network",
            "res_status": 3,
            "res_option_status": "0"
          }
        ],
        "machines": [
          {
            "uuid": "3027d7e8-6bd6-422a-8130-a12c809be254",
            "status": 10,
            "option_status": "00"
          }
        ]
      }
    ]
  }
}
```


2.6.2 group create

[機能説明]

グループを作成する。

[形式]

①group create <-g group_name> <-r resource-enum>

②group create <-y yaml_filename>

・ options

-g	作成するグループの名称を指定
-r	グループを構成するリソースを列挙。カンマ区切りでリソース名称を指定
-y	形式①の内容を記載した yaml ファイル名を指定

[補足事項]

● yaml ファイル例

```
groups:
  grp_name: 'group0'
  premachines: []
  resources:
  -
    res_specs:
    -
      res_type: 'compute'
      res_uuid: 'b9c59f7d-0a92-4af2-a28e-3778e5b922c1'
    -
      res_type: 'gpu'
      res_spec:
        condition:
        -
          column: 'info_model'
          operator: 'eq'
          value: 'a100'
      flavors: []
```

項目	説明
grp_name	グループの名称
res_type	グループを構成しているリソースの種別 compute:計算サーバ, storage:NVMe SSD, gpu:GPU, fpga:FPGA, network:NIC
res_uuid	リソースの UUID ※resource list コマンドなどで表示される res_uuid
res_spec	リソースのスペック (配列で複数条件を and で指定可) Column : info_model, info_product_name, info_vendor Operator : eq を設定 Value : 比較する値 ※詳細は 2.9 参照
premachines	不使用 ※指定なし([]を記載)
flavors	不使用 ※指定なし([]を記載)

※ res_spec,res_uuid は、res_type に対して、何れか 1 つを設定します。

[使用例]

リソース cpu0,gpu0,gpu1 を使用してグループ g0 を作成

```
# ./epcctl -u $URL group create -g g0 -r cpu0,gpu0,gpu1
```

```
200
```

```
{'grp_uuid': 'c50ec5e9-4b32-465b-8ff4-e0921a49b75a'}
```

2.6.3 group show

[機能説明]

グループの詳細情報を表示する。

[形式]

group show <-g group_name> [-o {JSON,LIST,YAML}]

・ options

-g	詳細情報を表示するグループの名称を指定
-o	表示形式 JSON:JSON 形式(default) LIST:LIST 形式 YAML:YAML 形式

[補足事項]

- 表示される詳細情報は以下になります。

項目	説明
grp_name	グループの名称
grp_uuid	グループの UUID
grp_id	グループの ID
grp_owner	グループの所有者
grp_status	グループのステータス
grp_op_status	グループのオプションステータス
networks	グループ内の NIC デバイス情報
uuid	UUID
uuid_type	UUID の種別
resources	グループを構成しているリソースの情報
res_uuid	リソースの UUID
res_name	リソースの名称
res_type	リソースの種別 compute:計算サーバ, storage:NVMe SSD, gpu:GPU, fpga:FPGA, network:NIC
res_status	リソースの状態 1:無効なリソース, 2:ファブリックフリープールのリソース 3:グループフリープールのリソース 4:論理サーバに組み込まれたリソース 5:削除されたリソース
res_option_status	リソースのオプションステータス 0:OK 1:Warning 2:Critical
grp_id_nonliquid	Don't Care
fbr_id	Don't Care
pod_id	Don't Care
shelf_id	Don't Care
node_id	Don't Care
grp_policy	Don't Care

[使用例]

グループ g0 の詳細情報を JSON 形式で表示

```
# ./epcctl -u $URL group show -g g0

200
{
  "data": {
    "groups": [
      {
        "grp_uuid": "c50ec5e9-4b32-465b-8ff4-e0921a49b75a",
        "shelf_id": null,
        "node_id": null,
        "fbr_id": 1,
        "pod_id": -1,
        "grp_id_nonliquid": null,
        "grp_id": 1,
        "grp_name": "g0",
        "grp_status": 1,
        "grp_op_status": "000",
        "grp_policy": null,
        "grp_owner": "Fujitsu",
        "networks": [
          {
            "uuid": "3b9f159b-3a67-466a-9c8c-c2933293a962",
            "uuid_type": 5
          }
        ],
        "resources": [
          {
            "res_uuid": "3ade12c8-bbd4-4306-8d5f-8c374dd82620",
            "res_name": "gpu0",
            "res_type": "gpu",
            "res_status": 3,
            "res_option_status": "0"
          },
          {
            "res_uuid": "4f6f4cac-1074-4cf0-ac6b-da37d17642f0",
            "res_name": "ssd0",
            "res_type": "storage",
            "res_status": 3,
            "res_option_status": "0"
          },
          {
            "res_uuid": "f59b7fc1-ad50-494b-9f45-2ff027c6cf55",
            "res_name": "pcpu3",
            "res_type": "compute",
            "res_status": 3,
            "res_option_status": "0"
          }
        ]
      }
    ]
  }
}
```

2.6.4 group destroy

[機能説明]

グループを解体する。

[形式]

group destroy <-g group_name>

・ options

-g	解体するグループの名称を指定
----	----------------

[補足事項]

なし

[使用例]

グループ g0 を解体

```
# ./epcctl -u $URL group destroy -g g0  
  
200  
{'grp_uuid': '3fa85f64-5717-4562-b3fc-2c963f66afa7'}
```

2.6.5 group modify

[機能説明]

グループへのリソースの追加または、グループからリソースの削除を行う。

[形式]

- ①group modify <-M {add, remove}> <-g group_name> <-r resource_enum>
- ②group modify <-M {add, remove}> <-y yaml_filename>

• options

-M	モード add:リソースの追加 remove:リソースの削除
-g	対象のグループの名称を指定
-r	グループを構成するリソースを列挙。カンマ区切りでリソースの名称を指定
-y	形式①のオプション -g -r の内容を記載した yaml ファイル名を指定

[補足事項]

```
groups:
  grp_uuid: '7cd208b6-8987-495f-b9bd-5f257ae850b7'
  resources:
  -
    res_type: 'compute'
    res_uuid: ' b9c59f7d-0a92-4af2-a28e-3778e5b922c1'
  -
    res_type: 'gpu'
    res_spec:
      condition:
      -
        column: 'info_model'
        operator: 'eq'
        value: 'a100'
```

項目	説明
grp_uuid	グループの UUID
res_type	リソースの種別 compute:計算サーバ, storage:NVMe SSD, gpu:GPU, fpga:FPGA, network:NIC
res_uuid	リソースの UUID ※resource list コマンドなどで表示される res_uuid
res_spec	リソースのスペック Column : info_model, info_product_name, info_vendor Operator : eq を設定 Value : 比較する値

※詳細は 2.9 参照

※ res_spec,res_uuid は、res_type に対して、何れか 1 つを設定します。

[使用例]

グループ g0 にリソース ssd を追加

```
# ./epcctl -u $URL group modify -M add -g g0 -r ssd

200
{'grp_uuid': '3fa85f64-5717-4562-b3fc-2c963f66afa7'}
```

2.7 イメージの操作

ここでは、イメージの操作に関連した CLI コマンドについて説明します。

2.7.1 image list

[機能説明]

ブートイメージ情報を表示する。

[形式]

image list [-o {JSON,LIST,YAML}]

・ options

-o	表示形式 JSON:JSON 形式(default) LIST:LIST 形式 YAML:YAML 形式
----	---

[補足事項]

- 表示されるイメージの情報は以下になります。

項目	説明
bootimg_uuid	ブートイメージの UUID
bootimg_owner	ブートイメージの所有者
bootimg_filename	ブートイメージのファイル名
bootimg_type	ブートイメージの種別 'img', 'iso'

[使用例]

ブートイメージの情報を JSON 形式で表示

```
# ./epcctl -u $URL image list

200
{
  "data": {
    "images": [
      {
        "bootimg_uuid": "0627d38b-5d5e-49d4-8439-0a2d4810d55c",
        "bootimg_owner": "n/a",
        "bootimg_filename": "ubuntu18-rx1-20221004.img",
        "bootimg_type": "img"
      },
      {
        "bootimg_uuid": "12e8c1fa-6dd4-4359-b03e-58cd6cf7acc2",
        "bootimg_owner": "n/a",
        "bootimg_filename": "ubuntu18-rx1-20221001.img",
        "bootimg_type": "img"
      }
    ]
  }
}
```

2.7.2 image add

[機能説明]

ブートイメージを登録する。

[形式]

image add <-t {img, iso}> <-I bootimage-filename>

・ options

-t	ブートイメージの種別 'img', 'iso'
-i	ブートイメージのファイル名を指定

[補足事項]

登録できるイメージの大きさは 16GB 以下になります。

[使用例]

bootimage.img をブート種別 img で登録

```
# ./epcctl -u $URL image add -t img -i bootimage.img

200
{
  'bootimg_uuid': '3fa85f64-5717-4562-b3fc-2c963f66afa6'
}
```


2.7.3 image show

[機能説明]

ブートイメージの詳細情報を表示する。

[形式]

image show <-I *bootimage-filename*> [-o {JSON,LIST,YAML}]

・ options

-i	ブートイメージファイル名称を指定
-o	表示形式 JSON:JSON 形式(default) LIST:LIST 形式 YAML:YAML 形式

[補足事項]

- 表示される詳細情報は以下になります。

項目	説明
bootimg_uuid	ブートイメージの UUID
bootimg_id	ブートイメージの ID
bootimg_date	ブートイメージを登録した日付
bootimg_owner	ブートイメージの所有者
bootimg_filename	ブートイメージファイル名
bootimg_type	ブートイメージの種別 'img', 'iso'

[使用例]

bootimage.img の詳細情報を JSON 形式で表示

```
# ./epcctl -u $URL image show -i bootimage.img

200
{
  "data": {
    "images": [
      {
        "bootimg_uuid": "0627d38b-5d5e-49d4-8439-0a2d4810d55c",
        "bootimg_id": 11,
        "bootimg_date": "2022-12-19T12:52:12+00:00",
        "bootimg_owner": "n/a",
        "bootimg_filename": "ubuntu18-rx1-20221004.img",
        "bootimg_type": "img"
      }
    ]
  }
}
```

2.7.4 image remove

[機能説明]

ブートイメージを削除する。

[形式]

image remove <-I *bootimage-filename*>

・ options

-i	ブートイメージファイルの名称を指定
----	-------------------

[補足事項]

なし

[使用例]

bootimage.img を削除

```
# ./epcctl -u $URL image remove -i bootimage.img  
{'bootimg_uuid': C3fa85f64-5717-4562-b3fc-2c963f66afa6'}
```

2.7.5 resource install

[機能説明]

イメージのインストールを行う。

[形式]

resource install <-r resource-name> <-b boot-image-name>

・ options

-r	インストール対象となるリソースの名称を指定
-b	インストールするブートイメージファイルの名称を指定

[補足事項]

論理サーバに組み込まれているリソースの種別(※1)が storage のリソース (NVMe SSD) がインストール対象になります。

※リソースの種別については、2.4.1 を参照

[使用例]

リソース `ssd0` にイメージ `ubuntu-20.04.4-live-server-amd64.iso` をインストール

```
# ./epcctl -u $URL resource install -r ssd0 -b ubuntu-20.04.4-live-server-amd64.iso

200
success
```

2.8 その他の操作

ここでは、2.1～2.7 以外の操作に関連した CLI コマンドについて説明します。

2.8.1 system dump

[機能説明]

- ①構成バックアップファイルをダウンロードする。
- ②構成バックアップファイルのリストを表示する。

[形式]

- ①system dump <-U, -F dump-file, 指定なし> [-O {path, download-file, path/download-file}]
- ②system dump -L

• options

-U	最新の構成バックアップファイルを生成してダウンロード
-L	構成バックアップファイルのリストを表示
-F	ファイル名を指定して構成バックアップファイルをダウンロード
-O	保存先を指定
指定なし	最新の構成バックアップファイルをダウンロード

[補足事項]

- 指定可能なオプションの組み合わせは以下になります。

組み合わせ	説明
-L	構成バックアップファイルのファイル名一覧を表示します
オプションなし	最新の構成バックアップファイルをカレントディレクトリにダウンロードします
-O path	最新の構成バックアップファイルを path で指定されたディレクトリにダウンロードします
-O download-file	最新の構成バックアップファイルを download-file で指定されたファイル名でカレントディレクトリにダウンロードします
-O path/download-file	最新の構成バックアップファイルを path で指定されたディレクトリに download-file で指定されたファイル名でダウンロードします
-U	最新の構成バックアップファイルを生成し、カレントディレクトリにダウンロードします
-U -O path	最新の構成バックアップファイルを生成し、path で指定されたディレクトリにダウンロードします
-U -O download-file	最新の構成バックアップファイルを生成し、download-file で指定されたファイル名でカレントディレクトリにダウンロードします
-U -O path/download-file	最新の構成バックアップファイルを生成し、path で指定されたディレクトリに download-file で指定されたファイル名でダウンロードします
-F dump-file	dump-file で指定された構成バックアップファイルを、カレントディレクトリにダウンロードします
-F dump-file -O path	Dump-file で指定された構成バックアップファイルを、path で指定されたディレクトリにダウンロードします
-F dump-file -O download-file	dump-file で指定された構成バックアップファイルを、download-file で指定されたファイル名でカレントディレクトリにダウンロードします
-F dump-file -O path/download-file	Dump-file で指定された構成バックアップファイルを、path で指定されたディレクトリに download-file で指定されたファイル名でダウンロードします

※ path は存在するディレクトリパスを指定してください。

[補足事項 (続き)]

- 構成バックアップファイルの名前は、{timestamp}.tar.gz の形式で、{timestamp}部分はバックアップを行った時のタイムスタンプになります。

※timestamp は以下の形式

yyyy-mm-dd-HH-MM-SSSSSSSSSS

Y Y Y Y : 年

mm : 月

d d : 日

HH : 時

MM : 分

SSSSSSSSSS : 分未満の数値

- 構成バックアップファイルには、以下のファイルが含まれています。

g_{grp_name}.yaml

グループ再構成に使用する yaml ファイルです。{grp_name}にはグループ名が入ります。

m_{grp_name}_{mach_name}.yaml

論理サーバ再構成に使用する yaml ファイルです。{grp_name}には論理サーバのグループ名、{mach_name}には論理サーバ名が入ります。

script.sh (構成ファイル)

グループ・論理サーバを再構成するためのコマンドリストを記述したファイルです。

[使用例]

最新の構成バックアップファイルをダウンロード

```
# ./epcctl $u system dump
```

```
200
```

```
20230301_184906.tar.gz
```

2.8.2 system log

[機能説明]

ログファイルの作成を行う。

[形式]

system log <-o {log-file, directory-name}, 指定なし>

・ options

-o	<ul style="list-style-type: none"> ・ log-file で指定したファイル名でログファイル（圧縮形式）を作成 ・ directory-name で指定したディレクトリにログファイル(log.tar.gz)を作成 ※指定したディレクトリが存在しない場合はエラーとなります。
指定なし	カレントディレクトリにログファイル(log.tar.gz)を作成 ※コマンドを実行したディレクトリに作成されます。

[補足事項]

- 作成されるログファイル（圧縮形式）には、以下のログが含まれています。
 - unicorn 関連ログ
 - unicorn_access.log, unicorn_error.log
 - nginx 関連ログ
 - nginx_access.log, nginx_error.log, nginx_conf_access.log, nginx_conf_error.log
 - resource manager 関連ログ
 - resource_manager_queue.log
 - RabbitMQ 関連ログ
 - rabbit@cdiv11.log
 - LIQID ログ
 - Liquid_log.tar.gz
- ログファイルの作成には、数 MB～数百 MB 程度の容量が必要になります。

[使用例]

ログファイルを/home/20230407.tar.gz を作成

```
# ./epcctl -u $URL system log -o /home/20230407.tar.gz

200
/home/20230407.tar.gz
```

2.8.3 resource erase

[機能説明]

リソース (NVMe SSD) に格納されたデータの削除を行う。

[形式]

```
resource erase <-r resource-name>
```

・ options

-r	データ削除の対象となるリソースの名称を指定
----	-----------------------

[補足事項]

論理サーバに組み込まれているリソースの種別(※1)が storage のリソース (NVMe SSD) が対象となります。

※リソースの種別については、2.4.1 を参照

[使用例]

リソース `ssd0` に格納されたデータを削除

```
# ./epcctl -u $URL resource erase -r ssd0  
  
200  
success
```

2.8.4 resource bmc

[機能説明]

- ① 計算サーバの iRMC 情報を表示する。
- ② 計算サーバの iRMC 情報の設定を行う。
- ③ 計算サーバの iRMC 情報の削除を行う。 ※iRMC : integrated Remote Management Controller (以下同じ)

[形式]

- ① resource bmc <-r resource-name> [-o {JSON,LIST,YAML}]
- ② resource bmc <-r resource-name> <-M add> <-i ip_address> <-u user_id>
- ③ resource bmc <-r resource-name> <-M remove>

• options

-r	表示するリソースの名称を指定
-o	表示形式 JSON:JSON 形式(default) LIST:LIST 形式 YAML:YAML 形式
-M	モード add:設定 remove:削除
-i	Ipmi IP アドレス ※Ipmi : Intelligent Platform Management Interface(以下同じ)
-u	Ipmi ユーザ ID

[補足事項]

- 表示される iRMC 情報は以下になります。

項目	説明
res_uuid_cpunode	リソースの UUID
cpunode_ipmi_address	Ipmi IP アドレス
cpunode_ipmi_userid	Ipmi ユーザ ID
cpunode_ipmi_password	Ipmi パスワード

- リソースの種別が compute のリソース（計算サーバ）が対象となります。
※リソースの種別については、2.4.1 を参照してください。

[使用例]

```
# ./epcctl -u $URL resource bmc -r pcpu3

200
{
  "data": {
    "resources": [
      {
        "res_uuid_cpunode": "f59b7fc1-ad50-494b-9f45-2ff027c6cf55",
        "cpunode_ipmi_address": "xxxxxx",
        "cpunode_ipmi_userid": "xxxxxx ",
        "cpunode_ipmi_password": "xxxxxx"
      }
    ]
  }
}
```


2.8.5 jbox add

[機能説明]

PCIe Box 情報を追加する。

[形式]

```
jbox add <-y yml-filename>
```

・ options

-y	PCIe Box 情報を記した yml ファイル名
----	---------------------------

[補足事項]

使用する yml ファイルについては、システム構築手順書「5.1.8 スペックファイルを編集する」および「5.1.12 PCIe Box テーブルを作成する」を参照してください。

[使用例]

jboxspec_10s_box0.yml の情報を JBOX テーブルに追加

```
# ./epcctl -u $URL jbox add -y jboxspec_10s_box0.yml  
  
{'data': {'jbox': [{'jbox_uuid': '0cbb6bc6-4f16-4357-9cc7-af2c5055e103'}]}}
```

2.8.6 jboxspecs add

[機能説明]

jboxspecs 情報を追加する。

[形式]

```
jboxspecs add <-y yaml-filename>
```

・ options

-y	jbox specs 情報を記した yaml ファイル名
----	------------------------------

[補足事項]

使用する yaml ファイルについては、システム構築手順書「5.1.11 PCIe BOX のスペックテーブルを作成する」を参照してください。

[使用例]

jboxspecs.yaml の情報を JBOX 情報に追加

```
# ./epcctl -u $URL jboxspecs add -y jboxspecs.yaml  
{'data': {'jboxspecs': [{'spec_uuid': 'fc44c85d-1f50-4eae-9d48-ca9122b43017'}]}}
```

2.8.7 spec list

[機能説明]

リソーススペックを表示する。

[形式]

spec list [-o {JSON,LIST,YAML}]

・ options

-o	表示形式 JSON:JSON 形式(default) LIST:LIST 形式 YAML:YAML 形式
----	---

[補足事項]

- 表示されるリソーススペック情報は以下になります。

項目	説明
resspec_uuid	リソーススペック UUID
productname	製品名
model	モデル名
vendor	ベンダ名
vid	ベンダ ID
did	デバイス ID
removable	removal
tdp_W	TDP
resource_count	-
fabric_free	ファブリックフリーリソース数
group_free	グループフリーリソース数
machine	論理サーバリソース数
type	タイプ名
memcapacity_GB	メモリ容量 ※リソースの種別が compute, gpu, fpga の場合に表示
corecount_counts	コア数 ※リソースの種別が compute, gpu の場合に表示
cpusocketcount_sockets	CPU ソケット数 ※リソースの種別が compute の場合に表示
memslotcount_slots	メモリスロット数 ※リソースの種別が compute の場合に表示
nicports_ports	LAN ポート数 ※リソースの種別が compute の場合に表示
subspec	未使用 ※リソースの種別が compute の場合に表示
capacity_GB	容量[GB] ※リソースの種別が storage の場合に表示
portcount_counts	ネットワークポート数 ※リソースの種別が network の場合に表示
bandwidth_Gbps	ネットワーク帯域幅 ※リソースの種別が network の場合に表示
clockspeed_GHz	クロック周波数[GHz] ※リソースの種別が gpu の場合に表示

※リソースの種別については、2.4.1 を参照してください。

[使用例]

リソーススペックを JSON 形式で表示

```
# ./epcctl $u spec list -o JSON

200
{
  "data": {
    "resspecs": [
      {
        "resspec_uuid": "190ff0c6-5d55-485e-8234-2fbd84312e61",
        "productname": "n/a",
        "model": "generic_processor",
        "vendor": "processor",
        "vid": "0x8086",
        "did": "0x0001",
        "removable": 0,
        "tdp_W": null,
        "cpusocketcount_sockets": null,
        "memslotcount_slots": null,
        "corecount_counts": null,
        "memcapacity_GB": null,
        "nicports_ports": null,
        "subspec": [],
        "type": "compute",
        "resource_count": {
          "fabric_free": 1,
          "group_free": 0,
          "machine": 0
        }
      },
      {
        "resspec_uuid": "21c6d94f-ab07-4cee-b8ef-f3b439565b45",
        "productname": "n/a",
        "model": "generic_processor",
        "vendor": "processor",
        "vid": "0x8086",
        "did": "0x0005",
        "removable": 0,
        "tdp_W": null,
        "cpusocketcount_sockets": null,
        "memslotcount_slots": null,
        "corecount_counts": null,
        "memcapacity_GB": null,
        "nicports_ports": null,
        "subspec": [],
        "type": "compute",
        "resource_count": {
          "fabric_free": 1,
          "group_free": 0,
          "machine": 0
        }
      }
    ]
  }
}
```

2.8.8 spec add

[機能説明]

スペック情報を追加する。

[形式]

spec add <-y *yaml-filename*>

・ options

-y	スペック情報を記載した yaml ファイル名を指定
----	---------------------------

[補足事項]

● yaml ファイル例

cpunodespecs:	
productname: PRIMERGYRX2530M6	← 製品名を設定
model: PRIMERGYRX2530M6	← 型名を設定
vendor: Fujitsu	← Fujitsu を設定
removable: 0	← 0 を設定
cpusocketcount_sockets: 2	← CPU 搭載数 (個数) を設定
memslotcount_slots: 8	← メモリ搭載数 (個数) を設定
corecount_counts: 2	← 総 CPU コア数 (個数) を設定
memcapacity_GB: 256	← 総メモリ容量 (GB) を設定
nicports_ports: 2	← NIC ポート数 (個数) を設定
tdp_W: 40	← 電力 (W) を設定
vid: "0x8086"	← ベンダ ID(0x8086 の固定値)
did: "0x0001"	← 接続した switch のポート番号

[使用例]

cpunode_spec.yaml に記載されたスペック情報をスペックテーブルに追加

```
# ./epcctl $u spec add -y cpunode_spec.yaml

200
{
  "resspec_uuid": "3fa85f64-5717-4562-b3fc-2c963f66afa6"
}
```

2.8.9 spec remove

[機能説明]

スペック情報を削除する。

[形式]

spec remove <-u *spec-uuid*>

・ options

-u	削除対象のスペックテーブルの UUID を指定
----	-------------------------

[補足事項]

なし

[使用例]

UUID 3fa85f64-5717-4562-b3fc-2c963f66afa6 のスペック情報を削除

```
# ./epcctl $u spec remove -u 3fa85f64-5717-4562-b3fc-2c963f66afa6  
  
200  
{  
  "resspec_uuid": "3fa85f64-5717-4562-b3fc-2c963f66afa6"  
}
```

2.8.10 login

[機能説明]

CDI 管理ソフトウェアへのログインを行う。

[形式]

login

- ・ options
なし

[補足事項]

- ログイン時にユーザ名とパスワードを入力します。
- ログイン後のシェルで CLI コマンドを実行します。
- 3 回以上続けてパスワードの入力に失敗した場合は、ユーザ ID は一時的に使用不可（ロック状態）となります。ロック状態は、一定時間後（10 分後）に解除されます。
- ログインしてからの CLI コマンド利用の有効期限は 30 分になります。

[使用例]

CDI 管理ソフトウェアにログイン

```
# ./epcctl -u $URL login
```

```
Enter Your Username : xxx ← ユーザ名を入力
```

```
Enter Your Password : *** ← パスワードを入力
```

```
login success
```

```
cdi:
```

2.9 YAML ファイル形式

ここでは、CLI コマンドで使用される YAML ファイル（パラメータファイル）について説明します。

2.9.1 フォーマット

各コマンドで使用される YAML ファイルのフォーマットは以下になります。

・グループの操作の YAML ファイル

groups:	← ハッシュキー（固定）
grp_name: 'group0'	← グループ名（ユーザ指定）
premachines: []	← ハッシュキー（固定）
resources:	← ハッシュキー（固定）
-	
res_specs:	← ハッシュキー（固定）
-	
res_type: 'compute'	← res_type を記載 (compute/storage/gpu/network/fpga から選択)
res_name: 'pcpu0'	← res_name を記載
-	
res_type: 'gpu'	← res_type を記載 (compute/storage/gpu/network/fpga から選択)
res_spec:	← リソースをスペック指定する場合
condition:	← リソースをスペック指定する場合
-	
column: 'info_model'	← 2.9.2 condition 指定方法を参照
operator: 'eq'	← 2.9.2 condition 指定方法を参照
value: 'a100'	← 2.9.2 condition 指定方法を参照
flavors: []	← ハッシュキー（固定）

・論理サーバの操作の YAML ファイル

groups:	← ハッシュキー（固定）	※machine modify 使用時には不要
grp_name: "group1"	← グループ名（ユーザ指定）	※machine modify 使用時には不要
machines:	← ハッシュキー（固定）	
- mach_name: "FJMachine"	← 論理サーバ名（ユーザ指定）	
resources:	← ハッシュキー（固定）	
- res_specs:	← ハッシュキー（固定）	
- res_type: "compute"	← res_type を記載 (compute/storage/gpu/network/fpga から選択)	
res_spec:	← リソースをスペック指定する場合	
condition:	← リソースをスペック指定する場合	
-		
column: "info_product_name"	← 2.9.2 condition 指定方法を参照	
operator: "eq"	← 2.9.2 condition 指定方法を参照	
value: "product node1"	← 2.9.2 condition 指定方法を参照	
- res_type: "storage"	← res_type を記載 (compute/storage/gpu/network/fpga から選択)	
res_name: "SSD1"	← res_name を記載	

2.9.2 condition の指定方法

YAML ファイルによる指定では、リソースのスペックを条件にして指定することが可能です。

※条件に合致するリソースが複数ある場合は、そのうちの 하나가選ばれます。

(1) 条件に使用できるスペック

2.8.7 のコマンド spec list により表示されるリソーススペックの model,productname,vendor の情報が指定に使用できます。

```
# ./epcctl -u $URL spec list -o YAML
```

```
- clockspeed_GHz: 765
  corecount_counts: 6920
  memcapacity_GB: 40
  model: a100
  productname: A100 PCIe 40GB
  removable: 0
  resource_count:
    fabric_free: 0
    group_free: 2
    machine: 0
  resspec_uuid: 38b15cc4-1f18-43cb-aed8-3c207e406f8c
  tdp_W: null
  type: gpu
  vendor: NVIDIA Corporation
```

(2) condition の指定

指定は以下の 3 つのパラメータで行います。

- Column
 - model の情報を条件に使用する場合 → column 名は “info_model” を設定
 - productname の情報を条件に使用する場合 → column 名は “info_product_name” を設定
 - vendor の情報を条件に使用する場合 → column 名は “info_vendor” を設定
- Operator
 - “eq” (Column で指定した情報が value で指定した値と等しいリソースを一つ選択) を設定
- Value
 - 比較する値を設定 (例 a100,NVIDIA Corporation など)

第3章 PCIe スイッチ制御コマンドリファレンス

本章では、CDI 管理ソフトウェアの PCIe スイッチ制御コマンドについて説明します。

3.1 コマンドの形式

CLI コマンドの形式は、以下になります。

```
swcfg subcmd options
```

パラメータ	説明
subcmd	コマンドの種類。指定は必須。
options	コマンドのオプション。subcmd のオプションを指定。

3.2 コマンドの種類

PCIe スイッチ制御コマンドには、以下の種類があります。詳細については、該当する章を参照してください。

- ・スイッチの操作 ([3.4](#))

3.3 コマンドの説明内容

各コマンドは、以下の内容について記載しています。

○ [機能説明]

コマンドの機能について説明しています。

○ [形式]

コマンドの形式について説明しています。なお、options の指定については以下のように表記しています。

- []で囲まれた項目 : 指定は任意
- { }で囲まれた項目 : 何れか 1 つを選択
- < >で囲まれた項目 : 必ず指定
- 斜体部分 : 斜体で書かれたそのままの意味に相当する適切な値を入力

○ [補足]

注意事項などの補足事項について説明しています。

○ [使用例]

コマンドの使用例を挙げています。

3.4 PCIe スイッチの操作

ここでは、PCIe スイッチの操作に関連した PCIe スイッチ制御コマンドについて説明します。

3.4.1 swcfg get

[機能説明]

PCIe スイッチの状態を表示する。

[形式]

swcfg get <-r hostname or IP address>

・ options

-r	FABRIC のホスト名または IP アドレス
----	-------------------------

[補足事項]

本コマンドは CDI 管理ソフトウェアの VM 上で動作します。

[使用例]

```
# swcfg get -r $REMOTE
Password: * * * ← LCC の root パスワードを入力
Liquid Express Fabric commander (built Mar 29 2023 23:20:57).
liquers: liqefcmd: 3.2.0.500
==== Switch ID: 0x00
Enabled ports: 8
  Port 0: HOST      GID=0000 L=G1x0/G4x16 MPS=128/2048 MRR=128
  Port 16: HOST     GID=0010 L=G1x0/G4x16 MPS=128/2048 MRR=128
  Port 32: FABRIC   GID=0020 L=G4x16/G4x16 MPS=128/2048 MRR=128
  Port 48: FABRIC   GID=0030 L=G4x16/G4x16 MPS=128/2048 MRR=128
  Port 64: HOST     GID=0040 L=G1x0/G4x16 MPS=128/2048 MRR=128
  Port 80: HOST     GID=0050 L=G1x0/G4x16 MPS=128/2048 MRR=128
```

3.4.2 swcfg set

[機能説明]

PCIe スイッチの設定を行う。

[形式]

```
swcfg set <-r hostname or IP address> <-c configuration pattern>
```

・ options

-r	FABRIC のホスト名または IP アドレス
-c	設定パターン

[補足事項]

- 本コマンドは CDI 管理ソフトウェアの VM 上で動作します。
- 設定パターンは、CDI 構成パターン名(※1)を以下から選択します。
SW1-BX1-A, SW1-BX2-A, SW1-BX3-A, SW2-BX1-A, SW2-BX2-A, SW2-BX3-A,
SW0-BX1-B, SW1-BX1-B, SW1-BX2-B, SW1-BX3-B, SW2-BX1-B, SW2-BX2-B
※1 システム構築手順書を参照

[使用例]

```
# swcfg set -r 192.168.1.133 -c $PATTERN
Password: * * * ← LCC の root パスワードを入力
```

3.4.3 swcfg check

[機能説明]

PCIe スイッチのリンク状態の表示。

[形式]

swcfg check <-r hostname or IP address> <-c configuration pattern> <-C check target>

・ options

-r	FABRIC のホスト名または IP アドレス
-c	設定パターン
-C	チェック対象 [SW, INT, EXT]

[補足事項]

- 本コマンドは CDI 管理ソフトウェアの VM 上で動作します。
- 設定パターンは、CDI 構成パターン名(※1)を以下から選択します。
 SW1-BX1-A, SW1-BX2-A, SW1-BX3-A, SW2-BX1-A, SW2-BX2-A, SW2-BX3-A,
 SW0-BX1-B, SW1-BX1-B, SW1-BX2-B, SW1-BX3-B, SW2-BX1-B, SW2-BX2-B
 ※1 システム構築手順書を参照

[使用例]

```
# swcfg check -r $REMOTE -c $PATTERN -C SW
Password: * * * ← パスワードを入力
Switch count good

# swcfg check -r 192.168.1.133 -c $PATTERN -C INT
Password: * * * ← パスワードを入力
INT check good

# swcfg check -r 192.168.1.133 -c $PATTERN -C EXT
Password: * * * ← パスワードを入力
EXT check good
```

第4章 Director 制御コマンドリファレンス

本章では、CDI 管理ソフトウェアの Director 制御コマンドについて説明します。

4.1 コマンドの形式

CLI コマンドの形式は、以下になります。

```
fabrctl subcmd options
```

パラメータ	説明
subcmd	コマンドの種類。指定は必須。
options	コマンドのオプション。subcmd のオプションを指定。

4.2 コマンドの種類

Director 制御コマンドには、以下の種類があります。詳細については、該当する章を参照してください。

- ・ Director の操作 ([4.4](#))

4.3 コマンドの説明内容

各コマンドは、以下の内容について記載しています。

○ [機能説明]

コマンドの機能について説明しています。

○ [形式]

コマンドの形式について説明しています。なお、options の指定については以下のように表記しています。

- []で囲まれた項目 : 指定は任意
- { }で囲まれた項目 : 何れか 1 つを選択
- < >で囲まれた項目 : 必ず指定
- 斜体部分 : 斜体で書かれたそのままの意味に相当する適切な値を入力

○ [補足]

注意事項などの補足事項について説明しています。

○ [使用例]

コマンドの使用例を挙げています。

4.4 Director の操作

ここでは、Director の操作に関連したコマンドについて説明します。

4.4.1 fabrctl state

[機能説明]

ファブリックシステムの状態表示を行う。

[形式]

```
fabrctl state <-u url> [-c certificate_filepath]
```

・ options

-u	FABRIC の URL(https : //Director の IP Address) 例 : https : //xxx.xxx.x.xxx
-c	Director 認証ファイルパス(Director の HTTPS 設定時)

[補足事項]

- 本コマンドは CDI 管理ソフトウェアの仮想環境で動作します。
- Director の IP Address は、システム構築手順書を参照してください。
- Director の認証ファイルパスは Director の HTTPS 設定をした場合のみ指定ください。

[使用例]

- 正常時

```
# fabrctl state -u $URL  
Success
```

- エラー時

```
# fabrctl state -u $URL  
Error: LiquidDegradedSteps: Should call 'reset'
```

※ メンテナンスモード (reset 呼び出しを推奨)

4.4.2 fabrctl shutdown

[機能説明]

Director のシャットダウンを行う。

[形式]

fabrctl shutdown <-u url> [-c certificate_filepath]

・ options

-u	FABRIC の URL(https : // <Director> の IP Address) 例 : https : //xxx.xxx.x.xxx
-c	Director 認証ファイルパス(Director の HTTPS 設定時)

[補足事項]

- 本コマンドは CDI 管理ソフトウェアの仮想環境で動作します。
- Director の IP Address は、システム構築手順書を参照してください。
- Director の認証ファイルパスは Director の HTTPS 設定をした場合のみ指定ください。

[使用例]

```
# fabrctl shutdown -u $URL
```


4.4.3 fabrctl reset

[機能説明]

Director のリセットを行う。

[形式]

```
fabrctl reset <-u url> [-c certificate_filepath]
```

・ options

-u	FABRIC の URL(https : // <Director> の IP Address >) 例 : https : //xxx.xxx.x.xxx
-c	Director 認証ファイルパス(Director の HTTPS 設定時)

[補足事項]

- 本コマンドは CDI 管理ソフトウェアの仮想環境で動作します。
- Director の IP Address は、システム構築手順書を参照してください。
- Director の認証ファイルパスは Director の HTTPS 設定をした場合のみ指定ください。

[使用例]

```
# fabrctl reset -u $URL  
Success
```

4.4.4 fabrctl restart

[機能説明]

Director の再起動を行う。

[形式]

```
fabrctl restart <-u url> <{-F, -S}> [-c certificate_filepath]
```

• options

-u	FABRIC の URL(https : // <Director> の IP Address>) 例 : https : //xxx.xxx.x.xxx
-F	ファブリック全体を再起動
-S	スイッチを再起動
-c	Director 認証ファイルパス(Director の HTTPS 設定時)

[補足事項]

- 本コマンドは CDI 管理ソフトウェアの仮想環境で動作します。
- Director の IP Address は、システム構築手順書を参照してください。
- Director の認証ファイルパスは Director の HTTPS 設定をした場合のみ指定ください。

[使用例]

```
# fabrctl restart -u $URL -F
Are you sure you want to restart fabric? [y/N]: y
Success

# fabrctl restart -u $URL -S
Are you sure you want to restart switch? [y/N]: y
Success
```

4.4.5 fabrctl upload

[機能説明]

Director のファームウェアの制御を行う。

[形式]

```
fabrctl upload <-u url> <{-f, -d, 指定なし}> [-c certificate_filepath]
```

• options

-u	FABRIC の URL(https : // <Director> の IP Address) 例 : https : //xxx.xxx.x.xxx
-f	アップロードするファームウェア (ファイル) の指定
-d	消去するファイルの指定
指定なし	-f, -d の省略時は、アップロードされたファイル一覧を表示
-c	Director 認証ファイルパス(Director の HTTPS 設定時)

[補足事項]

- 本コマンドは CDI 管理ソフトウェアの仮想環境で動作します。
- Director の IP Address は、システム構築手順書を参照してください。
- Director の認証ファイルパスは Director の HTTPS 設定をした場合のみ指定ください。

[使用例]

```
$ fabrctl upload -u $URL
Index Filename
-----
0   liqid-2.5.1.36-2.upr
1   liqid-3.0.0.106.upr

$ fabrctl upload -u $URL -f liqid-xxx.upr
Uploading liqid-xxx.upr
Success

$ fabrctl upload -u $URL -d liqid-xxx.upr
Deleting liqid-xxx.upr
Success
```

4.4.6 fabrctl upgrade

[機能説明]

Director のファームウェアのアップグレードを行う。

[形式]

fabrctl upgrade <-u url> <-i index> [-w] [-c certificate_filepath]

• options

-u	FABRIC の URL(https : // <Director> の IP Address) 例 : https : //xxx.xxx.x.xxx
-i	fabrctl upload コマンドで表示される Index を指定
-w	グループや論理サーバ情報を削除 ※省略時 (デフォルト) は保存
-c	Director 認証ファイルパス(Director の HTTPS 設定時)

[補足事項]

- 本コマンドは CDI 管理ソフトウェアの仮想環境で動作します。
- Director の IP Address は、システム構築手順書を参照してください。
- Director の認証ファイルパスは Director の HTTPS 設定をした場合のみ指定ください。

[使用例]

```
$ fabrctl upload -u $URL
Index Filename
-----
0    liqid-2.5.1.36-2.upr
1    liqid-3.0.0.106.upr

$ fabrctl upgrade -u http://xxxx/ -i 1
Are you sure you want to upgrade? [y/N]: y
Upgrading liqid-3.0.0.106
Success
```

4.4.7 fabrctl config

[機能説明]

Director の設定を行う。

[形式]

fabrctl config <-u url/> <{-K key -V value, 指定なし}> [-c certificate_filepath]

・ options

-u	FABRIC の URL(https : //<Director> の IP Address) 例 : https : //xxx.xxx.x.xxx
-K	設定を行うキー名を設定 process_devs, pdu_pass1, pdu_pass2, pdu_pass4
-V	設定値
指定なし	-K,-V の省略時は、キーと値を一覧として表示
-c	Director 認証ファイルパス(Director の HTTPS 設定時)

[補足事項]

- 本コマンドは、CDI 管理ソフトウェアの仮想環境で動作します。
- Director の IP Address は、システム構築手順書を参照してください。
- Director の認証ファイルパスは Director の HTTPS 設定をした場合のみ指定ください。

[使用例]

本コマンドはシステムの構築時に使用します。使用方法はシステム構築手順書「4.7 Director と PDU の追加設定」を参照してください。

第5章 fpclean コマンドリファレンス

本章では、CDI 管理ソフトウェアの fpclean コマンドについて説明します。

5.1 コマンドの形式

CLI コマンドの形式は、以下になります。

```
fpclean options
```

パラメータ	説明
options	コマンドのオプション。

5.2 コマンドの説明内容

各コマンドは、以下の内容について記載しています。

○ [機能説明]

コマンドの機能について説明しています。

○ [形式]

コマンドの形式について説明しています。なお、options の指定については以下のように表記しています。

- [] で囲まれた項目 : 指定は任意
- { } で囲まれた項目 : 何れか 1 つを選択
- < > で囲まれた項目 : 必ず指定
- 斜体部分 : 斜体で書かれたそのままの意味に相当する適切な値を入力

○ [補足]

注意事項などの補足事項について説明しています。

○ [使用例]

コマンドの使用例を挙げています。

5.3 Director の情報消去

ここでは、Director の情報消去コマンドについて説明します。

5.3.1 fpclean

[機能説明]

Director の情報消去を行う。

[形式]

```
fpclean <-r remote hostname or IP address>
```

・ options

-r	Director のホスト名または IP アドレス
----	---------------------------

[補足事項]

- 本コマンドは、CDI 管理ソフトウェアの仮想環境で動作します。
- Director のホスト名、IP Address は、システム構築手順書を参照してください。

[使用例]

本コマンドはシステムの構築時に使用します。使用方法はシステム構築手順書「4.9.8 Director の認証情報の消去」を参照してください。

第6章 Director 管理コマンドリファレンス

本章では、CDI 管理ソフトウェアの `fabradm` コマンドについて説明します。

6.1 コマンドの形式

CLI コマンドの形式は、以下になります。

```
fabradm subcmd options
```

パラメータ	説明
subcmd	コマンドの種類。指定は必須。
options	コマンドのオプション。subcmd のオプションを指定。

6.2 コマンドの種類

Director 管理コマンドには、以下の種類があります。詳細については、該当する章を参照してください。

- ・ Director の操作 ([6.4](#))

6.3 コマンドの説明内容

各コマンドは、以下の内容について記載しています。

○ [機能説明]

コマンドの機能について説明しています。

○ [形式]

コマンドの形式について説明しています。なお、options の指定については以下のように表記しています。

- [] で囲まれた項目 : 指定は任意
- { } で囲まれた項目 : 何れか 1 つを選択
- < > で囲まれた項目 : 必ず指定
- 斜体部分 : 斜体で書かれたそのままの意味に相当する適切な値を入力

○ [補足]

注意事項などの補足事項について説明しています。

○ [使用例]

コマンドの使用例を挙げています。

6.4 Director の管理

ここでは、Director の管理に関連したコマンドについて説明します。

6.4.1 fabradm shutdown

[機能説明]

Director の強制シャットダウンを行う。

[形式]

```
fabradm shutdown <-r hostname or IP address>
```

・ options

-r	Director のホスト名または IP アドレス
----	---------------------------

[補足事項]

- 本コマンドは、CDI 管理ソフトウェアの仮想環境で動作します。
- Director のホスト名、IP Address は、システム構築手順書を参照してください。

[使用例]

```
$ fabradm shutdown -r $REMOTE
Are you sure you want to hard shutdown? [y/N]: y
Warning: Permanently added 'xx.xx.xx.xx' (ED25519) to the list of known hosts.
root@xx.xx.xx.xx's password: * * * ← パスワードを入力
Success
```

6.4.2 fabradm restart

[機能説明]

Director の強制再起動を行う。

[形式]

fabradm restart <-r *hostname or IP address*>

・ options

-r	Director のホスト名または IP アドレス
----	---------------------------

[補足事項]

- 本コマンドは、CDI 管理ソフトウェアの仮想環境で動作します。
- Director のホスト名、IP Address は、システム構築手順書を参照してください。

[使用例]

```
$ fabradm restart -r $REMOTE
Are you sure you want to hard restart? [y/N]: y
Warning: Permanently added 'xx.xx.xx.xx' (ED25519) to the list of known hosts.
root@xx.xx.xx.xx's password: * * * ← パスワードを入力
Success
```

6.4.3 fabradm linkloss

[機能説明]

linkloss モードの設定を行う。

[形式]

fabradm linkloss <-r *hostname or IP address*> {-d, -o, -D, 指定なし}

・ options

-r	Director のホスト名または IP アドレス
-d	論理サーバに NMI を発行するモードに設定
-o	論理サーバの電源を OFF にするモードに設定
-D	何もしないモードに設定
指定なし	-d, -o, -D の省略時は論理サーバに NMI を発行するモードに設定

[補足事項]

- 本コマンドは、CDI 管理ソフトウェアの仮想環境で動作します。
- Director のホスト名、IP Address は、システム構築手順書を参照してください。

[使用例]

```
$ fabradm linkloss -r $REMOTE -d
Warning: Permanently added 'xx.xx.xx.xx' (ED25519) to the list of known hosts.
root@xx.xx.xx.xx's password: *** ← パスワードを入力
Success

$ fabradm linkloss -r $REMOTE -o
Warning: Permanently added 'xx.xx.xx.xx' (ED25519) to the list of known hosts.
root@xx.xx.xx.xx's password: *** ← パスワードを入力
Success

$ fabradm linkloss -r $REMOTE -D
Warning: Permanently added 'xx.xx.xx.xx' (ED25519) to the list of known hosts.
root@xx.xx.xx.xx's password: *** ← パスワードを入力
Success
```

6.4.4 fabradm fpclean

[機能説明]

Director の情報消去を行う。

[形式]

fabradm fpclean <-r *hostname or IP address*>

・ options

-r	Director のホスト名または IP アドレス
----	---------------------------

[補足事項]

- fpclean コマンドと同様。
- 本コマンドは、CDI 管理ソフトウェアの仮想環境で動作します。
- Director のホスト名、IP Address は、システム構築手順書を参照してください。

[使用例]

```
$ fabradm fpclean -r $REMOTE
Warning: Permanently added 'xx.xx.xx.xx' (ED25519) to the list of known hosts.
root@xx.xx.xx.xx's password: * * * ← パスワードを入力
Success
```

第7章 メッセージ

本章では、各コマンドのエラーメッセージについて説明します。

7.1 CLI コマンドのメッセージ一覧

画面に表示されるエラーのメッセージは以下になります。メッセージの先頭の数値（400,500 など）は CLI コマンドのレスポンスを表していて、200（正常終了）以外は何らかの対処が必要になります。

おり、対処により改善しない場合は富士通サポートにお問い合わせください。

※メッセージ内の xxx は、動的に表示される文字列

〇レスポンス：40x (x=0,1,2,...)

メッセージ	状況	推定原因・対処
'HTTPConnectionPool(host='xxx.xxx.xxx.xxx', port=xxxx): Max retries exceeded with url: xxxxx(Caused by NewConnectionError(xxx: Failed to establish a new connection: [Errno 113] No route to host'))	CDI 管理ソフトウェアに接続できない	<p><u>推定原因</u></p> <p>CDI 管理ソフトウェアの異常またはネットワークの障害</p> <p><u>対処①</u></p> <p>CDI 管理ソフトウェアを再起動してください。</p> <p>※システム運用管理者ガイド 「3.3 CDI 管理ソフトウェアを再起動する」</p> <p><u>対処②</u></p> <p>対処①で改善されない場合は、システム電源オフ後、ケーブル接続を確認し、システムを再起動してください。</p> <p>※システム構築手順書</p> <p>「4.9.2 電源切断手順」</p> <p>「3.6 Ethernet ケーブルの配線」</p> <p>「4.9.1 電源投入手順」</p> <p>※システム運用管理者ガイド</p> <p>「3.1.5 Director の FAN の状態確認」</p>
(メッセージなし)		
Response' object has no attribute 'json_data'	不正なレスポンスデータ受信	<p><u>推定原因</u></p> <p>URL 指定誤り</p> <p><u>対処</u></p> <p>CLI コマンドの-u パラメータを確認し、再度実行してください。</p>
Authentication not found. Please login	コマンド実行に失敗	<p><u>推定原因</u></p> <p>CDI 管理ソフトウェアにログインせずコマンドを実行した</p> <p><u>対処</u></p> <p>ログインを実行してください。</p>

レスポンス : 4 0 x (x=0,1,2,...)

メッセージ	状況	推定原因・対処
Unauthorized	ログインに失敗	<p><u>推定原因</u></p> <p>ユーザ名、パスワードの間違い</p> <p><u>対処</u></p> <p>ユーザ名、パスワードを確認し、再度ログインを実行してください。</p>
		<p><u>推定原因</u></p> <p>3 回以上ユーザ名、パスワードを間違えた</p> <p><u>対処</u></p> <p>10 分以上時間を置いてからログインを実行してください</p>
Authentication expired	コマンド実行に失敗	<p><u>推定原因</u></p> <p>ログイン後 30 分以上経過した</p> <p><u>対処</u></p> <p>ログアウトし、再度ログインしてからコマンドを実行してください。</p>
(上記以外のメッセージ)	パラメータエラー	<p><u>推定原因</u></p> <p>CLI コマンドラインオプション・パラメータが不正または yamI ファイルの内容が不正</p> <p><u>対処①</u></p> <p>ログアウトし、再度ログインしてからコマンドを実行してください。</p> <p><u>対処②</u></p> <p>CLI コマンドラインオプションが正しく指定されているか有効なパラメータが指定されているかを確認してから再度実行してください。</p> <p>yamI ファイルを指定している場合は、yamI ファイルの内容のフォーマットが正しいか、有効なパラメータが指定されているかを確認し、ファイルを修正して再実行してください。</p>

レスポンス : 500

メッセージ	状態	推定原因・対処
<p>'HTTPConnectionPool(host='xxx.xxx.xxx.xxx', port=xxxx): Max retries exceeded with url: xxxxx(Caused by NewConnectionError(xxx: Failed to establish a new connection: [Errno 113] No route to host')) (メッセージなし)</p>	<p>CDI 管理ソフトウェアに接続できない</p>	<p><u>推定原因</u> CDI 管理ソフトウェアの異常またはネットワークの障害</p> <p><u>対処①</u> CDI 管理ソフトウェアを再起動してください。 ※システム運用管理者ガイド 「3.3 CDI 管理ソフトウェアを再起動する」</p> <p><u>対処②</u> 対処①で改善されない場合は、管理 LAN のネットワーク接続を確認してください。 ※システム構築手順書 「3.6 Ethernet ケーブルの配線」</p>
<p>login fail</p>	<p>ログインに失敗した</p>	<p><u>推定原因</u> システムの異常または CDI 管理ソフトウェアの異常</p> <p><u>対処①</u> fabrctl state コマンドを実行し、"curl: (28) Failed to connect to xxx.xxx.xxx.xxx port xxx after xxx ms: 接続がタイムアウトしました"または、"curl: (7) Failed to connect to xxx.xxx.xxx.xxx port xxx:No route to host" と表示された場合は、以下(1)~(3)の手順でシステムを再起動してください。 (1)全論理サーバの使用停止 ※システム運用管理者ガイド 「4.1.2.1①」 (2) 全論理サーバの電源オフ ※システム運用管理者ガイド 「4.1.2.1②」 (3)システムの再起動 ※システム構築手順書 「4.9.2 電源切断手順の 3 (1,2,4,5,6 は不要)」 「4.9.1 電源投入手順の 2,3,4 (1,5 は不要)」 ※システム運用管理者ガイド 「3.1.5 Director の FAN の状態確認」</p> <p><u>対処②</u> 対処①に該当しない場合は CDI 管理ソフトウェア VM を再起動してください ※システム運用管理者ガイド 「3.3 CDI 管理ソフトウェアを再起動する」</p>
<p>Failed to xxx.</p>	<p>Director からの応答が不正</p>	<p><u>推定原因</u> システムの異常</p> <p><u>対処</u> システムを再起動してください ※システム運用管理者ガイド 「3.2 システムを終了する」 「3.1 システムを起動する」</p>

○レスポンス：番号なし

メッセージ	状態	推定原因・対処
usage: xxx	CLI コマンドがエラー	<p><u>推定原因</u></p> <p>CLI コマンドラインオプション指定の不正</p> <p><u>対処</u></p> <p>CLI コマンドラインオプション指定を確認して再度実行してください</p>

7.2 swcfg コマンドのメッセージ一覧

swcfg コマンドの実行で画面に表示されるエラーメッセージは以下になります。

※メッセージ内の xxx は、動的に表示される文字列

- 共通

メッセージ	状況	推定原因・対処
sshpas: command not installed	sshpas コマンドがインストールされていない	<p><u>推定原因</u></p> <p>CDI 管理ソフトウェア VM の異常</p> <p><u>対処①</u></p> <p>CDI 管理ソフトウェア VM で sshpas コマンドをインストールしてください。</p>
xxx: Unexpected command [get, check, set]	不正なコマンド	<p><u>推定原因</u></p> <p>コマンドラインオプション・パラメータが不正</p> <p><u>対処</u></p> <p>コマンドラインオプション・パラメータを確認し、再度実行してください。サブコマンドは get, check, set の中から指定してください。</p>
Required option missing: -r/--remote	パラメータエラー	<p><u>推定原因</u></p> <p>"-r"オプションが指定されていない</p> <p><u>対処</u></p> <p>コマンドラインに"-r"で Director の IP アドレスまたは FQDN を指定してください。</p>
ssh: connect to host xxx port 22: No route to host	パラメータエラー	<p><u>推定原因</u></p> <p>"-r"オプションで指定した Director の IP アドレスまたは FQDN の間違い</p> <p><u>対処</u></p> <p>正しい Director の IP アドレスまたは FQDN を指定してください。</p>
Permission denied, please try again.	パスワード間違い	<p><u>推定原因</u></p> <p>パスワードの入力間違い</p> <p><u>対処</u></p> <p>正しいパスワードを確認し、再度実行してください。</p>

● get サブコマンド

メッセージ	状況	推定原因・対処
<pre>swcfg: illegal option -- x Usage: swcfg get [OPTION]... -r/--remote remote remote hostname or ip [Required] -h/--help display this help -v/--version display version string</pre>	<p>オプションエラー</p>	<p><u>推定原因</u></p> <p>不正なオプションを指定した</p> <p><u>対処</u></p> <p>有効なオプションを確認し、再度コマンドを実行してください。</p>

● check サブコマンド

メッセージ	状況	推定原因・対処
<pre>swcfg: illegal option -- x Usage: swcfg check [OPTION]... -r/--remote remote remote hostname or ip [Required] -c/--config pattern configuration pattern [Required] [SW1-BX1-A SW1-BX2-A SW1- BX3-A SW2-BX1-A SW2-BX2-A SW2-BX3-A SW0-BX1-B SW1-BX1- B SW1-BX2-B SW1-BX3-B SW2-BX1-B SW2-BX2-B] -C/--check target check target [SW,INT,EXT] -h/--help display this help -v/--version display version string</pre>	オプションエラー	<p><u>推定原因</u></p> <p>不正なオプションを指定した</p> <p><u>対処</u></p> <p>有効なオプションを確認し、再度コマンドを実行してください。</p>
<pre>Required option missing: -c/--config, - C/--check</pre>	オプションエラー	<p><u>推定原因</u></p> <p>必須オプションが指定されていない</p> <p><u>対処</u></p> <p>有効なオプションを確認し、再度コマンドを実行してください。</p>
<pre>xxx: Unexpected pattern"</pre>	オプションエラー	<p><u>推定原因</u></p> <p>"-c" (必須オプション) で指定したパターンが間違っている</p> <p><u>対処</u></p> <p>有効なオプションを確認し、再度コマンドを実行してください。</p>
<pre>Switch count fail: Unexpected value xxx</pre>	スイッチ接続エラー	<p><u>推定原因</u></p> <p>Director と PCIe スイッチの接続に問題がある</p> <p><u>対処</u></p> <p>Director と PCIe スイッチの接続を確認して、再度コマンドを実行してください。</p>
<pre>Fail line: xxx</pre>	スイッチ接続エラー	<p><u>推定原因</u></p> <p>PCIe スイッチと PCIe Box の接続に問題がある</p> <p><u>対処</u></p> <p>PCIe スイッチと PCIe Box の接続を確認し、再度コマンドを実行してください。</p>

● set サブコマンド

メッセージ	状況	推定原因・対処
swcfg: illegal option -- x Usage: swcfg set [OPTION]... -r/--remote remote remote hostname or ip [Required] -c/--config pattern configuration pattern [Required] [SW1-BX1-A SW1-BX2-A SW1-BX3-A SW2-BX1-A SW2-BX2-A SW2-BX3-A SW0-BX1-B SW1-BX1-B SW1-BX2-B SW1-BX3-B SW2-BX1-B SW2-BX2-B] -h/--help display this help -v/--version display version string	オプションエラー	<u>推定原因</u> 不正なオプションを指定した <u>対処</u> 有効なオプションを確認し、再度コマンドを実行してください。
Required option missing: -c/--config	オプションエラー	<u>推定原因</u> 必須オプションが指定されていない <u>対処</u> 有効なオプションを確認し、再度コマンドを実行してください。
xxx: Unexpected pattern"	オプションエラー	<u>推定原因</u> "-c" (必須オプション) で指定したパターンが間違っている <u>対処</u> 有効なオプションを確認し、再度コマンドを実行してください。
Switch count fail: Unexpected value xxx	スイッチ接続エラー	<u>推定原因</u> Director と PCIe スイッチの接続に問題がある <u>対処</u> Director と PCIe スイッチの接続を確認して、再度コマンドを実行してください。
Error: command failed: /usr/bin/ssh root@xxx.xxx.xxx.xxx xxx	オプションエラーまたはパスワードエラー	<u>推定原因</u> オプションで指定した Director の IP アドレスまたは FQDN の間違い、または入力したパスワードの間違い <u>対処</u> Director の IP アドレスまたは FQDN、Director のパスワードを確認し、再度コマンドを実行してください。

7.3 fabrctl コマンドのメッセージ一覧

fabrctl コマンドの実行で画面に表示されるエラーメッセージは以下になります。

※メッセージ内の xxx は、動的に表示される文字列

● 共通

メッセージ	状況	推定原因・対処
xxx: Unexpected command [config, state, shutdown, reset, restart, upload, upgrade]"	不正なコマンド	<p><u>推定原因</u> コマンドラインオプション・パラメータが不正</p> <p><u>対処</u> コマンドラインオプション・パラメータを確認し、再度実行してください。サブコマンドは config, state, shutdown, reset, restart, upload, upgrade の中から指定してください。</p>
-u/--url: Required option missing	パラメータエラー	<p><u>推定原因</u> "-u"オプションが指定されていない</p> <p><u>対処</u> コマンドラインに"-u"オプションで Director のアクセス URL を指定してください。</p>
fabrctl: illegal option -- x	パラメータエラー	<p><u>推定原因</u> 不正なパラメータが指定された</p> <p><u>対処</u> コマンドラインオプション・パラメータを確認し、再度実行してください。</p>
curl: (7) Failed to connect to xxx.xxx.xxx.xxx port xxx after 1 ms: Connection refused	パラメータエラー	<p><u>推定原因</u> "-u"オプションで指定した URL が不正</p> <p><u>対処</u> Director のアクセス URL を確認し、再度実行してください。</p>
curl: (60) SSL certificate problem: self-signed certificate	SSL 認証エラー	<p><u>推定原因</u> Director の SSL 認証に失敗</p> <p><u>対処</u> Director の SSL 証明書を取得し、-c オプションで指定して、再度実行してください。</p>

● config サブコマンド

メッセージ	状況	推定原因・対処
--key and --value: Both options are required	パラメータエラー	<p><u>推定原因</u></p> <p>-K オプションと-V オプションが片方だけ指定されているか、またはどちらも指定されていない</p> <p><u>対処</u></p> <p>コマンドラインオプション・パラメータを確認し、再度実行してください。-K と-V はどちらも必須です。</p>
xxx: Unexpected key	パラメータエラー	<p><u>推定原因</u></p> <p>-K オプションで指定したキーが不正</p> <p><u>対処</u></p> <p>コマンドラインオプション・パラメータを確認し、再度実行してください。キーは process_devs, pdu_pass1, pdu_pass2, pdu_pass4 の中から指定してください。</p>

● state サブコマンド

メッセージ	状況	推定原因・対処
Error: LiquidDegradedSteps: Should call 'reset'	Director がエラー状態	<p><u>推定原因</u></p> <p>Director がエラー状態（メンテナンスモード）</p> <p><u>対処</u></p> <p>fabrctl reset コマンドを実行して Director をリセットした後再度実行してください。</p>

● shutdown サブコマンド

サブコマンドなし

● reset サブコマンド

メッセージ	状況	推定原因・対処
Error: reset: xxx	Director がエラー状態	<p><u>推定原因</u></p> <p>Director がエラー状態</p> <p><u>対処</u></p> <p>Director の状態を確認し、fabrctl restart を実行してください。</p>

● restart サブコマンド

メッセージ	状況	推定原因・対処
Either '--fabric' or '--switch' is valid"	パラメータエラー	<p><u>推定原因</u></p> <p>-F オプションと-S オプションが指定されていない、または両方指定されている</p> <p><u>対処</u></p> <p>コマンドラインオプション・パラメータを確認し、再度実行してください。-F オプションと-S オプションはどちらか一方だけが有効です。</p>
Error: upgrade: xxx	Director がエラー状態	<p><u>推定原因</u></p> <p>Director がエラー状態</p> <p><u>対処</u></p> <p>SmartPDU から Director の電源を再投入してください。</p>

● upload サブコマンド

メッセージ	状況	推定原因・対処
Either '--file' or '--delete' is valid	パラメータエラー	<p><u>推定原因</u></p> <p>-f オプションと-d オプションが両方指定されている</p> <p><u>対処</u></p> <p>コマンドラインオプション・パラメータを確認し、再度実行してください。-f オプションと-d オプションはどちらか一方だけが有効です。</p>
Error: xxx: Invalid upgrade file	ファイル指定エラー	<p><u>推定原因</u></p> <p>-f プションで指定したファイルが不正</p> <p><u>対処</u></p> <p>正しいアップグレードファイルを指定してください。</p>
Error: xxx	ファイル指定エラー	<p><u>推定原因</u></p> <p>-f オプションまたは、-d オプションで指定したファイル名が不正</p> <p><u>対処</u></p> <p>正しいファイル名を指定してください。</p>

● upgrade サブコマンド

メッセージ	状況	推定原因・対処
--index: Required option missing	パラメータエラー	<p><u>推定原因</u></p> <p>-i オプションが指定されていない</p> <p><u>対処</u></p> <p>コマンドラインオプション・パラメータを確認し、再度実行してください。-i オプションは必須です。</p>
Error: xxx: Index out of range	パラメータエラー	<p><u>推定原因</u></p> <p>-i オプションで指定したインデックスが不正</p> <p><u>対処</u></p> <p>アップグレードするファイルのインデックスをリスト表示されたインデックスの中から選択してください。</p>
Error: upgrade: xxx	アップグレードエラー	<p><u>推定原因</u></p> <p>Director の状態がエラー</p> <p><u>対処</u></p> <p>fabrctl restart -f で Director を再起動し、再度事項してください。</p>

7.4 fpclean コマンドのメッセージ一覧

fpclean コマンドの実行で画面に表示されるエラーメッセージは以下になります。

※メッセージ内の xxx は、動的に表示される文字列

- fpclean

メッセージ	状況	推定原因・対処
Required option missing: -r/--remote	パラメータエラー	<p><u>推定原因</u></p> <p>"-r"オプションが指定されていない</p> <p><u>対処</u></p> <p>コマンドラインに"-r"オプションで Director のホスト名または IP アドレスを指定してください。</p>
fpclean: illegal option -- x	パラメータエラー	<p><u>推定原因</u></p> <p>不正なパラメータが指定された</p> <p><u>対処</u></p> <p>コマンドラインオプション・パラメータを確認し、再度実行してください。</p>
ssh: connect to host xxx port 22: No route to host	パラメータエラー	<p><u>推定原因</u></p> <p>"-r"オプションで指定した Director の IP アドレスまたは FQDN の間違い</p> <p><u>対処</u></p> <p>正しい Director の IP アドレスまたは FQDN を指定してください。</p>
root@xx.xx.xx.xx.: Permission denied (publickey,gssapi-keyex,gssapi-with-mic,password).	パスワード間違い	<p><u>推定原因</u></p> <p>パスワードの入力間違い</p> <p><u>対処</u></p> <p>正しいパスワードを確認し、再度実行してください。</p>

7.5 fabradm コマンドのメッセージ一覧

fabradm コマンドの実行で画面に表示されるエラーメッセージは以下になります。

※メッセージ内の xxx は、動的に表示される文字列

● 共通

メッセージ	状況	推定原因・対処
Required option missing: -r/--remote	パラメータエラー	<p><u>推定原因</u></p> <p>"-r"オプションが指定されていない</p> <p><u>対処</u></p> <p>コマンドラインに"-r"オプションで Director のホスト名または IP アドレスを指定してください。</p>
xxx: Unexpected command [restart, shutdown, linkloss, fpclean]	不正なコマンド	<p><u>推定原因</u></p> <p>コマンドラインオプション・パラメータが不正</p> <p><u>対処</u></p> <p>コマンドラインオプション・パラメータを確認し、再度実行してください。サブコマンドは restart, shutdown, linkloss, fpclean の中から指定してください。</p>
fabradm: illegal option -- x	パラメータエラー	<p><u>推定原因</u></p> <p>不正なパラメータが指定された</p> <p><u>対処</u></p> <p>コマンドラインオプション・パラメータを確認し、再度実行してください。</p>
ssh: connect to host xxx port 22: No route to host	パラメータエラー	<p><u>推定原因</u></p> <p>"-r"オプションで指定した Director の IP アドレスまたは FQDN の間違い</p> <p><u>対処</u></p> <p>正しい Director の IP アドレスまたは FQDN を指定してください。</p>
root@xx.xx.xx.xx.: Permission denied (publickey,gssapi-keyex,gssapi-with-mic,password).	パスワード間違い	<p><u>推定原因</u></p> <p>パスワードの入力間違い</p> <p><u>対処</u></p> <p>正しいパスワードを確認し、再度実行してください。</p>

● linkloss

メッセージ	状況	推定原因・対処
One of these is required: -d/--diag, -o/--off, -D/--disable	パラメータエラー	<p><u>推定原因</u></p> <p>"-d"オプション,"-o"オプション,"-D"オプションの複数指定されている</p> <p><u>対処</u></p> <p>"-d"オプション,"-o"オプション,"-D"オプションの中の一つだけ指定して、再実行してください。</p>

