

FUJITSU AI

Zinrai ディープラーニング システム 200H/200E  
システム運用管理者ガイド

Copyright 2020 FUJITSU LIMITED

P3KD-1832-01

# 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

ライブメディアを作成する手順を説明します。

## 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 200H/200E 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 200H/200E 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しています。

- Zinrai ディープラーニング システム 200H を、GX2570 M5 と表記しています。
- Zinrai ディープラーニング システム 200E を、RX2540 M5 と表記しています。
- Graphics Processing Unit を、GPU と表記しています。

### 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
[ ]	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]

記号	意味
[ ]	コマンド入力で、パラメーターの選択肢を示します 例：[A B]

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>	

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な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。

2020年7月初版



# 目次

第1章 概要 .....	7
1.1 ソフトウェア構成 .....	7
1.2 サーバーのボリューム構成 .....	7
1.3 留意事項 .....	8
第2章 システム起動停止手順 .....	9
2.1 システム起動 .....	9
2.1.1 電源ボタンからシステム起動 .....	10
2.1.2 Web インターフェースからシステム起動 .....	10
2.1.3 ビデオリダイレクションからシステム起動 .....	11
2.2 システム停止 .....	12
2.2.1 OS からシステム停止 .....	12
2.2.2 Web インターフェースからシステム停止 .....	13
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 (GX2570 M5の場合) .....	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 iRMCのIPアドレス変更 (RX2540 M5の場合) .....	19
3.1.2.1 BIOS セットアップユーティリティでIPアドレスを変更 .....	19
3.1.2.2 Web インターフェースでIPアドレスを変更 .....	21
3.1.3 Zinrai ディープラーニングシステムのIPアドレス変更 .....	23
3.1.4 Docker 内部 LAN IP アドレス変更 .....	28
3.2 ポート番号変更 .....	29
第4章 Prometheus 監視関連 .....	31
4.1 監視システム管理画面 .....	31
4.1.1 監視システム管理画面へのログイン .....	31
4.2 監視システム設定 .....	32
4.3 アラート設定 .....	33
4.3.1 アラートルール .....	33
4.3.2 メールアラート設定 .....	33
4.3.3 IPTABLES 設定追加 .....	33
4.3.4 アラート機能有効化 .....	33
第5章 障害調査 .....	34
5.1 トラブルシューティング .....	34
5.2 ディスクの寿命確認 .....	35
5.2.1 寿命確認手順 .....	35
5.2.2 書込みデータ量の確認方法 .....	35

第 6 章	SSL サーバー証明書運用 .....	36
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 .....	36
6.2	自己署名の SSL サーバー証明書を利用する場合 .....	36
第 7 章	ソフトウェア RAID 管理 .....	37
7.1	RAID 構成情報の取得 .....	37
7.2	RAID ボリュームの詳細情報の取得 .....	38
7.3	内蔵ストレージの交換 .....	40
7.3.1	活性交換 .....	40
7.3.1.1	RAID 構成から内蔵ストレージを切り離す .....	40
7.3.1.2	RAID 構成に内蔵ストレージを組み込む .....	41
7.3.2	システム停止交換 .....	44
7.3.2.1	RAID 構成から内蔵ストレージを切り離す .....	44
7.3.2.2	RAID 構成に内蔵ストレージを組み込む .....	45
第 8 章	バックアップリストア .....	49
8.1	システムバックアップ .....	49
8.1.1	ファイルシステム情報採取 .....	50
8.1.2	ライブメディアから OS を起動 .....	51
8.1.2.1	ライブメディアから OS を起動 (GX2570 M5 の場合) .....	52
8.1.2.2	ライブメディアから OS を起動 (RX2540 M5 の場合) .....	55
8.1.3	システムバックアップ実施 .....	57
8.2	システムリストア .....	59
8.2.1	ライブメディアから OS を起動 .....	60
8.2.2	ファイルシステム作成 .....	61
8.2.3	システムリストア実施 .....	64
付録 A	コマンドリファレンス .....	69
付録 B	外部ストレージ接続 .....	71
付録 C	ライブメディア作成手順 .....	72

# 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。

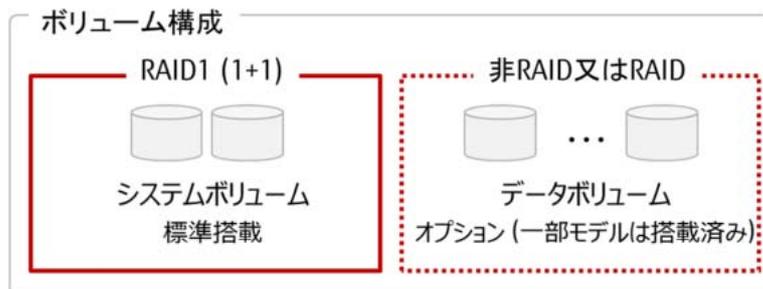
表 1：ソフトウェア構成

名称	
ホスト OS 環境	Ubuntu 18.04 LTS
	CUDA 10.2
	nvidia-container-toolkit 1.1.2
	Data Center GPU Manager 1.7.2
	Docker-CE 19.03
対話型学習環境 (フレームワーク)	対話型 Docker イメージ (TensorFlow 20.03-tf2-py3)
	対話型 Docker イメージ (PyTorch 20.03-py3)
	対話型 Docker イメージ (MxNet 20.03-py3)
監視サーバー	Prometheus 2.19

## 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、ゲスト OS、ソフトウェアなどを格納)
- データボリューム (一部モデル、またはオプション)  
ユーザー領域 (データセット、学習モデル、ユーザーデータなどを格納) としてユーザーが利用可能な領域です。データボリューム (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応をしてください。

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おりません。
- 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」\(P.12\)](#)の手順に従ってください。

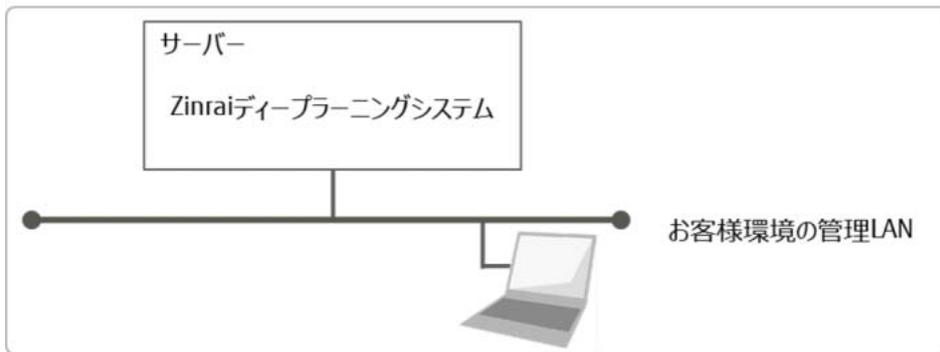
---

## 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を使用します。

お客様環境の管理 LAN などに端末を接続して、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」 \(P.10\)](#)
- [「2.1.2 Web インターフェースからシステム起動」 \(P.10\)](#)
- [「2.1.3 ビデオリダイレクションからシステム起動」 \(P.11\)](#)

#### ■ システム起動の確認

システムの起動は、ビデオリダイレクション機能の表示画面で確認します。そのため、システムを起動する前にビデオリダイレクション画面を開いてください。

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 18.04.4 LTS <hostname> tty1
<hostname> login:
```

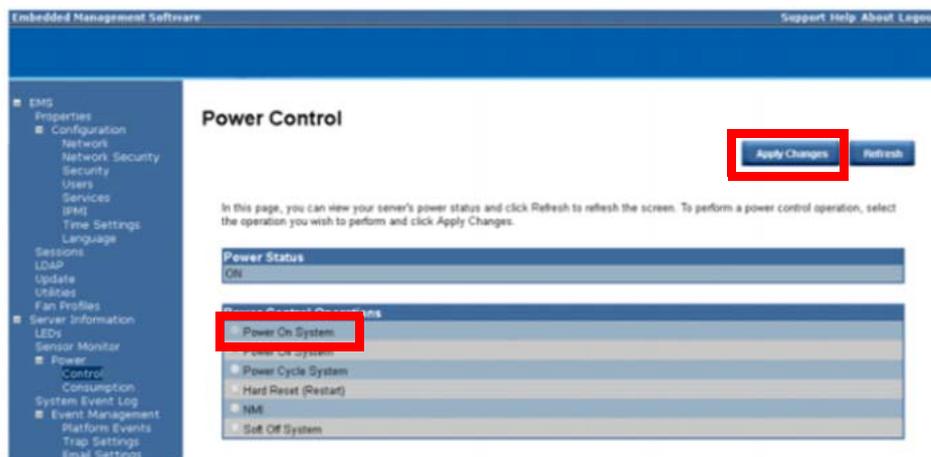
## 2.1.1 電源ボタンからシステム起動

装置前面の電源ボタンを押し、システムを起動します。



## 2.1.2 Web インターフェースからシステム起動

- GX2570 M5 の場合  
Web インターフェースの [Power Control] メニューから「Power On System」を選択し、[Apply Changes] ボタンをクリックします。



- RX2540 M5 の場合  
Web インターフェースの電源アイコンをクリックします。

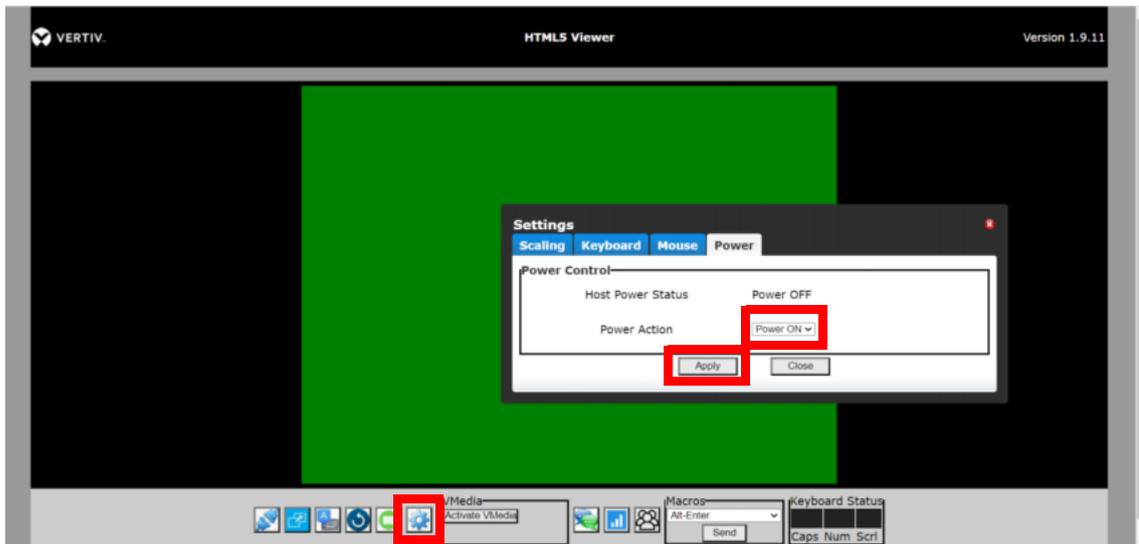


## 2.1.3 ビデオリダイレクションからシステム起動

- GX2570 M5 の場合

Web インターフェースの [Server Information] - [vKVM & vMedia] - [Launch] 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。

歯車アイコンをクリックし、[Power] メニューの [Power Action] で [Power ON] を選択して、[Apply] ボタンをクリックします。



- RX2540 M5 の場合

Web インターフェースの [Remote Control] メニューでビデオリダイレクションを起動し、[Power] メニューの「Power On Server」または電源アイコンをクリックします。



## 2.2 システム停止

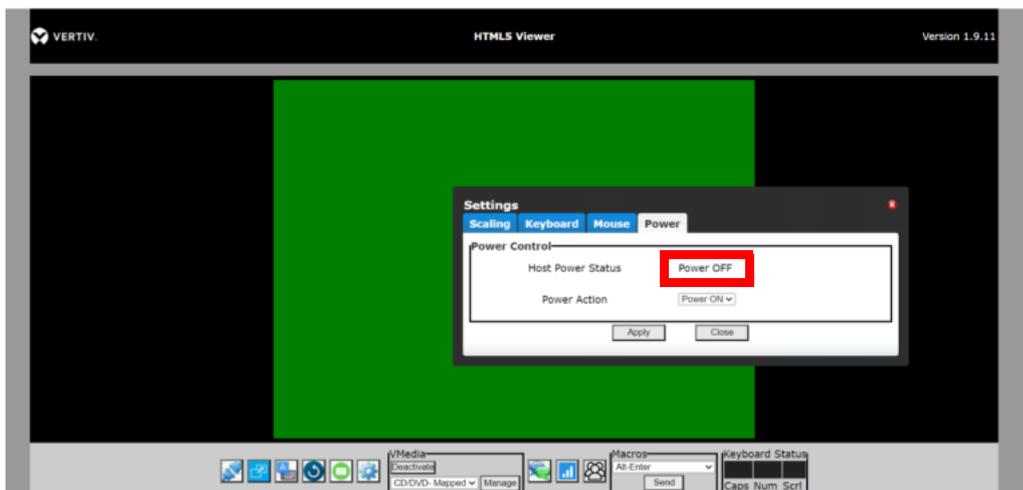
以下のどれかの操作でシステムを停止します。御利用環境に応じてシステムを停止してください。

- [2.2.1 OS からシステム停止] (P.12)
- [2.2.2 Web インターフェースからシステム停止] (P.13)
- [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  
[Power Control] メニューの 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
  - GX2570 M5 の場合  
[Power] メニューの Host Power Status が Power OFF になる



- RX2540 M5 の場合  
メニューバーにある電源アイコンが赤くなる



- 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

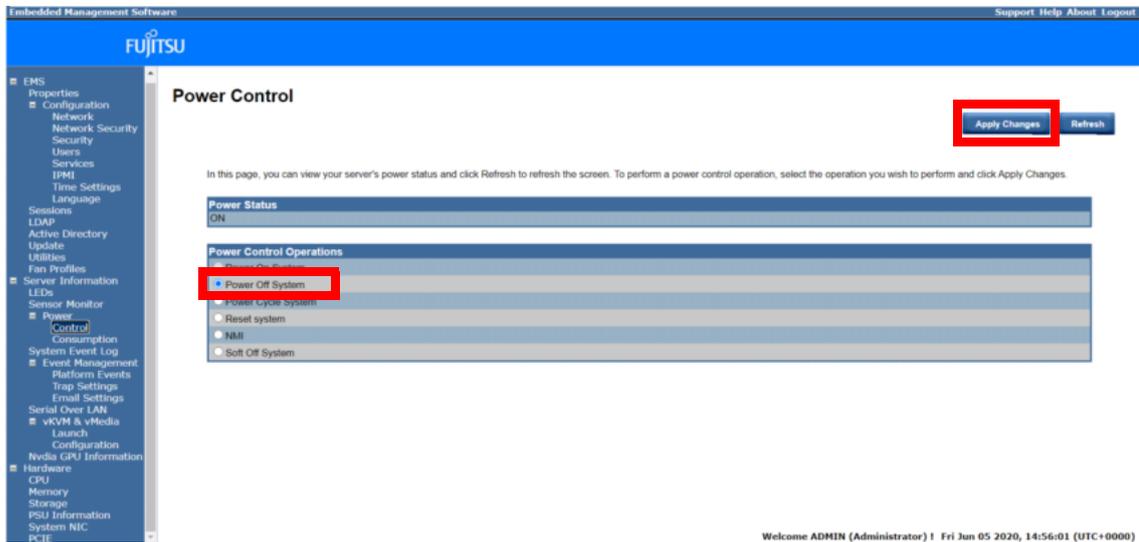
### 2.2.1 OS からシステム停止

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。

```
# shutdown -h now
```

## 2.2.2 Web インターフェースからシステム停止

- GX2570 M5 の場合  
Web インターフェースの [Power Control] メニューから「Power Off System」を選択し、[Apply Changes] ボタンをクリックします。



- RX2540 M5 の場合  
Web インターフェースの電源アイコンをクリックし、表示されるメニューから「即時電源オフ」を選択します。

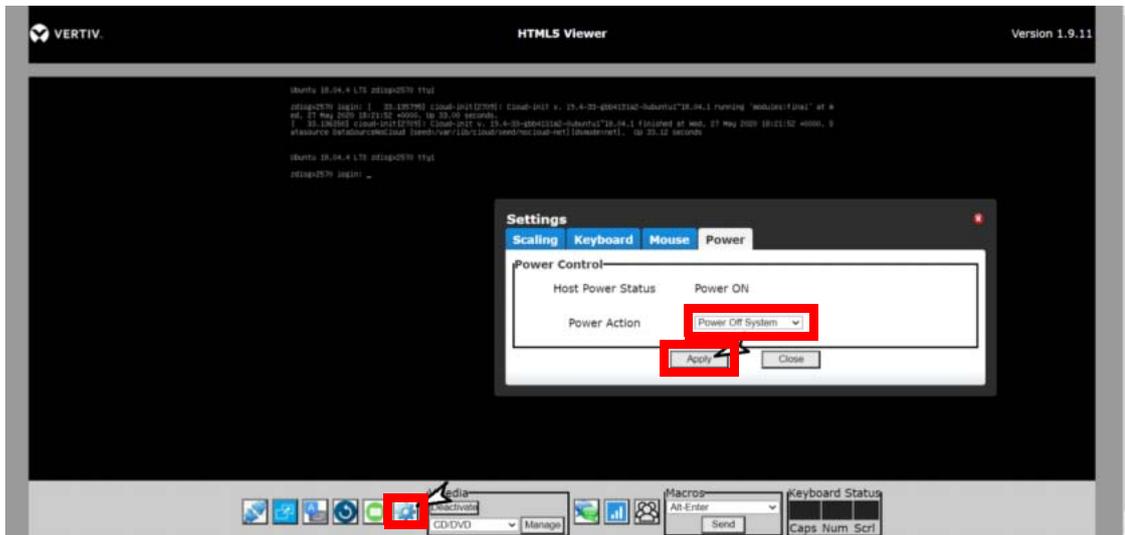


## 2.2.3 ビデオリダイレクションからシステム停止

- GX2570 M5 の場合

Web インターフェースの [Server Information] - [vKVM & vMedia] - [Launch] 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。

歯車アイコンをクリックし、[Power] メニューの [Power Action] で [Power Off System] を選択して、[Apply] ボタンをクリックします。



- RX2540 M5 の場合

Web インターフェースの [Remote Control] メニューでビデオリダイレクションを起動し、[Power] メニューの「Orderly Shutdown」または電源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

---

以下のどちらかの操作で IP アドレス情報を変更します。御利用環境に応じて参照してください。

- [「3.1.1 BMC の IP アドレス変更 \(GX2570 M5 の場合\)」 \(P.15\)](#)
- [「3.1.2 iRMC の IP アドレス変更 \(RX2540 M5 の場合\)」 \(P.19\)](#)

#### 3.1.1 BMC の IP アドレス変更 (GX2570 M5 の場合)

---

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  
装置前面の 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。

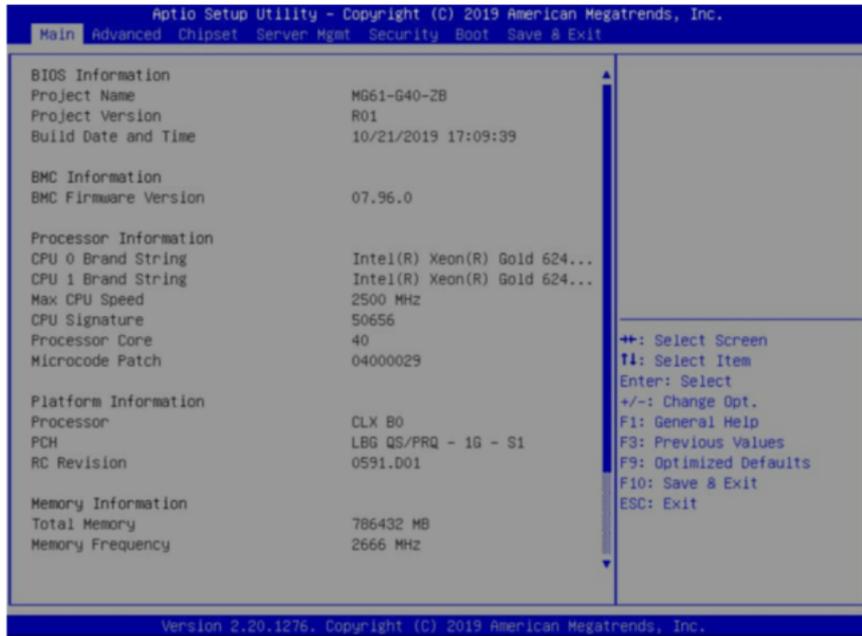
図 2：装置前面図 (VGA/USB ポート)



## 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 [Del (Delete)] 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。

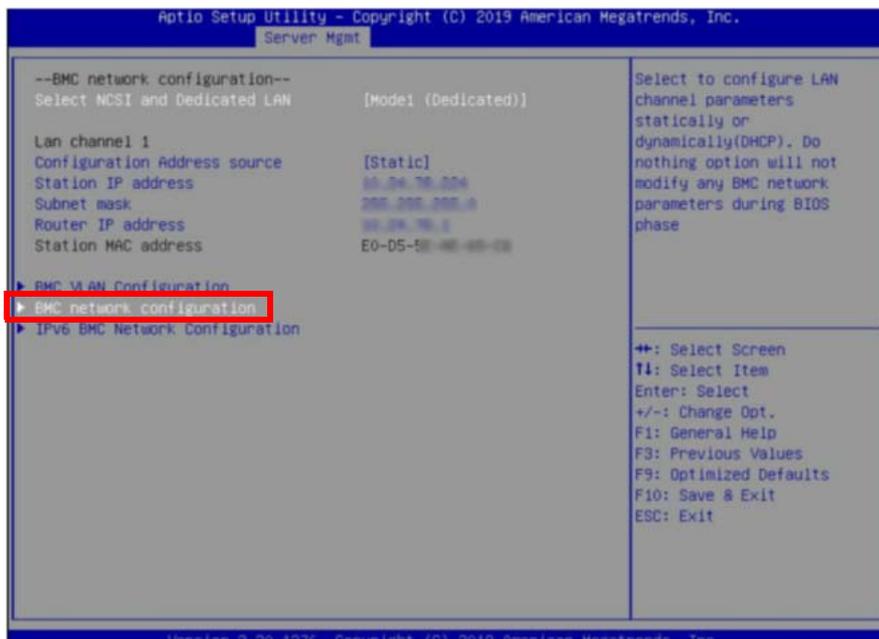
BIOS セットアップユーティリティを起動する前にシステムが起動してしまった場合は、装置を再起動してください。



## 3. Web インターフェースネットワークの設定

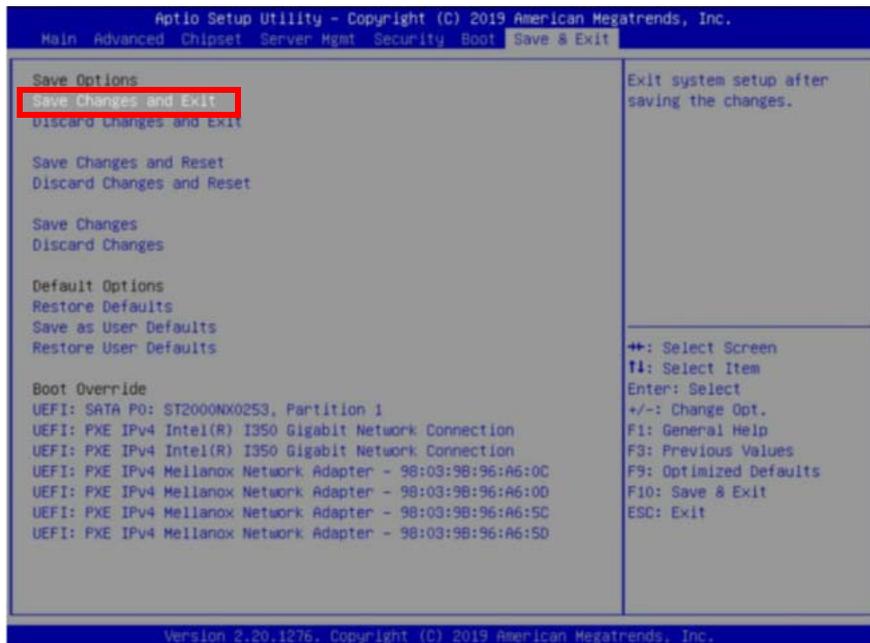
[Server Mgmt] メニューで「BMC network configuration」を選択し、環境に合わせて以下のネットワーク設定を行ってください。

Configuration Address source : Static / DynamicBmcDhcp  
IP address : xxx.xxx.xxx.xxx  
Subnet mask : yyy.yyy.yyy.yyy



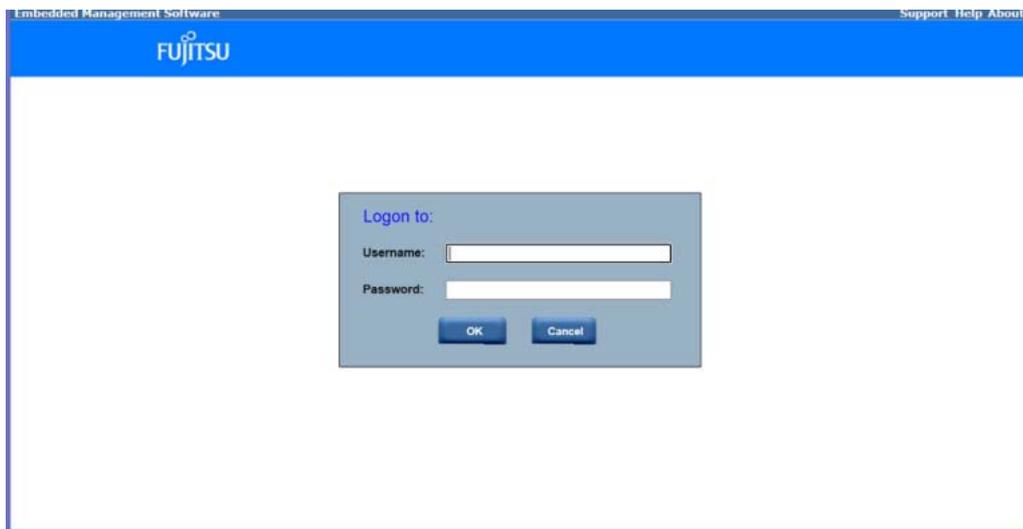
#### 4. 設定の保存

[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。

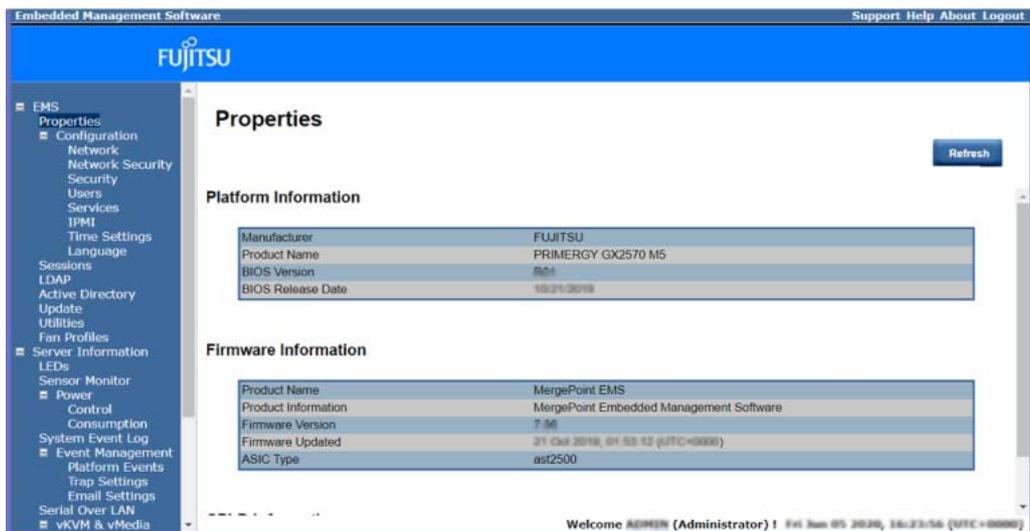


### 3.1.1.2 Web インターフェースで IP アドレスを変更

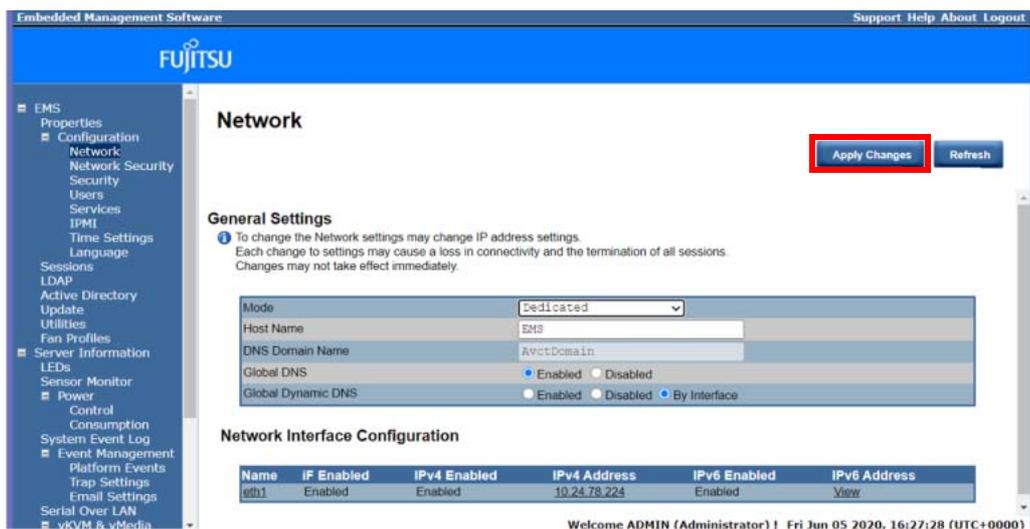
1. Web ブラウザーに「https://192.168.1.11/」を入力して、Web インターフェースにアクセスします。  
ログイン画面が表示されます。



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、[EMS] - [Configuration] - [Network] 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 Interface Configuration に設定するネットワーク情報を入力し、ページ右上部にある [Apply Changes] ボタンをクリックします。



5. 変更内容はすぐに表示されない場合があります。その場合は、[Refresh] ボタンをクリックすると表示が更新されます。
6. 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 iRMC の IP アドレス変更 (RX2540 M5 の場合)

---

Web インターフェースをもつシステムコントローラー (iRMC) 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。

iRMC 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。

- [「3.1.2.1 BIOS セットアップユーティリティで IP アドレスを変更」 \(P.19\)](#)
- [「3.1.2.2 Web インターフェースで IP アドレスを変更」 \(P.21\)](#)

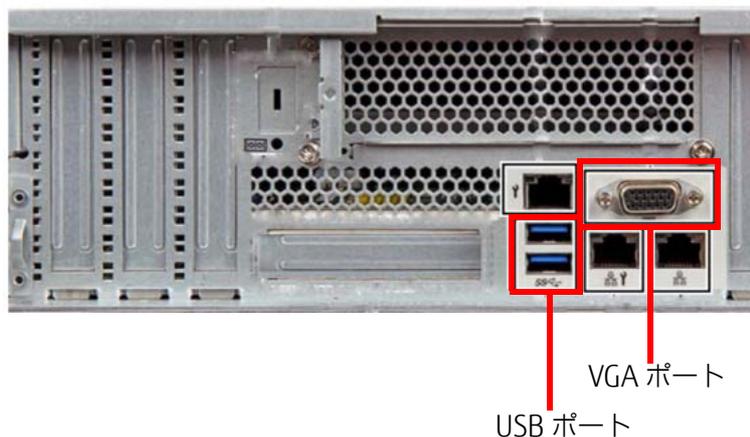
### 3.1.2.1 BIOS セットアップユーティリティで IP アドレスを変更

---

#### 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。

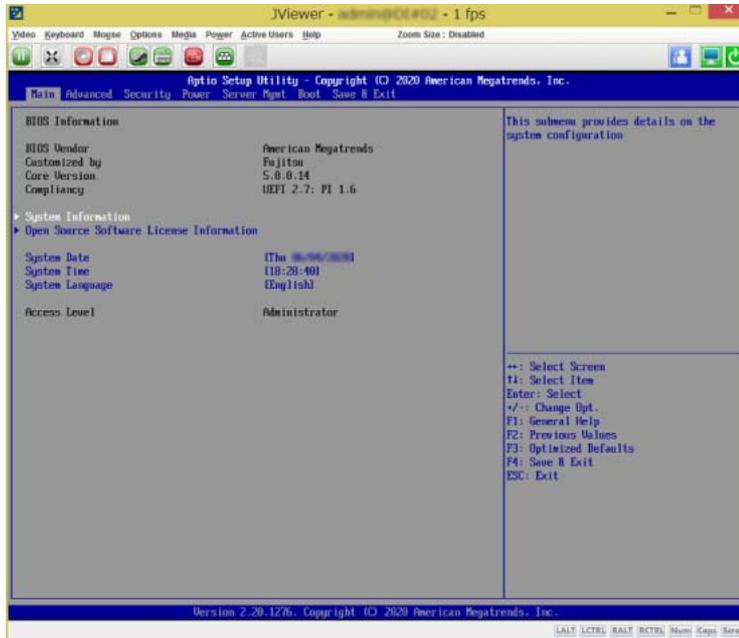
図 3：装置背面図 (VGA/USB ポート)



## 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。

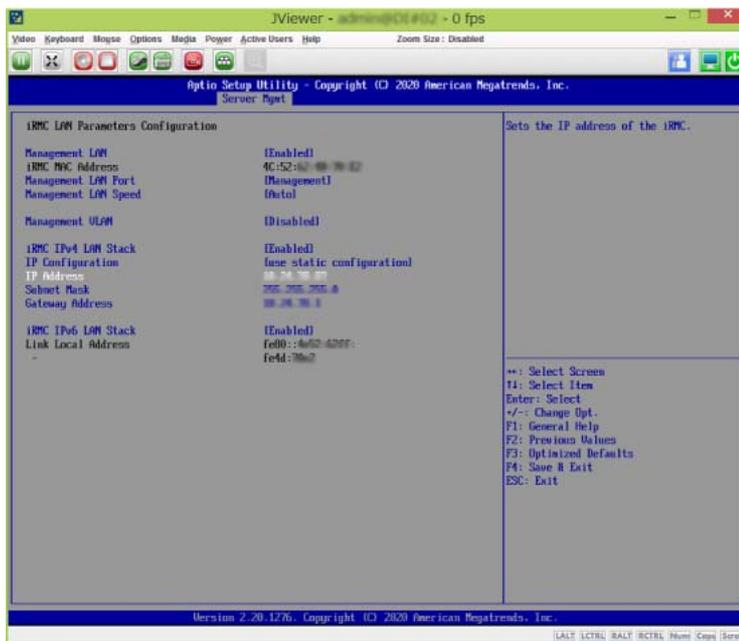
BIOS セットアップユーティリティを起動する前にシステムが起動してしまった場合は、装置を再起動してください。



## 3. Web インターフェースネットワークの設定

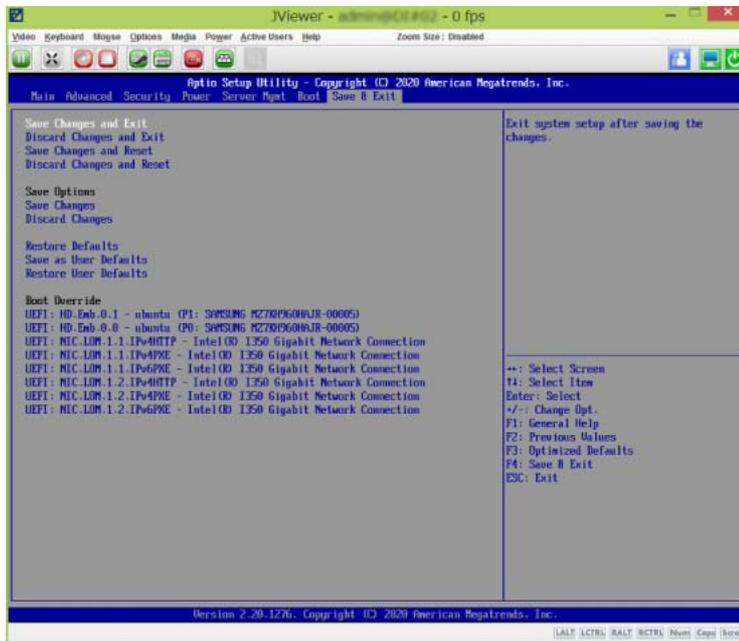
[Server Mgmt] メニューで「iRMC LAN Parameters Configuration」を選択し、環境に合わせて以下のネットワーク設定を行ってください。

IP Configuration : use static configuration  
IP address : xxx.xxx.xxx.xxx  
Subnet mask : yyy.yyy.yyy.yyy



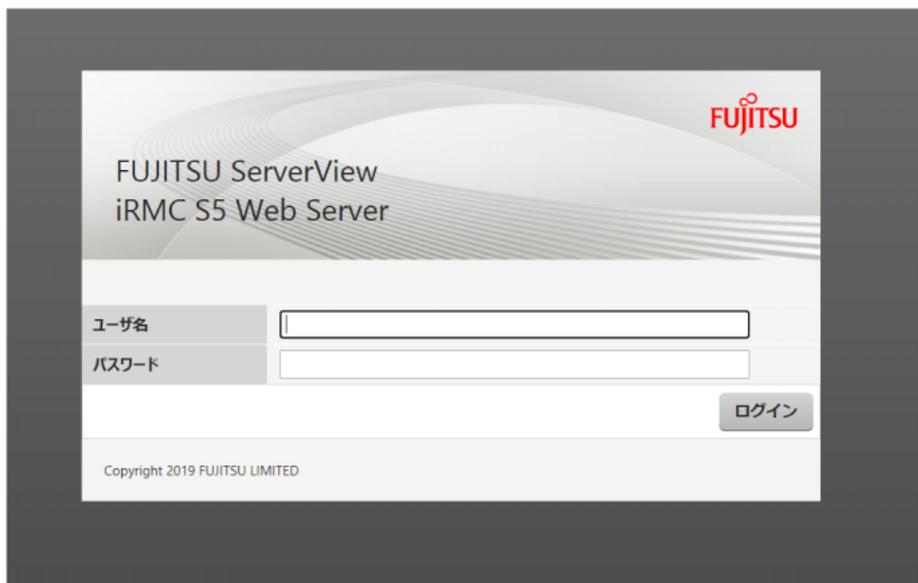
#### 4. 設定の保存

[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。



### 3.1.2.2 Web インターフェイスで IP アドレスを変更

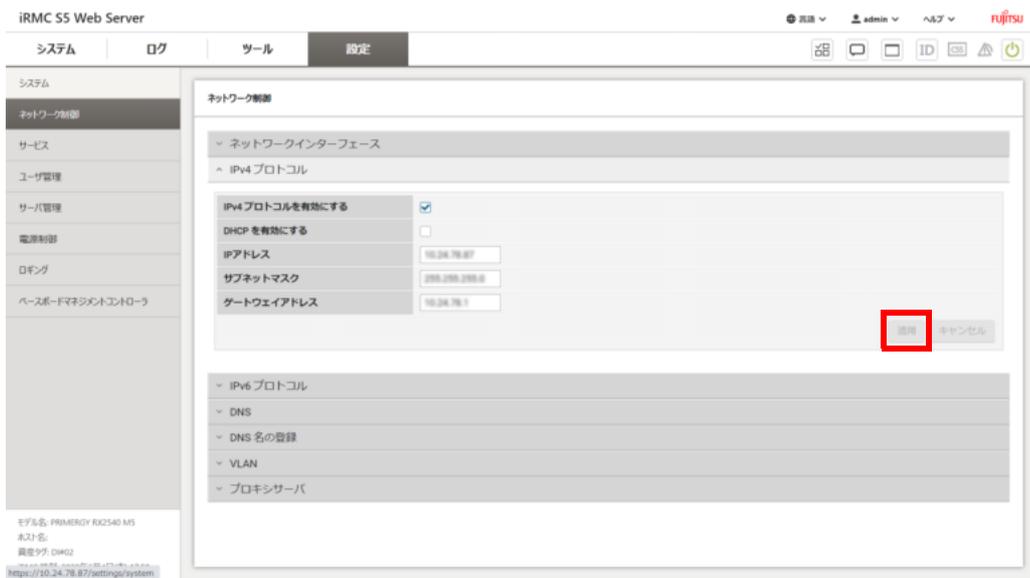
1. Web ブラウザーに「https://192.168.1.11/」を入力して、Web インターフェイスにアクセスします。  
ログイン画面が表示されます。



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. [設定] タブの [ネットワーク制御] 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. 各種プロトコル、DNS、プロキシサーバーなどに設定するネットワーク情報を入力し、各項目画面の下部にある [適用] 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。

### 3.1.3 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` により 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/hosts
- /etc/netplan/50-cloud-init.yaml
- /etc/prometheus/prometheus.yml
- /etc/grafana/grafana.ini

#### 1. /etc/hosts 変更

行	変更前	変更後
2	192.168.1.10 <hostname>	<HostOS IP address> <hostname>

#### 2. /etc/netplan/50-cloud-init.yaml のバックアップ

工場出荷時のファイル「50-cloud-init.yaml」を、以下のコマンドでバックアップします。

```
cp -p /etc/netplan/50-cloud-init.yaml /etc/netplan/50-cloud-init.yaml.org
```

#### 3. /etc/netplan/50-cloud-init.yaml の修正

お客様の環境に合わせて、「50-cloud-init.yaml」を修正します。

```
network:
  ethernets:
    eno1:
      addresses:
        - <HostOS IP/Subnet mask>
      gateway4: <Gateway address>
      nameservers:
        addresses: [<DNS addresses>]
    eno2:
      addresses:
        - <HostOS IP/Subnet mask>
      gateway4: <Gateway address>
      nameservers:
        addresses: [<DNS addresses>]
  version: 2
```

#### 4. /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: ['192.167.1.10:3903']	- targets: ['<HostOS IP address>:3903']

#### 5. /etc/grafana/grafana.ini 変更

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

#### 6. IPTABLES 変更

- (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 9090 -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 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
```

## 7. 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
```

## 8. 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 ACCEPT に変更します。

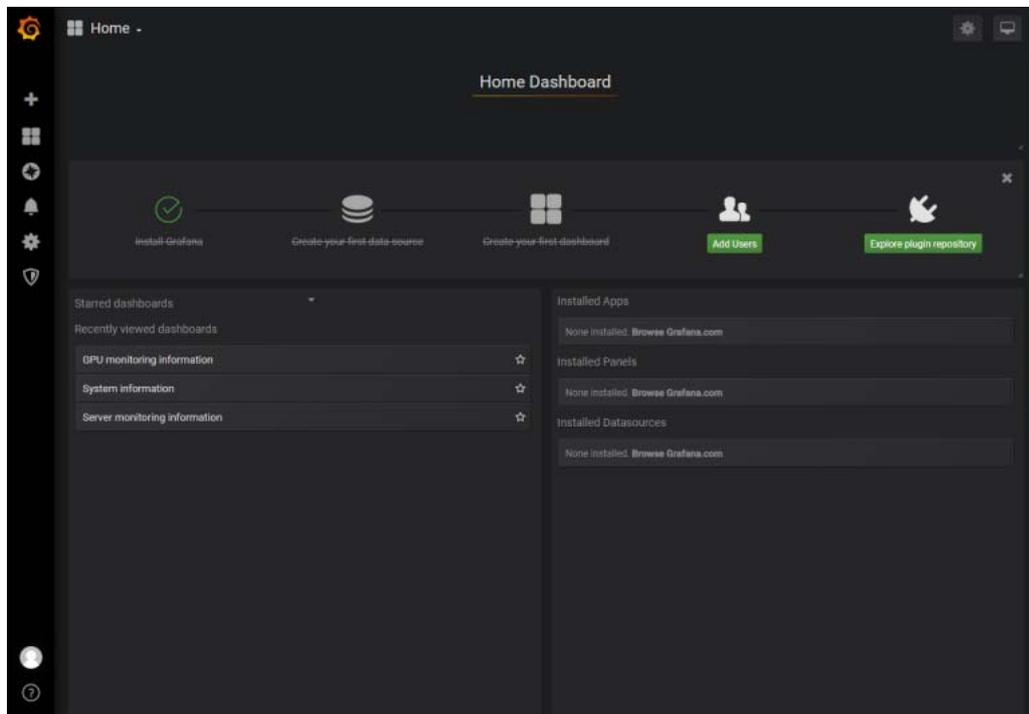
```
# iptables -P INPUT DROP
# /etc/init.d/netfilter-persistent save
* Saving netfilter rules...
run-parts: executing /usr/share/netfilter-persistent/plugins.d/15-ip4tables
save
run-parts: executing /usr/share/netfilter-persistent/plugins.d/25-ip6tables
save
[ OK ]

# /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 ]

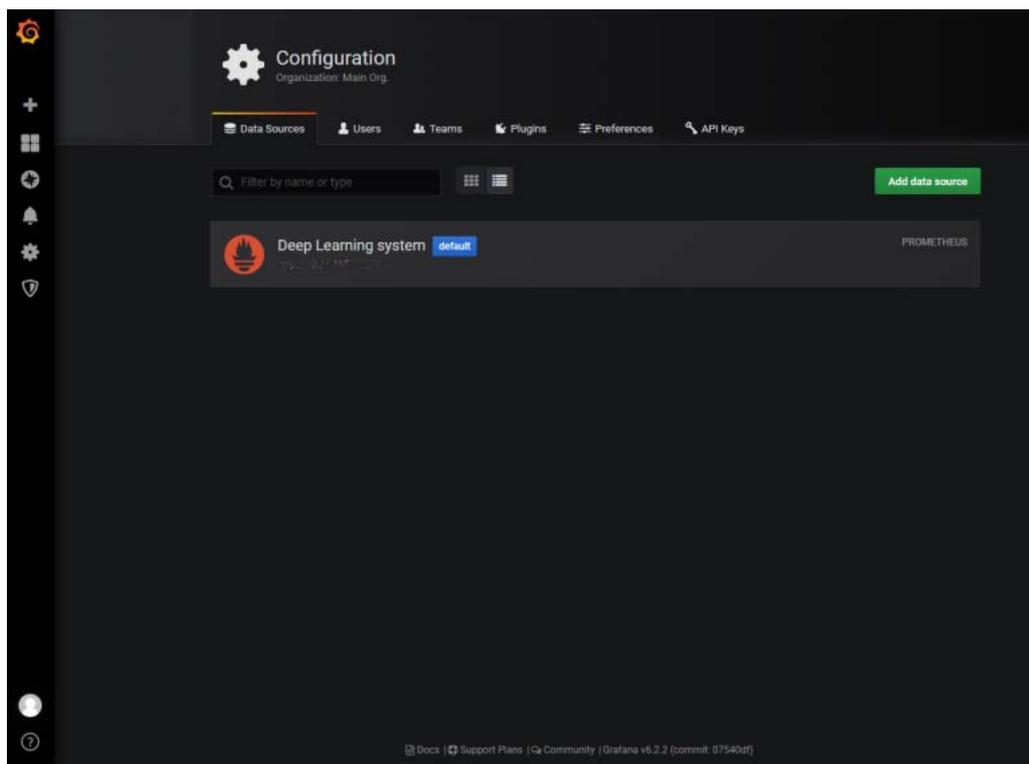
#
```

## 9. Grafana 設定変更

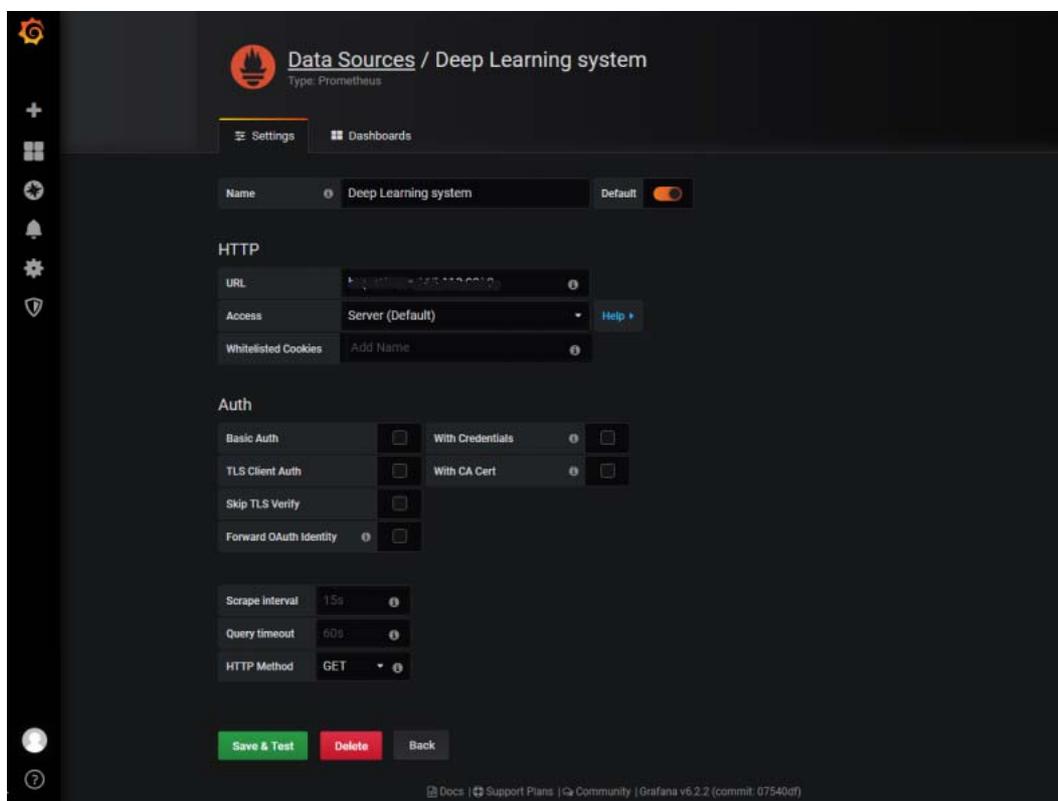
- (1) Web ブラウザーに、「<https://<HostOS IP address>:3000>」を入力し、Grafana の Home Dashboard 画面にアクセスします。



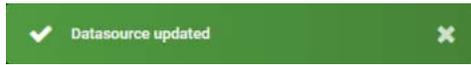
- (2) 左側メニューの [Configuration] - [Data Sources] を開きます。



- (3) [Deep Learning system] を開き、HTTP 項目の URL に設定したホスト 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。

### 3.1.4 Docker 内部 LAN 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
#
```

#### 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
# systemctl status 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 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>:8000>) し、各監視メトリクスが収集できていれば問題なく動作しています。

## 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れば問題なく動作しています。

---

## 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

---

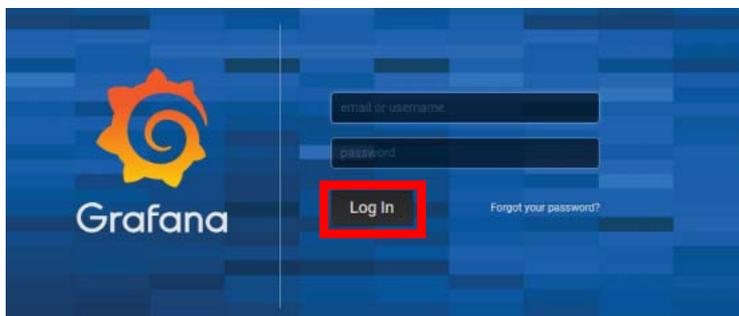
監視システム管理画面の利用について説明します。監視項目の設定追加、変更、および監視状況の確認は、監視システム管理画面から行います。

#### 4.1.1 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。

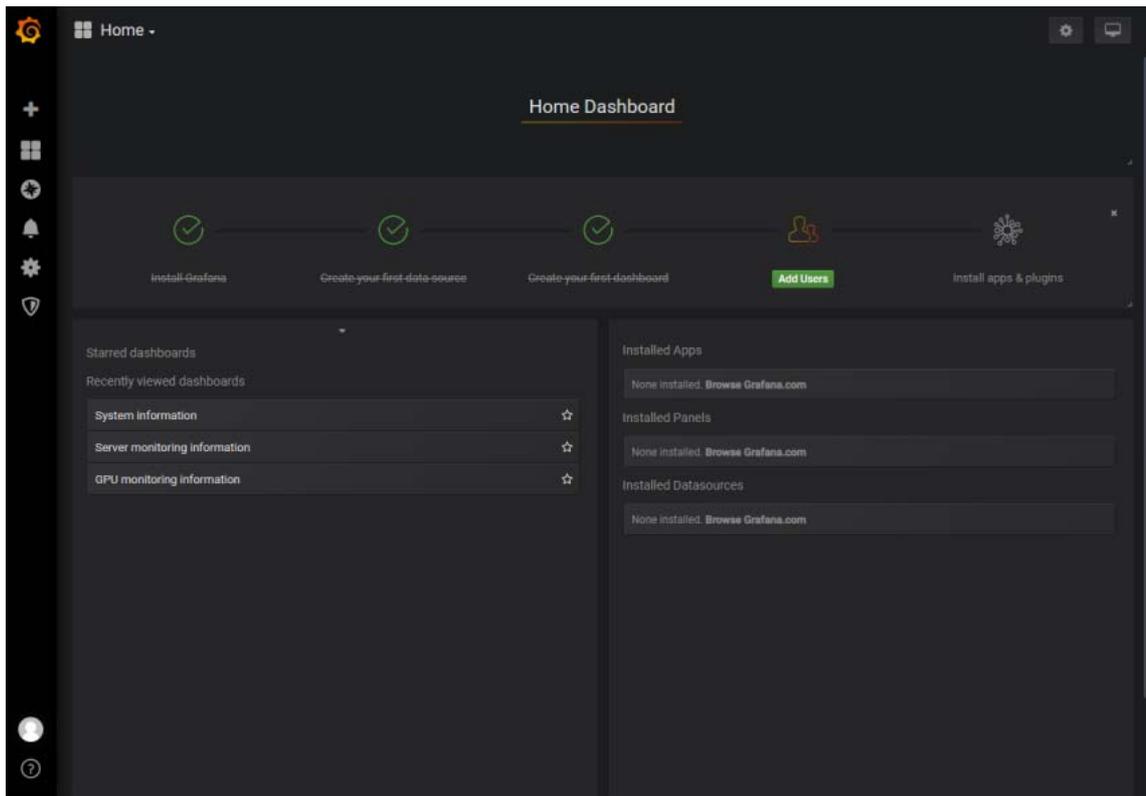


なお、工場出荷時は以下の username と password を設定しています。

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  
Web インターフェースで収集した温度や各 FRU のステータス
- GPU monitoring system  
GPU の消費電力、動作温度およびステータス

## 4.2 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/>

## 4.3 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。Zinrai ディープラーニングシステムにサンプルとして設定した、アラートメール内にあるリンクからはアクセスできません。

### 4.3.1 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/](https://prometheus.io/docs/prometheus/latest/configuration/alerting_rules/)

### 4.3.2 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/>

### 4.3.3 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
```

### 4.3.4 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 May 22 19:59 512D-8228 -> ../../sda1
lrwxrwxrwx 1 root root 11 May 22 19:59 706c0082-4a85-44a8-a54e-0e58706a7659 ->
../../md0p2
lrwxrwxrwx 1 root root 11 May 22 19:59 fbd34e24-809d-4350-b193-c2319b40451f ->
../../md0p1
#
```

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。

smartctl コマンドで確認します。

- SATA SSD-960GB の場合 (/dev/sda の例)

```
# smartctl -l devstat,7 /dev/sda
```

### 5.2.2 書き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、書き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、書き込み寿命となります。

対象SSD	確認項目	書き込み寿命	算出方法
PYBSS96NM6 : SATA SSD-960GB	Percent Lifetime Remaining	100	初期値はほぼ0% 書き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 6.6 2016-05-31 r4324 [x86_64-linux-4.15.0-106-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  N--  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]
      869923840 blocks super 1.2 [2/2] [UU]
      bitmap: 0/7 pages [0KB], 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
      869923840 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   sync   /dev/sda2
    -----
    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。

```
# 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
GNU Parted 3.2
Using /dev/sda
Welcome to GNU Parted! Type 'help' to view a list of commands.
(parted) 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
  2      538MB   960GB   960GB

(parted) quit
#
```

### ○ 備考

交換用の内蔵ストレージは、ここで確認したモデルと同じものを用意してください。

## 7.3 内蔵ストレージの交換

内蔵ストレージは、以下のどちらかの方法で交換してください。

- [「7.3.1 活性交換」\(P.40\)](#)
- [「7.3.2 システム停止交換」\(P.44\)](#)

内蔵ストレージを交換する場合は、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。交換作業は、当社サポート窓口にご連絡してください。

### ■ 注意

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」\(P.37\)](#) 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

交換対象の内蔵ストレージが搭載されているロケータ LED（赤）を点滅させます。故障モードによっては、LED は点滅しません。

```
# ledctl locate=/dev/sda
#
```

##### 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**
```

#### 4. 内蔵ストレージの取外し

サーバーから交換対象の内蔵ストレージを取り外します。

サーバーから内蔵ストレージを取り外すと、ロケータ LED は消灯します。

### 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、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. 内蔵ストレージの搭載

サーバーに内蔵ストレージを搭載します。

サーバーに内蔵ストレージを搭載すると、ロケータ LED の点滅が再開します。

#### 2. 内蔵ストレージのロケータ LED を消灯

```
# ledctl locate_off=/dev/sda
#
```

#### 3. 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
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.
>>> 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.
```

#### 4. 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 3.2
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
```

#### 5. 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
#
```

#### 6. 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>
```

#### 7. 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
#
```

## 8. 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
#
```

## 9. /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
# 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
```

## 10. 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-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
#
```

## 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. 内蔵ストレージ交換

システムを停止したあと、内蔵ストレージを交換し、システムを起動します。

### 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.
#
```

## 2. 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 3.2
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
```

## 3. 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/sda1
mdadm: added /dev/sda1
#
```

## 4. 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]
      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>
#
```

## 5. 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
#
```

## 6. 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* EFI: NIC.LOM.1.1.IPv4HTTP - Intel(R) I350 Gigabit Network
Connection
Boot0002* EFI: NIC.LOM.1.1.IPv4PXE - Intel(R) I350 Gigabit Network Connection
Boot0003* EFI: NIC.LOM.1.1.IPv6PXE - Intel(R) I350 Gigabit Network Connection
Boot0004* EFI: NIC.LOM.1.2.IPv4HTTP - Intel(R) I350 Gigabit Network
Connection
Boot0005* EFI: NIC.LOM.1.2.IPv6PXE - Intel(R) I350 Gigabit Network Connection
Boot0006* EFI: NIC.LOM.1.2.IPv6PXE - Intel(R) I350 Gigabit Network Connection
Boot0007* ubuntu redundancy
MirroredPercentageAbove4G: 0.00
MirrorMemoryBelow4GB: false
#
```

## 7. /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 10b101ea-8a01-4150-88f9-910870336c87 ->
../..md0p2
lrwxrwxrwx 1 root root 10 Jun 27 06:33 A1D7-999F -> ../..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
#

# 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-id/md-uuid-54703f0a:d8b12c34:d5922c8a:88fc32b2 none swap sw 0 0
# / was on /dev/md0 during curtin installation
/dev/disk/by-id/md-uuid-5b204118:c08814b7:78d6a33d:40e9b827 / ext4 defaults 0
0
# /boot/efi was on /dev/sda1 during curtin installation
/dev/disk/by-uuid/A1D7-999F /boot/efi vfat defaults 0 0
```

## 8. 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

バックアップリストアについて説明します。

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。御使用のハードウェア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 2.5 インチ SATA デバイス (/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、ディープラーニング処理を実施しないでください。

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

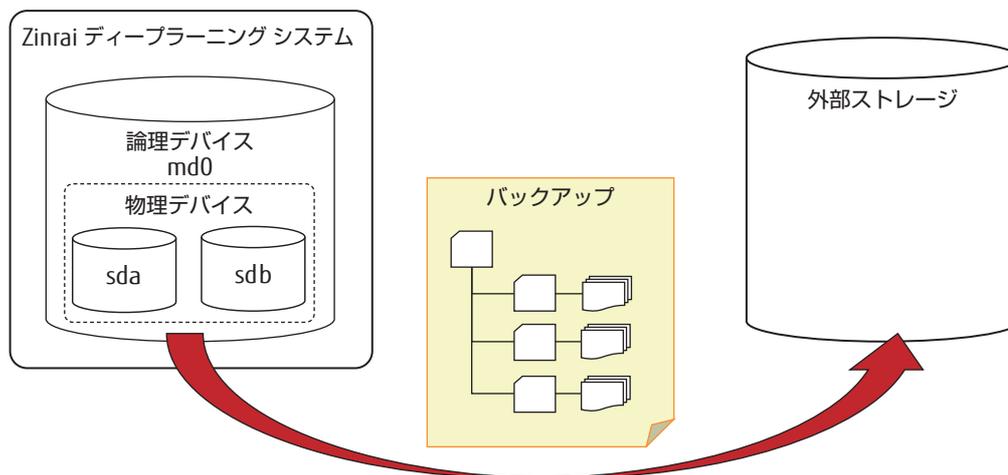
[\[8.1.1 ファイルシステム情報採取\] \(P.50\)](#)



[\[8.1.2 ライブメディアから OS を起動\] \(P.51\)](#)

### ● 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
```

以下の lsblk の例では、/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できます。

```
#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 ライブメディアから OS を起動

「付録 C ライブメディア作成手順」(P.72) で作成したライブメディアを起動した環境で、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。

### ■ 注意

シングルユーザーモードでバックアップを実施した場合、OS は稼働しているため、リストア時に予期しないエラーが発生することがあります。富士通では、シングルユーザーモードでバックアップを実施して、リストアしたシステムの動作は保証いたしておりません。

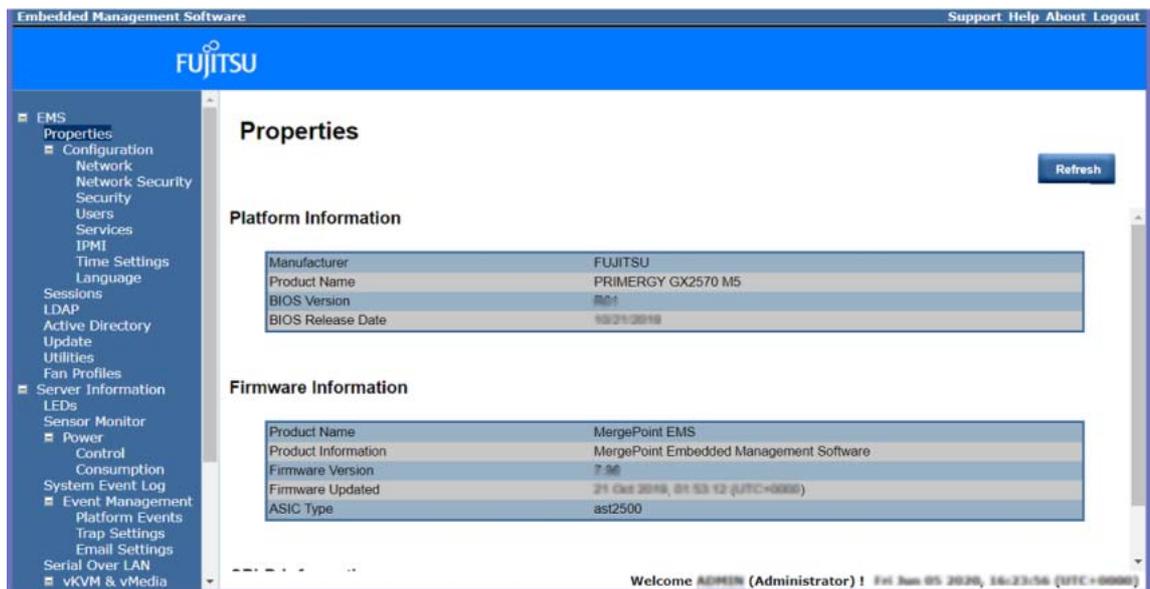
## 8.1.2.1 ライブメディアから OS を起動（GX2570 M5 の場合）

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。

- 「付録 C ライブメディア作成手順」(P.72) で作成した ISO イメージ
- 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（サーバーの空スロットに挿入してください）
- サーバーの Web インターフェースにログインできる端末

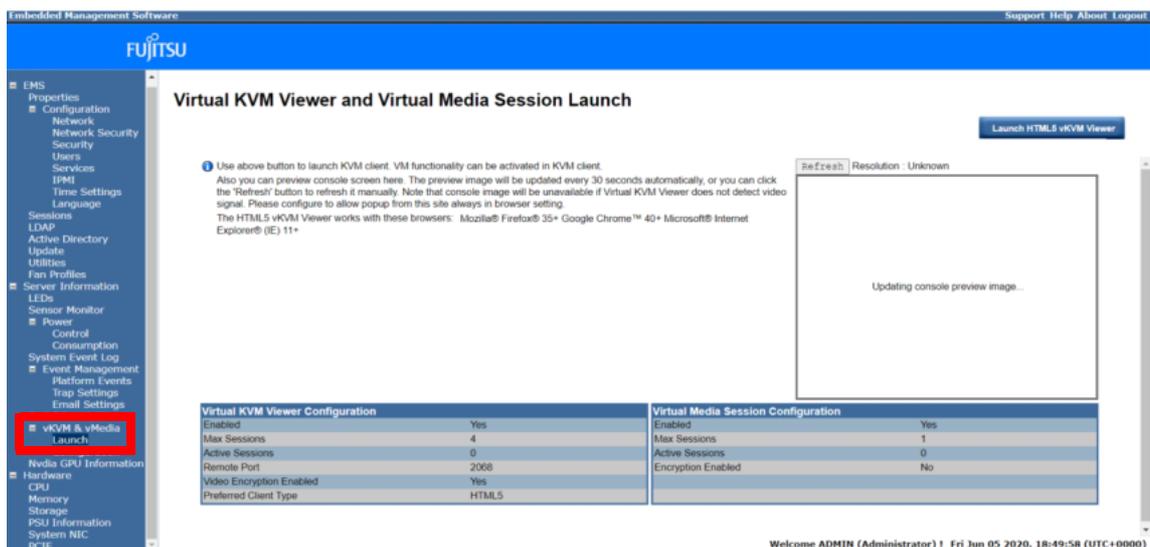
### 1. Web インターフェースにログイン

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

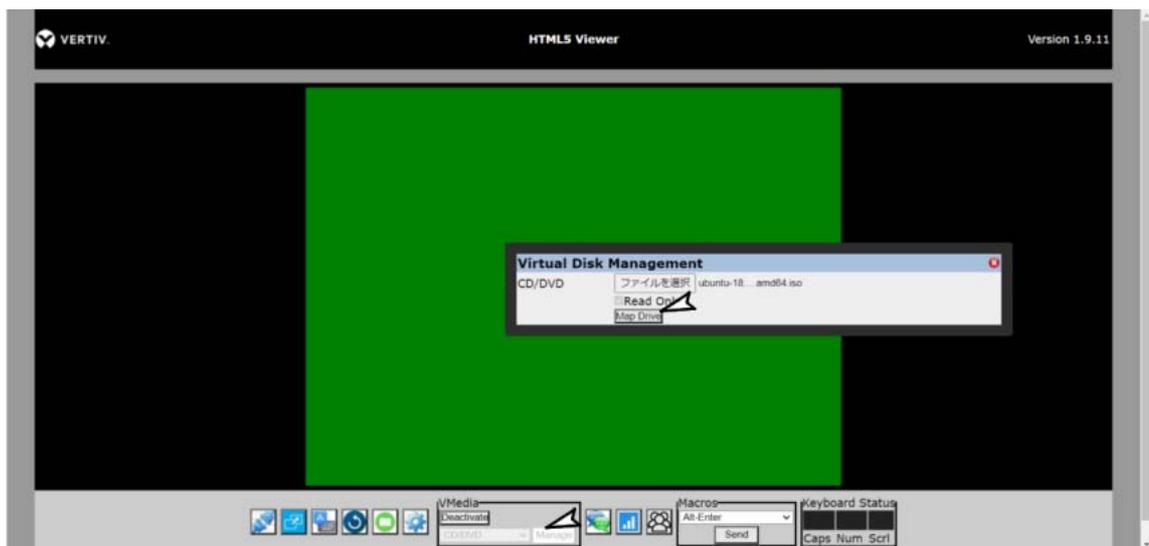
[Server Information] - [vKVM & vMedia] - [Launch] でビデオリダイレクション機能を起動します。



3. Virtual Media Wizard を開く  
[VMedia] 内で、[Activate VMedia] - [Manage] の順にクリックします。



4. ライブメディアをマウント  
Virtual Disk Management 画面で、[ファイルを選択] からライブメディアを選択して、[Map Drive] をクリックします (Read only モードでマウントされます)。

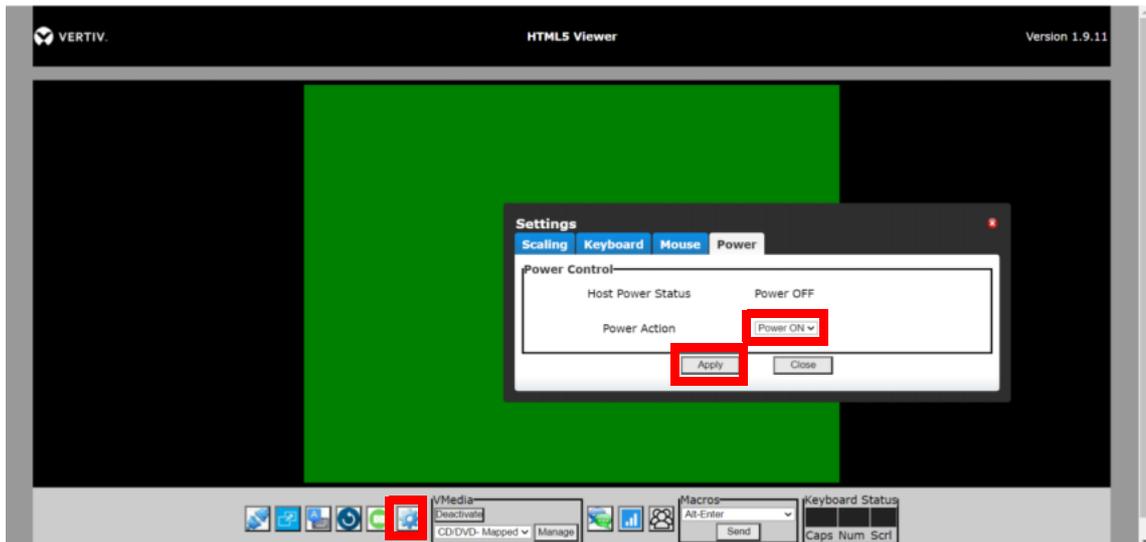


5. Virtual Disk Management 画面を閉じます。

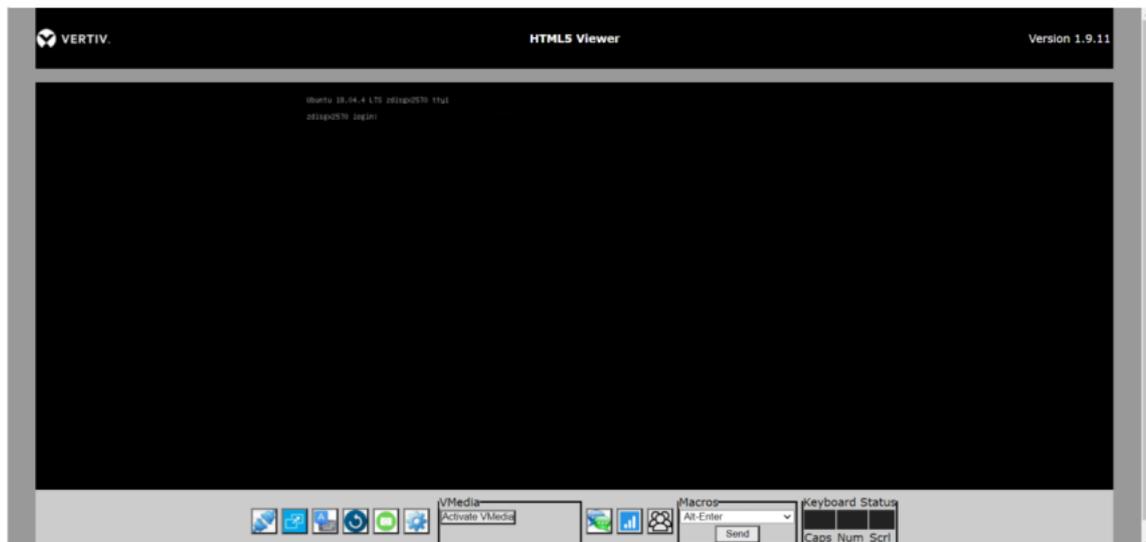
## 6. OS 起動

歯車アイコンをクリックし、[Power] メニューの [Power Action] で [Power ON] を選択して、[Apply] ボタンをクリックします。

内蔵ストレージを交換せずシステムリストアする場合は、[F2] キーを押して BIOS 設定画面を開き、[Save & Exit] メニューの「Boot Override:」で、「USB CD/DVD: MP EMS Virtual Media 0399」を選択してください。



OS の起動が完了すると、以下のログイン画面が表示されます。



## 8.1.2.2 ライブメディアから OS を起動 (RX2540 M5 の場合)

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。

- 「付録 C ライブメディア作成手順」 (P.72) で作成した ISO イメージ
- 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 (サーバーの空スロットに挿入してください)
- サーバーの Web インターフェースにログインできる端末

### 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。

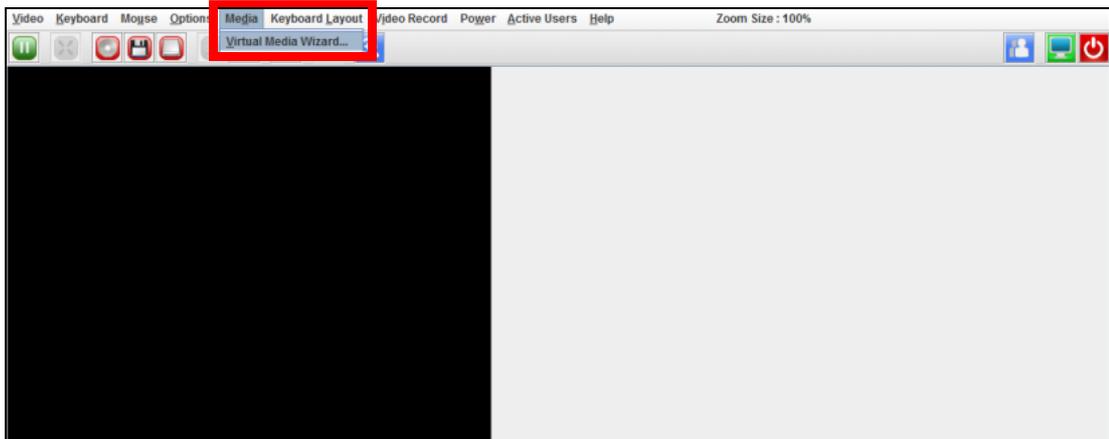


### 2. ビデオリダイレクションを開く

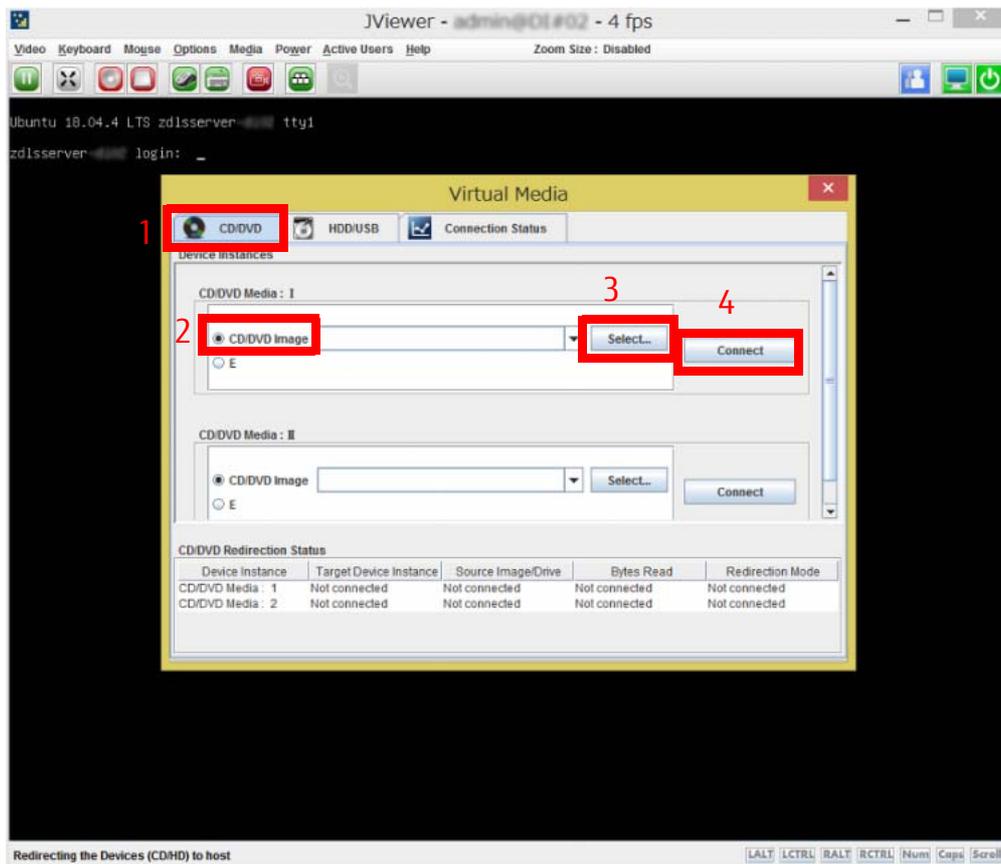
四角のアイコン (AVR の起動) をクリックして、ビデオリダイレクション機能を起動します。



3. Virtual Media Wizard を開く  
[Media] メニューから、Virtual Media Wizard を選択します。

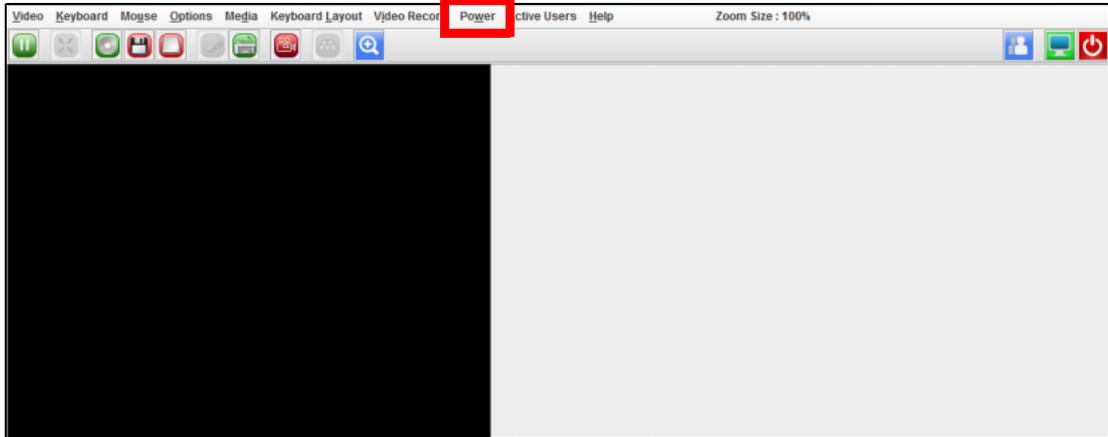


4. ライブメディアをマウント  
[CD/DVD] タブを開き、CD/DVD Media : I で CD/DVD Image を選択します。Select でライブメディアを選択し、[Connect] をクリックします (Read only モードでマウントされます)。



## 5. OS 起動

Virtual Media 画面を閉じ、[Power] メニューまたは  で OS を起動します。  
内蔵ストレージを交換せずシステムリストアする場合は、[F2] キーを押して BIOS 設定画面を開き、[Save & Exit] メニューの「Boot Override:」で、「AMI Virtual CDROM 1.00」を選択してください。



OS の起動が完了すると、以下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 18.04.4 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、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。

```
# mount /dev/sdc1 /media
# df -lh /media
Filesystem      Size  Used Avail Use% Mounted on
/dev/sdc1       1.8T   68M  1.7T   1% /media
# dump -0f /media/md0p2.dmp /dev/md0p2
DUMP: Date of this level 0 dump: Fri Jun 12 15:54:00 2020
DUMP: Dumping /dev/md0p2 (/) 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 50038330 blocks.
DUMP: Volume 1 started with block 1 at: Fri Jun 12 15:54:00 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 49973100 blocks (48801.86MB)
DUMP: Volume 1 took 0:02:46
DUMP: Volume 1 transfer rate: 301042 kB/s
DUMP: 49973100 blocks (48801.86MB) on 1 volume(s)
DUMP: finished in 152 seconds, throughput 328770 kBytes/sec
DUMP: Date of this level 0 dump: Fri Jun 12 15:54:00 2020
DUMP: Date this dump completed: Fri Jun 12 15:56:46 2020
DUMP: Average transfer rate: 301042 kB/s
DUMP: DUMP IS DONE
# ls -lh /media/md0p2.dmp
-rw-r--r-- 1 root root 48G Jun 12 06:56 /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。

```
# shutdown -r now
```

## 8.2 システムリストア

「8.1 システムバックアップ」(P.49) 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.60)



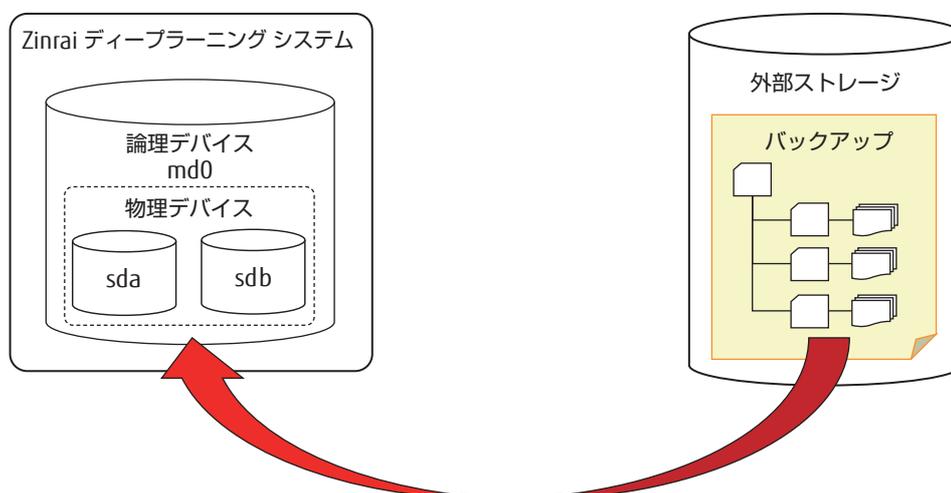
「8.2.2 ファイルシステム作成」(P.61)



「8.2.3 システムリストア実施」(P.64)

### ● 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。



## 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、「[8.1.2.1 ライブメディアから OS を起動 \(GX2570 M5 の場合\)](#)」(P.52)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 18.04.4 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
```

## 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があります)
# (rmコマンドでパーティションを削除します)
# wipefs -a /dev/md0
# (ラベル情報のクリアメッセージが表示されます)
# 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
# (rmコマンドでパーティションを削除します)
# parted /dev/sdb
# (rmコマンドでパーティションを削除します)
# wipefs -a /dev/sda
# wipefs -a /dev/sdb
# wipefs /dev/sda
# wipefs /dev/sdb
```

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、[\[8.1.1 ファイルシステム情報採取\] \(P.50\)](#) 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

/dev/sda に対してだけ、parted\_sda.out ファイルのパーティション情報を基に設定します。なお、/dev/sdb に対しては、システムをリストアしたあと、パーティションを設定します。

```
root@zdlserver-dl03:~# parted /dev/sda
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/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 1.44.1 (24-Mar-2018)
Discarding device blocks: done
Creating filesystem with 217513984 4k blocks and 54378496 inodes
Filesystem UUID: c176de18-f674-4863-b2e9-22b2e2b1d771
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632, 2654208,
    4096000, 7962624, 11239424

Allocating group tables: done
Writing inode tables: done
Creating journal (131072 blocks): done
Writing superblocks and filesystem accounting information: done

# parted /dev/sda
GNU Parted 3.2
Using /dev/sda
Welcome to GNU Parted! Type 'help' to view a list of commands.
(parted) 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

(parted) quit
#
```

## 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=855218540K  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 3.2
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? []? <blank>
File system type? [ext2]? linux-swap
Start? 2048
End? 134219775
(parted) mkpart
Partition name? []? <blank>
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 1.44.1 (24-Mar-2018)
Discarding device blocks: done
Creating filesystem with 16777216 4k blocks and 4202496 inodes
Filesystem UUID: 9d7c5ca3-2c6f-4d5a-8186-e90bb9cf0fb4
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632, 2654208,
    4096000, 7962624, 11239424

Allocating group tables: done
Writing inode tables: done
Creating journal (131072 blocks): done
Writing superblocks and filesystem accounting information: done

# mkswap /dev/md0p1
Setting up swapspace version 1, size = 64 GiB (68720521216 bytes)
no label, UUID=7af42c53-a3ef-4234-aed5-58078153b220
# ls -l /dev/disk/by-uuid | grep -i sda1
lrwxrwxrwx 1 root root 10 Jul 17 21:46 734C-F73E -> ../../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 10 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:2117eca2:2e9019 b7	ARRAY /dev/md0 metadata=1.2 UUID=a82ca9f3:8e3b40fe:2117eca2:2e9019 b7

#### 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). # # &lt;file system&gt; &lt;mount point&gt; &lt;type&gt; &lt;options&gt; &lt;dump&gt; &lt;pass&gt; /dev/disk/by-id/md-uuid-&lt;変更前のmd0p1の uuid&gt; none swap sw 0 0 # / was on /dev/md0 during curtin installation /dev/disk/by-id/md-uuid-&lt;変更前のmd0p2の uuid&gt; / ext4 defaults 0 0 # /boot/efi was on /dev/sda1 during curtin installation /dev/disk/by-uuid/&lt;変更前の/boot/efiのuuid&gt; /boot/efi vfat defaults 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). # # &lt;file system&gt; &lt;mount point&gt; &lt;type&gt; &lt;options&gt; &lt;dump&gt; &lt;pass&gt; /dev/disk/by-id/md-uuid- 7af42c53:a3ef4234:aed55807:8153b220 none swap sw 0 0 # / was on /dev/md0 during curtin installation /dev/disk/by-id/md-uuid- 9d7c5ca3:2c6f4d5a:8186e90b:b9cf0fb4 / ext4 defaults 0 0 # /boot/efi was on /dev/sda1 during curtin installation /dev/disk/by-uuid/734C-F73E /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 3653
-rwxr-xr-x 1 root root      162 Jul 18 07:35 grub.cfg
-rwxr-xr-x 1 root root 1134456 Jul 18 07:35 grubx64.efi
-rwxr-xr-x 1 root root 1269496 Jul 18 07:35 mmx64.efi
-rwxr-xr-x 1 root root 1334816 Jul 18 07:35 shimx64.efi
#
```

## 7. GRUB 更新

```
# update-grub2
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..
Warning: Setting GRUB_TIMEOUT to a non-zero value when GRUB_HIDDEN_TIMEOUT is
set is no longer supported.
/usr/sbin/grub-probe: warning: Couldn't find physical volume `(null)'. Some
modules may be missing from core image..
Found linux image: /boot/vmlinuz-4.4.0-150-generic
Found initrd image: /boot/initrd.img-4.4.0-150-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..
done
#
```

## 8. 初期 RAM 更新

```
# update-initramfs -k all -u
update-initramfs: Generating /boot/initrd.img-4.4.0-150-generic
W: Possible missing firmware /lib/firmware/ast_dp501_fw.bin for module ast
#
```

## 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。

```
# shutdown -h now
```

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。[Enter] キーを押し、ライブメディアをアンマウントして OS をシャットダウンしたあと、リストアしたシステムを起動してください。



システム起動後、root ユーザーに変更して、以降の作業を実施します。

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

## 11. /dev/sdb パーティション設定

parted\_sdb.out を基に、/dev/sdb にパーティションを設定します。設定手順は、[「8.2.2 ファイルシステム作成」\(P.61\)](#)の[手順 1.](#)を参照してください。

## 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.lmin 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 3653
-rwxr-xr-x 1 root root      162 Jul 18 08:14 grub.cfg
-rwxr-xr-x 1 root root 1134456 Jul 18 08:14 grubx64.efi
-rwxr-xr-x 1 root root 1269496 Jul 18 08:14 mmx64.efi
-rwxr-xr-x 1 root root 1334816 Jul 18 08:14 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
#
```

以上でシステムリストアは完了です。



## ■ 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:20.03-
tf2-py3

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

NVIDIA Release 20.03-tf2 (build 11026100)
TensorFlow Version 2.1.0

Container image Copyright (c) 2019, NVIDIA CORPORATION. All rights reserved.
Copyright 2017-2019 The TensorFlow Authors. All rights reserved.

Various files include modifications (c) NVIDIA CORPORATION. All rights reserved.
NVIDIA modifications are covered by the license terms that apply to the underlying
project or file.

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

root@94f6f1792efc:/workspace#
```

## ■ Docker コンテナへの割当てリソース確認

以下は、Docker コンテナで確認します。

```
root@8667ce506491:/workspace# nvidia-smi
Sat Jun 13 16:14:37 2020
+-----+
| NVIDIA-SMI 450.36.06      Driver Version: 450.36.06      CUDA Version: 11.0      |
+-----+
| GPU   Name           Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf    Pwr:Usage/Cap|      Memory-Usage | GPU-Util  Compute M. |
|====+=====+====+=====+====+=====+====+=====+====+=====+====+=====+
|    0  Tesla V100S-PCI...    On      | 00000000:18:00:0 Off  |            0         |
| N/A   40C    P0      27W / 250W | 0MiB / 32510MiB |    0%      Default  |
|                                           MIG M.           |
+-----+
+-----+
| Processes: |
| GPU   GI   CI        PID   Type   Process name                      GPU Memory |
|      ID   ID                                 |              | Usage      |
+-----+
| No running processes found |
+-----+
root@8667ce506491:/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)
/testfg01 (everyone)
/testvol01 (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 /testfg01 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:/testfg01 /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:/testfg01 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

---

ライブメディアを作成する手順を説明します。

以下の作成環境が必要です

- ubuntu 18.04 が動作しているシステム (Zinrai ディープラーニング システムで作業する必要はありません)  
ubuntu のリポジトリサーバーに接続できることを確認してください。
- Ubuntu-18.04.4-desktop-amd64.iso (ISO イメージ)  
デスクトップ環境の ISO イメージを用意してください。デスクトップ環境以外では、作成に失敗します。

## 1. パッケージインストール

ライブメディアの作成に必要なパッケージをインストールします。

```
$ sudo apt install uck syslinux syslinux-utils xorriso
```

## 2. ISO イメージコピー

手順 1. でパッケージをインストールしたシステムの任意のディレクトリに、ISO イメージを配置します。以下は、/tmp ディレクトリに ISO イメージを配置する例です。ubuntu のサイトから PC にダウンロードしたものを配置することもできます。

```
$ cd /tmp
$ curl -O http://releases.ubuntu.com/18.04/ubuntu-18.04.4-desktop-amd64.iso
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
 21 2028M    21  428M    0     0   996k      0  0:34:44  0:07:20  0:27:24 1040k
```

## 3. ISO イメージ展開

一般ユーザーで ISO イメージを展開します。以下は、zdlsadmin ユーザーで実施した例です。

```
$ sudo uck-remaster-unpack-iso /tmp/ubuntu-18.04.4-desktop-amd64.iso
Mounting ISO image...
mount: /home/zdlsadmin/tmp/remaster-iso-mount: WARNING: device write-
protected, mounted read-only.
Unpacking ISO image...
Unmounting ISO image...
$ sudo uck-remaster-unpack-rootfs
Mounting SquashFS image...
Unpacking SquashFS image...
Unmounting SquashFS image...
$
```

#### 4. 環境設定

展開した ISO イメージの動作環境を整備します。

```
$ sudo cp -p /home/zdlsadmin/tmp/remaster-root/etc/network/interfaces /home/
zdlsadmin/tmp/remaster-root/etc/network/interfaces.org
$ sudo cp -p /home/zdlsadmin/tmp/remaster-root/etc/netplan/01-network-manager-
all.yaml /home/zdlsadmin/tmp/remaster-root/etc/netplan/01-network-manager-
all.yaml.org
$ sudo cp -p /home/zdlsadmin/tmp/remaster-root/etc/apt/sources.list /home/zdlsadmin/
tmp/remaster-root/etc/apt/sources.list.org
$ sudo cp -p /etc/network/interfaces /home/zdlsadmin/tmp/remaster-root/etc/network/
$ sudo cp -p /etc/netplan/*.yaml /home/zdlsadmin/tmp/remaster-root/etc/netplan/
$ sudo cp -p /etc/apt/apt.conf /home/zdlsadmin/tmp/remaster-root/etc/apt/
$ sudo cp -p /etc/apt/sources.list /home/zdlsadmin/tmp/remaster-root/etc/apt/
$ sudo apt-key exportall > /tmp/repo.key
Warning: apt-key output should not be parsed (stdout is not a terminal)
zdlsadmin@zdlserver-di02:~$ sudo cp -p /tmp/repo.key /home/zdlsadmin/tmp/remaster-
root/etc/apt/repo.key
```

#### 5. ISO イメージにログイン

手順 4. で展開した ISO イメージに chroot でログインします。

```
$ sudo uck-remaster-chroot-rootfs
Mounting /proc
Mounting /sys
Mounting /dev/pts
Mounting /tmp
Mounting /home/zdlsadmin/tmp/remaster-root-home
Mounting /home/zdlsadmin/tmp/remaster-apt-cache
Mounting /run
Copying fstab/mtab...
Creating DBUS uuid...
Deactivating initctl...
mv: cannot stat '/sbin/initctl': No such file or directory
Deactivating update-grub...
Deactivating grub-probe...
Hacking grub-probe postinst/postrm...
Remembering kernel update state...
# (← chroot でログインした場合、プロンプトが"##"に変更されます)
```

以下のメッセージが表示され、chroot に失敗することがあります。

```
cp: '/etc/mtab' and '/home/guest/tmp/remaster-root/etc/mtab' are the same file
```

/usr/lib/uck/remaster-live-cd.sh ファイルの以下の箇所をコメントアウトしたあと、再度 uck-remaster-chroot-rootfs してください。

行	変更前	変更後
370	cp -f /etc/mtab "\$REMASTER_DIR/etc/mtab"	#cp -f /etc/mtab "\$REMASTER_DIR/etc/mtab"
371	failure "Failed to copy mtab, error=\$?"	# failure "Failed to copy mtab, error=\$?"

#### 6. リポジトリ情報の更新

```
# apt-key add /etc/apt/repo.key
OK
# apt update
```

## 7. 不要なソフトウェアの削除

ライブメディアのサイズを小さくするため、リストア作業に不必要なソフトウェアを削除します。

```
# apt remove --purge -y \  
ubiquity \  
libreoffice* \  
thunderbird* \  
firefox* \  
bluez* \  
libbluetooth* \  
gnome-bluetooth \  
cheese* \  
cups \  
cups-* \  
libcups* \  
libfontembed* \  
python3-cupshelpers \  
enchant \  
libenchant* \  
example-content \  
hplip* \  
modemmanager \  
sane-utils \  
libsane* \  
shotwell*
```

不要な依存パッケージを削除します。

```
# apt autoremove -y
```

## 8. パッケージインストール

リストアに必要なパッケージをインストールします。

```
# apt install -y ncd u nfs-common dump mdadm gparted openssh-server
```

## 9. ISO イメージ作成

```
# exit  
exit  
Restoring kernel update state...  
Reactivating initctl...  
mv: cannot stat '/sbin/initctl.uck_blocked': No such file or directory  
Reactivating update-grub...  
Reactivating grub-probe...  
Reactivating grub-probe postinst/postrm...  
xauth: file /home/zdlsadmin/.Xauthority does not exist  
Removing /home/username directory...  
Removing generated machine uuid...  
Removing generated fstab/mtab...  
Removing crash reports...  
Unmounting /home/zdlsadmin/tmp/remaster-root/var/cache/apt...  
Unmounting /home/zdlsadmin/tmp/remaster-root/tmp...  
Unmounting /home/zdlsadmin/tmp/remaster-root/sys...  
Unmounting /home/zdlsadmin/tmp/remaster-root/run...  
Unmounting /home/zdlsadmin/tmp/remaster-root/root...  
Unmounting /home/zdlsadmin/tmp/remaster-root/proc...  
Unmounting /home/zdlsadmin/tmp/remaster-root/dev/pts...  
Cleaning up temporary directories...  
$ (← ISOイメージからログアウトすると"$"プロンプトに変更されます)  
$ sudo uck-remaster-pack-rootfs -c  
[sudo] password for zdlsadmin:  
Updating files lists...  
Packing SquashFS image...  
dpkg-query: package 'squashfs-tools' is not available  
Use dpkg --info (= dpkg-deb --info) to examine archive files,  
and dpkg --contents (= dpkg-deb --contents) to list their contents.  
/usr/lib/uck/remaster-live-cd.sh: line 514: [: =: unary operator expected  
Parallel mksquashfs: Using 64 processors  
Creating 4.0 filesystem on /home/zdlsadmin/tmp/remaster-iso/casper/  
filesystem.squashfs, block size 131072.  
[=====  
=====\] 98241/98241 100%
```

```

Exportable Squashfs 4.0 filesystem, gzip compressed, data block size 131072
  compressed data, compressed metadata, compressed fragments, compressed
  xattrs
    duplicates are removed
Filesystem size 1398126.12 Kbytes (1365.36 Mbytes)
  48.36% of uncompressed filesystem size (2890802.43 Kbytes)
Inode table size 1093500 bytes (1067.87 Kbytes)
  27.73% of uncompressed inode table size (3943063 bytes)
Directory table size 1124605 bytes (1098.25 Kbytes)
  41.68% of uncompressed directory table size (2698357 bytes)
Xattr table size 43 bytes (0.04 Kbytes)
  53.75% of uncompressed xattr table size (80 bytes)
Number of duplicate files found 12464
Number of inodes 115536
Number of files 89385
Number of fragments 5282
Number of symbolic links 12642
Number of device nodes 7
Number of fifo nodes 0
Number of socket nodes 0
Number of directories 13502
Number of ids (unique uids + gids) 28
Number of uids 11
  root (0)
  zdlsadmin (1000)
  sshd (110)
  landscape (108)
  unknown (118)
  nvidia-persistenced (111)
  systemd-network (100)
  syslog (102)
  _apt (104)
  man (6)
  uuuid (106)
Number of gids 23
  root (0)
  zdlsadmin (1000)
  dip (30)
  shadow (42)
  uuuid (110)
  nogroup (65534)
  audio (29)
  systemd-network (102)
  utmp (43)
  tty (5)
  crontab (105)
  mlocate (109)
  postfix (115)
  messagebus (107)
  staff (50)
  disk (6)
  man (12)
  ntp (117)
  syslog (106)
  landscape (112)
  adm (4)
  systemd-journal (101)
  mail (8)
$ sudo uck-remaster-pack-iso zdls_livemedia.iso -h -g -d "LiveMedia for
recovery ZDLS"
Preparing directory for new files
Updating md5sums...
~/tmp/remaster-iso ~
~

```

```

Packing ISO image...
ISO description set to: LiveMedia for recovery ZDLS
Size of boot image is 4 sectors -> No emulation
 0.67% done, estimate finish Sun Jun 14 06:04:21 2020
 1.34% done, estimate finish Sun Jun 14 06:04:21 2020
 2.01% done, estimate finish Sun Jun 14 06:04:21 2020
    <snip>
 98.52% done, estimate finish Sun Jun 14 06:04:22 2020
 99.19% done, estimate finish Sun Jun 14 06:04:22 2020
 99.86% done, estimate finish Sun Jun 14 06:04:22 2020
Total translation table size: 2048
Total rockridge attributes bytes: 64213
Total directory bytes: 188416
Path table size(bytes): 960
Max brk space used a7000
746063 extents written (1457 MB)
Making your ISO hybrid...
isohybrid: Warning: more than 1024 cylinders: 1458
isohybrid: Not all BIOSes will be able to boot this device
Generating md5sum for newly created ISO...
$ sudo mount /home/zdlsadmin/tmp/remaster-new-files/zdls_livemedia.iso /media
$ sudo grub-mkrescue -c isolinux/boot.cat -b isolinux/isolinux.bin -no-emul-
boot -boot-load-size
 4 -boot-info-table -eltorito-alt-boot -e boot/grub/efi.img -no-emul-boot -
isohybrid-gpt-basdat -o /tmp/zdls_livemedia_efiboot.
iso /media
$ sudo umount /media
$

```

/tmp/ 配下に、ライブメディアの ISO イメージ (zdls\_livemedia.iso) が作成されます。

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

発行日 2020 年 7 月  
Copyright 2020 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

本文中の表記に一部誤りがありましたので、下記に変更箇所と内容を示します。読み替えをお願いいたします。

誤	【Page 47】 7.3.2.2 RAID 構成に内蔵ストレージを組み込む - 7. /boot/efi エントリー更新(不要な文字あり)
正	7. /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。 <pre># ls -l /dev/disk/by-uuid/ total 0 lrwxrwxrwx 1 root root 11 Jun 27 06:33 &lt;交換後の/md0p2 の uuid&gt; -&gt; ../../md0p2 lrwxrwxrwx 1 root root 10 Jun 27 06:33 &lt;交換後の/sda1 の uuid&gt; -&gt; ../../sda1 lrwxrwxrwx 1 root root 11 Jun 27 06:33 &lt;交換後の/md0p1 の uuid&gt; -&gt; ../../md0p1 lrwxrwxrwx 1 root root 10 Jun 27 06:33 5D69-B7E8 -&gt; ../../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). # # &lt;file system&gt; &lt;mount point&gt; &lt;type&gt; &lt;options&gt; &lt;dump&gt; &lt;pass&gt;  /dev/disk/by-uuid/&lt;交換後の/md0p1 の uuid&gt; none swap sw 0 0 # / was on /dev/md0 during curtin installation /dev/disk/by-uuid/&lt;交換後の/md0p2 の uuid&gt; / ext4 defaults 0 0 # /boot/efi was on /dev/sda1 during curtin installation /dev/disk/by-uuid/&lt;交換後の/sda1 の uuid&gt; /boot/efi vfat defaults 0 0</pre>

<b>誤</b>	<p>【Page 64】</p> <p>8.2.3 システムリストア実施 - 3. RAID 情報更新 表中の記載が誤っていました。(行頭の#漏れ)</p>				
<b>正</b>	<p>3. RAID 情報更新</p> <p>/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。</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: center;">変更前</th> <th style="text-align: center;">変更後</th> </tr> </thead> <tbody> <tr> <td>ARRAY /dev/md0 metadata=1.2UUID=00000000:8e3b40fe:2117eca2:2e9019b7MAILADDR root</td> <td># ARRAY /dev/md0 metadata=1.2UUID=00000000:8e3b40fe:2117eca2:2e9019b7MAILADDR root</td> </tr> </tbody> </table>	変更前	変更後	ARRAY /dev/md0 metadata=1.2UUID=00000000:8e3b40fe:2117eca2:2e9019b7MAILADDR root	# ARRAY /dev/md0 metadata=1.2UUID=00000000:8e3b40fe:2117eca2:2e9019b7MAILADDR root
変更前	変更後				
ARRAY /dev/md0 metadata=1.2UUID=00000000:8e3b40fe:2117eca2:2e9019b7MAILADDR root	# ARRAY /dev/md0 metadata=1.2UUID=00000000:8e3b40fe:2117eca2:2e9019b7MAILADDR root				

<b>誤</b>	<p>【Page 65】</p> <p>8.2.3 システムリストア実施 - 4. fstab 情報更新 表中の UUID の表記が誤っていました。</p>				
<b>正</b>	<p>4. fstab 情報更新</p> <p>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。</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: center;">変更前</th> <th style="text-align: center;">変更後</th> </tr> </thead> <tbody> <tr> <td> <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). # # &lt;file system&gt; &lt;mount point&gt; &lt;type&gt; &lt;options&gt; &lt;dump&gt; &lt;pass&gt; /dev/disk/by-uuid/&lt;変更前のmd0p1のuuid&gt; none swap sw 0 0 # / was on /dev/md0 during curtin installation /dev/disk/by-uuid/&lt;変更前のmd0p2のuuid&gt;/ ext4 defaults 0 0 # /boot/efi was on /dev/sda1 during curtin installation /dev/disk/by-uuid/&lt;変更前の/boot/efiのuuid&gt; /boot/efi vfat defaults 0 0</pre> </td> <td> <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). # # &lt;file system&gt; &lt;mount point&gt; &lt;type&gt; &lt;options&gt; &lt;dump&gt; &lt;pass&gt; /dev/disk/by-uuid/&lt;変更後のmd0p1のuuid&gt; none swap sw 0 0 # / was on /dev/md0 during curtin installation /dev/disk/by-uuid/&lt;変更後のmd0p2のuuid&gt; / ext4 defaults 0 0 # /boot/efi was on /dev/sda1 during curtin installation /dev/disk/by-uuid/&lt;変更後の/boot/efiのuuid&gt; /boot/efi vfat defaults 0 0</pre> </td> </tr> </tbody> </table>	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). # # &lt;file system&gt; &lt;mount point&gt; &lt;type&gt; &lt;options&gt; &lt;dump&gt; &lt;pass&gt; /dev/disk/by-uuid/&lt;変更前のmd0p1のuuid&gt; none swap sw 0 0 # / was on /dev/md0 during curtin installation /dev/disk/by-uuid/&lt;変更前のmd0p2のuuid&gt;/ ext4 defaults 0 0 # /boot/efi was on /dev/sda1 during curtin installation /dev/disk/by-uuid/&lt;変更前の/boot/efiのuuid&gt; /boot/efi vfat defaults 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). # # &lt;file system&gt; &lt;mount point&gt; &lt;type&gt; &lt;options&gt; &lt;dump&gt; &lt;pass&gt; /dev/disk/by-uuid/&lt;変更後のmd0p1のuuid&gt; none swap sw 0 0 # / was on /dev/md0 during curtin installation /dev/disk/by-uuid/&lt;変更後のmd0p2のuuid&gt; / ext4 defaults 0 0 # /boot/efi was on /dev/sda1 during curtin installation /dev/disk/by-uuid/&lt;変更後の/boot/efiのuuid&gt; /boot/efi vfat defaults 0 0</pre>
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). # # &lt;file system&gt; &lt;mount point&gt; &lt;type&gt; &lt;options&gt; &lt;dump&gt; &lt;pass&gt; /dev/disk/by-uuid/&lt;変更前のmd0p1のuuid&gt; none swap sw 0 0 # / was on /dev/md0 during curtin installation /dev/disk/by-uuid/&lt;変更前のmd0p2のuuid&gt;/ ext4 defaults 0 0 # /boot/efi was on /dev/sda1 during curtin installation /dev/disk/by-uuid/&lt;変更前の/boot/efiのuuid&gt; /boot/efi vfat defaults 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). # # &lt;file system&gt; &lt;mount point&gt; &lt;type&gt; &lt;options&gt; &lt;dump&gt; &lt;pass&gt; /dev/disk/by-uuid/&lt;変更後のmd0p1のuuid&gt; none swap sw 0 0 # / was on /dev/md0 during curtin installation /dev/disk/by-uuid/&lt;変更後のmd0p2のuuid&gt; / ext4 defaults 0 0 # /boot/efi was on /dev/sda1 during curtin installation /dev/disk/by-uuid/&lt;変更後の/boot/efiのuuid&gt; /boot/efi vfat defaults 0 0</pre>				