

FUJITSU AI

Zinrai ディープラーニング システム 200M

システム運用管理者ガイド

Copyright 2021 FUJITSU LIMITED

P3KD-1952-02

はじめに

本書は、FUJITSU AI Zinrai ディープラーニング システムを運用するために必要な操作方法、および監視方法を説明します。

本書の読者

本書はZinrai ディープラーニング システムを、運用、監視するシステム管理者を対象として説明します。本書を読むにあたって、以下の知識が必要です。

- Linux に関する基本的な知識
- 運用するサーバー、ストレージ、およびネットワーク機器に関する基本的な知識

本書の構成

本書の構成は以下のとおりです。

- 第1章 概要
本製品の概要について説明します。
- 第2章 システム起動停止手順
サーバー機器の起動と停止に関する手順について説明します。
- 第3章 ネットワーク設定変更
IP アドレス情報、およびポート番号の変更手順について説明します。
- 第4章 Prometheus 監視関連
Prometheus を使用したシステム監視および監視設定について説明します。
- 第5章 障害調査
メッセージやトラブルが発生した場合の対処または調査方法について説明します。
- 第6章 SSL サーバー証明書運用
システムの証明書を入れ替える方法について説明します。
- 第7章 ソフトウェア RAID 管理
RAID 構成情報の取得、および内蔵ストレージの交換について説明します。
- 第8章 バックアップリストア
システムのバックアップおよびリストアについて説明します。
- 付録 A コマンドリファレンス
Docker コンテナに関するコマンドについて説明します。
- 付録 B 外部ストレージ接続
NAS ストレージとの接続について説明します。

● 付録 C ライブメディア作成手順

起動用ライブメディアをカスタマイズする手順について説明します。必要に応じて参照してください。

関連ドキュメント

関連ドキュメントとして、以下のマニュアルがあります。必要に応じて参照してください。

ドキュメント名称	概要
FUJITSU AI Zinrai ディープラーニング システム 200M システム運用管理者ガイド（本書）	Zinrai ディープラーニング システムを運用するために必要な操作方法、および監視方法を記載しています
FUJITSU AI Zinrai Deep Learning System Software License Terms	Zinrai ディープラーニング システムで使用している、OSS（オープンソースソフトウェア）の使用許諾条件、および OSS の一覧を記載しています

- 参考
Prometheus マニュアル（お使いのバージョンを御確認ください）
<https://prometheus.io/docs/>

そのほか、各構成品のマニュアルを必要に応じて参照してください。

本書の表記について

本書では、略称および記号を使用しています。

製品名／技術名の略称について

本書では、製品名／技術名を以下のように表記する場合があります。

- GPU
NVIDIA 社製 GPU
- Web インターフェース
BMC のブラウザ操作画面
（本書では英語表示で説明している部分があります。言語設定で日本語を選択している場合は、適宜読み替えてください。）
- ビデオリダイレクション
BMC のリモート KVM 機能

記号について

本書では、参照先、キー、メニューなどを表記するために、以下のように記号を使用します。

記号	意味
「 」	本書内の参照先のタイトル、画面での設定値を「 」で囲んでいます
『 』	参照先がほかのマニュアルの場合、マニュアル名を『 』で囲んでいます

記号	意味
[]	画面のボタン名、タブ名、ドロップダウンメニュー、およびキーボードのキー名を示します 例：[設定] ダイアログボックス、[ファイル] メニュー、[項目名]、 [OK] ボタン、[Enter] キー
[] - []	画面のメニューとメニューの階層を示します 例：[New] メニューの [Terminal] の場合 [New] - [Terminal]

コマンドインターフェースの説明では、以下のような記号を使用します。

記号	意味
XXXX	値や文字列が可変であることを表す場合、斜体（イタリック体）の文字を使用、または < > で囲みます
<XXXX>	
文字色が黒	ユーザーが入力するコマンドを示します
文字色がグレー	システムが返すレスポンスを示します
[]	パラメーターの選択肢を示します 例：[A B]
␣	半角空白が入ることを示します 例：shutdown ␣ now
↵	[Enter] キーを押すことを示します

また、以下のアイコンを使用します。

■ 注 意

操作や設定を行ううえで制限される内容や注意が必要な内容を記載しています。

○ 備 考

操作や設定を行ううえで知っておくと便利な機能や使い方など、本文を補足する内容を記載しています。

輸出管理規制について

本ドキュメントを輸出または第三者へ提供する場合は、お客様が居住する国および米国輸出管理関連法規等の規制をご確認のうえ、必要な手続きをおとりください。

高度な安全性が要求される用途への使用について

本製品は、一般事務用、パーソナル用、家庭用、通常の産業等の一般的用途を想定して開発・設計・製造されているものであり、原子力施設における核反応制御、航空機自動飛行制御、航空交通管制、大量輸送システムにおける運行制御、生命維持のための医療用機器、兵器システムにおけるミサイル発射制御など、極めて高度な安全性が要求され、仮に当該安全性が確保されない場合、直接生命・身体に対する重大な危険性を伴う用途（以下「ハイセイフティ用途」という）に使用されるよう開発・設計・製造されたものではありません。

お客様は本サービスを必要な安全性を確保する措置を施すことなくハイセイフティ用途に使用しないでください。また、お客様がハイセイフティ用途に本サービスを使用したことにより発生する、お客様または第三者からのいかなる請求または損害賠償に対しても富士通株式会社およびその関連会社は一切責任を負いかねます。

商標

- Linux® は米国及びその他の国における Linus Torvalds の登録商標です。
- Ubuntu は、Canonical Ltd. の登録商標です。
- Docker は、米国およびその他の国における Docker, Inc. の商標 / または登録商標です。
- NVIDIA、CUDA は、米国および / または他国の NVIDIA Corporation の商標および / または登録商標です。
- その他の会社名、各製品名などの固有名詞は、各社の商号、登録商標または商標です。
- その他、会社名、システム名、製品名などには必ずしも商標表示を付記しておりません。

2021 年 12 月 2 版

改版履歷表

版数	日付	変更箇所	変更内容
初版	2021 年 8 月	全体	新規作成
2 版	2021 年 12 月	1.1（変更） 付録 A（変更）	ソフトウェア版数変更
		3.1.2（追加）	IPTABLES 設定変更に関する補足を追加

目次

第1章 概要	8
1.1 ソフトウェア構成	8
1.2 サーバーのボリューム構成	8
1.3 留意事項	9
第2章 システム起動停止手順	10
2.1 システム起動	10
2.1.1 電源ボタンからシステム起動	11
2.1.2 Web インターフェースからシステム起動	11
2.1.3 ビデオリダイレクションからシステム起動	12
2.2 システム停止	13
2.2.1 OS からシステム停止	13
2.2.2 Web インターフェースからシステム停止	14
2.2.3 ビデオリダイレクションからシステム停止	14
第3章 ネットワーク設定変更	15
3.1 IP アドレス情報変更	15
3.1.1 BMC の IP アドレス変更	15
3.1.1.1 BIOS セットアップユーティリティで IP アドレスを変更	15
3.1.1.2 Web インターフェースで IP アドレスを変更	17
3.1.2 Zinrai ディープラーニング システムの IP アドレス変更	19
3.1.3 Docker 内部 LAN の IP アドレス変更	24
3.2 ポート番号変更	26
第4章 Prometheus 監視関連	27
4.1 監視システムの起動	27
4.1.1 Prometheus 起動	27
4.1.2 Prometheus 起動確認	27
4.1.3 Prometheus 停止	28
4.1.4 Prometheus 再開	28
4.2 監視システム管理画面	28
4.2.1 監視システム管理画面へのログイン	28
4.3 監視システム設定	29
4.4 アラート設定	30
4.4.1 アラートルール	30
4.4.2 メールアラート設定	30
4.4.3 IPTABLES 設定追加	30
4.4.4 アラート機能有効化	30
第5章 障害調査	31
5.1 トラブルシューティング	31
5.2 ディスクの寿命確認	32
5.2.1 寿命確認手順	32

5.2.2	書込みデータ量の確認方法	32
第 6 章	SSL サーバー証明書運用	33
6.1	認証局によって正式に署名された SSL サーバー証明書を利用する場合	33
6.2	自己署名の SSL サーバー証明書を利用する場合	33
第 7 章	ソフトウェア RAID 管理	34
7.1	RAID 構成情報の取得	34
7.2	RAID ボリュームの詳細情報の取得	35
7.3	内蔵ストレージの交換	36
7.3.1	活性交換	36
7.3.1.1	RAID 構成から内蔵ストレージを切り離す	37
7.3.1.2	RAID 構成に内蔵ストレージを組み込む	38
7.3.2	システム停止交換	41
7.3.2.1	RAID 構成から内蔵ストレージを切り離す	41
7.3.2.2	RAID 構成に内蔵ストレージを組み込む	43
第 8 章	バックアップリストア	46
8.1	システムバックアップ	47
8.1.1	ファイルシステム情報採取	48
8.1.2	起動用ライブメディアから OS を起動	50
8.1.3	システムバックアップ実施	51
8.2	システムリストア	53
8.2.1	ライブメディアから OS を起動	54
8.2.2	ファイルシステム作成	55
8.2.3	システムリストア実施	59
付録 A	コマンドリファレンス	66
付録 B	外部ストレージ接続	69
付録 C	ライブメディア作成手順	70
C.1	作成環境の事前要件	70
C.2	必要なツールのインストール	70
C.3	ひな型メディアの入手と作成準備	71
C.3.1	ISO イメージファイルのダウンロード	71
C.3.2	ISO イメージファイルの展開	71
C.3.3	作業環境の準備	72
C.4	ライブメディア環境のカスタマイズ	73
C.4.1	パッケージの追加と削除	73
C.4.2	クリーンナップ処理	74
C.5	ライブメディアイメージの作成	74
C.5.1	ファイルシステムの構築	74
C.5.2	メディアイメージの書き出し	75

第 1 章 概要

1.1 ソフトウェア構成

本製品のソフトウェア構成を以下に示します。

表 1：ソフトウェア構成

名称	
ホスト OS 環境	Ubuntu 20.04.3 LTS
	nvidia-driver 470.57.02
	nvidia-docker2 2.6.0-1
	Docker-CE 20.10.10
フレームワーク	Docker イメージ (TensorFlow 21.10-tf2-py3)
	Docker イメージ (PyTorch 21.10-py3)
	Docker イメージ (MxNet 21.09-py3)
監視サーバー	Prometheus 2.30.1

1.2 サーバーのボリューム構成

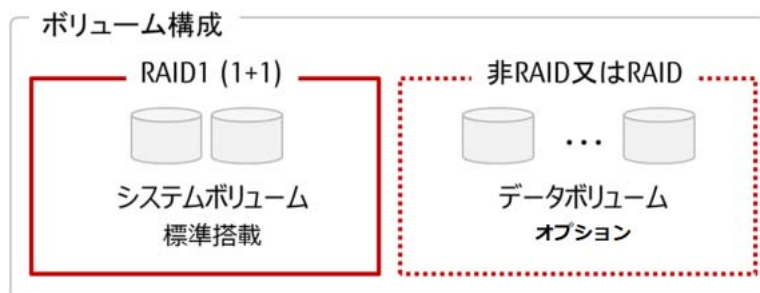
- システムボリューム
システム管理領域（OS 環境、コンテナ環境、ログ、そのほかソフトウェアを格納）
- データボリューム（オプション）
ユーザー領域（データセット、学習モデルなどのユーザーデータを格納）としてユーザーが利用可能な領域です。内蔵 SSD（オプション）または、外部ストレージを利用してください。外部ストレージの接続方法については、[「付録 B 外部ストレージ接続」](#)を参照してください。

システム管理者はディスク使用状況を適宜確認し、使用状況に応じて不要なデータを削除するなどの対処をしてください。

■ 注 意

本書では日常的なバックアップについては記載していません。御利用の環境に合わせた方法で、お客様自身でシステムボリューム、データボリュームとも必ず実施してください。

図 1 : Zinrai ディープラーニング システム サーバーのボリューム構成



1.3 留意事項

- Zinrai ディープラーニング システムで使用しているハードウェアおよびソフトウェアについて、お客様の判断による設定変更で発生した影響については、富士通では保証いたしていません。
- 正規のシステム停止手順ではない方法でサーバーの電源を切断した場合、保存されているデータが、破損または失われる可能性があります。電源を切断する場合は、[「2.2 システム停止」\(P.13\)](#)の手順に従ってください。

第 2 章 システム起動停止手順

サーバー機器の操作には、ターミナルエミュレータおよび Web ブラウザーを使用します。
お客様環境の管理 LAN などにサーバーと PC を接続して、サーバーへアクセスしてください。



2.1 システム起動

以下のどれかの操作でシステムを起動します。御利用環境に応じてシステムを起動してください。

- [「2.1.1 電源ボタンからシステム起動」\(P.11\)](#)
- [「2.1.2 Web インターフェースからシステム起動」\(P.11\)](#)
- [「2.1.3 ビデオリダイレクションからシステム起動」\(P.12\)](#)

■ システム起動の確認

システムの起動状態は、ビデオリダイレクション機能の表示画面で確認します。
システム起動処理が完了すると、ビデオリダイレクション画面上にログインプロンプトが表示されます。

```
Ubuntu 20.04.3 LTS <hostname> tty1
<hostname> login:
```

○ 備考

ログイン方法は、製品添付の DVD に格納されている『はじめに』を参照してください。

2.1.1 電源ボタンからシステム起動

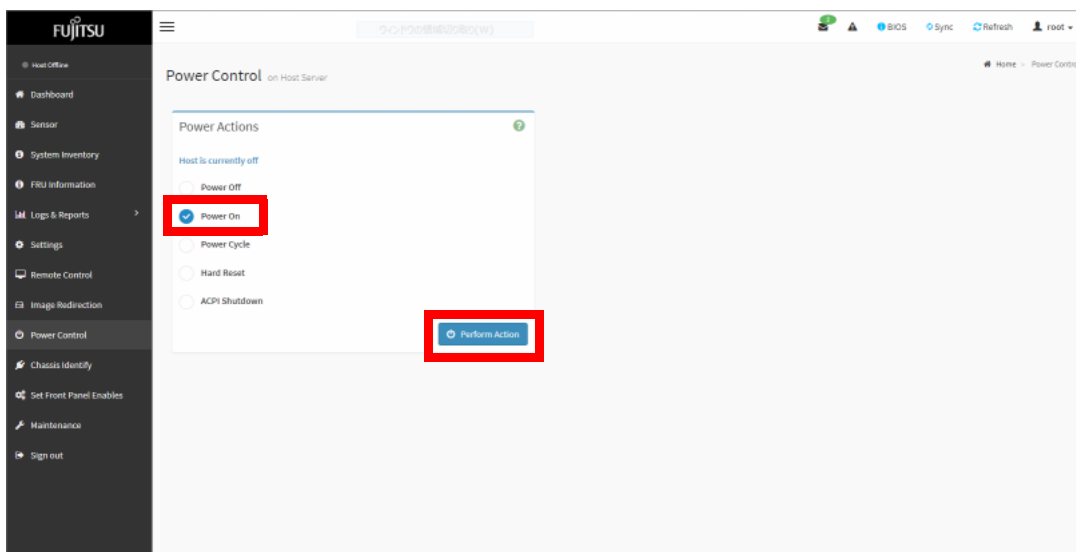
装置前面の電源ボタンを押し、システムを起動します。



電源ボタン

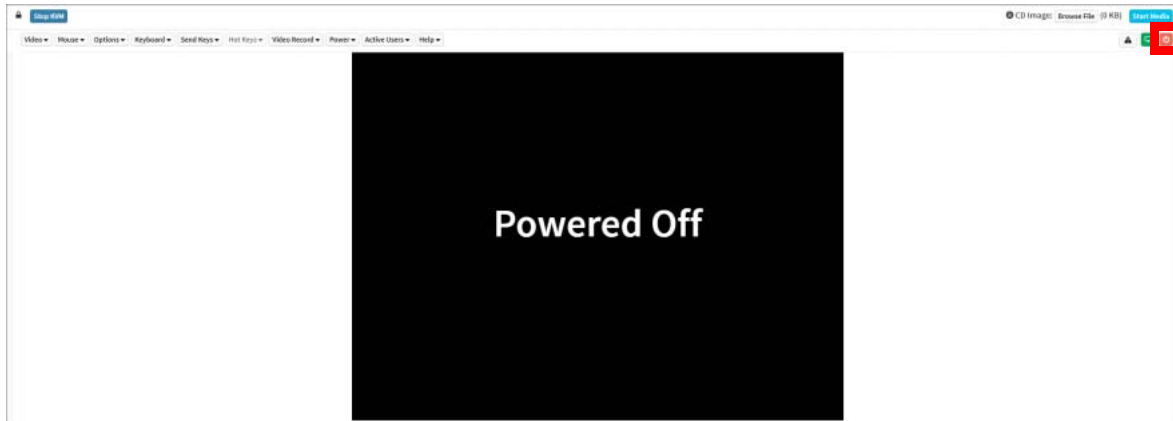
2.1.2 Web インターフェースからシステム起動

Web インターフェースの [Power Control] メニューから「Power On」を選択し、[Perform Action] ボタンをクリックします。Are you sure to perform this operation? のポップアップが表示されたら [OK] をクリックします



2.1.3 ビデオリダイレクションからシステム起動

Web インターフェースの [Remote Control] - [Launch KVM (HTML5)] メニューを選択し、ビデオリダイレクションを起動します。
右上の橙色の電源ボタンアイコンをクリックします。



2.2 システム停止

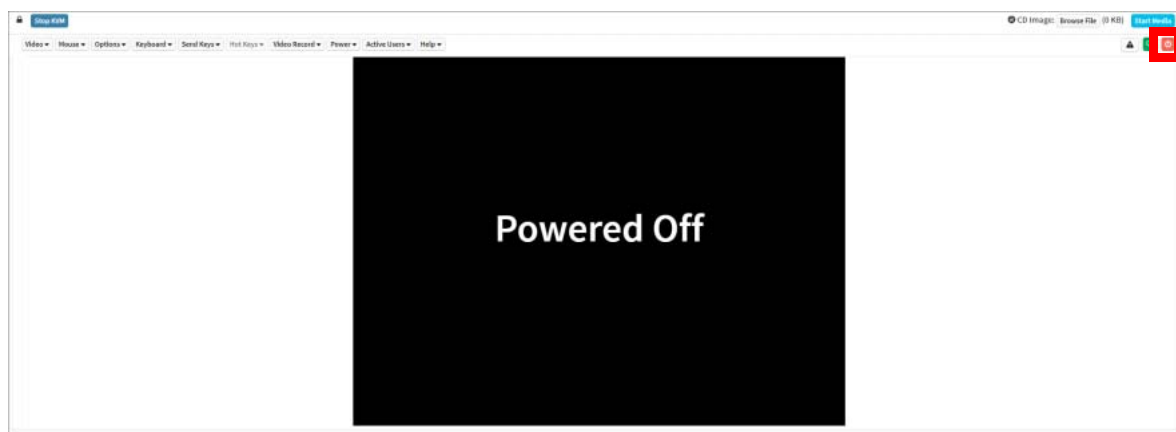
以下のどれかの操作でシステムを停止します。御利用環境に応じてシステムを停止してください。

- [「2.2.1 OS からシステム停止」 \(P.13\)](#)
- [「2.2.2 Web インターフェースからシステム停止」 \(P.14\)](#)
- [「2.2.3 ビデオリダイレクションからシステム停止」 \(P.14\)](#)

■ システム停止の確認

以下のどれかの方法で、システムが停止したことを確認します。

- Web インターフェースのダッシュボード
Dashboard の Device information の Device Power Status が off になる
- ビデオリダイレクション機能のメニュー
メニューバー右端の電源ボタンアイコンが橙色になる



- サーバー前面部の LED
オンボード LAN のステータス LED 以外すべて消灯する

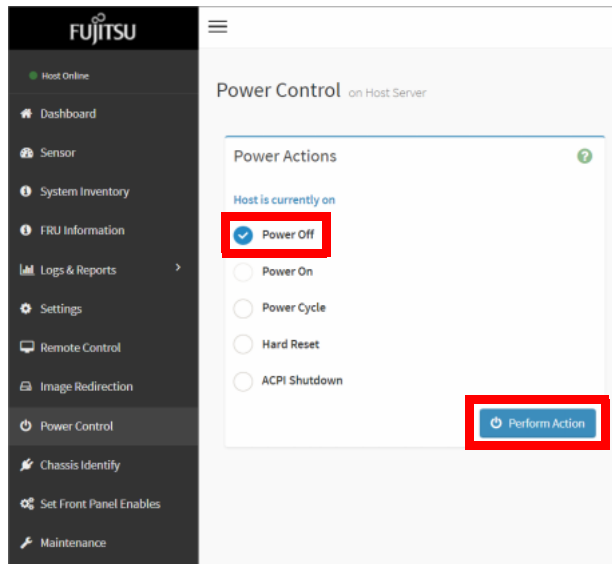
2.2.1 OS からシステム停止

ターミナルから、sudo 権限で shutdown コマンドを実行し、システムを停止します。
通常はこの方法でシステムを停止してください。

```
# shutdown _ now
```

2.2.2 Web インターフェースからシステム停止

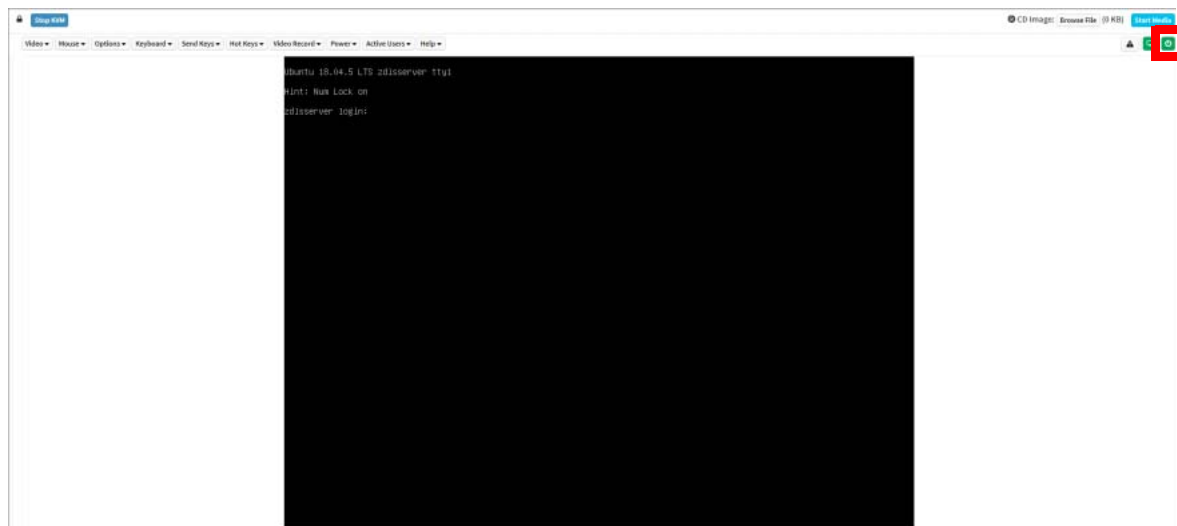
Web インターフェースの [Power Control] メニューから「Power Off」を選択し、[Perform Action] ボタンをクリックします。



2.2.3 ビデオリダイレクションからシステム停止

Web インターフェースの [Remote Control] - [Launch KVM (HTML5)] メニューを選択し、ビデオリダイレクションを起動します。

右上の緑色の電源ボタンアイコンをクリックします。



第 3 章 ネットワーク設定変更

3.1 IP アドレス情報変更

3.1.1 BMC の IP アドレス変更

Web インターフェースをもつシステムコントローラー（BMC）に設定した、工場出荷時の IP アドレスを変更する手順を説明します。

BMC に設定した IP アドレスの変更方法は、以下の 2 通りあります。御利用環境に応じて選択してください。

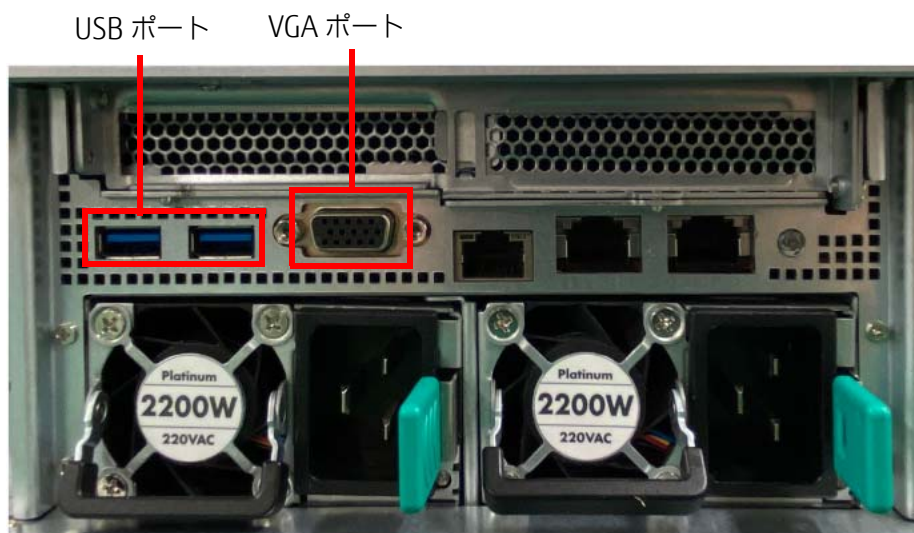
- [「3.1.1.1 BIOS セットアップユーティリティで IP アドレスを変更」 \(P.15\)](#)
- [「3.1.1.2 Web インターフェースで IP アドレスを変更」 \(P.17\)](#)

3.1.1.1 BIOS セットアップユーティリティで IP アドレスを変更

1. コンソールの準備

装置背面の VGA ポート、USB ポートに、ディスプレイおよびキーボードを接続します。

図 2：装置背面図（VGA/USB ポート）



2. BIOS セットアップユーティリティの起動

装置の電源を投入するか、または再起動し、POST 実行中に [F2] キーを押して BIOS セットアップユーティリティを起動します。

BIOS セットアップユーティリティを起動する前にシステムが起動してしまった場合は、装置を再起動してください。



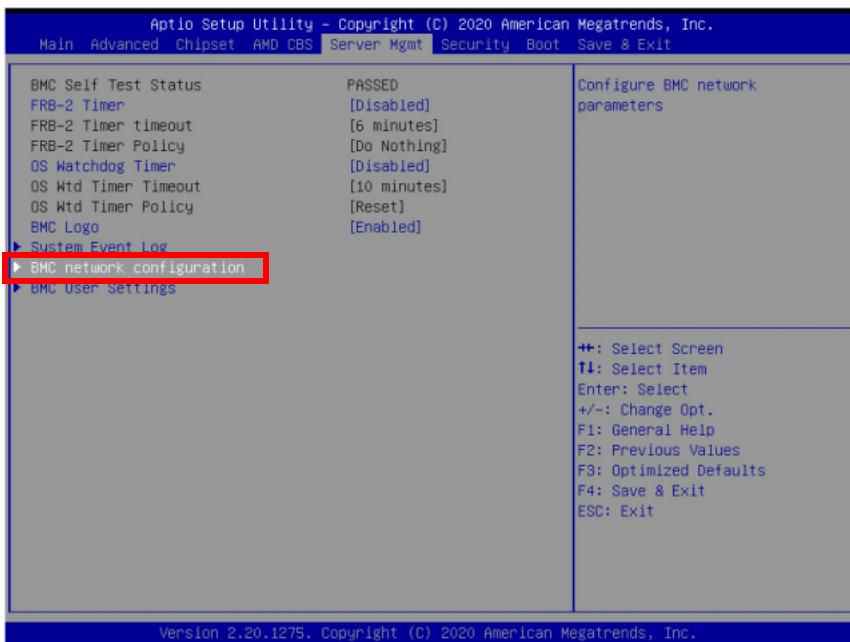
3. Web インターフェースネットワークの設定

[Server Mgmt] メニューで「BMC network configuration」を選択し、環境に合わせて以下のネットワーク設定を行ってください。

Configuration Address source : Static (DHCP 運用時は DynamicBmcDhcp)

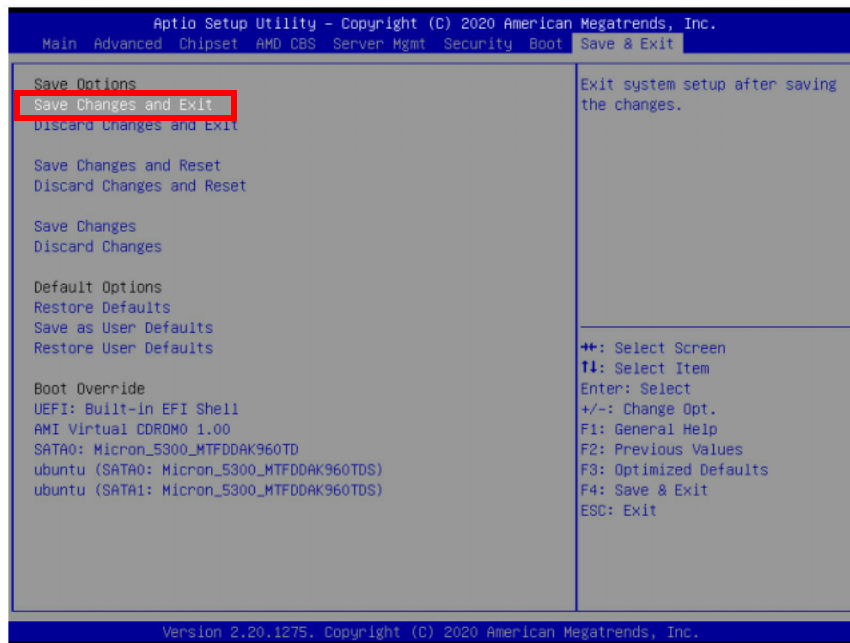
IP address : xxx.xxx.xxx.xxx (初期値 :192.168.1.11)

Subnet mask : yyy.yyy.yyy.yyy (初期値 :255.255.255.0)



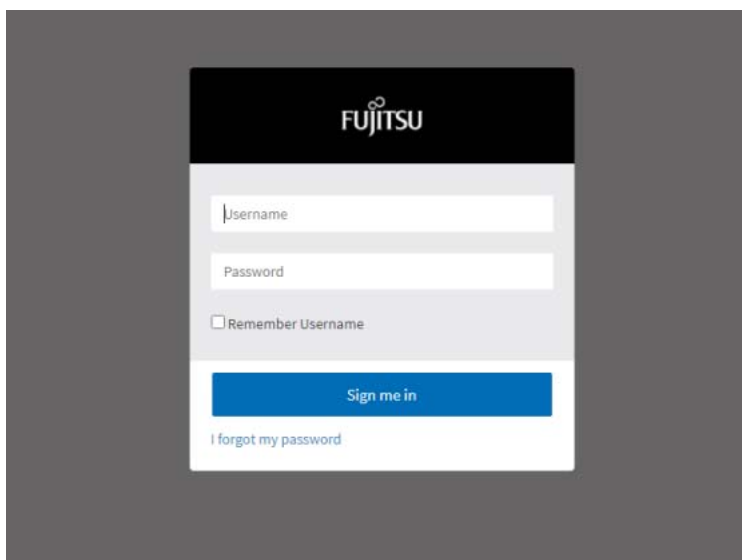
4. 設定の保存

[Save & Exit] メニューで「Save Changes and Exit」を選択し、設定を保存してシステムのセットアップを終了します。

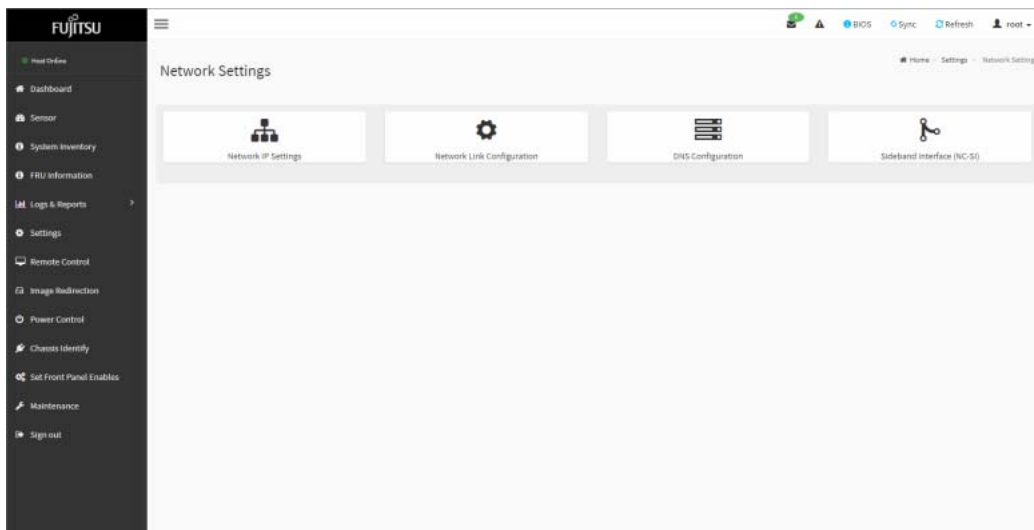


3.1.1.2 Web インターフェースで IP アドレスを変更

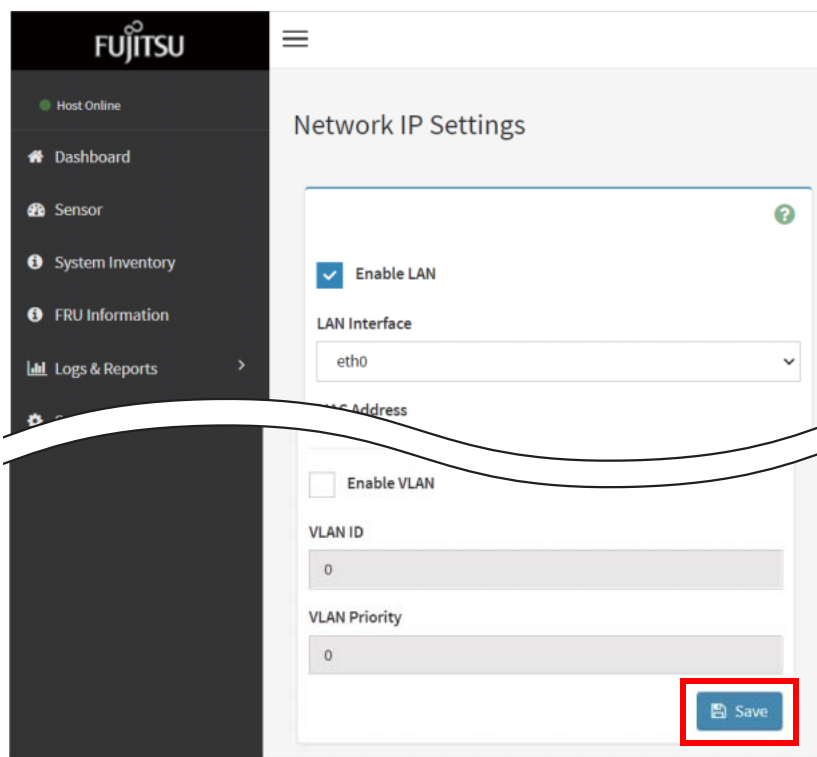
1. Web ブラウザーに「https://192.168.1.11/」を入力して、Web インターフェースにアクセスします。
ログイン画面が表示されます。



2. ユーザー名とパスワードを入力します。
Web インターフェースの初期画面が表示されます。



3. 左側のメニューバーから、[Settings] - [Network Setting] - [Network IP Settings] を選択し、ネットワーク設定メニューにアクセスします。
4. Network IP Settings に設定するネットワーク情報を入力し、ページ最下部にある [Save] ボタンをクリックします。



5. 変更した IP アドレスにアクセスし、ログイン画面が表示されることを確認します。

3.1.2 Zinrai ディープラーニング システムの IP アドレス 変更

工場出荷時は、ホスト OS に IP アドレスを設定しています。

ここでは、工場出荷時にホスト OS に設定した IP アドレス（192.168.1.10）を、御利用のネットワーク環境に応じた IP アドレスに変更する手順を説明します。

ホスト OS の管理ユーザーでログインしたあと、`sudo su` コマンドで root ユーザーに変更してください。すでに Zinrai ディープラーニング システムを運用しており、ネットワーク環境を変更している場合は、設定されている内容に応じた変更を実施してください。

以下のファイルと IPTABLES 設定を変更します。

- /etc/netplan/00-installer-config.yaml
- /etc/prometheus/prometheus.yml
- /etc/grafana/grafana.ini

1. /etc/netplan/00-installer-config.yaml のバックアップ
工場出荷時のファイル「00-installer-config.yaml」を、以下のコマンドでバックアップします。

```
cp -p /etc/netplan/00-installer-config.yaml /etc/netplan/00-installer-config.yaml.org
```

2. /etc/netplan/00-installer-config.yaml の修正
お客様の環境に合わせて、「00-installer-config.yaml」を修正します。

```
# This is the network config written by 'subiquity'
network:
  ethernets:
    enp97s0f0:
      addresses:
        - <HostOS IP address>/<Subnet mask>
      gateway4: <Gateway IP address>
      nameservers:
        addresses: [<DNS IP address>]
        search: []
    enp97s0f1:
      dhcp4: true
  version: 2
```

3. /etc/prometheus/prometheus.yml 変更

行	変更前	変更後
11	- targets: ['192.168.1.10:9093']	- targets: ['<HostOS IP address>:9093']
28	- targets: ['192.168.1.10:9090']	- targets: ['<HostOS IP address>:9090']
35	- targets: ['192.168.1.10:9100']	- targets: ['<HostOS IP address>:9100']
46	- targets: ['192.168.1.10:9290']	- targets: ['<HostOS IP address>:9290']
57	- targets: ['192.168.1.10:9116']	- targets: ['<HostOS IP address>:9116']
67	- targets: ['192.168.1.10:3903']	- targets: ['<HostOS IP address>:3903']
78	- targets: ['192.168.1.10:9400']	- targets: ['<HostOS IP address>:9400']

4. /etc/grafana/grafana.ini 変更

行	変更前	変更後
35	http_addr = 192.168.1.10	http_addr = <HostOS IP address>

5. IPTABLES 設定変更

iptables の第 3 引数のインデックス番号または順番が以下の内容と異なる場合は、適宜実際の値に読み替えてください。

- (1) Filter テーブルの INPUT チェインに設定した DROP ポリシーを、ACCEPT ポリシーに変更します。

```
# iptables -P INPUT ACCEPT
```

- (2) Filter テーブルの INPUT チェインの設定内容を変更します。

```
# iptables -R INPUT 7 -p tcp -s 0.0.0.0/0 -d <HostOS IP address>/32 -m state --state NEW -m tcp --dport 22 -j ACCEPT
# iptables -R INPUT 8 -p udp -s 0.0.0.0/0 -d <HostOS IP address>/32 -m state --state NEW -m udp --dport 123 -j ACCEPT
# iptables -R INPUT 10 -p udp -s 0.0.0.0/0 -d <HostOS IP address>/32 -m state --state NEW -m udp --dport 162 -j ACCEPT
# iptables -R INPUT 12 -p tcp -s 0.0.0.0/0 -d <HostOS IP address>/32 ! --dport 3000 -j PROMETHEUS
# iptables -R INPUT 13 -p tcp -s 0.0.0.0/0 -d <HostOS IP address>/32 --dport 3000 -j GRAFANA
#
```

- (3) Filter テーブルの GRAFANA チェインの設定内容を変更します。

```
# iptables -R GRAFANA 1 -p tcp -s 0.0.0.0/0 -d <HostOS IP address>/32 --dport 3000 -m state --state NEW -m tcp -j ACCEPT
#
```

- (4) Filter テーブルの PROMETHEUS チェインの設定内容を変更します。

```
# iptables -R PROMETHEUS 1 -p tcp -s <HostOS IP address>/32 -d <HostOS IP address>/32 --dport 3903 -m state --state NEW -m tcp -j ACCEPT
# iptables -R PROMETHEUS 2 -p tcp -s 172.17.0.0/16 -i docker0 -d <HostOS IP address>/32 --dport 3903 -m state --state NEW -m tcp -j ACCEPT
# iptables -R PROMETHEUS 3 -p tcp -s <HostOS IP address>/32 -d <HostOS IP address>/32 --dport 9100 -m state --state NEW -m tcp -j ACCEPT
# iptables -R PROMETHEUS 4 -p tcp -s 172.17.0.0/16 -i docker0 -d <HostOS IP address>/32 --dport 9100 -m state --state NEW -m tcp -j ACCEPT
# iptables -R PROMETHEUS 5 -p tcp -s <HostOS IP address>/32 -d <HostOS IP address>/32 --dport 9116 -m state --state NEW -m tcp -j ACCEPT
# iptables -R PROMETHEUS 6 -p tcp -s 172.17.0.0/16 -i docker0 -d <HostOS IP address>/32 --dport 9116 -m state --state NEW -m tcp -j ACCEPT
# iptables -R PROMETHEUS 7 -p tcp -s <HostOS IP address>/32 -d <HostOS IP address>/32 --dport 9290 -m state --state NEW -m tcp -j ACCEPT
# iptables -R PROMETHEUS 8 -p tcp -s 172.17.0.0/16 -i docker0 -d <HostOS IP address>/32 --dport 9290 -m state --state NEW -m tcp -j ACCEPT
# iptables -R PROMETHEUS 9 -p tcp -s <HostOS IP address>/32 -d <HostOS IP address>/32 --dport 9400 -m state --state NEW -m tcp -j ACCEPT
# iptables -R PROMETHEUS 10 -p tcp -s 172.17.0.0/16 -i docker0 -d <HostOS IP address>/32 --dport 9400 -m state --state NEW -m tcp -j ACCEPT
# iptables -R PROMETHEUS 11 -p tcp -s <HostOS IP address>/32 -d <HostOS IP address>/32 --dport 9090 -m state --state NEW -m tcp -j ACCEPT
# iptables -R PROMETHEUS 12 -p tcp -s 172.17.0.0/16 -i docker0 -d <HostOS IP address>/32 --dport 9090 -m state --state NEW -m tcp -j ACCEPT
#
```

(5) NAT テーブルの PREROUTING チェインの設定内容を変更します。

```
# iptables -t nat -R PREROUTING 1 -p tcp -s 172.17.0.0/16 -
i docker0 -d <HostOS IP address>/32 --dport 9090 -j DOCKER
# iptables -t nat -R PREROUTING 2 -p tcp -s <HostOS IP address>/32 -
d <HostOS IP address>/32 --dport 9090 -j DOCKER
#
```

(6) 変更内容を保存します。

```
# /etc/init.d/netfilter-persistent save
* Saving netfilter rules...
run-parts: executing /usr/share/netfilter-persistent/plugins.d/15-ip4tables
save
run-parts: executing /usr/share/netfilter-persistent/plugins.d/25-ip6tables
save
[ OK ]

# /etc/init.d/netfilter-persistent reload
* Loading netfilter rules...
run-parts: executing /usr/share/netfilter-persistent/plugins.d/15-ip4tables
start
run-parts: executing /usr/share/netfilter-persistent/plugins.d/25-ip6tables
start
[ OK ]

#
```

(7) 変更内容を確認します。

```
# iptables -nvL --line-numbers
# iptables -t nat -nvL --line-numbers
```

6. システム再起動

システムを再起動して、IP アドレスの変更をシステムに反映します。

```
# shutdown -r now
```

7. ポリシー変更

Filter テーブルの INPUT チェインに設定するポリシーを DROP に戻します。

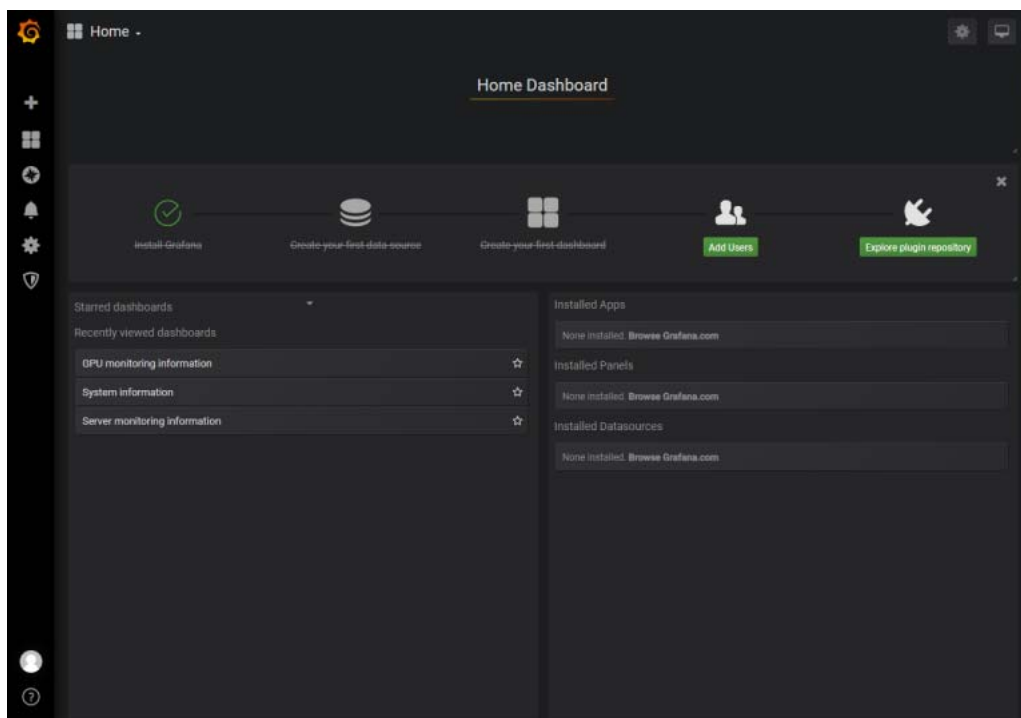
```
# iptables -P INPUT DROP
# /etc/init.d/netfilter-persistent save
* Saving netfilter rules...
run-parts: executing /usr/share/netfilter-persistent/plugins.d/15-ip4tables
save
run-parts: executing /usr/share/netfilter-persistent/plugins.d/25-ip6tables
save
[ OK ]

# /etc/init.d/netfilter-persistent reload
* Loading netfilter rules...
run-parts: executing /usr/share/netfilter-persistent/plugins.d/15-ip4tables
start
run-parts: executing /usr/share/netfilter-persistent/plugins.d/25-ip6tables
start
[ OK ]

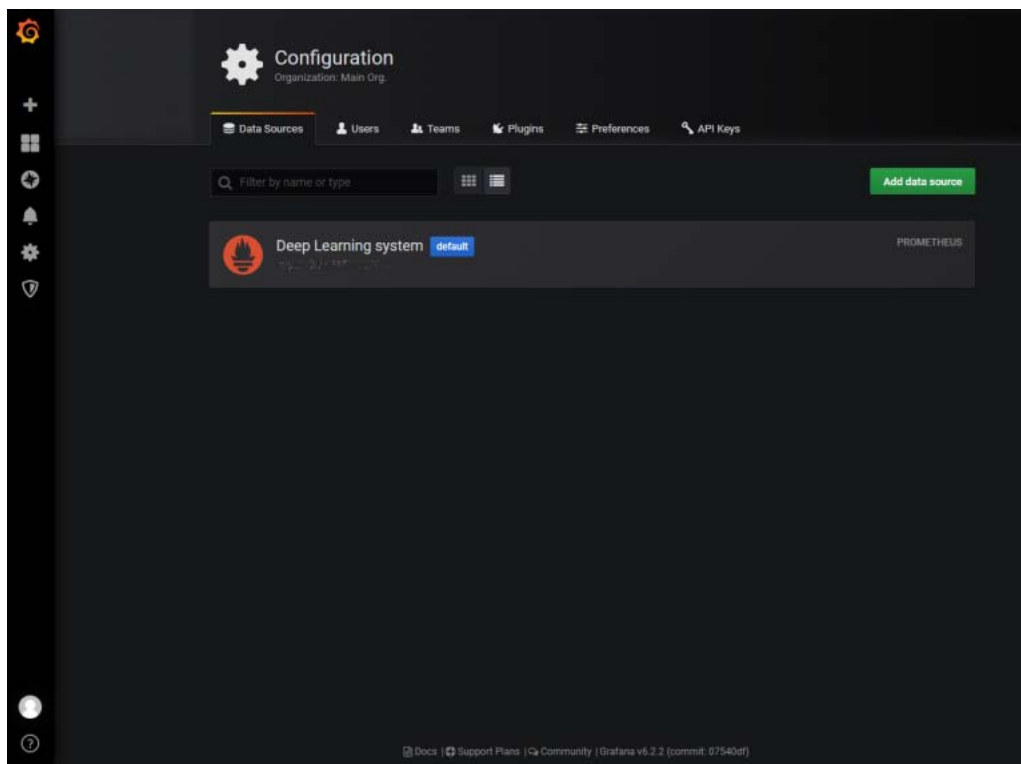
#
```

8. Grafana 設定変更

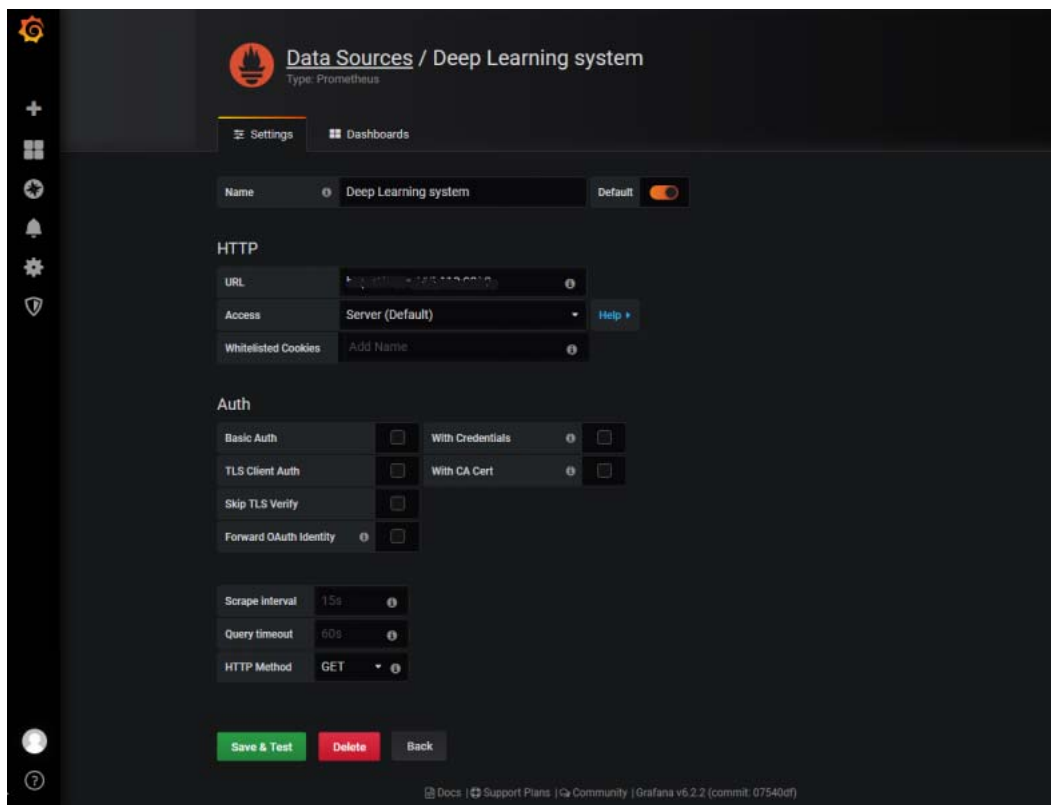
- (1) Web ブラウザーに、「https://<HostOS IP address>:3000」を入力し、Grafana の Home Dashboard 画面にアクセスします。



- (2) 左側メニューの [Configuration] - [Data Sources] を開きます。



- (3) [Deep Learning system] を開き、HTTP 項目の URL に設定した Host OS の IP アドレスを入力し、ページ下部にある [Save & Test] ボタンをクリックします。



Prometheus との連携に問題がなければ、以下のポップアップが表示されます。



- (4) Web ブラウザーに、「https://<HostOS IP address>:3000」を入力し、Grafana から Prometheus との通信ができることを確認します。

3.1.3 Docker 内部 LAN の IP アドレス変更

工場出荷時は、Docker 内部 LAN の IP アドレスに、「172.17.0.1/16」を設定しています。御利用のネットワーク環境により変更が必要な場合は、Docker 内部 LAN の IP アドレスを変更してください。

ホスト OS の管理ユーザーでログインしたあと、`sudo su` により root ユーザーに変更してください。ここでは例として、`docker0` に「182.18.0.1/16」を設定し、Zinrai ディープラーニングシステムに反映する手順を説明します。

1. Docker コンテナの停止

以下は、Prometheus コンテナを停止し、Docker コンテナが起動していない状態にしています。

```
# docker stop prometheus
prometheus
# docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED
STATUS              PORTS              NAMES
```

2. Docker 内部 LAN の定義の変更

/etc/docker/daemon.json ファイルを変更します。

行	変更前	変更後
2	"bip": "172.17.0.1/16",	"bip": "182.18.0.1/16",

3. Docker に反映

Docker サービスを再起動して反映します。

```
# systemctl daemon-reload
# systemctl restart docker.service
# docker network inspect bridge <記載省略>

    "IPAM": {
      "Driver": "default",
      "Options": null,
      "Config": [
        {
          "Subnet": "182.18.0.0/16",
          "Gateway": "182.18.0.1"
        }
      ]
    } <記載省略>

#
```

4. IPTABLES 設定変更

- (1) Filter テーブルの PROMETHEUS チェインに設定している内容を変更します。

```
# iptables -R PROMETHEUS 2 -p tcp -s 182.18.0.0/16 -i docker0 -  
d <HostOS IP address>/32 --dport 3903 -m state --state NEW -m tcp -  
j ACCEPT  
# iptables -R PROMETHEUS 4 -p tcp -s 182.18.0.0/16 -i docker0 -  
d <HostOS IP address>/32 --dport 9100 -m state --state NEW -m tcp -  
j ACCEPT  
# iptables -R PROMETHEUS 6 -p tcp -s 182.18.0.0/16 -i docker0 -  
d <HostOS IP address>/32 --dport 9116 -m state --state NEW -m tcp -  
j ACCEPT  
# iptables -R PROMETHEUS 8 -p tcp -s 182.18.0.0/16 -i docker0 -  
d <HostOS IP address>/32 --dport 9290 -m state --state NEW -m tcp -  
j ACCEPT  
# iptables -R PROMETHEUS 10 -p tcp -s 182.18.0.0/16 -i docker0 -  
d <HostOS IP address>/32 --dport 9400 -m state --state NEW -m tcp -  
j ACCEPT  
# iptables -R PROMETHEUS 12 -p tcp -s 182.18.0.0/16 -i docker0 -  
d <HostOS IP address>/32 --dport 9090 -m state --state NEW -m tcp -  
j ACCEPT  
#
```

- (2) NAT テーブルの PREROUTING チェインに設定している内容を変更します。

```
# iptables -t nat -R PREROUTING 1 -p tcp -s 182.18.0.0/16 -  
i docker0 -d <HostOS IP address>/32 --dport 9090 -j DOCKER  
#
```

- (3) 設定を反映します。

```
# /etc/init.d/netfilter-persistent save  
* Saving netfilter rules...  
run-parts: executing /usr/share/netfilter-persistent/plugins.d/15-ip4tables  
save  
run-parts: executing /usr/share/netfilter-persistent/plugins.d/25-ip6tables  
save  
[ OK ]  
  
# /etc/init.d/netfilter-persistent reload  
* Loading netfilter rules...  
run-parts: executing /usr/share/netfilter-persistent/plugins.d/15-ip4tables  
start  
run-parts: executing /usr/share/netfilter-persistent/plugins.d/25-ip6tables  
start  
[ OK ]  
#
```

- (4) 設定を確認します。

```
# iptables -nvL  
# iptables -t nat -nvL
```

5. 動作確認

Grafana にログイン (<https://<HostOS IP Address>:3000>) し、各監視メトリクスが収集できていることを確認してください。

3.2 ポート番号変更

ここでは例として、工場出荷時に設定されている Grafana のポート番号（3000）を、「8000」に変更する手順を説明します。御利用のネットワーク環境により変更が必要な場合は、Grafana のポート番号の設定を変更してください。

ホスト OS の管理ユーザーでログインしたあと、`sudo su` により root ユーザーに変更してください。

1. Grafana 設定ファイルを変更

`/etc/grafana/grafana.ini` ファイルの以下の箇所を変更します

行	変更前	変更後
38	<code>http_port = 3000</code>	<code>http_port = 8000</code>

2. Grafana サービスに反映

```
# systemctl restart grafana-server.service
# systemctl status grafana-server.service
● grafana-server.service - Grafana instance
   Loaded: loaded (/usr/lib/systemd/system/grafana-server.service; enabled;
   vendor preset: enabled)
   Active: active (running) since Wed 2019-05-29 17:03:19 JST; 6s ago
           <記載省略>
May 29 17:03:19 <hostname> grafana-server[45680]: t=2019-05-29T17:03:19+0900
lvl=info msg="HTTP Server Listen" logger=http.server address=0.0.0.0:8000
protocol=https subUrl= socket=...
```

3. IPTABLES 設定変更

(1) Filter テーブルの INPUT チェインに設定した内容を変更します。

```
# iptables -R INPUT 12 -p tcp -s 0.0.0.0/0 -d <HostOS IP address>/
32 ! --dport 8000 -j PROMETHEUS
# iptables -R INPUT 13 -p tcp -s 0.0.0.0/0 -d <HostOS IP address>/
32 --dport 8000 -j GRAFANA
#
```

(2) Filter テーブルの GRAFANA チェインに設定した内容を変更します。

```
# iptables -R GRAFANA 1 -p tcp -s 0.0.0.0/0 -d <HostOS IP address>/
32 --dport 8000 -m state --state NEW -m tcp -j ACCEPT
#
```

(3) 設定を反映します。

```
# /etc/init.d/netfilter-persistent save
* Saving netfilter rules...
run-parts: executing /usr/share/netfilter-persistent/plugins.d/15-ip4tables
save
run-parts: executing /usr/share/netfilter-persistent/plugins.d/25-ip6tables
save
[ OK ]

# /etc/init.d/netfilter-persistent reload
* Loading netfilter rules...
run-parts: executing /usr/share/netfilter-persistent/plugins.d/15-ip4tables
start
run-parts: executing /usr/share/netfilter-persistent/plugins.d/25-ip6tables
start
[ OK ]

#
```

4. 動作確認

Grafana にログイン (`https://<HostOS IP Address>:8000`) し、各監視メトリクスが収集できていることを確認してください。

第 4 章 Prometheus 監視関連

Zinrai ディープラーニング システムでは、Prometheus を使った監視システムを提供しています。工場出荷時は、サンプルの監視項目を設定しています。利用要件に合わせて、各項目を変更してください。

4.1 監視システムの起動

御購入直後は監視機能が停止状態になっています。起動／停止方法は以下のとおりです。

○ 備考

- 画面上の“¥”は御利用の PC 環境によっては“\”（バックスラッシュ）で表示されることもあります。
- Prometheus の起動オプションを変更する場合は、docker kill コマンドでコンテナを停止後、docker rm コマンドでコンテナを削除してください。

4.1.1 Prometheus 起動

docker run コマンドを実行して、Prometheus を起動します。

```
# docker run -d -p 9090:9090 --restart=always --name prometheus ¥
-v /etc/prometheus/prometheus.yml:/etc/prometheus/prometheus.yml:ro ¥
-v /etc/alertmanager/rules.d:/etc/alertmanager/rules.d:ro ¥
-v /var/log/metrics:/prometheus/data:rw ¥
-v /etc/localtime:/etc/localtime:ro ¥
prom/prometheus-linux-amd64 ¥
--config.file="/etc/prometheus/prometheus.yml" ¥
--storage.tsdb.path="/prometheus/data" ¥
--web.console.libraries="/usr/share/prometheus/console_libraries" ¥
--web.console.templates="/usr/share/prometheus/consoles"
#
```

4.1.2 Prometheus 起動確認

docker ps -a コマンドを実行して、NAMEs が prometheus のコンテナの STATUS を確認してください。“Up xx Seconds”などが表示されたら正常に起動しています。

```
# docker ps -a
CONTAINER ID        IMAGE               PORTS              COMMAND              NAMES              CREATED
STATUS
d18c63987251       prom/prometheus-linux-amd64  "/bin/prometheus --c..."  13 hours ago
Up 6 minutes       0.0.0.0:9090->9090/tcp    prometheus
```

4.1.3 Prometheus 停止

docker ps コマンドで Prometheus コンテナの CONTAINER ID を確認し、docker stop コマンドで指定してください。

再度、docker ps コマンドを実行します。何も表示されなければ、Prometheus は停止しています。

```
# docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED
STATUS
d18c63987251      prom/prometheus-linux-amd64  "/bin/prometheus --c..."  8 hours ago
Exited(0) 9 minutes ago    prometheus
# docker stop <CONTAINER ID>
<CONTAINER ID>
# docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED
STATUS
#
```

4.1.4 Prometheus 再開

停止中の Prometheus を再開するには、docker start コマンドを実行します。

```
# docker start <CONTAINER ID>
<CONTAINER ID>
#
```

4.2 監視システム管理画面

監視システム管理画面の利用について説明します。監視項目の設定追加、変更、および監視状況の確認は、監視システム管理画面から行います。

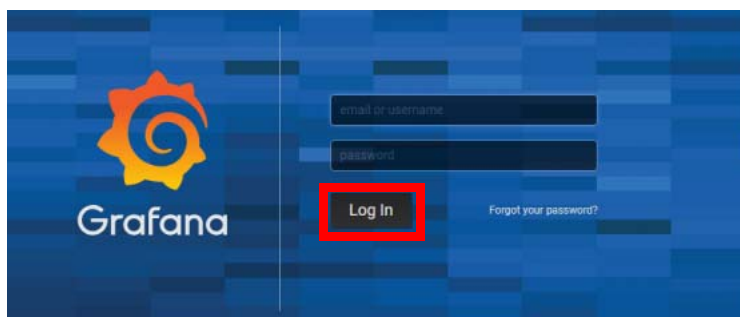
4.2.1 監視システム管理画面へのログイン

監視システム管理画面には、以下の URL でアクセスします。

ポート番号を初期値（3000）から変更している場合は、変更したポート番号でアクセスしてください。

<https://<HostOS IP Address>:3000/>

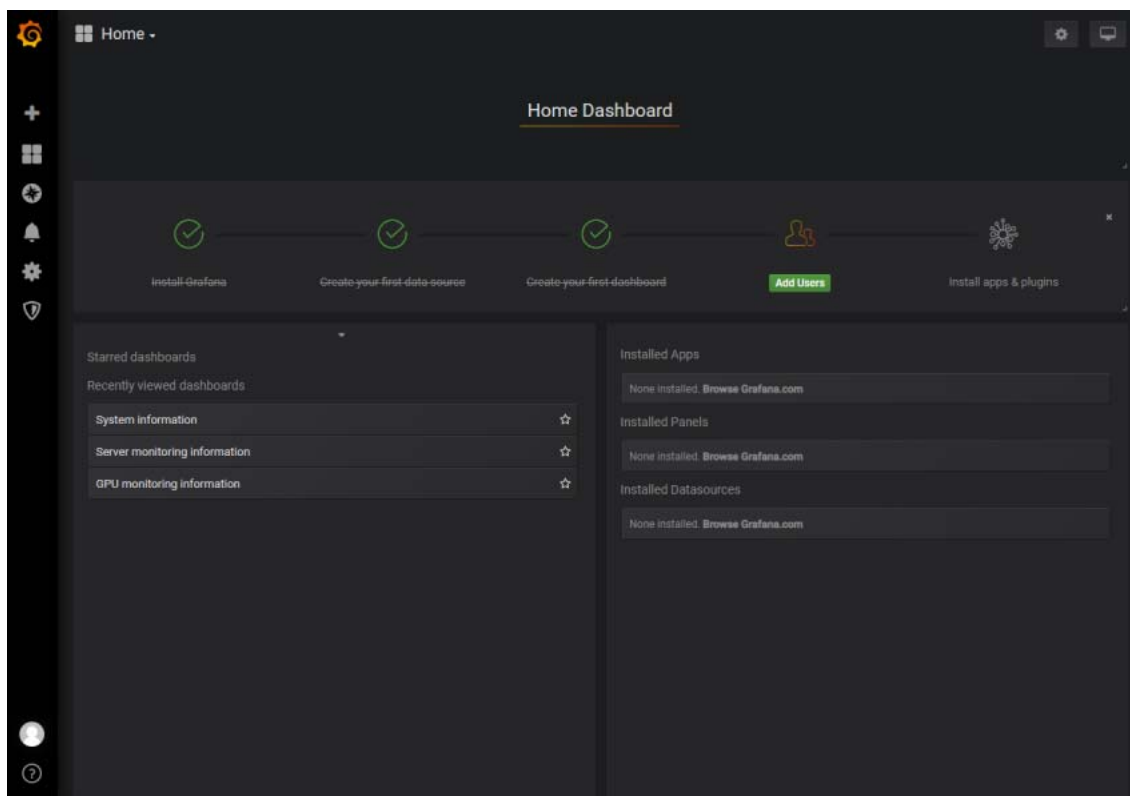
1. 監視システム管理画面へログインします。
ログイン画面で username と password を入力し、[Log in] ボタンをクリックします。



なお、工場出荷時は以下の username と password を設定しています。

username : admin

password : admin
ログインすると、ダッシュボード画面が表示されます。



ダッシュボードにはテンプレートとして以下を準備しています。
左上の [Home] ボタンから DL フォルダーを選択して、テンプレートを確認できます。

- System information
ホスト OS から収集した CPU、メモリ、内蔵ストレージの利用状況
- Server monitoring information
装置内の温度や各 FRU のステータス
- NVIDIA DCGM Exporter Dashboard
GPU の消費電力、動作温度およびステータス

4.3 監視システム設定

監視システムへのアクセスユーザーおよびパスワードを変更する場合、または監視項目を変更する場合の操作については、以下のサイトで確認してください。

- Prometheus
<https://prometheus.io/docs/introduction/overview/>
- Grafana
<https://grafana.com/docs/features/datasources/prometheus/>

4.4 アラート設定

工場出荷時は、Prometheus のアラート機能を提供していますが、サービスを無効にしています。アラート機能を利用する場合は、設置環境に合わせて以下の手順を参考にしてください。

4.4.1 アラートルール

工場出荷時は、Exporter サービスの停止をトリガとするアラートルールのサンプルを提供しています。詳細は、`/etc/alertmanager/rules.d/exporter_alert.rules` ファイルを参照してください。アラートルールの詳細は、以下を参照してください。

https://prometheus.io/docs/prometheus/latest/configuration/alerting_rules/

4.4.2 メールアラート設定

工場出荷時は、メールサーバーに送信するサンプルを提供しています。詳細は、`/etc/alertmanager/alertmanager.yml` ファイルを参照してください。御利用環境に合わせてメールアラート環境を変更する場合は、以下を参照してください。

<https://prometheus.io/docs/alerting/configuration/>

4.4.3 IPTABLES 設定追加

工場出荷時は、Prometheus からシステム外部への通信は遮断しています。メールアラートを利用する場合は、IPTABLES 設定で Filter テーブルの INPUT チェインにあるログ採取ルール(工場出荷設定の rule num は 14) の前に以下を追加してください。

```
# iptables -I INPUT <rule_num> -p tcp -s <PrometheusコンテナのIPアドレス> -  
d <HostOS IP Address> -m state --state NEW -m tcp --dport 9093 -j ACCEPT
```

IPTABLE にルールを追加したあとは、以下を実施して設定を保存してください。

```
# /etc/init.d/netfilter-persistent save  
# /etc/init.d/netfilter-persistent reload
```

4.4.4 アラート機能有効化

アラート機能を有効にする場合は、以下を実施してください。

```
# systemctl enable alertmanager  
Created symlink from /etc/systemd/system/multi-user.target.wants/alertmanager.  
service to /etc/systemd/system/alertmanager.service.  
# systemctl start alertmanager  
#
```

第 5 章 障害調査

5.1 トラブルシューティング

- サーバーの故障 LED が点灯または点滅している場合

サーバー故障 LED が点灯または点滅している場合は、ハードウェア故障が発生した可能性があります。

当社サポート窓口にご連絡してください。

故障に関する LED の詳細については、サーバーの『オペレーティングマニュアル』を参照してください。

- ホスト OS が正常に起動しない場合

- Emergency mode で起動したとき

Web インターフェースからビデオリダイレクション機能を起動し、以下のメッセージが表示されている場合、内蔵ストレージの故障により、/boot/efi ディレクトリ配下をマウントできていない可能性があります。

```
Welcome to emergency mode! After logging in type "journal -xb" to view
system logs, "systemctl reboot" to reboot, "systemctl default" or ^D to
Press Enter for maintenance
(or press Control-D to continue):
```

ビデオリダイレクション上で Enter を入力してホスト OS にログインし、/boot/efi にマウントするデバイスの UUID を確認して /etc/fstab ファイルに反映し、再起動してください。

- UUID 確認手順

```
# ls -l /dev/disk/by-uuid/
total 0
lrwxrwxrwx 1 root root 10 Jun 14 18:53 4650-2BDF -> ../../sda1
lrwxrwxrwx 1 root root 11 Jun 14 18:53 5d8f27e4-1e49-46ad-9f23-fae39ed7b1c9 ->
../../md0p2
lrwxrwxrwx 1 root root 11 Jun 14 18:53 a4f2e3b8-9297-46a2-ba33-085e374125d3 ->
../../md0p1
lrwxrwxrwx 1 root root 10 Jun 14 18:53 C21C-23B3 -> ../../sdb1
#
```

上記以外のトラブルが発生したときは、当社サポート窓口にお問い合わせください。

5.2 ディスクの寿命確認

故障または寿命の判断を確認する操作方法について説明します。

5.2.1 寿命確認手順

smartmontools でディスクの寿命を確認します。

ホスト OS の管理ユーザーでログインしたあと、sudo su により root ユーザーで実行してください。

5.2.2 書込みデータ量の確認方法

ホスト OS 上で smartctl コマンドを実行して、VALUE を確認します。

この値は SSD の寿命設計値に対する、書き込まれたデータ量を示します。御購入時はほぼ「0」の状態が表示され、書込みデータ量が蓄積することで値が上昇していきます。「100」またはそれ以上になると、書込み寿命となります。

対象SSD	確認項目	書込み寿命	算出方法
SATA SSD-960GB	Value	100	初期値はほぼ0% 書込みデータの累積に伴い、値が上昇

以下のように表示されます。

```
# smartctl -l devstat,7 /dev/sda
smartctl 7.1 2019-12-30 r5022 [x86_64-linux-5.4.0-42-generic] (local build)
Copyright (C) 2002-16, Bruce Allen, Christian Franke, www.smartmontools.org

Device Statistics (GP Log 0x04)
Page  Offset Size      Value Flags Description
0x07  =====  =      =    ==  == Solid State Device Statistics (rev 1) ==
0x07  0x008  1      0    ---  --- Percentage Used Endurance Indicator
                        |||_  C monitored condition met
                        ||__  D supports DSN
                        |___  N normalized value
```

第 6 章 SSL サーバー証明書運用

6.1 認証局によって正式に署名された SSL サーバー証明書を利用する場合

■ 注 意

システムの初期状態では、有効期限が十分に長い自己証明書が格納されています。必要に応じて、信頼された認証局の発行する SSL サーバー証明書を入れ替えて運用してください。

6.2 自己署名の SSL サーバー証明書を利用する場合

自己署名の SSL サーバー証明書を利用する環境では、必要に応じて CA 証明書（ルート証明書）を登録してください。

CA 証明書（ルート証明書）は、以下に格納されています。

`/etc/certs.d/<hostname>.crt`

第 7 章 ソフトウェア RAID 管理

工場出荷時は、サーバーのシステムボリュームをソフトウェア RAID 構成で構築しています。
ここでは、mdadm コマンドを使用したシステムボリュームの RAID 操作について説明します。

7.1 RAID 構成情報の取得

RAID の構成情報や動作状態は、以下の操作で確認します。

1. ホスト OS にログイン

ホスト OS には、Web インターフェースにログインしてビデオリダイレクション機能からログインするか、TeraTerm などのターミナルエミュレータを使用して、SSH 接続でログインします。
ホスト OS の管理ユーザーでログインしたあと、`sudo su`により root ユーザーに変更してください。

2. RAID 情報の確認

• 正常時

```
# cat /proc/mdstat
Personalities : [raid1] [linear] [multipath] [raid0] [raid6] [raid5] [raid4]
[raid10]
md0 : active raid1 sda2[0] sdb2[1]
      937033726 blocks super 1.2 [2/2] [UU]
      bitmap: 1/7 pages [1KB], 65536KB chunk

unused devices: <none>
#
```

• 異常時

```
# cat /proc/mdstat
Personalities : [raid1] [linear] [multipath] [raid0] [raid6] [raid5] [raid4]
[raid10]
md0 : active raid1 sdb2[1] ← sda2が表示されない
      937033728 blocks super 1.2 [2/1] [U]
      bitmap: 2/7 pages [0KB], 65536KB chunk

unused devices: <none>
#
```

7.2 RAID ボリュームの詳細情報の取得

RAID ボリュームの詳細情報と内蔵ストレージのデバイス情報は、以下の操作で確認します。

1. ホスト OS にログイン

ホスト OS には、Web インターフェースにログインしてビデオリダイレクション機能からログインするか、TeraTerm などのターミナルエミュレータを使用して、SSH 接続でログインします。ホスト OS の管理ユーザーでログインしたあと、`sudo su`により root ユーザーに変更してください。

2. RAID ボリュームの詳細情報の確認

```
# mdadm --detail /dev/md0
/dev/md0:
    Version : 1.2
    Creation Time : Mon Jun  8 20:05:58 2020
    Raid Level : raid1
    Array Size : 937033728 (893.63 GiB 959.52 GB)
    Used Dev Size : 937033728 (893.63 GiB 959.52 GB)
    Raid Devices : 2
    Total Devices : 2
    Persistence : Superblock is persistent

    Intent Bitmap : Internal

    Update Time : Fri Jun 12 13:17:37 2020
    State : clean
    Active Devices : 2
    Working Devices : 2
    Failed Devices : 0
    Spare Devices : 0

    Consistency Policy : bitmap

    Name : ubuntu-server:0
    UUID : 5b204118:c08814b7:78d6a33d:40e9b827
    Events : 1571

    Number Major Minor RaidDevice State
       0       8        2        0     active sync  /dev/sda2
       1       8       18        1     active sync  /dev/sdb2
#
```

3. 内蔵ストレージの詳細情報の確認

内蔵ストレージのマウント情報を確認します（文字化けする場合は `lsblk -i`）。

```
# lsblk
NAME        MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
loop0        7:0    0   89.1M 1 loop /snap/core/8268
sda          8:0    0  894.3G 0 disk
├─sda1       8:1    0   512M 0 part /boot/efi
└─sda2       8:2    0  893.8G 0 part
   └─md0      9:0    0  893.6G 0 raid1
      ├─md0p1 259:0   0    64G 0 md    [SWAP]
      └─md0p2 259:1   0  829.6G 0 md    /
sdb          8:16   0  894.3G 0 disk
├─sdb1       8:17   0   512M 0 part
└─sdb2       8:18   0  893.8G 0 part
   └─md0      9:0    0  893.6G 0 raid1
      ├─md0p1 259:0   0    64G 0 md    [SWAP]
      └─md0p2 259:1   0  829.6G 0 md    /
#
```

内蔵ストレージのデバイス情報を確認します。

```
# parted /dev/sda print
Model: ATA Micron 5300_MTFD (scsi)
Disk /dev/sda: 960GB
Sector size (logical/physical): 512B/4096B
Partition Table: gpt
Disk Flags:
Number  Start   End     Size    File system  Name  Flags
  1      1049kB  538MB   537MB   fat32         boot, esp
  2      538MB   960GB   960GB
```

7.3 内蔵ストレージの交換

ここでは、システムボリュームを構成する内蔵ストレージの交換手順について説明します。
交換の流れには以下の 2 つの方法があります。システムの状態に応じて、どちらかを選択してください。

- [「7.3.1 活性交換」\(P.36\)](#)
- [「7.3.2 システム停止交換」\(P.41\)](#)

システムボリュームを構成する内蔵ストレージを交換する場合は、お客様自身で交換対象の内蔵ストレージを RAID 構成から切り離してください。そのほかの内蔵ストレージを交換する場合は、お客様自身でシステムから切り離してください。

保守部品との交換作業は、当社サポート技術員が実施します。交換に際しては当社サポート窓口にご連絡してください。

■ 注 意

システムボリュームを構成する内蔵ストレージを交換する前に、必ず [「7.1 RAID 構成情報の取得」\(P.34\)](#) を実施し、交換対象の内蔵ストレージを特定してください。交換対象を誤るとデータを損失するおそれがあります。

7.3.1 活性交換

ここでは、システムが稼働した状態で、システムボリュームを構成する内蔵ストレージを交換する手順を説明します。

7.3.1.1 RAID 構成から内蔵ストレージを切り離す

ここでは、RAID 構成から /dev/sda を切り離す手順について説明します。

1. RAID 構成情報の確認

/proc/mdstat の情報を取得します。

```
# cat /proc/mdstat
Personalities : [raid1] [linear] [multipath] [raid0] [raid6] [raid5] [raid4] [raid10]
md0 : active raid1 sdb2[1] ← sda2が表示されない
      869923840 blocks super 1.2 [2/1] [U]
      bitmap: 2/7 pages [0KB], 65536KB chunk

unused devices: <none>
#
```

2. 内蔵ストレージの搭載位置の確認

交換対象のシステムボリュームを構成する内蔵ストレージの搭載位置について、以下を参照して確認してください。



3. 内蔵ストレージの切離し

内蔵ストレージの故障モードにより、RAID ボリュームに組み込まれた状態になっている場合があります。以下の手順で、RAID 構成から交換対象の内蔵ストレージを切り離してください。

(1) 内蔵ストレージを非アクティブにします。

```
# mdadm --fail /dev/md* /dev/sd** ← *, **は交換箇所の値を指定
```

(2) 内蔵ストレージを切り離します。

```
# mdadm --remove /dev/md* /dev/sd**
```

7.3.1.2 RAID 構成に内蔵ストレージを組み込む

内蔵ストレージを交換したあと、交換対象の内蔵ストレージを RAID 構成へ組み込み、RAID 機能の復旧を行います。

ここでは、/dev/sda の内蔵ストレージを交換したあと、RAID 構成に内蔵ストレージを組み込む手順について説明します。

1. 内蔵ストレージの交換
当社サポート技術員が内蔵ストレージを交換します。
2. 内蔵ストレージのパーティション設定
稼働している /dev/sdb の内蔵ストレージのパーティション情報をコピーします。

```
# sfdisk -d /dev/sdb | sfdisk /dev/sda --force
Checking that no-one is using this disk right now ... OK

Disk /dev/sda: 894.3 GiB, 960197124096 bytes, 1875385008 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 4096 bytes
I/O size (minimum/optimal): 4096 bytes / 4096 bytes
Disklabel type: gpt
Disk identifier: 72A22159-C3C8-4B91-8E0B-EF9C16AC9E85

Old situation:

Device          Start          End      Sectors   Size Type
/dev/sda1        2048        1050623    1048576   512M Linux filesystem
/dev/sda2    1050624    1875382271  1874331648 893.8G Linux filesystem

>>> Script header accepted.
>>> Script header accepted.
>>> Script header accepted.
>>> Script header accepted.
>>> Script header accepted.
>>> Script header accepted.
>>> Created a new GPT disklabel (GUID: 72A22159-C3C8-4B91-8E0B-EF9C16AC9E85).
Created a new partition 1 of type 'EFI System' and of size 512 MiB.
/dev/sda2: Created a new partition 2 of type 'Linux filesystem' and of size
829.8 GiB.

New situation:
Disklabel type: gpt
Disk identifier: 72A22159-C3C8-4B91-8E0B-EF9C16AC9E85

Device          Start          End      Sectors   Size Type
/dev/sdb1        2048        1050623    1048576   512M Linux filesystem
/dev/sdb2    1050624    1875382271  1874331648 893.8G Linux filesystem

The partition table has been altered.
Calling ioctl() to re-read partition table.
Syncing disks.
```

3. パーティション情報の確認

コピーしたパーティション情報を確認します。

```
# parted /dev/sdb
GNU Parted 3.3
Using /dev/sdb
Welcome to GNU Parted! Type 'help' to view a list of commands.
(parted) unit s
(parted) print
Model: ATA Micron 5200 MTFD (scsi)
Disk /dev/sdb: 1875385008s
Sector size (logical/physical): 512B/4096B
Partition Table: gpt
Disk Flags:

Number      Start          End              Size            File system  Name  Flags
  1          2048s         1050623s        1048576s        fat32
  2          1050624s      1875382271s    1874331648s

(parted) select /dev/sda
Using /dev/sda
(parted) print
Model: ATA Micron 5200 MTFD (scsi)
Disk /dev/sda: 1875385008s
Sector size (logical/physical): 512B/4096B
Partition Table: gpt
Disk Flags:

Number      Start          End              Size            File system  Name  Flags
  1          2048s         1050623s        1048576s        fat32
  2          1050624s      1875382271s    1874331648s

(parted) quit
```

4. 内蔵ストレージの組み込み

交換した内蔵ストレージを RAID ボリュームに組み込みます。

```
# mdadm --add /dev/md0 /dev/sda2
mdadm: added /dev/sda2
#
```

5. RAID 構成情報の確認

内蔵ストレージの同期状況を確認します。同期が完了する時間の目安は以下の finish で確認してください。対象 RAID へのアクセス状況により完了時間が長くなりますので、御注意ください。

```
# cat /proc/mdstat
Personalities : [raid1] [linear] [multipath] [raid0] [raid6] [raid5] [raid4] [raid10]
md0 : active raid1 sda2[2] sdb2[1]
      937033728 blocks super 1.2 [2/1] [_U]
      [====>.....] recovery = 20.1% (175056640/937033728)
      finish=56.1min speed=206114K/sec
      bitmap: 4/7 pages [16KB], 65536KB chunk

unused devices: <none>
```

6. grub インストール

交換した内蔵ストレージに grub をインストールします。

```
# umount /boot/efi
# mkfs.fat -F32 /dev/sda1
mkfs.fat 4.1 (2017-01-24)
# mount /dev/sda1 /boot/efi
# grub-install --target=x86_64-efi --efi-directory=/boot/efi
Installing for x86_64-efi platform.
Installation finished. No error reported.
# ls -l /boot/efi/EFI/ubuntu/
total 3644
-rwxr-xr-x 1 root root    108 Jun  8 20:10 BOOTX64.CSV
-rwxr-xr-x 1 root root    157 Jun  8 20:10 grub.cfg
-rwxr-xr-x 1 root root 1116536 Jun  8 20:10 grubx64.efi
-rwxr-xr-x 1 root root 1269496 Jun  8 20:10 mmx64.efi
-rwxr-xr-x 1 root root 1334816 Jun  8 20:10 shimx64.efi
#
```


7. EFI Boot マネージャーにエントリー追加

```
# efibootmgr --create --disk /dev/sda --label "ubuntu" --loader \
"\\EFI\\ubuntu\\shimx64.efi"
BootCurrent: 0000
Timeout: 1 seconds
BootOrder: 0007,0000,0001,0002,0003,0004,0005,0006
Boot0000* ubuntu
Boot0001* UEFI: NIC.LOM.1.1.IPv4HTTP - Intel(R) I350 Gigabit Network
Connection
Boot0002* UEFI: NIC.LOM.1.1.IPv4PXE - Intel(R) I350 Gigabit Network Connection
Boot0003* UEFI: NIC.LOM.1.1.IPv6PXE - Intel(R) I350 Gigabit Network Connection
Boot0004* UEFI: NIC.LOM.1.2.IPv4HTTP - Intel(R) I350 Gigabit Network
Connection
Boot0005* UEFI: NIC.LOM.1.2.IPv4PXE - Intel(R) I350 Gigabit Network Connection
Boot0006* UEFI: NIC.LOM.1.2.IPv6PXE - Intel(R) I350 Gigabit Network Connection
Boot0007* ubuntu redundancy
MirroredPercentageAbove4G: 0.00
MirrorMemoryBelow4GB: false
#
```

8. /etc/fstab エントリー更新

/boot/efi ディレクトリに交換前の内蔵ストレージをマウントして運用していた場合、交換した内蔵ストレージの UUID を /etc/fstab ファイルに反映します。

```
# ls -l /dev/disk/by-uuid/
total 0
lrwxrwxrwx 1 root root 11 Jun 27 06:33 10b101ea-8a01-4150-88f9-910870336c87 ->
../..md0p2
lrwxrwxrwx 1 root root 10 Jun 27 06:33 46E3-E6BA -> ../..sda1
lrwxrwxrwx 1 root root 11 Jun 27 06:33 a24cac56-fcd2-42df-a9df-eb9d5767040a ->
../..md0p1
lrwxrwxrwx 1 root root 10 Jun 27 06:33 BBFA-B001 -> ../..sdb1
```

編集後の /etc/fstab の内容です。ドライブ増設や、外部ストレージと接続している場合などは、該当部分を適宜読み替えてください。

```
# cat /etc/fstab
# /etc/fstab: static file system information.
#
# Use 'blkid' to print the universally unique identifier for a
# device; this may be used with UUID= as a more robust way to name devices
# that works even if disks are added and removed. See fstab(5).
#
# <file system> <mount point> <type> <options> <dump> <pass>
/dev/md0p1 none swap sw 0 0
# / was on /dev/md0p2 during curtin installation
/dev/md0p2 / ext4 defaults 0 0
# /boot/efi was on /dev/sda1 during curtin installation
/dev/disk/by-uuid/46E3-E6BA /boot/efi vfat defaults 0 0
```

9. システム更新

交換した内蔵ストレージの同期完了後、ミニルートとブートローダを更新します。

```
# cat /proc/mdstat
Personalities : [raid1] [linear] [multipath] [raid0] [raid6] [raid5] [raid4] [raid10]
md0 : active raid1 sda2[0] sdb2[1]
      869923840 blocks super 1.2 [2/2] [UU]
      bitmap: 0/7 pages [0KB], 65536KB chunk

unused devices: <none>
# update-initramfs -u
update-initramfs: Generating /boot/initrd.img-5.8.0-55-generic
I: The initramfs will attempt to resume from /dev/md0p2
I: (UUID=c05a80c0-7435-4b17-8c0f-6326678a1cab)
I: Set the RESUME variable to override this.
# update-grub
Sourcing file `/etc/default/grub'
Sourcing file `/etc/default/grub.d/kdump-tools.cfg'
Generating grub configuration file ...
Found linux image: /boot/vmlinuz-5.8.0-55-generic
Found initrd image: /boot/initrd.img-5.8.0-55-generic
Adding boot menu entry for EFI firmware configuration
done
#
```

7.3.2 システム停止交換

ここでは、システム停止してシステムボリュームを構成する内蔵ストレージを交換する手順を説明します。

7.3.2.1 RAID 構成から内蔵ストレージを切り離す

システムを停止する前に、RAID 構成から /dev/sda を切り離します。

1. RAID 構成情報の確認

/proc/mdstat の情報を取得します。

```
# cat /proc/mdstat
Personalities : [raid1] [linear] [multipath] [raid0] [raid6] [raid5] [raid4] [raid10]
md0 : active raid1 sda2[0] sdb2[1]
      869923840 blocks super 1.2 [2/2] [UU]
      bitmap: 0/7 pages [0KB], 65536KB chunk

unused devices: <none>
#
```

2. UUID を確認

/boot/efi にマウントするパーティション (/dev/sdb1) の UUID を確認します。

```
# ls -l /dev/disk/by-uuid/
total 0
lrwxrwxrwx 1 root root 11 Jun 27 06:33 10b101ea-8a01-4150-88f9-910870336c87 ->
../..md0p2
lrwxrwxrwx 1 root root 10 Jun 27 06:33 A1D7-249C -> ../..sda1
lrwxrwxrwx 1 root root 11 Jun 27 06:33 a24cac56-fcd2-42df-a9df-eb9d5767040a ->
../..md0p1
lrwxrwxrwx 1 root root 10 Jun 27 06:33 5D69-B7E8 -> ../..sdb1
#
```

3. マウント情報を変更

/etc/fstab ファイルの /boot/efi にマウントするパーティション情報を変更します。

- 変更前

```
# <file system> <mount point>    <type>  <options>          <dump>  <pass>
/dev/md0p1 none swap sw 0 0
# / was on /dev/md0p2 during curtin installation
/dev/md0p2 / ext4 defaults 0 0
# /boot/efi was on /dev/sda1 during curtin installation
/dev/disk/by-uuid/A1D7-249C/boot/efi vfat defaults 0 0
```

- 変更後

```
# <file system> <mount point>    <type>  <options>          <dump>  <pass>
/dev/md0p1 none swap sw 0 0
# / was on /dev/md0p2 during curtin installation
/dev/md0p2 / ext4 defaults 0 0
# /boot/efi was on /dev/sda1 during curtin installation
/dev/disk/by-uuid/5D69-B7E8/boot/efi vfat defaults 0 0
```

4. 内蔵ストレージの切離し

以下の手順で、RAID 構成から交換対象の内蔵ストレージを切り離します。

すでに交換対象の内蔵ストレージが RAID ボリュームから切り離されている場合は、[手順 5.](#)に進みます。

```
# mdadm --fail /dev/md0 /dev/sda2
mdadm: set /dev/sda2 faulty in /dev/md0
# mdadm --remove /dev/md0 /dev/sda2
mdadm: hot removed /dev/sda2 from /dev/md0
# cat /proc/mdstat
Personalities : [raid1] [linear] [multipath] [raid0] [raid6] [raid5] [raid4] [raid10]
md0 : active raid1 sdb2[1]
      869923840 blocks super 1.2 [2/1] [ U]
      bitmap: 2/7 pages [0KB], 65536KB chunk

unused devices: <none>
#
```

5. システム停止

sudo 権限で shutdown コマンドを実行し、システムを停止します。

7.3.2.2 RAID 構成に内蔵ストレージを組み込む

システム起動後、交換した内蔵ストレージを RAID 構成へ組み込み、RAID 機能の復旧を行います。
ここでは、内蔵ストレージを交換したあと、RAID 構成に内蔵ストレージを組み込む手順について説明します。

1. 内蔵ストレージの交換

当社サポート技術員が内蔵ストレージを交換します。完了したらお客様がシステムを起動します。

2. 内蔵ストレージのパーティション設定

稼働している内蔵ストレージのパーティション情報をコピーします。

```
# sfdisk -d /dev/sdb | sfdisk /dev/sda --force
Checking that no-one is using this disk right now ... OK

Disk /dev/sda: 894.3 GiB, 960197124096 bytes, 1875385008 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 4096 bytes
I/O size (minimum/optimal): 4096 bytes / 4096 bytes
Disklabel type: gpt

<省略>

Device      Start          End      Sectors    Size Type
/dev/sdb1    2048          1050623    1048576    512M Linux filesystem
/dev/sdb2   1050624    1875382271 1874331648 893.8G Linux filesystem

The partition table has been altered.
Calling ioctl() to re-read partition table.
Syncing disks.
#
```

3. パーティション情報の確認

コピーしたパーティション情報を確認します。

```
# parted /dev/sdb
GNU Parted 3.3
Using /dev/sdb
Welcome to GNU Parted! Type 'help' to view a list of commands.
(parted) unit s
(parted) print
Model: ATA Micron 5200 MTFD (scsi)
Disk /dev/sdb: 1875385008s
Sector size (logical/physical): 512B/4096B
Partition Table: gpt
Disk Flags:

Number  Start      End          Size          File system  Name  Flags
  1      2048s     1050623s     1048576s      fat32
  2     1050624s 1875382271s 1874331648s

(parted) select /dev/sda
Using /dev/sda
(parted) print
Model: ATA Micron 5200 MTFD (scsi)
Disk /dev/sda: 1875385008s
Sector size (logical/physical): 512B/4096B
Partition Table: gpt
Disk Flags:

Number  Start      End          Size          File system  Name  Flags
  1      2048s     1050623s     1048576s      fat32
  2     1050624s 1875382271s 1874331648s

(parted) quit
```

4. 内蔵ストレージの組み込み

交換した内蔵ストレージを RAID ボリュームに組み込みます。

```
# mdadm _ --add _ /dev/md0 _ /dev/sda2
mdadm: added /dev/sda2
#
```

5. RAID 構成情報の確認

内蔵ストレージの同期状況を確認します。同期が完了する時間の目安は以下の finish の示す時間で確認してください。

```
# cat _ /proc/mdstat
Personalities : [raid1] [linear] [multipath] [raid0] [raid6] [raid5] [raid4] [raid10]
md0 : active raid1 sda2[2] sdb2[1]
      869399552 blocks super 1.2 [2/1] [_U]
      [====>.....] recovery = 20.1% (175056640/869399552)
finish=56.1min speed=206114K/sec
      bitmap: 4/7 pages [16KB], 65536KB chunk

unused devices: <none>
#
```

6. grub インストール

交換した内蔵ストレージに grub をインストールします。

```
# umount _ /boot/efi
# mkfs.fat _ -F32 _ /dev/sda1
mkfs.fat 4.1 (2017-01-24)
# mount _ /dev/sda1 _ /boot/efi
# grub-install _ --target=x86_64-efi _ --efi-directory=/boot/efi
Installing for x86_64-efi platform.
Installation finished. No error reported.
# ls _ -l _ /boot/efi/EFI/ubuntu/
total 3644
-rwxr-xr-x 1 root root      108 Jun  8 20:10 BOOTX64.CSV
-rwxr-xr-x 1 root root      157 Jun  8 20:10 grub.cfg
-rwxr-xr-x 1 root root 1116536 Jun  8 20:10 grubx64.efi
-rwxr-xr-x 1 root root 1269496 Jun  8 20:10 mmx64.efi
-rwxr-xr-x 1 root root 1334816 Jun  8 20:10 shimx64.efi
#
```

7. EFI Boot マネージャーにエントリー追加

```
# efibootmgr _ --create _ --disk _ /dev/sda _ --label _ "ubuntu" _ --loader _
"¥¥EFI¥¥ubuntu¥¥shimx64.efi"

BootCurrent: 0000
Timeout: 1 seconds
BootOrder: 0007,0000,0001,0002,0003,0004,0005,0006
Boot0000* ubuntu
Boot0001* UEFI: NIC.LOM.1.1.IPv4HTTP - Intel(R) I350 Gigabit Network
Connection
Boot0002* UEFI: NIC.LOM.1.1.IPv4PXE - Intel(R) I350 Gigabit Network Connection
Boot0003* UEFI: NIC.LOM.1.1.IPv6PXE - Intel(R) I350 Gigabit Network Connection
Boot0004* UEFI: NIC.LOM.1.2.IPv4HTTP - Intel(R) I350 Gigabit Network
Connection
Boot0005* UEFI: NIC.LOM.1.2.IPv4PXE - Intel(R) I350 Gigabit Network Connection
Boot0006* UEFI: NIC.LOM.1.2.IPv6PXE - Intel(R) I350 Gigabit Network Connection
Boot0007* ubuntu redundancy
MirroredPercentageAbove4G: 0.00
MirrorMemoryBelow4GB: false
#
```

8. /boot/efi エントリー更新

内蔵ストレージの運用状態を交換前の状態に戻す場合、/etc/fstab ファイルの /boot/efi にマウントするパーティションの UUID を変更します。

```
# ls -l /dev/disk/by-uuid/
total 0
lrwxrwxrwx 1 root root 11 Jun 27 06:33 <交換後の/md0p2のuuid> -> ../../md0p2
lrwxrwxrwx 1 root root 10 Jun 27 06:33 <交換後の/sda1のuuid> -> ../../sda1
lrwxrwxrwx 1 root root 11 Jun 27 06:33 <交換後の/md0p1のuuid> -> ../../md0p1
lrwxrwxrwx 1 root root 10 Jun 27 06:33 5D69-B7E8 -> ../../sdb1
#

# cat /etc/fstab
# /etc/fstab: static file system information.
#
# Use 'blkid' to print the universally unique identifier for a
# device; this may be used with UUID= as a more robust way to name devices
# that works even if disks are added and removed. See fstab(5).
#
# <file system> <mount point> <type> <options> <dump> <pass>
/dev/disk/by-uuid/<交換後の/md0p1のuuid> none swap sw 0 0
# / was on /dev/md0 during curtin installation
/dev/disk/by-uuid/<交換後の/md0p2のuuid> / ext4 defaults 0 0
# /boot/efi was on /dev/sda1 during curtin installation
/dev/disk/by-uuid/<交換後の/sda1のuuid> /boot/efi vfat defaults 0 0
```

9. システム更新

交換した内蔵ストレージの同期完了後、ミニルートとブートローダを更新します。

```
# cat /proc/mdstat
Personalities : [raid1] [linear] [multipath] [raid0] [raid6] [raid5] [raid4] [raid10]
md0 : active raid1 sda2[0] sdb2[1]
      937033728 blocks super 1.2 [2/2] [UU]
      bitmap: 0/7 pages [0KB], 65536KB chunk

unused devices: <none>
# update-initramfs -u
update-initramfs: Generating /boot/initrd.img-4.15.0-101-generic
I: The initramfs will attempt to resume from /dev/md0p2
I: (UUID=c05a80c0-7435-4b17-8c0f-6326678a1cab)
I: Set the RESUME variable to override this.
# update-grub
Sourcing file `/etc/default/grub'
Sourcing file `/etc/default/grub.d/kdump-tools.cfg'
Generating grub configuration file ...
Found linux image: /boot/vmlinuz-4.15.0-101-generic
Found initrd image: /boot/initrd.img-4.15.0-101-generic
Adding boot menu entry for EFI firmware configuration
done
#
```

第 8 章 バックアップリストア

ここでは、システムボリュームの環境が壊れた場合などに、システムをバックアップ時点の環境に復元する手順を説明します。

事前に以下の留意事項をご確認ください。

- 起動用ライブメディアの準備
システムボリューム全体をバックアップリストアするために、起動用ライブメディアを使用します。製品添付の DVD から「起動用ライブメディア」を作業用 PC にコピーしておいてください。起動用ライブメディアのファイル名などは、DVD の README.txt を参照してください。
- ユーザーデータのバックアップ
学習データ、ネットワークモデルなど日常的に発生するユーザーデータのバックアップについては、お客様の運用に適した方法を御検討いただき、必ず実施してください。
- システムのサポートについて
ここに記載した手順以外で取得したバックアップデータを使用してリストアした場合、富士通ではシステムのサポートをすることができません。
- データの復旧について
本書に記載した手順はシステムをバックアップ時点の環境に復元するものですが、システム動作やすべてのデータの復旧を保証するものではありません。
復旧できない場合は、システム構築手順書に従って工場出荷時点の環境に戻していただくことになります。

なお、システムボリュームを構成する片方の内蔵ストレージが故障した場合は、[「第 7 章 ソフトウェア RAID 管理」\(P.34\)](#) の手順で復旧してください。

ここで説明しているコマンド入力や出力結果などは一例です。御使用のシステム構成に合わせてデバイス名を変更し、実際の出力結果を確認してください。

8.1 システムバックアップ

ここでは、dump コマンドを使用して、サーバーに搭載したバックアップ用ストレージ (/dev/sdc) に、オフラインバックアップする手順を例に説明します。バックアップ処理中は、ソフトウェアバージョンアップや、アプリケーションを実行しないでください。

御購入いただいた Zinrai ディープラーニングシステムの構成によっては、バックアップファイルの保存領域はありません。システムディスク以外のストレージ領域に、バックアップファイルを保存してください。

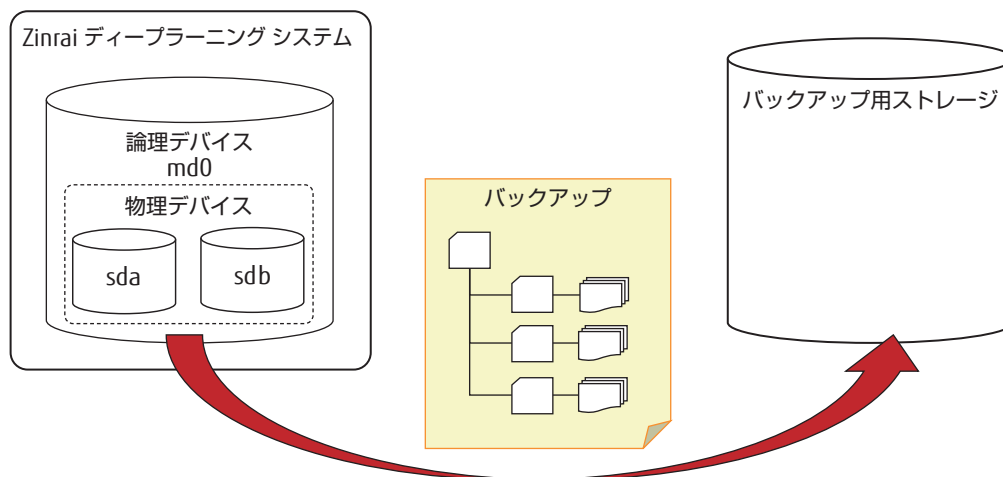
「8.1.1 ファイルシステム情報採取」(P.48)



「8.1.2 起動用ライブメディアから OS を起動」(P.50)

● システムバックアップのイメージ

Zinrai ディープラーニングシステムの /root ディレクトリに、dump コマンドを使用してバックアップを採取します。



8.1.1 ファイルシステム情報採取

リストアする場合、事前に以下の情報を保存します。`/root` ディレクトリに保存する手順を例に説明します。

- ファイルシステム情報
- パーティション情報
- UUID

なお、ファイルシステムの情報採取は、以下により `root` ユーザーに変更して作業を実施してください。

```
$ sudo su
[sudo] password for zdlsadmin:
#
```

1. ファイルシステム情報

`/etc/fstab` 情報を取得します。

```
# cp /etc/fstab /root/fstab.bak
# cat /root/fstab.bak
```

`lsblk` コマンドで、`/boot/efi` に物理パーティション `"sda1"` が、ルート (`/`) に論理パーティション `"md0p2"` が、`[SWAP]` に論理パーティション `"md0p1"` が割り当てられていることを確認してください。

```
# lsblk
NAME            MAJ:MIN RM  SIZE RO TYPE  MOUNTPOINT
loop0           7:0    0   89.1M 1 loop  /snap/core/8268
sda             8:0    0 894.3G 0 disk
├─sda1          8:1    0   512M 0 part  /boot/efi
├─sda2          8:2    0 893.8G 0 part
│ └─md0         9:0    0 893.6G 0 raid1
│   ├─md0p1    259:0   0    64G 0 md    [SWAP]
│   └─md0p2    259:1   0 829.6G 0 md    /
sdb            8:16    0 894.3G 0 disk
├─sdb1          8:17    0   512M 0 part
├─sdb2          8:18    0 893.8G 0 part
│ └─md0         9:0    0 893.6G 0 raid1
│   ├─md0p1    259:0   0    64G 0 md    [SWAP]
│   └─md0p2    259:1   0 829.6G 0 md    /
#
```

2. パーティション情報

パーティション情報を取得します。

```
# parted /dev/sda unit s print > /root/parted_sda.out
# parted /dev/sdb unit s print > /root/parted_sdb.out
# parted /dev/md0 unit s print > /root/parted_md0.out
```

以下は parted_sda.out ファイルと parted_md0.out ファイルの例です。パーティション情報と各パーティションのセクター情報を確認します。

```
# cat /root/parted_sda.out
Model: ATA Micron 5200 MTFD (scsi)
Disk /dev/sda: 1875385008s
Sector size (logical/physical): 512B/4096B
Partition Table: gpt
Disk Flags:

Number   Start      End          Size          File system  Name  Flags
  1       2048s     1050623s     1048576s      fat32
  2       1050624s  1875382271s  1874331648s

# cat /root/parted_md0.out
Model: Linux Software RAID Array (md)
Disk /dev/md0: 1874067456s
Sector size (logical/physical): 512B/4096B
Partition Table: loop
Disk Flags:

Number   Start      End          Size          File system  Flags
  1       2048s     134219775s   134217728s    linux-swap(v1)
  2       134219776s 1874061311s  1739841536s    ext4
#
```

3. UUID

ファイルシステムの UUID を取得します。

```
# blkid /root/blkid.out
```

以下は blkid.out ファイルの例です。システムを構成するパーティションの UUID を確認します。

```
# blkid
/dev/sda1: UUID="46E3-E6BA" TYPE="vfat" PARTUUID="0a510967-a4ea-4e8d-bce1-701362a5be89"
/dev/sda2: UUID="a82ca9f3-8e3b-40fe-2117-eca22e9019b7" UUID_SUB="5652a722-fb4c-a259-5410-6e13122349ca" LABEL="ubuntu-server:0" TYPE="linux_raid_member" PARTUUID="e1042acf-ae3-40f9-9201-f0f100a79659"
/dev/sdb1: UUID="BBFA-B001" TYPE="vfat" PARTUUID="0a510967-a4ea-4e8d-bce1-701362a5be89"
/dev/sdb2: UUID="a82ca9f3-8e3b-40fe-2117-eca22e9019b7" UUID_SUB="d9a0d78d-43f8-b9df-b100-eda4cf05d86c" LABEL="ubuntu-server:0" TYPE="linux_raid_member" PARTUUID="e1042acf-ae3-40f9-9201-f0f100a79659"
/dev/sdd: UUID="8ad4918b-1728-45af-8d7a-e9108e445548" TYPE="ext4"
/dev/md0p1: UUID="a24cac56-fcd2-42df-a9df-eb9d5767040a" TYPE="swap" PARTUUID="fec8a927-02a8-4002-a166-2ff06a57d061"
/dev/md0p2: UUID="10b101ea-8a01-4150-88f9-910870336c87" TYPE="ext4" PARTUUID="55e379c1-9b35-41d2-8ab0-89ad22006e0a"
/dev/loop0: TYPE="squashfs"

/dev/md0: PTUUID="423a3b48-0e86-4efd-a3c5-659fbd756c1b" PTTYPE="gpt"
```

8.1.2 起動用ライブメディアから OS を起動

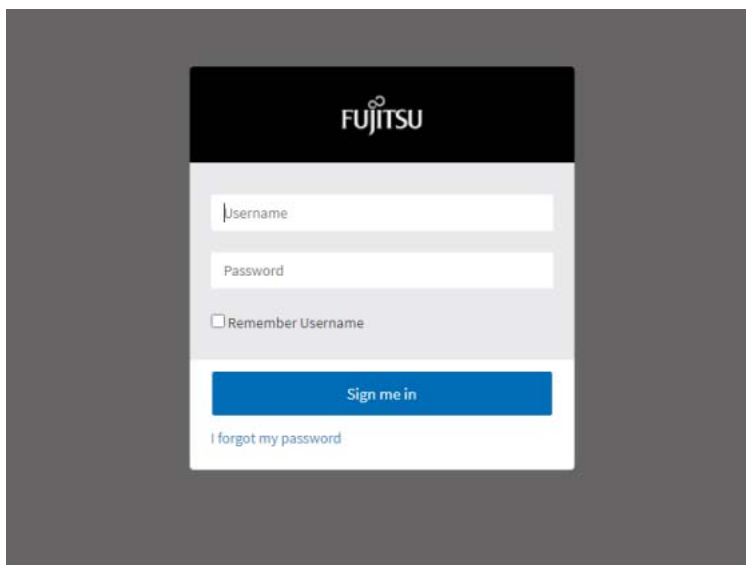
起動用ライブメディアから作業用のシステムを起動した環境で、Zinrai ディープラーニング システムのオフラインバックアップを実施します。バックアップの保存先には、バックアップ対象の内蔵ストレージの使用容量と同程度の空き容量を確保してください。

起動用ライブメディアから OS を起動します。事前に以下の準備をしてください。

- バックアップファイルを保存するファイルシステム作成済みの内蔵ストレージ
(外部ストレージを NFS でマウントして使うこともできます)
- サーバーの Web インターフェースにログインできる作業用 PC

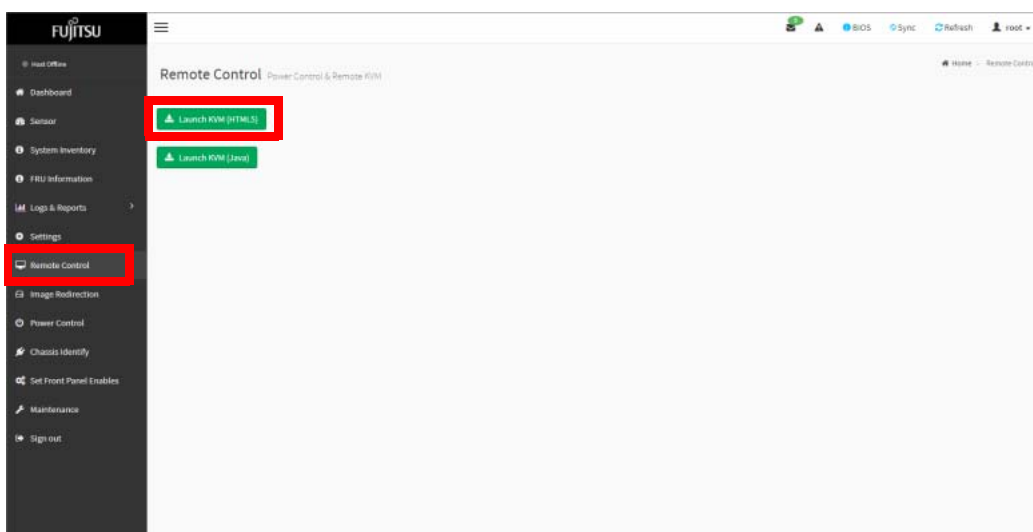
1. Web インターフェースにログイン

Web インターフェースにログインすると、以下の画面が表示されます。



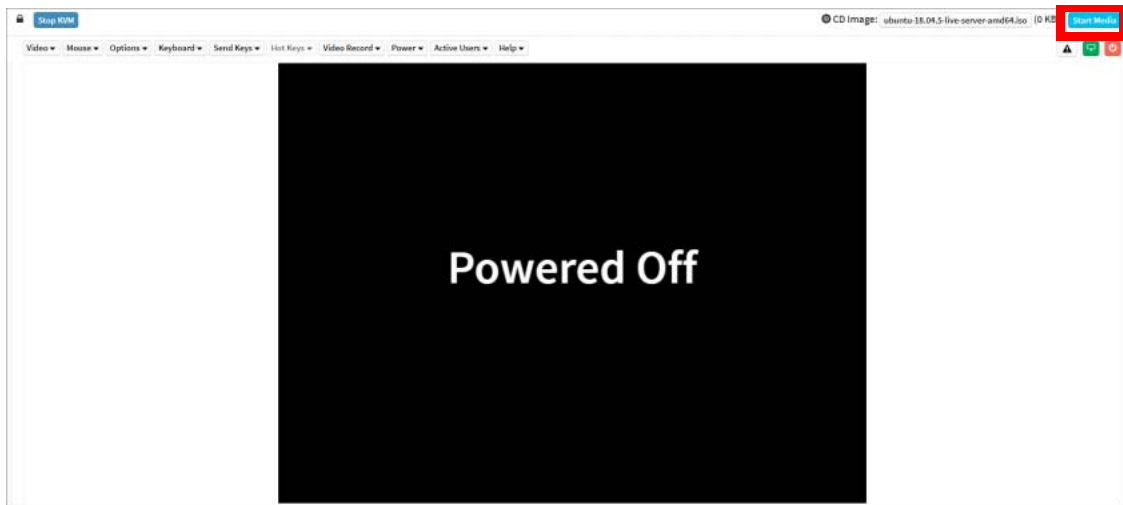
2. ビデオリダイレクション機能を開く

[Remote Control] - [Launch KVM (HTML5)] でビデオリダイレクション機能を起動します。



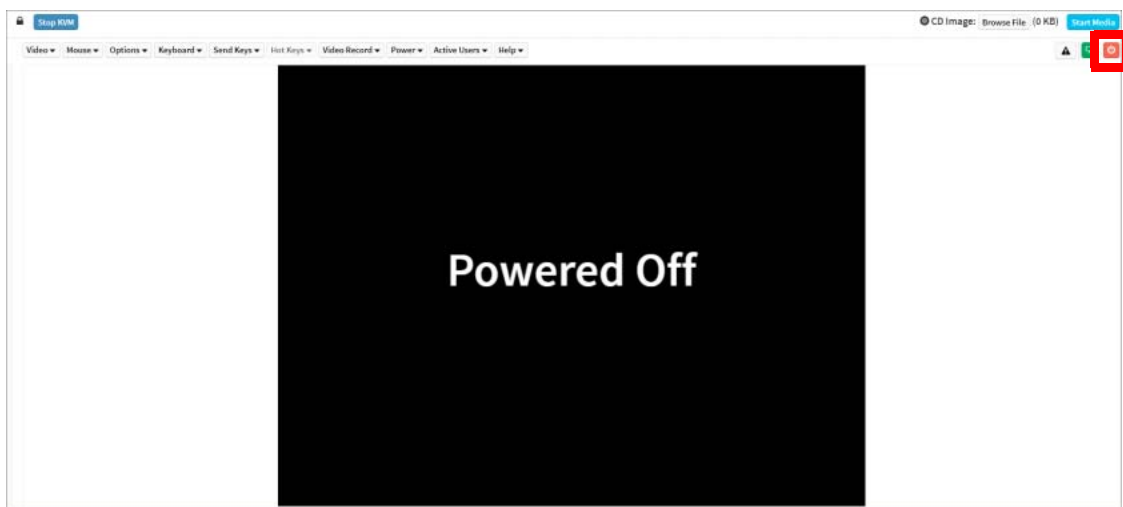
3. CD Image をマウントする

右上の [Browse] ボタンをクリックしてライブメディアを選択し、[Start Media] ボタンをクリックします。



4. OS 起動

メニューバー右端の橙色の電源ボタンアイコンをクリックして電源を投入します。POST 画面に Fujitsu ロゴが表示されたら [F11] キーを押して boot 選択画面を開き、UEFI: AMI Virtual CD-ROM0 からの起動を選択してください。



起動が完了すると、OS のログイン画面が表示されます。

8.1.3 システムバックアップ実施

システムのバックアップは、ビデオリダイレクション機能の表示画面で行います。ライブメディアから OS が起動すると、ビデオリダイレクション画面上にログインプロンプトが表示されます。

```
Ubuntu 20.04.2 LTS ubuntu tty1
ubuntu login:
```

1. ログイン

ライブメディアの環境には、以下でログインします。

アカウント：ubuntu
パスワード：<未入力>

ログイン後は、以下で root ユーザーに変更して、バックアップを実施します。

```
ubuntu@ubuntu:~$ sudo su
root@ubuntu:/home/ubuntu#
```

ライブメディアで起動した環境では、キーボードレイアウトは「Generic 105-key (intl) PC」が設定されています。必要に応じて以下のコマンドを入力し、キーボードレイアウトを変更してください。

```
# dpkg-reconfigure keyboard-configuration
```

2. システムバックアップ

/dev/sdc1 を /media ディレクトリにマウントし、dump コマンドでシステムのバックアップを採取する例を示します。SATA SSD にシステムを構築した環境で、ファイルサイズが約 50GB の場合、バックアップ採取時間の目安は約 3 分です。

```
# parted /dev/sdc mklabel gpt
# parted /dev/sdc mkpart sdc1 0% 100%
# mkfs.ext4 /dev/sdc1
mke2fs 1.45.5 (07-Jan-2020)
Discarding device blocks: done
Creating filesystem with 468843264 4k blocks and 117211136 inodes
Filesystem UUID: 18178c4e-b530-4cb1-90a3-edc7436a68d13
Superblock backups stored on blocks:
<snip>
Allocating group tables: done
Writing inodes tables: done
Creating journal (262144 blocks): done
Writing superblocks and filesystem accounting information: done
#

# mount /dev/sdc1 /media
# df -lh /media
Filesystem Size Used Avail Use% Mounted on
/dev/sdc1 1.8T 77M 1.7T 1% /media
# dump -0f /media/md0p2.dmp /dev/md0p2
DUMP: Date of this level 0 dump: Tue Dec 22 04:50:19 2020
DUMP: Dumping /dev/md0p2 (an unlisted file system) to /media/md0p2.dmp
DUMP: Label: none
DUMP: Writing 10 Kilobyte records
DUMP: mapping (Pass I) [regular files]
DUMP: mapping (Pass II) [directories]
DUMP: estimated 53494363 blocks.
DUMP: Volume 1 started with block 1 at: Tue Dec 22 04:50:20 2020
DUMP: dumping (Pass III) [directories]
DUMP: dumping (Pass IV) [regular files]
DUMP: Closing /media/md0.dmp
DUMP: Volume 1 completed at: Fri Jun 12 15:56:46 2020
DUMP: Volume 1 53433980 blocks (52181.62MB)
DUMP: Volume 1 took 0:02:46
DUMP: Volume 1 transfer rate: 243990 kB/s
DUMP: 53433980 blocks (52181.62MB) on 1 volume(s)
DUMP: finished in 219 seconds, throughput 243990 kBytes/sec
DUMP: Date of this level 0 dump: Tue Dec 22 04:50:19 2020
DUMP: Date this dump completed: Tue Dec 22 04:53:59 2020
DUMP: Average transfer rate: 301042 kB/s
DUMP: DUMP IS DONE
# ls -lh /media/md0p2.dmp
-rw-r--r-- 1 root root 51G Dec 22 04:53 /media/md0p2.dmp
#
```

3. システム復帰

システムを再起動して、運用可能な状態に戻します。

```
# reboot
```

8.2 システムリストア

「8.1 システムバックアップ」(P.47) で採取したバックアップファイルから、システムをリストアする手順を説明します。ここでは、バックアップファイルを HDD に保存していることを前提に説明します。

「8.2.1 ライブメディアから OS を起動」(P.54)



「8.2.2 ファイルシステム作成」(P.55)

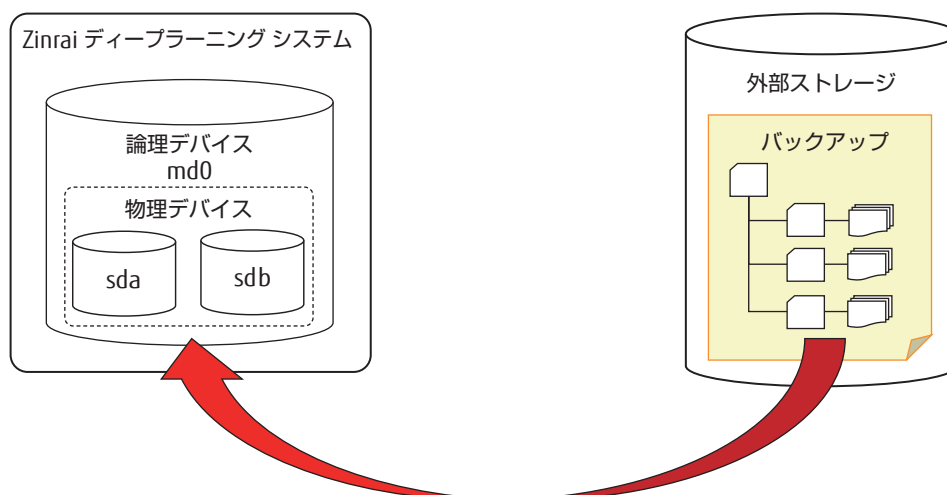


「8.2.3 システムリストア実施」(P.59)

● システムリストアのイメージ

Zinrai ディープラーニング システムの外部に保存しているバックアップファイルからシステムリストアを実施します。

ここでは、物理デバイスの /dev/sda を初期化して、バックアップファイルから /dev/sda にシステムをリストアしたあとに、/dev/sdb を初期化してミラーリングで同期をとる手順を説明します。実際の状況に合わせて作業の順序を入れ替える場合は、適宜読み替えてください。



8.2.1 ライブメディアから OS を起動

バックアップファイルを保存している内蔵ストレージを搭載して、「[8.1.2 起動用ライブメディアから OS を起動](#)」(P.50) の手順で、ライブメディアから OS を起動します。

システムのリストアは、ビデオリダイレクション機能の表示画面で行います。

ライブメディアから OS が起動すると、ビデオリダイレクション画面上にログインプロンプトが表示されます。

```
Ubuntu 20.04.2 LTS <hostname> tty1
<hostname> login:
```

1. ログイン

ライブメディアの環境には、以下でログインします。

アカウント：ubuntu
パスワード：<未入力>

ログイン後は、以下で root ユーザーに変更して、以降のシステムリストア作業はすべて root 権限で実行します。

```
ubuntu@ubuntu:~$ sudo su
root@ubuntu:/home/ubuntu#
```

2. キーボード変更

ライブメディアで起動した環境では、キーボードレイアウトは「Generic 105-key (intl) PC」が設定されています。以下のコマンドを実行して、御利用の環境に合ったキーボードレイアウトに変更してください。

```
# dpkg-reconfigure keyboard-configuration
```

3. 内蔵ストレージをマウント

以下の例では、バックアップファイルを格納した内蔵ストレージ (/dev/sdc1、ファイルシステム :ext4) を /media ディレクトリにマウントします。

```
# mount /dev/sdc1 /media

# lsblk
NAME            MAJ:MIN RM  SIZE RO TYPE  MOUNTPOINT
loop0            7:0    0   1.5G  1 loop  /rofs
loop1            7:1    0  29.9M  1 loop  /snap/snapd/8542
loop2            7:2    0    55M  1 loop  /snap/core18/1880
loop3            7:3    0 255.6M  1 loop  /snap/gnome-3-34-1804/36
loop4            7:4    0   62.1M  1 loop  /snap/gtk-common-themes/1506
loop5            7:5    0   49.8M  1 loop  /snap/snap-store/467
sda              8:0    0 894.3G  0 disk
├─sda1            8:1    0   512M  0 part
├─sda2            8:2    0 893.8G  0 part
│   └─md0          9:0    0 893.6G  0 raid1
│       ├─md0p1    259:0    0    64G  0 part
│       └─md0p2    259:1    0 829.6G  0 part
└─sdb             8:16   0 894.3G  0 disk
    ├─sdb1         8:17    0   512M  0 part
    ├─sdb2         8:18    0 893.8G  0 part
    │   └─md0       9:0    0 893.6G  0 raid1
    │       ├─md0p1 259:0    0    64G  0 part
    │       └─md0p2 259:1    0 829.6G  0 part
    └─sdc          8:32    0   1.8T  0 disk
        └─sdc1     8:33    0   1.8T  0 part  /media
sr0             11:0    1   2.1G  0 rom   /cdrom
sr1             11:1    1 1024G  0 rom
```

8.2.2 ファイルシステム作成

内蔵ストレージにパーティション情報を設定します。

内蔵ストレージを交換せず、バックアップファイルからシステムを復旧する場合、以下のコマンドを実行して RAID 情報、ラベル情報、およびパーティション情報をクリアしてください。

```
# parted /dev/md0 (← /dev/md127と認識している場合は、以降のmd0をmd127に読み替えてください)
GNU Parted 3.3
Using /dev/md0
Welcome to GNU Parted! Type 'help' to view a list of commands.
(parted) print
Model: Linux Software RAID Array (md)
Disk /dev/md0: 960GB
Sector size (logical/physical): 512B/4096B
Partition Table: gpt
Disk Flags:

Number  Start   End     Size    File system  Name  Flags
  1      1049kB  68.7GB  68.7GB  linux-swaps(v1)  swap
  2      68.7GB  960GB   891GB   ext4

(parted) rm_2
(parted) rm_1
(parted) print
Model: Linux Software RAID Array (md)
Disk /dev/md0: 960GB
Sector size (logical/physical): 512B/4096B
Partition Table: gpt
Disk Flags:

Number  Start   End     Size    File system  Name  Flags
(parted) quit
Information: You may need to update /etc/fstab.
# wipefs -a /dev/md0
/dev/md0: 8 bytes were erased at offset 0x00000200 (gpt): 45 46 49 20 50 41 52 54
/dev/md0: 8 bytes were erased at offset 0xdf67fffe00 (gpt): 45 46 49 20 50 41 52 54
/dev/md0: 2 bytes were erased at offset 0x000001fe (PMBR): 55 aa
/dev/md0: calling ioctl to re-read partition table: Success
#
# wipefs /dev/md0
(何も表示されなければラベル情報はクリアされています)
# mdadm --stop /dev/md0
# mdadm --misc --zero-superblock /dev/sda2 (← /dev/sda2のRAID情報をクリア)
# mdadm --misc --zero-superblock /dev/sdb2 (← /dev/sdb2のRAID情報をクリア)
# parted /dev/sda
GNU Parted 3.3
Using /dev/sda
Welcome to GNU Parted! Type 'help' to view a list of commands.
(parted) print
Model: ATA Micron_5300_MTFS (scsi)
Disk /dev/sda: 960GB
Sector size (logical/physical): 512B/4096B
Partition Table: gpt
Disk Flags:

Number  Start   End     Size    File system  Name  Flags
  1      1049kB  538MB   537MB   fat32         boot, esp
  2      538MB   960GB   960GB

(parted) rm_2
(parted) rm_1
(parted) quit
Information: You may need to update /etc/fstab.
# wipefs -a /dev/sda
# wipefs /dev/sda
```


SATA SSD-960GB でシステムボリュームを構成している場合の手順を例に説明します。必ず、[「8.1.1 ファイルシステム情報採取」\(P.48\)](#) でバックアップ時に採取したパーティション情報に合わせて、設定してください。

1. 物理デバイス設定

parted_sda.out ファイルを基に、/dev/sda にパーティション情報を設定します。
/dev/sdb はシステムをリストアしたあとにパーティション情報を設定するため、ここでは設定しません。

```
# parted /dev/sda
GNU Parted 3.3
Using /dev/sda
Welcome to GNU Parted! Type 'help' to view a list of commands.
(parted) unit s
(parted) mklabel gpt
(parted) mkpart
Partition name? []? ↵
File system type? [ext2]? ↵
Start? 2048
End? 1050623
(parted) set 1 esp on
(parted) mkpart
Partition name? []? ↵
File system type? [ext2]? ↵
Start? 1050624
End? 1875382271
(parted) set 2 raid on
(parted) print
Model: ATA Micron 5200 MTFD (scsi)
Disk /dev/sda: 1875385008s
Sector size (logical/physical): 512B/4096B
Partition Table: gpt
Disk Flags:

Number  Start      End          Size          File system  Name  Flags
  1      2048s      1050623s     1048576s      ext2         boot, esp
  2      1050624s   1875382271s  1874331648s   ext2         raid

(parted) quit
Information: You may need to update /etc/fstab.
#
```

パーティション設定時、"Warning: The resulting partition is not properly aligned for best performance." と表示された場合、「Ignore」を入力してください。

2. 物理デバイスのファイルシステム作成

物理デバイスに設定したパーティションに、ファイルシステムを作成します。

```
# mkfs.fat -F32 /dev/sda1
mkfs.fat 4.1 (2017-01-24)
# mkfs.ext4 -F /dev/sda2
mke2fs 1.45.5 (07-Jan-2020)
Discarding device blocks: done
Creating filesystem with 234291200 4k blocks and 58687200 inodes
Filesystem UUID: 427c14ed-8690-472c-ac50-67c6ce572697
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632, 2654208,
    4096000, 7962624, 11239424, 20480000, 23887872, 71663616, 78675968,
    102400000, 214990848
Allocating group tables: done
Writing inode tables: done
Creating journal (262144 blocks): done
Writing superblocks and filesystem accounting information: done
# parted /dev/sda print
Model: ATA Micron 5200_MTFD (scsi)
Disk /dev/sda: 960GB
Sector size (logical/physical): 512B/4096B
Partition Table: gpt
Disk Flags:


| Number | Start  | End   | Size  | File system | Name | Flags     |
|--------|--------|-------|-------|-------------|------|-----------|
| 1      | 1049kB | 538MB | 537MB | fat32       |      | boot, esp |
| 2      | 538MB  | 960GB | 960GB | ext4        |      | raid      |


#
```

3. RAID デバイス作成

raid フラグを設定したパーティションで RAID デバイスを作成します。

```
# mdadm --create /dev/md0 --metadata=1.2 --level=1 --raid-
devices=2 --missing /dev/sda2
mdadm: /dev/sda2 appears to contain an ext2fs file system
       size=937164800K mtime=Thu Jan 1 00:00:00 1970
Continue creating array? y
mdadm: array /dev/md0 started.
#
```

4. RAID デバイスのパーティション設定

以下で設定するパーティション情報は、バックアップファイルの parted_md0.out ファイルを基にしてください。

```
# parted /dev/md0
GNU Parted 3.3
Using /dev/md0
Welcome to GNU Parted! Type 'help' to view a list of commands.
(parted) unit s
(parted) mklabel gpt
(parted) mkpart
Partition name? []? ↵
File system type? [ext2]? linux-swap
Start? 2048
End? 134219775
(parted) mkpart
Partition name? []? ↵
File system type? [ext2]? ext4
Start? 134219776
End? 1874061311
(parted) print
Model: Linux Software RAID Array (md)
Disk /dev/md0: 1874067456s
Sector size (logical/physical): 512B/4096B
Partition Table: gpt
Disk Flags:


| Number | Start      | End         | Size        | File system | Name | Flags |
|--------|------------|-------------|-------------|-------------|------|-------|
| 1      | 2048s      | 134219775s  | 134217728s  | linux-swap  | (v1) |       |
| 2      | 134219776s | 1874061311s | 1739841536s | ext4        |      |       |


(parted) quit
Information: You may need to update /etc/fstab.

#
```

パーティション設定時、"Warning: The resulting partition is not properly aligned for best performance."と表示された場合、「Ignore」を入力してください。

5. RAID デバイスのファイルシステム作成

RAID デバイスのパーティションに、ファイルシステムを作成します。

```
# mkfs.ext4 /dev/md0p2
mke2fs 1.45.5 (07-Jan-2020)
Discarding device blocks: done
Creating filesystem with 217480192 4k blocks and 54370304 inodes
Filesystem UUID: 80f99cd3-6d70-4952-b4ef-f0a1b51f5fb6
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632, 2654208,
    4096000, 7962624, 11239424, 20480000, 23887872, 71663616, 78675968,
    102400000, 214990848

Allocating group tables: done
Writing inode tables: done
Creating journal (262144 blocks): done
Writing superblocks and filesystem accounting information: done

# mkswap /dev/md0p1
Setting up swapspace version 1, size = 64 GiB (68719472640 bytes)
no label, UUID=262d5940-8587-4541-874b-1dd22de77547
# ls -l /dev/disk/by-uuid | grep -i sda1
lrwxrwxrwx 1 root root 10 Dec 22 21:46 AF7E-AC09 -> ../../sda1
#
```

"Proceed anyway?"と表示された場合は、「y」を入力してください。

注 意

ここで表示される /dev/md0p1、/dev/md0p2、および /dev/sda1 の UUID は、リストアで必要になるため、必ずメモしてください。

8.2.3 システムリストア実施

システムをリストアする手順を説明します。バックアップするサイズや内蔵ストレージの種類により、数時間かかる場合があります。

1. ルート (/) 領域マウント

以下の例では、RAID デバイスのルートパーティション (md0p2) を /mnt ディレクトリにマウントします。

```
# mount /dev/md0p2 /mnt
```

2. システムリストア

```
# cd /mnt
# restore -rf /media/md0p2.dmp
restore: ./lost+found: File exists
# rm restoresymtable
#
```

約 50GB のシステムを SATA SSD の内蔵ストレージにリストアする時間の目安は、約 3 分です。

3. RAID 情報更新

/mnt/etc/mdadm/mdadm.conf ファイルの「ARRAY /dev/md0 ~」の定義をコメントアウトします。

変更前	変更後
ARRAY /dev/md0 metadata=1.2 UUID=00000000:8e3b40fe:2117eca2:2e9019b7 MAILADDR root	# ARRAY /dev/md0 metadata=1.2 UUID=00000000:8e3b40fe:2117eca2:2e9019b7 MAILADDR root

mdadm.conf 情報を更新します。

```
# mdadm --detail --scan >> /mnt/etc/mdadm/mdadm.conf
```

/mnt/etc/mdadm/mdadm.conf ファイル最下行に追加した RAID 情報を変更します。

最下行にある「name=ubuntu-server:0」を削除します。

変更前	変更後
ARRAY /dev/md0 metadata=1.2 name=ubuntu-server:0 UUID=a82ca9f3:8e3b40fe:2117eca2:2e9019b7	ARRAY /dev/md0 metadata=1.2 UUID=a82ca9f3:8e3b40fe:2117eca2:2e9019b7

4. fstab 情報更新

ファイルシステム作成時にメモした UUID を /mnt/etc/fstab ファイルに反映します。

変更前	変更後
<pre># /etc/fstab: static file system information. # # Use 'blkid' to print the universally unique identifier for a # device; this may be used with UUID= as a more robust way to name devices # that works even if disks are added and removed. See fstab(5). # # <file system> <mount point> <type> <options> <dump> <pass> /dev/disk/by-id/<変更前のmd0p1>md-uuid> none swap sw 0 0 # / was on /dev/md0 during curtin installation /dev/disk/by-id/<変更前のmd0p2>md-uuid> /ext4 defaults 0 0 # /boot/efi was on /dev/sda1 during curtin installation /dev/disk/by-uuid/<変更前の/boot/efiの uuid> /boot/efi vfat defaults 0 0 /swap.img none swap sw 0 0</pre>	<pre># /etc/fstab: static file system information. # # Use 'blkid' to print the universally unique identifier for a # device; this may be used with UUID= as a more robust way to name devices # that works even if disks are added and removed. See fstab(5). # # <file system> <mount point> <type> <options> <dump> <pass> /dev/disk/by-uuid/262d5940-8587-4541- 874b-1dd22de77547 none swap sw 0 0 # / was on /dev/md0 during curtin installation /dev/disk/by-uuid/80f99cd3-6d70-4952- b4ef-f0a1b51f5fb6 /ext4 defaults 0 0 # /boot/efi was on /dev/sda1 during curtin installation /dev/disk/by-uuid/AF7E-AC09 /boot/efi vfat defaults 0 0</pre>

5. ブートローダインストール準備

リストアしたシステムに chroot でログインします。

```
# mount --bind /dev /mnt/dev
# mount --bind /dev/pts /mnt/dev/pts
# mount --bind /proc /mnt/proc
# mount --bind /sys /mnt/sys
# mount --bind /run /mnt/run
# chroot /mnt
#
```

6. ブートローダインストール

chroot でログインした環境で、/dev/sda1 にブートローダをインストールします。/dev/sdb1 には、RAID 組込み時にインストールします。

```
# mount /dev/sda1 /boot/efi
# grub-install --target=x86_64-efi --efi-directory=/boot/efi
Installing for x86_64-efi platform.
grub-install: warning: Couldn't find physical volume `(null)'. Some modules
may be missing from core image..
grub-install: warning: Couldn't find physical volume `(null)'. Some modules
may be missing from core image..
efibootmgr: EFI variables are not supported on this system.
efibootmgr: EFI variables are not supported on this system.
Installation finished. No error reported.
# ls -l /boot/efi/EFI/ubuntu/
total 4196
-rwxr-xr-x 1 root root 108 Dec 22 15:49 BOOTX64.CSV
-rwxr-xr-x 1 root root 162 Dec 22 15:49 grub.cfg
-rwxr-xr-x 1 root root 1681280 Dec 22 15:49 grubx64.efi
-rwxr-xr-x 1 root root 1269496 Dec 22 15:49 mmx64.efi
-rwxr-xr-x 1 root root 1334816 Dec 22 15:49 shimx64.efi
#
```

7. GRUB 更新

```
# update-grub2
Sourcing file `/etc/default/grub'
Sourcing file `/etc/default/grub.d/init-select.cfg'
Sourcing file `/etc/default/grub.d/kdump-tools.cfg'
Generating grub configuration file ...
grub-probe: warning: Couldn't find physical volume `(null)'. Some modules may
be missing from core image..
/usr/sbin/grub-probe: warning: Couldn't find physical volume `(null)'. Some
modules may be missing from core image..
/usr/sbin/grub-probe: warning: Couldn't find physical volume `(null)'. Some
modules may be missing from core image..
/usr/sbin/grub-probe: warning: Couldn't find physical volume `(null)'. Some
modules may be missing from core image..
/usr/sbin/grub-probe: warning: Couldn't find physical volume `(null)'. Some
modules may be missing from core image..
Found linux image: /boot/vmlinuz-5.4.0-42-generic
Found initrd image: /boot/initrd.img-5.4.0-42-generic
/usr/sbin/grub-probe: warning: Couldn't find physical volume `(null)'. Some
modules may be missing from core image..
/usr/sbin/grub-probe: warning: Couldn't find physical volume `(null)'. Some
modules may be missing from core image..
/usr/sbin/grub-probe: warning: Couldn't find physical volume `(null)'. Some
modules may be missing from core image..
/usr/sbin/grub-probe: warning: Couldn't find physical volume `(null)'. Some
modules may be missing from core image..
Adding boot menu entry for UEFI Firmware Settings
done
#
```

8. 初期 RAM 更新

```
# update-initramfs -k all -u
update-initramfs: Generating /boot/initrd.img-5.4.0-52-generic
#
```

9. chroot からログアウト

```
# exit
# umount -lf /mnt/run
# umount -lf /mnt/sys
# umount -lf /mnt/proc
# umount -lf /mnt/dev/pts
# umount -lf /mnt/dev
# umount -lf /mnt
#
```

10. システム起動

リストアしたシステムを起動します。

ビデオリダイレクション機能コンソール上で、以下の画面が表示されます。ライブメディアがアンマウントされていることを確認して [Enter] キーを押し、リストアしたシステムを起動してください。



POST 画面に Fujitsu ロゴが表示されたら [F11] キーを押して boot 選択画面を開き、リストアしたデバイス (/dev/sda の場合は "ubuntu(SATA0: ...)") を選択して起動してください。

システム起動後、意図した環境がリストアされていることを確認してください。

以降は、root ユーザーに変更して作業を実施してください。

```
$ sudo su
[sudo] password for zdlsadmin:
#
```

11. /dev/sdb パーティション設定

ミラー先の /dev/sdb のファイルシステムを初期化するため、parted_sdb.out ファイルを基に、/dev/sdb にパーティション情報を設定します。

```
# parted /dev/sdb
GNU Parted 3.3
Using /dev/sdb
Welcome to GNU Parted! Type 'help' to view a list of commands.
(parted) print
Model: ATA Micron 5300_MTF8 (scsi)
Disk /dev/sda: 960GB
Sector size (logical/physical): 512B/4096B
Partition Table: gpt
Disk Flags:
Number  Start      End          Size         File system  Name  Flags
  1      1049kB     538MB        537MB        fat32                boot, esp
  2      538MB     960GB        960GB
(parted) rm 2
(parted) rm 1
(parted) quit
Information: You may need to update /etc/fstab.

# wipefs -a /dev/sdb
/dev/md0: 8 bytes were erased at offset 0x00000200 (gpt): 45 46 49 20 50 41 52 54
/dev/md0: 8 bytes were erased at offset 0xdf67ffe0 (gpt): 45 46 49 20 50 41 52 54
/dev/md0: 2 bytes were erased at offset 0x000001fe (PMBR): 55 aa
/dev/md0: calling ioctl to re-read partition table: Success
#
# wipefs /dev/sdb
(何も表示されなければラベル情報はクリアされています)
# parted /dev/sdb
GNU Parted 3.2
Using /dev/sda
Welcome to GNU Parted! Type 'help' to view a list of commands.
(parted) unit s
(parted) mklabel gpt
(parted) mkpart
Partition name? []? ↵
File system type? [ext2]? ↵
Start? 2048
End? 1050623
(parted) set 1 esp on
(parted) mkpart
Partition name? []? ↵
File system type? [ext2]? ↵
Start? 1050624
End? 1875382271
(parted) set 2 raid on
(parted) print
Model: ATA Micron 5200 MTFD (scsi)
Disk /dev/sdb: 1875385008s
Sector size (logical/physical): 512B/4096B
Partition Table: gpt
Disk Flags:
Number  Start      End          Size         File system  Name  Flags
  1      2048s     1050623s     1048576s     ext2                boot, esp
  2     1050624s  1875382271s  1874331648s  ext2                raid
(parted) quit
Information: You may need to update /etc/fstab.
#
```


12. RAID 組込み

/dev/sdb に対して、/dev/md0 への組込みとブートローダをインストールします。

```
# mdadm --add /dev/md0 /dev/sdb2
mdadm: added /dev/sdb2
# cat /proc/mdstat
Personalities : [raid1] [linear] [multipath] [raid0] [raid6] [raid5] [raid4]
[raid10]
md0 : active raid1 sda2[2] sdb2[1]
      937033728 blocks super 1.2 [2/1] [_U]
      [====>.....] recovery = 20.1% (175056640/937033728)
      finish=56.1min speed=206114K/sec
      bitmap: 4/7 pages [16KB], 65536KB chunk

unused devices: <none>
# mkfs.fat -F32 /dev/sdb1
mkfs.fat 4.1 (2017-01-24)
# umount /boot/efi
# mount /dev/sdb1 /boot/efi
# grub-install --target=x86_64-efi --efi-directory=/boot/efi
Installing for x86_64-efi platform.
Installation finished. No error reported.
# ls -l /boot/efi/EFI/ubuntu/
total 4196
-rwxr-xr-x 1 root root 108 Dec 22 15:49 BOOTX64.CSV
-rwxr-xr-x 1 root root 162 Dec 22 15:49 grub.cfg
-rwxr-xr-x 1 root root 1681280 Dec 22 15:49 grubx64.efi
-rwxr-xr-x 1 root root 1269496 Dec 22 15:49 mmx64.efi
-rwxr-xr-x 1 root root 1334816 Dec 22 15:49 shimx64.efi
# umount /boot/efi
# mount /dev/sda1 /boot/efi
#
```

cat /proc/mdstat で /dev/sdb2 へのコピー状況と完了時間を確認してください。

13. エントリー追加

/dev/sdb1 でシステム起動するエントリー情報を追加します。

```
# efibootmgr --create --disk /dev/sdb --label "ubuntu redundancy" --loader
"\\EFI\\ubuntu\\shimx64.efi"
BootCurrent: 0000
Timeout: 1 seconds
BootOrder: 0007,0000,0001,0002,0003,0004,0005,0006
Boot0000* ubuntu
Boot0001* UEFI: NIC.LOM.1.1.IPv4HTTP - Intel(R) I350 Gigabit Network Connection
Boot0002* UEFI: NIC.LOM.1.1.IPv4PXE - Intel(R) I350 Gigabit Network Connection
Boot0003* UEFI: NIC.LOM.1.1.IPv6PXE - Intel(R) I350 Gigabit Network Connection
Boot0004* UEFI: NIC.LOM.1.2.IPv4HTTP - Intel(R) I350 Gigabit Network Connection
Boot0005* UEFI: NIC.LOM.1.2.IPv4PXE - Intel(R) I350 Gigabit Network Connection
Boot0006* UEFI: NIC.LOM.1.2.IPv6PXE - Intel(R) I350 Gigabit Network Connection
Boot0007* ubuntu redundancy
MirroredPercentageAbove4G: 0.00
MirrorMemoryBelow4GB: false
#
```

○ 備考

ご利用の環境によっては、UEFI のエントリー情報に余分なものが残っていたり、意図しない順番になっていたりとすることがあります。その場合は以下の手順を参考に、適切な状態に整理してください。

1. エントリーの一覧を表示する

```
# efibootmgr
BootCurrent: 0009
Timeout: 1 seconds
BootOrder: 0009,0001,000A,0008,0005,000B
Boot0000* AMI FWUpdate
Boot0001* ubuntu redundancy
Boot0005* UEFI: Built-in EFI Shell
Boot0008* USB
Boot0009* ubuntu
Boot000A* Hard Drive
Boot000B* <不要なエントリー名>
```

2. 不要なエントリーを削除する

不要なエントリーの Boot Num（上記 BootXXXX で表示されている XXXX 部分）を指定して inactive にします。Boot Num の * 表示が消えます。そのあと、削除コマンドで削除します。

```
# efibootmgr --inactive --bootnum 000B
BootCurrent: 0009
Timeout: 1 seconds
BootOrder: 0009,0001,000A,0008,0005,000B
Boot0000* AMI FWUpdate
Boot0001* ubuntu redundancy
Boot0005* UEFI: Built-in EFI Shell
Boot0008* USB
Boot0009* ubuntu
Boot000A* Hard Drive
Boot000B <不要なエントリー名>
# efibootmgr --bootnum 000B --delete-bootnum
BootCurrent: 0009
Timeout: 1 seconds
BootOrder: 0009,0001,000A,0008,0005
Boot0000* AMI FWUpdate
Boot0001* ubuntu redundancy
Boot0005* UEFI: Built-in EFI Shell
Boot0008* USB
Boot0009* ubuntu
Boot000A* Hard Drive
```

3. エントリーの起動順番を修正する

エントリー Boot Num 0009 から起動したい場合は、0009 を先頭に記載します。

```
# efibootmgr
BootCurrent: 0001
Timeout: 1 seconds
BootOrder: 0001,0009,000A,0008,0005
Boot0000* AMI FWUpdate
Boot0001* ubuntu redundancy
Boot0005* UEFI: Built-in EFI Shell
Boot0008* USB
Boot0009* ubuntu
Boot000A* Hard Drive
# efibootmgr --bootorder 0009,0001,000A,0008,0005
BootCurrent: 0009
Timeout: 1 seconds
BootOrder: 0009,0001,000A,0008,0005
Boot0000* AMI FWUpdate
Boot0001* ubuntu redundancy
Boot0005* UEFI: Built-in EFI Shell
Boot0008* USB
Boot0009* ubuntu
Boot000A* Hard Drive
```

以上でシステムリストアは完了です。

付録 A コマンドリファレンス

各確認項目の、出力例を説明します。モデルによって出力結果は異なります。

■ 格納済み Docker イメージを確認

現在格納されている Docker イメージを確認できます。

```
# docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
prom/prometheus-linux-amd64  latest            86ea6f86fc57       4 weeks ago       185MB
nvcr.io/nvidia/pytorch    21.10-py3         d88ec2b67660       7 weeks ago       14.3GB
nvcr.io/nvidia/mxnet      21.09-py3         be1e703caa5e       7 weeks ago       10.7GB
nvcr.io/nvidia/tensorflow 21.10-tf2-py3     08e720630243       7 weeks ago       10.6GB
#
```

■ Docker コンテナの運用状況を確認

```
# docker ps
CONTAINER ID   IMAGE                                COMMAND              NAMES
STATUS        PORTS
03ace78d4501   prom/prometheus-linux-amd64        "/bin/prometheus --c..." 2 days ago
Up 2 days     0.0.0.0:9090->9090/tcp, :::9090->9090/tcp  prometheus
```

■ GPU カードの使用状況を確認

```
# nvidia-smi
Thu Jun 17 17:34:34 2021
+-----+
| NVIDIA-SMI 470.57.02      Driver Version: 470.57.02      CUDA Version: 11.4      |
+-----+-----+
| GPU   Name               Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf    Pwr:Usage/Cap|      Memory-Usage | GPU-Util  Compute M. |
|                                           MIG M. |
+-----+-----+
|  0   A100-PCIE-80GB      On          | 00000000:01:00.0 Off |           0         |
| N/A   32C    P0         44W / 250W | 0MiB / 40536MiB |      0%      Default |
|                                           Disabled          |
+-----+-----+
|  1   A100-PCIE-80GB      On          | 00000000:25:00.0 Off |           0         |
| N/A   32C    P0         44W / 250W | 0MiB / 40536MiB |      0%      Default |
|                                           Disabled          |
+-----+-----+
|  2   A100-PCIE-80GB      On          | 00000000:81:00.0 Off |           0         |
| N/A   33C    P0         43W / 250W | 0MiB / 40536MiB |      0%      Default |
|                                           Disabled          |
+-----+-----+
|  3   A100-PCIE-80GB      On          | 00000000:c7:00.0 Off |           0         |
| N/A   33C    P0         43W / 250W | 0MiB / 40536MiB |      0%      Default |
|                                           Disabled          |
+-----+-----+

+-----+
| Processes: |
| GPU   GI   CI          PID    Type   Process name                      GPU Memory |
|      ID   ID                 |              |           Usage |
+-----+-----+
| No running processes found |
+-----+
```

■ GPU カードの MIG モードの設定と解除

- 設定

```
# nvidia-smi -mig 1
Warning: MIG mode is in pending enable state for GPU 00000000:01:00.0:In use by
another client
00000000:01:00.0 is currently being used by one or more other processes (e.g.
CUDA application or a monitoring application such as another instance of nvidia-
smi). Please first kill all processes using the device and retry the command or
reboot the system to make MIG mode effective.
Warning: MIG mode is in pending enable state for GPU 00000000:25:00.0:In use by
another client
00000000:25:00.0 is currently being used by one or more other processes (e.g.
CUDA application or a monitoring application such as another instance of nvidia-
smi). Please first kill all processes using the device and retry the command or
reboot the system to make MIG mode effective.
Warning: MIG mode is in pending enable state for GPU 00000000:81:00.0:In use by
another client
00000000:81:00.0 is currently being used by one or more other processes (e.g.
CUDA application or a monitoring application such as another instance of nvidia-
smi). Please first kill all processes using the device and retry the command or
reboot the system to make MIG mode effective.
Warning: MIG mode is in pending enable state for GPU 00000000:C7:00.0:In use by
another client
00000000:C7:00.0 is currently being used by one or more other processes (e.g.
CUDA application or a monitoring application such as another instance of nvidia-
smi). Please first kill all processes using the device and retry the command or
reboot the system to make MIG mode effective.
All done.
```

- 解除

```
# nvidia-smi -mig 0
Disabled MIG Mode for GPU 00000000:01:00.0
Disabled MIG Mode for GPU 00000000:25:00.0
Disabled MIG Mode for GPU 00000000:81:00.0
Disabled MIG Mode for GPU 00000000:C7:00.0
All done.
```

■ Docker コンテナを起動

以下は、GPU を 1 つ、CPU を 8 スレッド、メモリ容量を 64GB 割り当てて、TensorFlow の Docker コンテナを起動するコマンド例です。

```
# docker run --gpus 1 --cpuset-cpus=0-7 --memory=64g --shm-size=1g --ulimit
memlock=1 --it --rm -v /home/zdlsadmin:/workspace nvcr.io/nvidia/
tensorflow:21.10-tf2-py3

=====
== TensorFlow ==
=====

NVIDIA Release 21.10-tf2 (build 22382848)
TensorFlow Version 2.4.0

Container image Copyright (c) 2021, NVIDIA CORPORATION. All rights reserved.
Copyright 2017-2021 The TensorFlow Authors. All rights reserved.

NVIDIA Deep Learning Profiler (dlprof) Copyright (c) 2021, NVIDIA CORPORATION. All
rights reserved.

Various files include modifications (c) NVIDIA CORPORATION. All rights reserved.

This container image and its contents are governed by the NVIDIA Deep Learning
Container License.
By pulling and using the container, you accept the terms and conditions of this
license:
https://developer.nvidia.com/ngc/nvidia-deep-learning-container-license

NOTE: Legacy NVIDIA Driver detected. Compatibility mode ENABLED.

NOTE: MOFED driver for multi-node communication was not detected.
Multi-node communication performance may be reduced.

root@cf0eb09422d8:/workspace#
```

■ Docker コンテナへの割当てリソース確認

以下は、Docker コンテナで確認した例です。

```
root@cf0eb09422d8:/workspace# nvidia-smi
Thu Jun 17 08:46:50 2021
+-----+
| NVIDIA-SMI 470.57.02      Driver Version: 470.57.02      CUDA Version: 11.4      |
+-----+-----+-----+-----+-----+-----+
| GPU   Name               Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf    Pwr:Usage/Cap|      Memory-Usage | GPU-Util  Compute M. |
|                                           MIG M. |
+-----+-----+-----+-----+-----+-----+
|    0   A100-PCIE-40GB         On      | 00000000:01:00.0 Off  |             0      |
| N/A   32C    P0      44W / 250W | 0MiB / 40536MiB |      0%      Default |
|                                           Disabled  |
+-----+-----+-----+-----+-----+-----+

+-----+
| Processes: |
| GPU   GI    CI          PID    Type    Process name                        GPU Memory |
| ID     ID                                 Name                                Usage     |
+-----+-----+-----+-----+-----+-----+
| No running processes found |
+-----+

root@cf0eb09422d8:/workspace#
```

docker の詳細は以下を参照してください。

docker : <https://docs.docker.com/>

付録 B 外部ストレージ接続

Zinrai ディープラーニング システムでは、NAS ストレージとの接続が可能です。

Zinrai ディープラーニング システムでは、以下の仕様の NAS ストレージとの接続を推奨しています。

- インターフェース：Ethernet
- ファイルシステム：NFS

NAS ストレージとの接続手順例を以下に示します。実際の接続は、御使用の機器、環境に合わせて NAS ストレージと接続してください。

1. NAS ボリューム確認

サーバーから NAS ストレージ (IP : 192.168.11.11) の NAS ボリュームを確認します。

```
# showmount -e 192.168.11.11
Export list for 192.168.11.11:
/                (everyone)
/datafg01        (everyone)
/datavol01       (everyone)
#
```

2. NAS ボリュームのマウント設定

サーバーの /mnt ディレクトリに、NAS ストレージの /datafg01 をマウントする以下の設定を /etc/fstab ファイルに追記します。

```
192.168.11.11:/datafg01 /mnt nfs defaults 0 0
```

3. NAS ボリュームマウント

```
# mount -a
# df -h
<記載省略>
192.168.11.11:/datafg01 9.5T 6.4T 3.2T 67% /mnt
#
```

必要に応じて、IPTABLES で filter テーブルの INPUT チェインにアクセス許可を追加してください。

付録 C ライブメディア作成手順

boot 可能な ISO イメージをライブメディアと呼びます。

本製品に添付の DVD に格納されている起動用ライブメディアをカスタマイズする際の、ライブメディアの作成手順を説明します。必要に応じて適宜読み替えてください。

C.1 作成環境の事前要件

ライブメディアの作成には以下の環境が必要です。

Zinrai ディープラーニングシステム上で作成する必要はありません。

- ubuntu の公開リポジトリに接続できる環境
- ubuntu 20.04 が動作しているシステム

○ 備考

ここでは、Zinrai ディープラーニングシステムのバックアップやリストアを行うためのブート可能なメディアを作成する手順を説明しています。

[「第 8 章 バックアップリストア」\(P.46\)](#) で説明している手順以外でバックアップリストアを行う場合は、ライブメディアの作成は必須ではありません。

C.2 必要なツールのインストール

ライブメディアの作成環境に、必要なパッケージをインストールします。

```
$ sudo apt install squashfs-tools gpxboot xorriso mttools isolinux
```

C.3 ひな型メディアの入手と作成準備

C.3.1 ISO イメージファイルのダウンロード

ひな型メディアとしてデスクトップ環境の ISO イメージファイルを入手します。

ここでは curl コマンドでダウンロードする手順を示していますが、URL やファイル名が異なる場合は適宜読み替えてください。

また、ダウンロード済みの ISO イメージファイルがある場合は、以下の ~/livemedia ディレクトリに ISO イメージファイルをコピーしてください。

```
~$ mkdir ~/livemedia
~$ cd ~/livemedia
~/livemedia$ curl -O http://releases.ubuntu.com/20.04/ubuntu-20.04.2-desktop-
amd64.iso
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           %             0         0    942k         0  0:48:06  0:01:42  0:46:24   911k
:
```

C.3.2 ISO イメージファイルの展開

1. ISO イメージファイルマウント用、および ISO イメージファイル展開用の作業用ディレクトリを作成します。

```
~/livemedia$ mkdir mnt
~/livemedia$ mkdir extract-cd
```

2. ISO イメージファイルをマウントポイントにマウントして、展開用ディレクトリに必要な内容をコピーします。

```
~/livemedia$ sudo mount -o loop ubuntu-20.04.2-desktop-amd64.iso mnt
mount: /home/zdlsadmin/livemedia/mnt: WARNING: device write-protected, mounted
read-only.
~/livemedia$ sudo rsync --exclude=/casper/filesystem.squashfs -a mnt/ _
extract-cd
~/livemedia$ sudo unsquashfs mnt/casper/filesystem.squashfs
Parallel unsquashfs: Using 64 processors
182221 inodes (202230 blocks) to write

[=====
=====] 202230/202230 100%

created 153164 files
created 18759 directories
created 29017 symlinks
created 9 devices
created 0 fifos
~/livemedia$ sudo mv squashfs-root edit
```


C.3.3 作業環境の準備

ひな型メディアにパッケージを追加 / 削除して、ライブメディアの内容をカスタマイズするために、展開した ISO イメージに chroot でログインします。

■ 注 意

- 作業環境のネットワーク設定を ISO イメージにコピーするため、この ISO イメージを使って起動したシステムは作業環境と同じ IP アドレスになります。
IP アドレスを変更する場合は、`/etc/netplan/00-installer-config.yaml` を編集してください。
- 必要に応じて、お使いの環境の proxy 設定および iptables の設定変更を実施してください。

```
~/livemedia$ sudo _mount_ -o _bind_ /run/ _edit/run
~/livemedia$ sudo _mount_ --bind_ /dev/ _edit/dev
~/livemedia$ sudo _cp_ /etc/resolv.conf _edit/etc/
~/livemedia$ sudo _cp_ -p_ /etc/netplan/00-installer-config.yaml _edit/etc/
netplan/
~/livemedia$ sudo _-E_ chroot_ edit
/# dig google.com

; <<>> DiG 9.16.1-Ubuntu <<>> google.com ;; global options: +cmd ;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 63906 ;; flags: qr rd ra;
QUERY: 1, ANSWER: 0, AUTHORITY: 1, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
;; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;google.com.                IN      A

;; AUTHORITY SECTION:
google.com.                 71      IN      A      142.250.XXX.XXX

;; Query time: 16 msec
;; SERVER:192.168.XX.XX#53(192.168.XX.XX)
;; WHEN: Tue Jul 06 16:16:26 JST 2021
;; MSG SIZE rcvd: 144

/# mount_ -t_ proc_ none_ /proc
/# mount_ -t_ sysfs_ none_ /sys
/# mount_ -t_ devpts_ none_ /dev/pts
/# export_ HOME=/root
/# export_ LC_ALL=C
```

C.4 ライブメディア環境のカスタマイズ

ひな型メディアにパッケージを追加 / 削除してライブメディアの内容をカスタマイズします。

C.4.1 パッケージの追加と削除

1. リポジトリの情報を更新します。

```
/# apt update
Hit:1 http://archive.ubuntu.com/ubuntu focal InRelease
Get:2 http://security.ubuntu.com/ubuntu focal-security InRelease [109 kB]
Get:3 http://archive.ubuntu.com/ubuntu focal-updates InRelease [114 kB]
Get:4 http://security.ubuntu.com/ubuntu focal-security/main amd64 DEP-11 Metadata [24.3 kB]
Get:5 http://archive.ubuntu.com/ubuntu focal-updates/main amd64 DEP-11 Metadata [263 kB]
Get:6 http://archive.ubuntu.com/ubuntu focal-updates/main DEP-11 64x64 Icons [87.9 kB]
Fetched 598 kB in 3s (234 kB/s)
Reading package lists... Done
Building dependency tree
Reading state information... Done
359 packages can be upgraded. Run 'apt list --upgradable' to see them.
```

2. ISO イメージファイルから不要なパッケージを削除します。

```
/# apt remove --purge -y ^bluez* ^cheese* ^cups* ^firefox* ^girl.2-
gnomebluetooth.* ^hyphen.* ^libfontembed.* ^libgnome-games-support.*
^ubiquity.* ^thunder* ^libreoffice* ^yaru-theme* ^tegaki* ^samba*
^fonts* ^language-pack* ^guile* ^mythes* ^mozc* ^ubuntu-wallpapers*
^libfreerdp* ^openprinting* ^libhangul* ^shotwell* ^gnome-* ^xcursor*
^modem* ^xserver* ^printer*
:
Purging configuration files for libreoffice-calc (1:6.4.4-0ubuntu0.20.04.2) ...
Processing triggers for ufw (0.36-6) ...
Processing triggers for systemd (245.4-4ubuntu3.2) ...
Running in chroot, ignoring request: daemon-reload
Processing triggers for dbus (1.12.16-2ubuntu2.1) ...
Processing triggers for fontconfig (2.13.1-2ubuntu3) ...
ERROR couldn't connect to zsys daemon: connection error: desc = "transport: Error
while dialing dial unix /run/zsysd.sock: connect: no such file or directory"
/# apt autoremove -y
:
Processing triggers for gnome-menus (3.36.0-1ubuntu1) ...
Processing triggers for libglib2.0-0:amd64 (2.64.3-1~ubuntu20.04.2) ...
Processing triggers for libc-bin (2.31-0ubuntu9) ...
Processing triggers for man-db (2.9.1-1) ...
Processing triggers for dbus (1.12.16-2ubuntu2.1) ...
Processing triggers for udev (245.4-4ubuntu3.2) ...
Processing triggers for install-info (6.7.0.dfsg.2-5) ...
Processing triggers for desktop-file-utils (0.24-1ubuntu3) ...
ERROR couldn't connect to zsys daemon: connection error: desc = "transport: Error
while dialing dial unix /run/zsysd.sock: connect: no such file or directory"
/#
```

3. 追加パッケージをインストールします。

```
/# apt install -y software-properties-common
:
/# apt-add-repository universe
'universe' distribution component enabled for all sources.
Hit:1 http://archive.ubuntu.com/ubuntu focal InRelease
Get:2 http://security.ubuntu.com/ubuntu focal-security InRelease [109 kB]
Get:3 http://archive.ubuntu.com/ubuntu focal-updates InRelease [114 kB]
:
Get:24 http://archive.ubuntu.com/ubuntu focal-updates/universe amd64 c-n-f
Metadata [13.7 kB]
Fetched 30.8 MB in 25s (1,219 kB/s)
Reading package lists... Done
/# apt install -y ncdu nfs-common dump mduadm gparted openssh-server
```

C.4.2 クリーンアップ処理

```
#!/# umount /proc || umount -lf /proc
#!/# umount /sys
#!/# umount /dev/pts
#!/# umount /dev
#!/# exit
exit
~/livemedia$
```

C.5 ライブメディアイメージの作成

ライブメディアイメージを作成します。

C.5.1 ファイルシステムの構築

1. ファイルシステムの作成準備をします。

```
~/livemedia$ sudo chmod +w extract-cd/casper/filesystem.manifest
~/livemedia$ sudo -sE
livemedia/# chroot edit dpkg-query -W --showformat='${Package}
${Version}\n' > extract-cd/casper/filesystem.manifest
livemedia/# printf $(sudo du -sx --block-size=1 edit | cut -f1) >
extract-cd/casper/filesystem.size
livemedia/# exit
exit
~/livemedia$ sudo cp -v extract-cd/casper/filesystem.manifest extract-cd/
casper/filesystem.manifest-desktop
~/livemedia$ REMOVE='ubiquity ubiquity-frontend-gtk ubiquity-frontend-kde
casper lupin-casper live-initramfs user-setup discover1 xresprobe
os-prober libdebian-installer4'
~/livemedia$ for i in $REMOVE
~/livemedia$ do
~/livemedia$ sudo sed -i "${i}/d" extract-cd/casper/filesystem.manifest-
desktop
~/livemedia$ done
```

2. 読み取り専用ファイルシステムを作成します。

```
~/livemedia$ sudo mksquashfs edit extract-cd/casper/filesystem.squashfs
-noappend -comp xz -e edit/boot
Parallel mksquashfs: Using 64 processors
Creating 4.0 filesystem on extract-cd/casper/filesystem.squashfs, block size
131072.
[-----] 322/300021 0%
Unrecognised xattr prefix system.posix_acl_access
:
unknown (130)
unknown (127)
unknown (124)
tss (111)
adm (4)
syslog (110)
```

付加情報を作成します。

```
~/livemedia$ cd _extract-cd
~/livemedia/extract-cd$ sudo _rm _md5sum.txt
~/livemedia/extract-cd$ find _-type _f _-print0 _|_sudo _xargs _-0 _
md5sum _|_grep _-v _isolinux/boot.cat _|_sudo _tee _md5sum.txt
:
fc3d6a8927f4fb1f2a760bcdabd39556c ./dists/focal/main/binary-i386/Release
51b97824001923dac66e7386d9b83a87 ./dists/focal/main/binary-i386/Packages.gz
4d2da0407037856fd92564a2e3e29383 ./dists/focal/main/binary-amd64/Release
0274b0831ed88ad7f18e21fe2f8112d7 ./dists/focal/main/binary-amd64/Packages.gz
~/livemedia/extract-cd$
```

C.5.2 メディアイメージの書き出し

作成した ISO イメージから、ハイブリッドブート可能なファイルを作成します。

```
~/livemedia/extract-cd# sudo _xorriso _-as _mkisofs _-r _-iso-level _3 _-V _
'ZDLS _boot' _-J _-joliet-long _-boot-info-table _-eltorito-alt-boot _--efi-boot _
boot/grub/efi.img _-append_partition _2 _0xef _boot/grub/efi.img _
-partition_cyl_align_all _-b isolinux/isolinux.bin _-c _isolinux/boot.cat _-no-
emul-boot _-boot-load-size _4 _-boot-info-table _-isohybrid-mbr _/usr/lib/ISOLINUX/
isohdpx.bin _-isohybrid-gpt-basdat _-o ../ubuntu-20.04.2-desktop-amd64-zdls.iso _
xorriso 1.4.8 : RockRidge filesystem manipulator, libburnia project.

Drive current: -outdev 'stdio:../ubuntu-20.04.2-desktop-amd64-zdls3.iso'
Media current: stdio file, overwriteable
Media status : is blank
Media summary: 0 sessions, 0 data blocks, 0 data, 725g free
xorriso : WARNING : -volid text problematic as automatic mount point name
xorriso : WARNING : -volid text does not comply to ISO 9660 / ECMA 119 rules
Added to ISO image: directory '/'='/home/zdlsadmin/livemedia/extract-cd'
xorriso : UPDATE : 925 files added in 1 seconds
:
ISO image produced: 1095440 sectors
Written to medium : 1095440 sectors at LBA 0
Writing to 'stdio:../ubuntu-20.04.2-desktop-amd64-zdls3.iso' completed successfully.
livemedia/extract-cd#
```

~/livemedia/ 配下に、ライブメディアの ISO イメージ (ubuntu-20.04.2-desktop-amd64-zdis.iso) が作成されます。作業用の PC などにコピーして保管してください。

○ 備考

以下の画面が表示されているときに [Ctrl]+[C] を押すと、ファイルシステムのチェック処理がスキップされ、起動時間を短縮できます。



FUJITSU AI
Zinrai ディープラーニング システム 200M
システム運用管理者ガイド

発行日 2021 年 12 月
Copyright 2021 FUJITSU LIMITED

- 本書の内容は、改善のため事前連絡なしに変更することがあります。
- 本書の無断転載を禁じます。

正誤表

下記の内容が変更になっていますので適宜読み替えをお願いします。

項番	場所	元記事	訂正内容	備考
1	P.71	<pre>~\$ mkdir ~/livemedia ~\$ cd ~/livemedia ~/livemedia\$ curl -O http://releases.ubuntu.com/20.04/ubuntu-20.04.2-desktop-amd64.iso</pre>	<pre>~\$ mkdir ~/livemedia ~\$ cd ~/livemedia ~/livemedia\$ curl -O https://old-releases.ubuntu.com/releases/20.04.2/ubuntu-20.04.2.0-desktop-amd64.iso</pre>	ダウンロード URL 変更
2	P.19	<pre># This is the network config written by 'subiquity' network: ethernets: enp97s0f0: addresses: - <HostOS IP address>/<Subnet mask> gateway4: <Gateway IP address> nameservers: addresses: [<DNS IP address>] search: [] enp97s0f1: dhcp4: true version: 2</pre>	<pre># This is the network config written by 'subiquity' network: ethernets: eno1: addresses: - <HostOS IP address>/<Subnet mask> gateway4: <Gateway IP address> nameservers: addresses: [<DNS IP address>] search: [] eno2: dhcp4: true version: 2</pre>	デバイス名変更