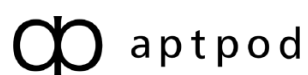


AMI を使った intdash サーバー構築手順 (intdash All-in-One 利用者様向け)

intdash All-in-One Ver. 202209

第 7 版 (2022 年 11 月)



目次

| | |
|---|-----------|
| 01 はじめに | 3 |
| 1.1 対象読者 | 3 |
| 1.2 前提事項 | 3 |
| 1.3 このドキュメントの構成 | 3 |
| 02 intdash インスタンスを作成・起動する | 4 |
| 2.1 準備 | 4 |
| 2.2 intdash インスタンスを作成する | 4 |
| 2.3 intdash インスタンスが作成されたことを確認する | 8 |
| 2.4 intdash インスタンスに SSH で接続できることを確認する | 9 |
| 03 intdash サーバーの設定を行う | 10 |
| 3.1 ドメイン名とサーバー証明書を設定する | 10 |
| 3.2 管理者ユーザーのアカウントを設定する | 18 |
| 3.3 Google Maps の API キーを設定する (Google Maps を使用する場合のみ) | 22 |
| 04 動作を確認する | 24 |
| 4.1 ユーザーアカウントを作成する | 24 |
| 4.2 エッジアカウントを作成する | 24 |
| 4.3 ダッシュボードを準備する | 24 |
| 4.4 iPhone からリアルタイム動画を送信する | 27 |
| 4.5 Data Visualizer で表示を確認する | 28 |
| 05 付録 : intdash を構成するサービスと設定ファイル | 29 |
| 5.1 独自アプリケーション | 29 |
| 5.2 オープンソースアプリケーション | 31 |
| 06 付録 : 旧バージョンの AMI を使用している場合のソフトウェアアップデート手順 | 32 |
| 6.1 アップデートのための準備をする | 32 |
| 6.2 AMI バージョン 202009 のソフトウェアをバージョン 202103 相当にアップデートする | 33 |
| 6.3 AMI バージョン 202103 のソフトウェアをバージョン 202106 相当にアップデートする | 35 |
| 6.4 AMI バージョン 202106 のソフトウェアをバージョン 202112 相当にアップデートする | 37 |
| 6.5 AMI バージョン 202112 のソフトウェアをバージョン 202203 相当にアップデートする | 43 |
| 6.6 AMI バージョン 202203 のソフトウェアをバージョン 202206 相当にアップデートする | 46 |
| 6.7 AMI バージョン 202206 のソフトウェアをバージョン 202209 相当にアップデートする | 49 |

01 はじめに

このドキュメントでは、AWS Marketplace で提供されている Amazon マシンイメージ (以下 AMI) を使用して、intdash サーバー (以下 intdash インスタンス) を構築する手順を説明します。

1.1 対象読者

このドキュメントは、intdash インスタンスの構築および管理を行う方を対象に書かれています。

このドキュメントを使って intdash 環境を構築する方は、ネットワーク管理やサーバー管理のほか、IaaS(AWS)でのインスタンス構築についての基礎知識を持っていることを前提とします。

1.2 前提事項

AMI を使って intdash インスタンスを構築し、設定を行うには以下のソフトウェアが必要です。

- SSH 接続が可能なターミナルソフトウェア
- ウェブブラウザ Google Chrome

1.3 このドキュメントの構成

次の章以降は、以下のような構成になっています。

intdash インスタンスを作成・起動する (p. 4)

AWS Marketplace で AMI を選択し起動します。AWS コンソールでの操作が中心です。

intdash サーバーの設定を行う (p. 10)

起動した intdash インスタンスにおいて認証やアカウント関連の設定を行います。SSH 接続によるターミナル操作と、設定用の専用ウェブアプリケーションでの操作です。

動作を確認する (p. 24)

intdash インスタンスが正常に起動できたことを確認するため、簡単な動作確認を行います。リアルタイムデータを実際に送信して表示します。

付録：intdash を構成するサービスと設定ファイル (p. 29)

intdash アプリケーションに関する技術的な補足事項を記載しています。

付録：旧バージョンの AMI を使用している場合のソフトウェアアップデート手順 (p. 32)

以前のバージョンの AMI を使用している方がインスタンス内のソフトウェアをアップデートする手順を記載しています。

02 intdash インスタンスを作成・起動する

AWS Marketplace で提供されている AMI を使用して、intdash 用の EC2 インスタンスを作成し、起動します。

2.1 準備

以下の情報を事前に準備してください。

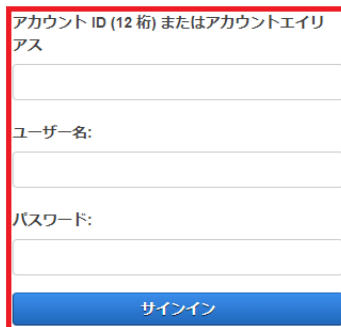
- AWS アカウント
- intdash インスタンス用のネットワーク設定
 - VPC 設定
 - サブネット設定
 - ルーティングテーブル
- Elastic IP（外部に公開する場合のみ必要）

2.2 intdash インスタンスを作成する

AWS コンソールを開き、intdash インスタンスを作成します。

1. AWS にサインインします。

IAM ユーザーとしてサインイン

A screenshot of the AWS IAM sign-in page for the root user. The page is titled "IAM ユーザーとしてサインイン". It contains three input fields: "アカウント ID (12 桁) またはアカウントエイリアス", "ユーザー名:", and "パスワード:". Below these fields is a blue button labeled "サインイン". The entire form is enclosed in a red rectangular border.

ルートユーザーの E メールを使用したサインイン
パスワードをお忘れですか?

図 1 AWS へのサインイン

2. [コンピューティング] > [EC2] を選択します。



図 2 [EC2] を選択

3. ダッシュボードから [インスタンスを起動] をクリックします。



図 3 インスタンスを起動

4. intdash を検索し、AWS Marketplace より [intdash All-in-One] を選択します。

- 検索ワード : intdash
- タブ: AWS Marketplace



図 4 intdash の検索と選択

注釈: AMI のバージョン番号のハイフンの後の数字 (バージョン「202209-2」の場合の「2」) は、AMI に軽微な修正が行われた際にインクリメントされます。複数の AMI バージョンが表示される場合は、最新のものを選択してください。

5. 内容を確認し、[Continue] をクリックします。



図 5 内容を確認し [Continue] をクリック

6. intdash インスタンスのタイプを選択します。インスタンスタイプはさまざまな CPU、メモリ、ストレージ、ネットワークキャパシティの組み合わせです。この例では「m5.large」を選択し、[次のステップ] をクリックします。

ステップ 2: インスタンスタイプの選択

| | | | | |
|-------------------------------------|----|------------|----|-----|
| <input type="checkbox"/> | 汎用 | m5d.metal | 96 | 384 |
| <input checked="" type="checkbox"/> | 汎用 | m5.large | 2 | 8 |
| <input type="checkbox"/> | 汎用 | m5.xlarge | 4 | 16 |
| <input type="checkbox"/> | 汎用 | m5.2xlarge | 8 | 32 |
| <input type="checkbox"/> | 汎用 | m5.4xlarge | 16 | 64 |

図 6 インスタンスタイプの選択

7. intdash インスタンスの詳細を設定します。インスタンスが所属するネットワーク、IP アドレスの割り当て、その他の詳細設定を行います。

この例では「ネットワーク」「サブネット」「自動割り当てパブリック IP」の 3 つを設定します。必要に応じてその他の項目も設定してください。設定が終わったら、[次のステップ] をクリックします。

インスタンス数 ① 1 Auto Scaling グループに作成する ①

購入のオプション ① ☐ スポットインスタンスのリクエスト

ネットワーク ① vpc- intdash-ami

サブネット ① subnet- 1018 個の IP アドレスが利用可能

自動割り当てパブリック IP ① 有効

IPv6 IP の自動割り当て ① サブネット設定を使用 (無効)

配置グループ ① ☐ インスタンスをプレースメントグループに追加します。

キャパシティの予約 ① 開く

IAM ロール ① なし

図 7 インスタンスの詳細の設定

8. intdash インスタンスにアタッチするストレージのサイズを選択します。この例では必要最小限のサイズである 8GB としています。なお、ストレージのサイズはインスタンス作成後に一般的な手法で増やすことも可能です。設定が終わったら、[次のステップ] をクリックします。

ステップ 4: ストレージの追加

インスタンスは次のストレージデバイス設定を使用して作成されます。インスタンスに追加の EBS ボリュームやインスタンスストアボリュームをアタッチするか、ルートボリュームの設定を編集することができます。また、インスタンスを作成してから追加の EBS ボリュームをアタッチすることもできますが、インスタンスストアボリュームはアタッチできません。Amazon EC2 のストレージオプションに関する [詳細](#) はこちらをご覧ください。

| ボリュームタイプ ⓘ | デバイス ⓘ | スナップショット ⓘ | サイズ (GiB) ⓘ | ボリュームタイプ ⓘ | IOPS ⓘ | スループット (MB/秒) ⓘ | 終了時に削除 ⓘ |
|------------|-----------|------------|-------------|--------------|------------|-----------------|-------------------------------------|
| ルート | /dev/xvda | snap- | 8 | 汎用 SSD (gp2) | 100 / 3000 | 該当なし | <input checked="" type="checkbox"/> |

[新しいボリュームの追加](#)

図 8 ストレージの追加

9. intdash インスタンスに適用するタグを追加します。タグの追加は任意です。この例では「Name」タグを追加し、値を「intdash」にしています。設定が終わったら、[次のステップ] をクリックします。

ステップ 5: タグの追加

タグは、大文字と小文字が区別されるキーと値のペアから構成されます。たとえば、キーに「Name」、値に「Webserver」を使用してタグを定義することができます。タグのコピーは、ボリューム、インスタンス、またはその両方に適用できます。タグは、すべてのインスタンスとボリュームに適用されます。Amazon EC2 リソースのタグ付けに関する [詳細はこちら](#)。

| キー (最大 128 文字) | 値 (最大 256 文字) | インスタンス ⓘ |
|----------------|---------------|-------------------------------------|
| Name | intdash | <input checked="" type="checkbox"/> |

図 9 タグの追加

10. intdash インスタンスに適用するセキュリティグループを設定します。セキュリティグループでは以下を許可してください。設定が終わったら、[確認と作成] をクリックします。

- SSH(TCP 22 番ポート)
- HTTP(TCP 80 番ポート)
- HTTPS(TCP 443 番ポート)

セキュリティグループの割り当て: ☒ 新しいセキュリティグループを作成する
☐ 既存のセキュリティグループを選択する

セキュリティグループ名:
説明:

| タイプ ⓘ | プロトコル ⓘ | ポート範囲 ⓘ | ソース ⓘ | 説明 ⓘ |
|-------|---------|---------|----------------|-----------|
| SSH | TCP | 22 | カスタム 0.0.0.0/0 | for SSH |
| HTTP | TCP | 80 | カスタム 0.0.0.0/0 | for HTTP |
| HTTPS | TCP | 443 | カスタム 0.0.0.0/0 | for HTTPS |

図 10 セキュリティグループの設定

注釈: 「SSH(TCP 22 番ポート)」は、SSH を利用してサーバソフトウェアの設定作業を行うために許可します。必要に応じ接続元を制限することを推奨します。

11. 設定の確認画面が表示されます。内容を確認し、問題がなければ [起動] をクリックします。
12. 使用するキーペアを選択します。

キーペアは intdash インスタンスに接続するために必要な秘密鍵と公開鍵のペアです。既存のキーペアを使用するか、新しいキーペアを作成してください。新しいキーペアを作成する場合は、ここで必ず秘密鍵を自分のコンピュータにダウンロードしてください。



図 11 キーペアを新規作成する場合の例

13. [インスタンスの作成] をクリックします。

2.3 intdash インスタンスが作成されたことを確認する

1. AWS コンソールで、左側のメニューバーから [インスタンス]>[インスタンス] を選択し、インスタンスの一覧を表示します。



図 12 インスタンスを表示する

2. intdash インスタンスが表示されていることを確認します。またインスタンスの状態およびステータスチェックが以下の状態となっていることを確認します。
 - インスタンスの状態: 「running」
 - ステータスチェック: 「2/2 のチェックに合格しました」



図 13 作成したインスタンスが表示されていることを確認

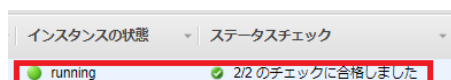


図 14 [インスタンスの状態] と [ステータスチェック] を確認

注釈: Elastic IP を使用する場合は、Elastic IP アドレスの割り当てを行ってください。

2.4 intdash インスタンスに SSH で接続できることを確認する

手元の PC のターミナルから intdash インスタンスに SSH で接続できることを確認します。ターミナルで以下のようにコマンドを実行してください。

```
$ ssh -i PATH-TO-PRIVATE-KEY ec2-user@INSTANCE-IP-ADDRESS
```

PATH-TO-PRIVATE-KEY

秘密鍵のファイルパス

INSTANCE-IP-ADDRESS

intdash インスタンスの IP アドレス

例えば、秘密鍵のファイルパスが `./ssh/intdash-keypair.pem` で、intdash インスタンスの IP アドレスが `203.0.113.123` の場合は、コマンドは `ssh -i ./ssh/intdash-keypair.pem ec2-user@203.0.113.123` のようになります。

正常に接続できた場合は、以下のような画面が表示されます。

```
$ ssh -i ./ssh/intdash-keypair.pem ec2-user@203.0.113.123
Last login: .....

  __/  __/_ )
 _/ (    /  Amazon Linux 2 AMI
---/\---/---/

https://aws.amazon.com/amazon-linux-2/
NNN package(s) needed for security, out of NNN available
Run "sudo yum update" to apply all updates.

-          - /  ---          -
( ) - _ _ / / _ / / _ \  _ _ _ _ / / _
/ / / ' \ / _/ / / / / / \ ' / _/ / ' \
/ / / / / / _ / / / / / ( / / \ _ \ / / /
/_/ / / / _ \ / / _ _ / \ _ _ / _ _ / / _ / /
/

https://www.aptpod.co.jp/products/intdash/

[ec2-user@ip-10-0-0-123 ~]$
```

以上でインスタンスが作成できました。[intdash サーバーの設定を行う](#) (p. 10) に進んでください。

03 intdash サーバーの設定を行う

intdash インスタンスを作成・起動する (p. 4) の手順に従って intdash インスタンスを起動できたら、使用する環境に合わせて intdash の設定を行います。

3.1 ドメイン名とサーバー証明書を設定する

このドキュメントでは、以下のドメイン名と証明書ファイルを例として説明します。

| 項目 | この説明で使用する例 |
|---|--------------------------------------|
| intdash 用のドメイン名 | intdash.example.com |
| 第三者認証局により発行された SSL サーバー証明書 (サーバー証明書と中間証明書が連結されたもの) | /etc/pki/intdash/certs/intdash.pem |
| SSL サーバー秘密鍵 | /etc/pki/intdash/private/intdash.pem |

注釈: intdash 用ドメイン名は、組織で所有・管理しているドメイン名に対しサブドメインとして払い出す方法が一般的です。例えば、example.com を所有、管理している場合は、intdash 用のドメイン名は intdash.example.com のようにします。

詳細については、組織のドメイン名管理者にご相談ください。

注釈:

- サーバー証明書は、SSL/TLS 証明書とも呼ばれ、クライアントとサーバーの間の接続情報を保護し、第三者からの読み取りや変更を防ぐものです。
- セキュリティ上の理由から、自己署名証明書ではなく第三者認証局により発行された証明書を使用してください。自己署名証明書を使用すると、クライアントからの接続時にサーバー証明書の検証に失敗します。サーバー証明書の検証をスキップした場合、通信相手の認証をしないことになり、中間者攻撃等による盗聴・改ざんを許すことになります。
- 証明書を発行する認証局の例としては、Cybertrust、DigiCert、Let's Encrypt 等があります。[Let's Encrypt](#) では、無料で証明書の発行を受けることができます。

3.1.1 ウェブサーバーの設定を行う

ウェブサーバー nginx の設定ファイルで、使用するドメイン名とサーバー証明書を設定します。

1. SSH で intdash インスタンスに接続し、設定ファイルのディレクトリに移動します。

```
# cd /etc/nginx/conf.d
```

2. デフォルトの設定ファイルをバックアップします。

```
# cp -p intdash.conf intdash.conf.org
```

3. テキストエディターで設定ファイルを開きます。

```
# vi intdash.conf
```

4. ポート 80 (HTTP) とポート 443 (HTTPS) の [server_name] にドメイン名を設定します。

```
...

server {
    listen      80;
    listen      [::]:80;
    server_name intdash.example.com;  <--
    access_log  off;

    location / {
        return 301 https://$host$request_uri;
    }
}

server {
    listen      443 ssl http2;
    listen      [::]:443 ssl http2;
    server_name intdash.example.com;  <--
    root        /usr/share/nginx/html;

    ssl_certificate      /etc/pki/intdash/certs/intdash.pem;
    ssl_certificate_key  /etc/pki/intdash/private/intdash.pem;

    ...
}
```

5. 同じ設定ファイル内でポート 443 の [ssl_certificate] にサーバー証明書、[ssl_certificate_key] にサーバー証明書の秘密鍵を指定します。その後設定ファイルを閉じます。

```
...

server {
    listen      80;
    listen      [::]:80;
```

(次のページに続く)

(前のページからの続き)

```
server_name  intdash.example.com;
access_log   off;

location / {
    return 301 https://$host$request_uri;
}
}

server {
    listen      443 ssl http2;
    listen      [::]:443 ssl http2;
    server_name intdash.example.com;
    root        /usr/share/nginx/html;

    ssl_certificate      /etc/pki/intdash/certs/intdash.pem;      <--
    ssl_certificate_key  /etc/pki/intdash/private/intdash.pem;    <--

    ...
}
```

6. 設定を反映させるため、以下のコマンドで nginx を再起動します。

```
# systemctl restart nginx
```

7. 再起動後、nginx が正常に起動していることを確認します。

```
# systemctl status nginx
```

正常に動作していれば Active: active (running) と表示されます。

3.1.2 エッジルーター Traefik の設定を行う

intdash の API サービスや、マイクロサービスへのプロキシを担う Traefik の設定を行います。

1. 前の手順に引き続き SSH で intdash インスタンスに接続し、設定ファイルのディレクトリに移動します。

```
# cd /etc/intdash/traefik.d
```

2. デフォルトの設定ファイルをバックアップします。

```
# cp -p middleware.toml middleware.toml.org
```

3. テキストエディターで設定ファイルを開きます。

```
# vi middleware.toml
```

4. 以下の個所にドメイン名を入力します。その後、設定ファイルを閉じます。

```
[http]
  [http.middlewares]
```

(次のページに続く)

(前のページからの続き)

```
...

[http.middlewares.cors]
  [http.middlewares.cors.headers]

...

  accessControlAllowOriginList = ["https://intdash.example.com", "http://intdash.example.com",
  ↪ "wss://intdash.example.com:443/api/v1/ws/measurements"]  <--

...
```

5. 設定を反映させるため、以下のコマンドで intdash-api-gateway サービスを再起動します。

```
# systemctl restart intdash-api-gateway
```

6. 再起動後、intdash-api-gateway が正常に起動していることを確認します。

```
# systemctl status intdash-api-gateway
```

正常に動作していれば Active: active (running) と表示されます。

3.1.3 認証処理を行う intdash-micro-auth の設定を行う

intdash で認証処理を行うマイクロサービス (intdash-micro-auth) の設定を行います。

1. 前の手順に引き続き SSH で intdash インスタンスに接続し、設定ファイルのディレクトリに移動します。

```
# cd /etc/intdash
```

2. デフォルトの設定ファイルをバックアップします。

```
# cp -p authd.conf authd.conf.org
```

3. テキストエディターで設定ファイルを開きます。

```
# vi authd.conf
```

4. 以下の個所にドメイン名を入力します。その後、設定ファイルを閉じます。

- [email] セクション
 - verification-page-uri

```
[email]
from = "noreply@example.com"
from-display-name = "VM2M"
reply-to = ""
limit-per-user = 1
```

(次のページに続く)

(前のページからの続き)

```
verification-expiration-period = "3h"
verification-page-uri = "https://intdash.example.com/email/activate" <--
...
```

- [oauth2] セクション
 - issuer
 - sign-in-page-uri
 - change-password-page-uri

```
[oauth2]
  issuer = "https://intdash.example.com" <--
  access-token-expiration-period = "1h"
  refresh-token-expiration-days = 30
  sign-in-page-uri = "https://intdash.example.com/signin/" <--
  change-password-page-uri = "https://intdash.example.com/oauth2/authorization/api/password-change" <-
  <--
  ...
```

- [oauth2.default-client] サブセクション
 - password-recovery-redirect-base-url
 - web-redirect-base-urls

```
[oauth2.default-client]
  password-recovery-redirect-base-url = "https://intdash.example.com" <--
  web-redirect-base-urls = [
    "http://localhost:8080",
    "https://localhost:8080",
    "https://intdash.example.com", <--
  ]
  ...
```

5. oauth2 関連のパラメーターをデータベースに反映させるため、以下のコマンドを実行します。

```
# sudo -u intdash authd db migrate -c /etc/intdash/authd.conf
```

6. 設定を反映させるため、以下のコマンドで intdash-micro-auth サービスを再起動します。

```
# systemctl restart intdash-micro-auth
```

7. 再起動後、intdash-micro-auth が正常に起動していることを確認します。

```
# systemctl status intdash-micro-auth
```

正常に動作していれば Active: active (running) と表示されます。

3.1.4 intdash-micro-auth の認証用の JWT 秘密鍵や RSA 秘密鍵を固定化する

intdash-micro-auth サービスの設定ファイル `/etc/intdash/authd.conf` で、認証用の JWT 秘密鍵や RSA 秘密鍵を設定します。

注釈: ここで固定値を設定しない場合、intdash-micro-auth サービスが再起動されるたびに、認証用の JWT 秘密鍵や RSA 秘密鍵が自動生成されます。その場合、サービスを再起動するたびに、以下のようなことが起こります。

- API トークンが無効になる
- ユーザーのログイン状態がリセットされる
- メールアドレス確認用のメールに含まれるリンクが無効になる

そのため、固定の値を設定することを推奨します。

- `[keys.oauth2-rsa.private-key]` サブセクション
 - `key` : PEM 形式の RSA 秘密鍵を設定します。

```
[keys]
[keys.oauth2-rsa]
  [keys.oauth2-rsa.private-key]
    driver = "static"
  [keys.oauth2-rsa.private-key.static]
    key = "-----BEGIN RSA PRIVATE KEY-----\nXXXXXX...XXXXXX\n-----END RSA PRIVATE KEY-----" <--
```

- `[keys.oauth2-hmac]`, `[keys.api-token]`, `[keys.password-recovery]`, `[keys.email-verification]` サブセクション
 - `hash-secret` : ランダムで生成した文字列を設定します。

```
[keys.oauth2-hmac]
  hash-secret = "XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX" <--
  rotated-hash-secrets = [""]
[keys.api-token]
  hash-secret = "XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX" <--
  rotated-hash-secrets = [""]
[keys.password-recovery]
  hash-secret = "XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX" <--
  rotated-hash-secrets = [""]
[keys.email-verification]
  hash-secret = "XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX" <--
  rotated-hash-secrets = [""]
```

この設定後は intdash-micro-auth サービスを再起動してください。また、この設定変更のデータベースへの反映は不要です。

注釈: [keys.oauth2-rsa.private-key] サブセクションでは、driver = "file" を指定することで、PEM 形式の文字列を直接書き込む代わりに RSA 秘密鍵のファイルパスを指定して読み込ませることも可能です。秘密鍵ファイルは intdash ユーザーで動作する intdash-micro-auth サービスから読み込まれます。intdash ユーザーからの読み込みができるように、ファイルのパーミッション、オーナー、グループを設定してください。

```
[keys]
[keys.oauth2-rsa]
  [keys.oauth2-rsa.private-key]
    driver = "file"
  [keys.oauth2-rsa.private-key.file]
    path = "/etc/intdash/authd.keys/privkey.pem" <--
```

3.1.5 intdash-micro-broker の iSCP チケットの秘密鍵を固定化する

intdash-micro-broker サービスの設定ファイル /etc/intdash/brokerd.conf で、iSCP チケットの秘密鍵を固定化します。

注釈: ここで固定値を設定しない場合、intdash-micro-broker サービスが起動するたびに秘密鍵が自動生成されますが、その場合も利用上の問題はありません。

1. 前の手順に引き続き SSH で intdash インスタンスに接続し、設定ファイルのディレクトリに移動します。

```
# cd /etc/intdash
```

2. デフォルトの設定ファイルをバックアップします。

```
# cp -p brokerd.conf brokerd.conf.org
```

3. テキストエディターで設定ファイルを開きます。

```
# vi brokerd.conf
```

4. 以下の個所を設定します。その後、設定ファイルを閉じます。

- [ticket] セクション
- secret : ランダムで生成した文字列を設定します。

```
[ticket]
secret = "XXXXXXXXXXXXXXXXXXXXXXXXXXXX" <--
```

5. 設定を反映させるため、以下のコマンドで intdash-micro-broker サービスを再起動します。

```
# systemctl restart intdash-micro-broker
```

6. 再起動後、intdash-micro-broker が正常に起動していることを確認します。


```
# systemctl status intdash-micro-broker
```

正常に動作していれば Active: active (running) と表示されます。

3.1.6 intdash-web-oauth2 の設定を行う

intdash の OAuth2 認証におけるウェブ UI クライアントの認証を担う UI サービス (intdash-web-oauth2) の設定を行います。

1. 前の手順に引き続き SSH で intdash インスタンスに接続し、設定ファイルのディレクトリに移動します。

```
# cd /etc/sysconfig
```

2. デフォルトの設定ファイルをバックアップします。

```
# cp -p intdash-web-oauth2 intdash-web-oauth2.org
```

3. テキストエディターで設定ファイルを開きます。

```
# vi intdash-web-oauth2
```

4. AUTHORIZATION_HOST と CLIENT_HOST にドメイン名を入力します。その後、設定ファイルを閉じます。

```
HOST="127.0.0.1"
PORT="13003"

API_HTTP_URL="http://127.0.0.1:8080"
AUTHORIZATION_HOST="https://intdash.example.com" <--
CLIENT_HOST="https://intdash.example.com" <--
CLIENT_ID="533bc9ea_authorization_code.aptpod.co.jp"
RETURN_TO_URL="/"
ALLOW_LOGGING="false"
```

5. 設定を反映させるため、以下のコマンドで intdash-web-oauth2 サービスを再起動します。

```
# systemctl restart intdash-web-oauth2
```

6. 再起動後、intdash-web-oauth2 が正常に起動していることを確認します。

```
# systemctl status intdash-web-oauth2
```

正常に動作していれば Active: active (running) と表示されます。

3.2 管理者ユーザーのアカウントを設定する

初期状態では、intdash という名前の管理者ユーザーアカウントが存在します。本セクションではこのアカウントのパスワードとメールアドレスを設定します。あわせて、メールサーバーの設定も行います。

注釈: ユーザー intdash の初期パスワードは、intdash インスタンスのインスタンス ID です。インスタンス ID は EC2 コンソールで確認できます。(例: i-1234567890abcdef0)

3.2.1 管理者ユーザーのパスワードを設定する

管理者ユーザーのパスワードを設定します。

1. ウェブブラウザで Visual M2M Data Visualizer (以下 Data Visualizer) <https://intdash.example.com/> にアクセスします。(ドメイン名はご利用の環境に合わせてください。)
2. ユーザー名 intdash と、初期パスワードを入力し、利用者ガイドラインに同意してから、[ログイン] をクリックします。



図 15 サインイン画面

3. サインインすると、以下のようにパスワード設定画面が表示されますので、任意のパスワードを入力して、[パスワードの変更] をクリックしてください。



図 16 パスワードの設定

注意: 認証情報は、十分に注意して適切に管理してください。

3.2.2 メールサーバーと通知メール送信元を設定する

パスワードを忘れた場合の認証の際などに、intdash サーバーが使用者にメールを送信することがあります。その際に使用する送信元のメールアドレスと SMTP サーバーを設定します

1. SSH で intdash インスタンスに接続し、設定ファイルのディレクトリに移動します。

```
# cd /etc/intdash
```

2. テキストエディターで設定ファイルを開きます。

```
# vi authd.conf
```

4. [email] セクションに SMTP サーバーの情報と送信メールの情報を入力します。SMTP サーバーの情報はお使いの環境に合わせて設定してください。その後、設定ファイルを閉じます。

- [email] セクション
 - from: 送信元として使用されるメールアドレス
 - reply-to: Reply-to ヘッダーに使用されるメールアドレス（設定は必須ではありません）
- [email.smtp] サブセクション
 - ご使用の環境の SMTP サーバーの情報

```
[email]
from = "noreply@example.com"
from-display-name = "VM2M"
reply-to = ""

...
```

(次のページに続く)

(前のページからの続き)

```
[email.smtp]
address = "127.0.0.1:25"
hostname = ""
enable-tls = false
enable-starttls-auto = false
insecure-tls = false
authentication = ""
username = ""
password = ""
```

5. 設定を反映させるため、以下のコマンドで intdash-micro-auth サービスを再起動します。

```
# systemctl restart intdash-micro-auth
```

6. 再起動後、intdash-micro-auth が正常に起動していることを確認します。

```
# systemctl status intdash-micro-auth
```

正常に動作していれば Active: active (running) と表示されます。

3.2.3 管理者ユーザーのメールアドレスを設定する

1. ウェブブラウザで <https://intdash.example.com/users/me/> にアクセスします。
2. サインイン画面が表示されたら、管理者ユーザー名 intdash と、さきほど設定したパスワードでサインインします。

ユーザー intdash のマイページが表示されます。



図 17 マイページ

3. [メールアドレス] をクリックして、管理者ユーザーのメールアドレスを入力し、[変更を保存] をクリックします。

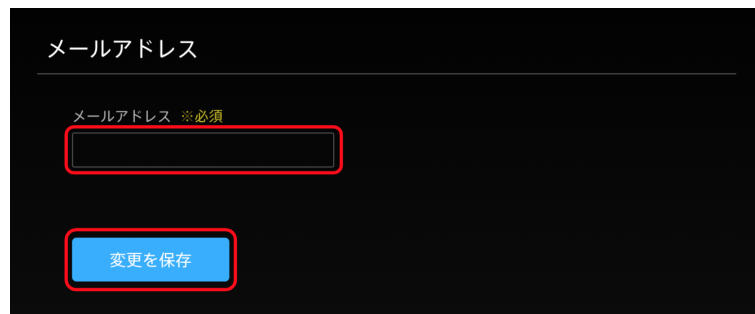


図 18 メールアドレスを設定

4. 以下のメッセージが表示されます。メールアドレスが正しいことを確認し、[閉じる] をクリックします。

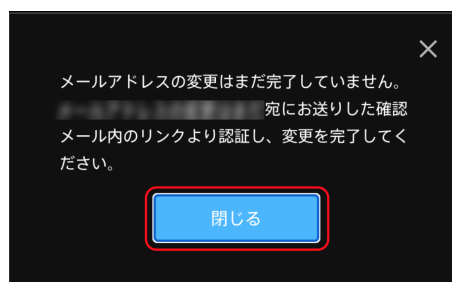
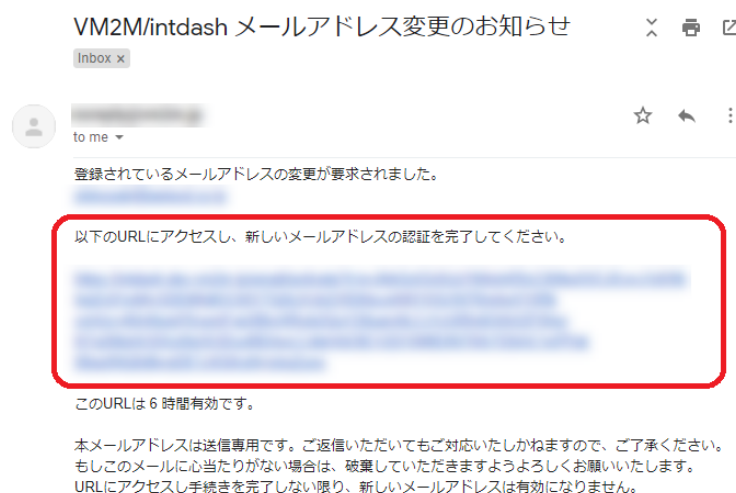


図 19 メールアドレスを確認

メールアドレス宛に確認メールが送信されます。

5. お使いのメールクライアントで確認メールを開きます。

メールに記載されているアクティベーション用 URL にアクセスします。



6. 再度、マイページの [メールアドレス] を開き、以下のように [認証済み] となっていれば設定は完了です。

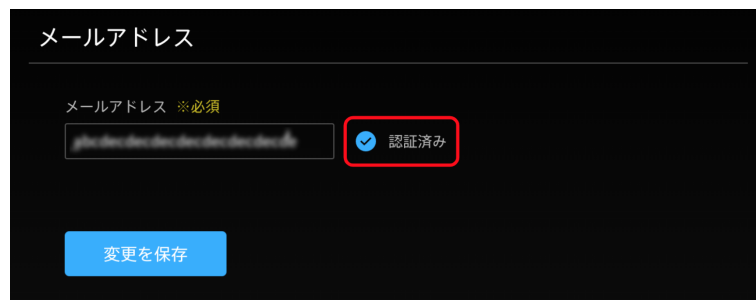


図 20 メールアドレス認証済み

3.3 Google Maps の API キーを設定する（Google Maps を使用する場合のみ）

Data Visualizer で Google Maps ビジュアルパーツを使用するためには、Google Maps の API キーを intdash サーバーに設定しておく必要があります。

注釈:

- Data Visualizer では、Open Street Map を使用することもできます。Open Street Map を使用する場合、ここで API キーを設定する必要はありません。
- Google Maps の詳細に関しては [Google Maps 公式サイト](#) を参照してください。

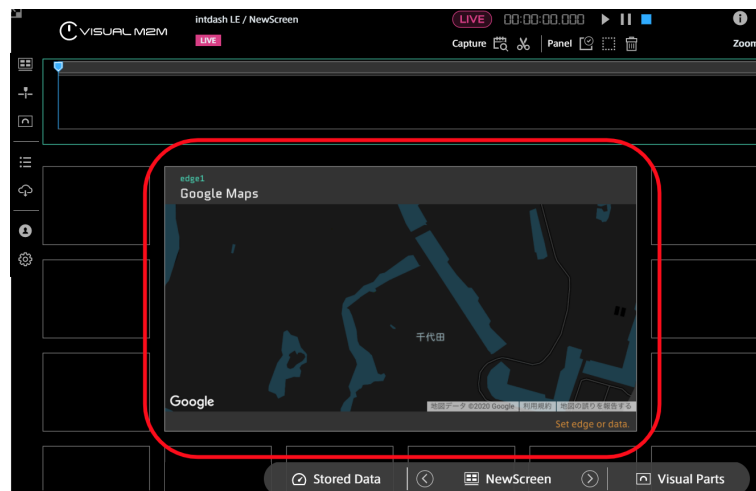


図 21 Google Maps ビジュアルパーツ

Google Maps の API キーの設定は、intdash サーバーの設定用 API に管理者ユーザーとしてアクセスして行います。

1. SSH で intdash インスタンスに接続します。
2. 下記のコマンドで、管理者ユーザー intdash のパスワードをシェル変数に設定します。

```
# read -s PASSWORD  
<-- Enter the password and press <Enter>.
```

3. 下記のコマンドで、intdash にサインインします。

```
# curl -X POST http://127.0.0.1:8080/api/auth/oauth2/token \  
-H "Content-Type: application/x-www-form-urlencoded" \  
-d "username="intdash" \  
-d "password"="${PASSWORD}" \  
-d "grant_type"="password" \  
-d "client_id"="99dcf67c_default.aptopd.co.jp"
```

4. 下記のコマンドで、前の手順でのレスポンスの JSON に含まれる access_token の値をシェル変数に設定します。

```
# read -s ACCESS_TOKEN  
<-- Enter the access token and press <Enter>.
```

5. 下記のコマンドで、Google Maps API キーを設定します。

お持ちの Google Maps API キーを、googleMapsApiKey の値として入力してください。

以下では、-d '{"content": ... の行は表示の都合で折り返して表示されていますが、実際はこの行内に改行は不要です。

```
# curl -X PUT http://127.0.0.1:8080/api/v1/configs/vm2m-2nd-variables \  
-H "Content-Type: application/json" \  
-H "Authorization: Bearer ${ACCESS_TOKEN}" \  
-d '{"content":{"googleMapsApiKey":"XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX","  
↪ "googleMapsApiClientID":"\""}'}
```

注釈: API キーではなく Client ID をお持ちの場合は、下記のように googleMapsApiClientID の値として入力してください。

```
-d '{"content":{"googleMapsApiKey":"\", \"googleMapsApiClientID\":\"  
↪ XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX\"}}'
```

以上で設定は完了です。[動作を確認する](#) (p. 24) に進んでください。

04 動作を確認する

本章では、簡易的な動作確認として以下を行います。

- 動作確認用に新しくユーザー（user1）とエッジ（edge1）を作成する。
- iPhone で intdash Motion アプリケーションを起動し、edge1 として動画を撮影しながら intdash サーバーに送信する。
- PC で Data Visualizer アプリケーションを使って動画を確認する。
- Data Visualizer で過去のデータを再生する。


4.1 ユーザーアカウントを作成する

管理者ユーザーとして Admin Console にログインし、ユーザーアカウント（user1）を作成します。

1. ウェブブラウザで Admin Console <https://intdash.example.com/admin/users> にアクセスします。
2. サインイン画面が表示されたら、管理者ユーザー名 intdash と、さきほど設定したパスワードでサインインします。
3. [ユーザーを作成] をクリックし必要事項を入力して、[作成] をクリックします。
 - 名前: user1
 - ロール: member（一般ユーザー）
4. ユーザーが作成され、一時パスワードが作成されます。一時パスワードをメモしておきます。

4.2 エッジアカウントを作成する

引き続き Admin Console で、エッジアカウントを作成します。

1. [エッジ一覧] で、[エッジを作成] をクリックし、必要事項を入力して、[作成] をクリックします。
 - 名前: edge1UUID とクライアントシークレットが表示されますが、動作確認ではこれらは必要ありません。
2. 右上の  をクリックし、ログアウトします。以降の手順では新しいユーザー user1 を使用します。

4.3 ダッシュボードを準備する

1. Chrome ブラウザーで Visual M2M Data Visualizer <https://intdash.example.com/vm2m/> にアクセスし、ユーザー user1 としてログインします。

初めてログインする場合、パスワード再設定する必要があります。画面の指示に従ってパスワードを再設定してください。
2. [Links]>[Download DAT File] から、動画データのデータ設定ファイル（DAT ファイル）をダウンロードします。

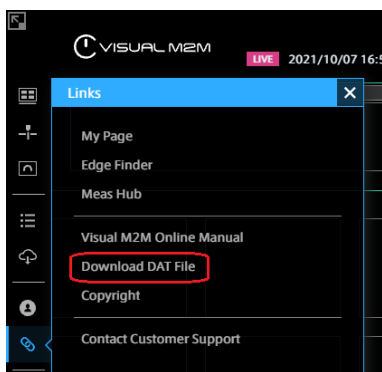


図 22 DAT ファイルダウンロードページを開く

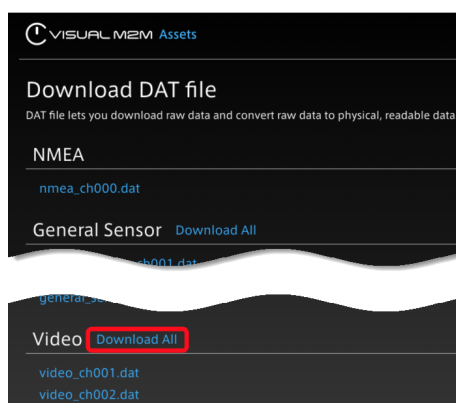


図 23 動画用の DAT ファイルをダウンロード

3. [Data Settings]>[Import] をクリックし、ダウンロードしたデータ設定ファイルをインポートします。

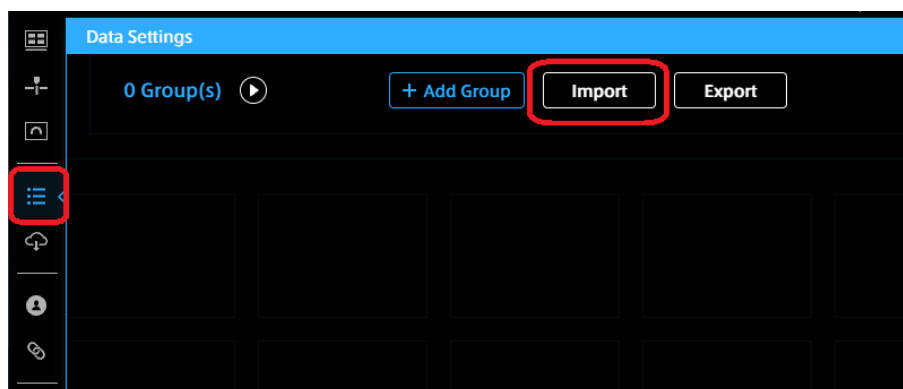


図 24 DAT ファイルのインポート

インポートが成功すると、以下のように「Video」というグループが作成されます。

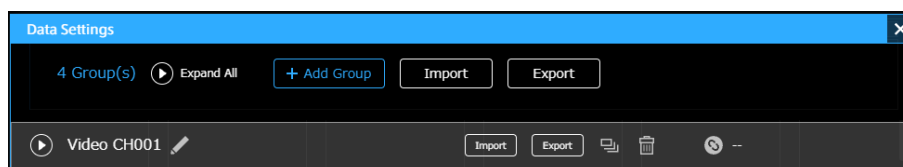


図 25 Video グループ

4. ダッシュボード上にパネルを作成します。



図 26 パネルを作成

5. パネルをクリックし、データ送信元として edge1 を選択し、Video:ch_001 をバインドします。

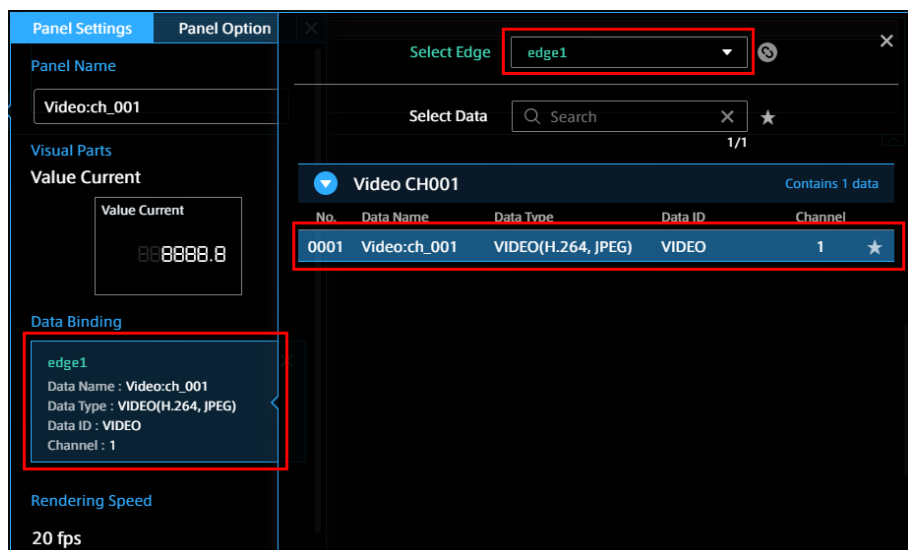


図 27 データ送信元を選択し、動画データをバインド

6. ビジュアルパーツは、Video Player を選択します。

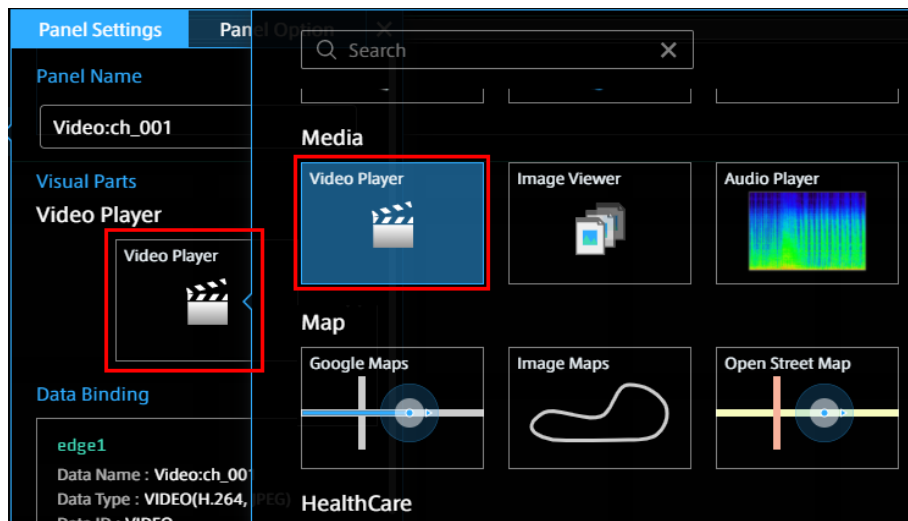


図 28 Video Player を選択

4.4 iPhone からリアルタイム動画を送信する

iPhone で intdash Motion アプリケーションを起動し、動画を送信します。

1. iPhone で intdash Motion を起動し、ログインします。

ログインには以下の情報を使用します。

- URL: intdash インスタンスの URL <https://intdash.example.com>
- ユーザー名: user1
- パスワード: user1 に設定したパスワード

注釈: ログインは、エッジとしてではなくユーザー（user1）として行います。

2. [Settings] を開いて、以下のように設定します。

- [Send Data As]: edge1
- [Video]: オン
 - [Stream to Server]: オン
 - [Save to Server]: オン
 - [Channel]: 1
 - [Codec and Options]: H.264

3. Main 画面に戻り、▶ をタップして、撮影と送信を開始します。

4.5 Data Visualizer で表示を確認する

1. Data Visualizer で LIVE 再生モードにして、[Play] をクリックします。

intdash Motion で撮影している動画がリアルタイムに表示されます。



2. 動作確認ができたなら、intdash Motion で■をタップし、計測を終了します。
3. Data Visualizer で、[Stored Data]>[Measurements] をクリックすると、過去に取得したデータが表示されます。上の手順で撮影したデータが表示されているはずですので、それをクリックして再生します。

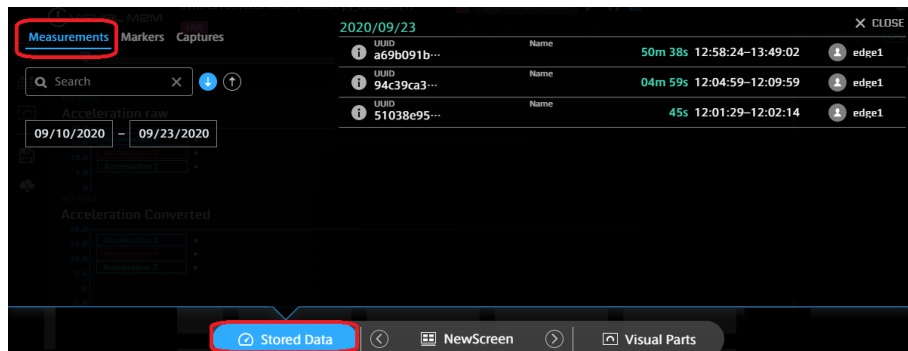


図 29 過去に取得したデータのリストから再生

以上で動作確認は終了です。

05 付録：intdash を構成するサービスと設定ファイル

intdash を構成する各サービスの設定ファイル、サービスが使用するポート、用途は以下の通りです。

5.1 独自アプリケーション

intdash を構成している独自アプリケーションは以下の通りです。

- **intdash-api**
 - 設定ファイル: `/etc/intdash/intdashd.conf`
 - サービスポート: 8097/tcp、8179/tcp
 - 用途: intdash の API サービス。各種マイクロサービスへのプロキシも担う。
- **intdash-micro-auth**
 - 設定ファイル: `/etc/intdash/authd.conf`
 - サービスポート: 8094/tcp
 - 用途: 認証を担うマイクロサービス
- **intdash-micro-broker**
 - 設定ファイル: `/etc/intdash/brokerd.conf`
 - サービスポート: 8180/tcp、8178/tcp
 - 用途: 計測データのリアルタイム処理を担うマイクロサービス
- **intdash-micro-measurement**
 - 設定ファイル: `/etc/intdash/measurementd.conf`
 - サービスポート: 8095/tcp
 - 用途: 計測データの保存や過去計測の読み込みを担うマイクロサービス
- **intdash-micro-config**
 - 設定ファイル: `/etc/intdash/configd.conf`
 - サービスポート: 8096/tcp
 - 用途: スマートフォンアプリの設定を管理するマイクロサービス
- **intdash-micro-media**
 - 設定ファイル: `/etc/intdash/mediad.toml`
 - サービスポート: 8085/tcp
 - 用途: H.264 動画の計測データを取り扱うサービス
- **intdash-web-oauth2-redirector**
 - 設定ファイル: `/etc/sysconfig/intdash-web-oauth2-redirector`
 - サービスポート: UNIX domain socket
 - `/var/run/intdash/intdash-web-oauth2-redirector.sock`
 - 用途: Oauth2 認証クライアントアプリ用のリダイレクト URL を生成する。

- **intdash-web-signin-redirector**
 - 設定ファイル: /etc/sysconfig/intdash-web-signin-redirector
 - サービスポート: UNIX domain socket
 - /var/run/intdash/intdash-web-signin-redirector.sock
 - 用途: 認証系におけるサインイン機能のリダイレクト URL を生成する。
- **intdash-web-me**
 - 設定ファイル: /etc/sysconfig/intdash-web-me
 - サービスポート: 13000/tcp
 - 用途: ログインしているユーザー（自分自身）についての情報を表示するウェブアプリケーション「My Page」
- **intdash-web-edges**
 - 設定ファイル: /etc/sysconfig/intdash-web-edges
 - サービスポート: 13001/tcp
 - 用途: エッジの情報表示のためのウェブアプリケーション「Edge Finder」
- **intdash-web-oauth2**
 - 設定ファイル: /etc/sysconfig/intdash-web-oauth2
 - サービスポート: 13003/tcp
 - 用途: OAuth2 認証においてクライアント側の処理を行うサービス
- **intdash-web-measurements**
 - 設定ファイル: /etc/sysconfig/intdash-web-measurements
 - サービスポート: 13004/tcp
 - 用途: 計測管理のためのウェブアプリケーション「Meas Hub」
- **intdash-web-widget-app-links**
 - 設定ファイル: /etc/sysconfig/intdash-web-widget-app-links
 - サービスポート: 13005/tcp
 - 用途: 他のウェブアプリケーションへのリンクの生成（アプリケーションアイコンによるリンク）を担うサービス
- **intdash-web-admin**
 - 設定ファイル: /etc/sysconfig/intdash-web-admin
 - サービスポート: 13006/tcp
 - 用途: ユーザーとエッジ管理のためのウェブアプリケーション「Admin Console」
- **vm2m-dataviz-backend**
 - 設定ファイル: /etc/sysconfig/vm2m-dataviz-backend
 - サービスポート: UNIX domain socket
 - 用途: 計測の可視化ウェブアプリケーションの設定等を保持するサービス

5.2 オープンソースアプリケーション

intdash では以下の OSS(Open Source Software) を利用しています。

- ウェブサーバー nginx (<https://nginx.org/en/>)
 - 設定ファイル: /etc/nginx 以下 (intdash ウェブサーバーの設定: conf.d/intdash.conf、各アプリケーションの設定: intdash.d 以下)
 - サービスポート: 80/tcp、443/tcp
- エッジルーター Traefik (<https://doc.traefik.io/traefik/>)
 - 設定ファイル: /etc/intdash 以下 (メインの設定: traefik.toml、各アプリケーションの設定: traefik.d 以下)
 - サービスポート: 8080/tcp、18080/tcp
- リレーショナルデータベース PostgreSQL (<https://www.postgresql.org/>)
 - 設定ファイル: /var/lib/pgsql/data 以下
 - サービスポート: UNIX domainsocket:(Peer 認証:5432)
- 時系列データベース InfluxDB (<https://www.influxdata.com/>)
 - 設定ファイル: /etc/influxdb/influxdb.conf
 - サービスポート: 8086/tcp

06 付録：旧バージョンの AMI を使用している場合のソフトウェアアップデート手順

以前のバージョンの AMI を利用している方は、アプトポッドのパッケージリポジトリから新しいバージョンのソフトウェアを取得しインストールすることができます。

アプトポッドのパッケージリポジトリを利用するには、事前に申し込みが必要です。

6.1 アップデートのための準備をする

アップデートを行うためには、リポジトリの認証情報を取得し、リポジトリの情報を intdash インスタンス内のリポジトリ定義ファイルに設定します。

6.1.1 パッケージリポジトリの利用申し込みを行う

アプトポッドのパッケージリポジトリを利用するため申し込みを行い、認証情報を取得します。

1. ウェブブラウザで、アプトポッドの認証情報発行申し込みサイト <https://www.aptpod.co.jp/request-auth-info/> にアクセスします。
2. 必要事項を入力し、送信します。
3. 入力したメールアドレスに受付を完了した旨のメールが送付されます。アプトポッド担当者より認証情報が送付されるまでお待ちください。

6.1.2 認証情報を設定する

パッケージリポジトリにアクセスするためのユーザー名とパスワードを受け取ったら、intdash インスタンス内のリポジトリ定義ファイルに認証情報を設定します。

1. SSH で intdash インスタンスに接続し、リポジトリ定義ファイルのディレクトリに移動します。

```
# cd /etc/yum.repos.d
```

2. テキストエディターでリポジトリ定義ファイルを開きます。

```
# vi intdash.repo
```

3. 以下の個所を編集します。その後、設定ファイルを閉じます。

- [intdash] セクション
 - username
 - password
 - enabled

```
[intdash]
```

(次のページに続く)

(前のページからの続き)

```
...  
username=<user_name>  
password=<password>  
enabled=1
```

4. リポジトリに正常にアクセスできることを確認します。

```
# yum search intdash
```

正常にアクセスできれば、パッケージリストが表示されます。

6.2 AMI バージョン 202009 のソフトウェアをバージョン 202103 相当にアップデートする

AMI バージョン 202009 を使って作成したインスタンスを使用している方は、以下の手順によりソフトウェアをバージョン 202103 相当にアップデートすることができます。

注釈: 以下の手順を実行するとインスタンス内のソフトウェアはアップデートされますが、EC2 コンソールで表示される AMI のバージョン名は変更されません。

1. SSH で intdash インスタンスに接続します。
2. 新しいバージョンのソフトウェアと、新規で追加するソフトウェアをインストールします。

```
# yum install \  
intdash-api \  
intdash-client \  
intdash-media-h264 \  
intdash-micro-auth \  
intdash-micro-broker \  
intdash-micro-config \  
intdash-micro-measurement \  
intdash-nginx \  
intdash-web-edges \  
intdash-web-edges-admin \  
intdash-web-email \  
intdash-web-me \  
intdash-web-measurements \  
intdash-web-oauth2 \  
intdash-web-password \  
intdash-web-shared-assets \  
intdash-web-signin \  
intdash-web-utility-tools \  
intdash-web-widget-app-links \  
vm2m-assets \  
vm2m-dataviz \  
vm2m-dataviz-backend
```

(次のページに続く)

(前のページからの続き)

```
vm2m-dataviz-plugins-core
```

設定ファイルを更新する必要がある場合、このコマンドの出力に警告が表示されます。

3. 設定ファイルを更新します。

設定ファイルを更新する必要がある場合、既存の設定ファイルと同じ階層に、`.rpmnew` という拡張子が付与された新しいデフォルトの設定ファイルが作成されます。既存の設定ファイルの内容と `.rpmnew` の内容を比較し、設定ファイルに新しい設定を反映してください。設定内容については、[intdash サーバーの設定を行う](#) (p. 10) を参考にしてください。

注釈: 例えば、設定ファイル `intdash.conf` と同じ階層に、`intdash.conf.rpmnew` という新しい設定ファイルが作成された場合は以下のようにします。

1. 使用中の設定ファイル `intdash.conf` と新しい設定ファイル `intdash.conf.rpmnew` の差分を確認します。
2. `intdash.conf.rpmnew` で変更されている個所について、内容を確認しながら `intdash.conf` に反映します。
3. `intdash.conf.rpmnew` を削除します。

4. 以下のコマンドを実行してデータベーススキーマを更新します。

```
# sudo -u intdash intdashd db migrate -c /etc/intdash/intdashd.conf
# sudo -u intdash authd db migrate -c /etc/intdash/authd.conf
# sudo -u intdash measurementd db migrate -c /etc/intdash/measurementd.conf
# sudo -u intdash configd db migrate -c /etc/intdash/configd.conf
# sudo -u postgres psql -d template1 -c 'CREATE DATABASE "intdash-micro-broker" OWNER "intdash";'
# sudo -u intdash brokerd db migrate -c /etc/intdash/brokerd.conf
# sudo -u intdash mediad-h264 db migrate -c /etc/intdash/mediad-h264.toml
```

5. 以下のクエリを実行してデータベースの内容を更新します。(バージョン 202103 では一部の機能制限が撤廃されたため、それに合わせるための更新です。)

```
# sudo -u intdash psql -d intdash-micro-auth
intdash-micro-auth=> UPDATE "setting_master" SET "edge_limit" = 50;
intdash-micro-auth=> UPDATE "edge_authority" SET "authority" = '{"allows":["**:*"],"denys":[]}' WHERE
↪edge_id = (SELECT id FROM "edge" WHERE uuid = 'a95d8502-7814-487c-8f8c-f774b55b20d2');

# sudo -u intdash psql -d intdash-api
intdash-api=> DELETE FROM "config" WHERE name IN ('vm2m-2nd-config', 'vm2m-2nd-visual-parts');
```

6. 以下のコマンドを実行して、追加されたサービスが自動起動するように設定します。(新規に追加されたサービスは、デフォルトでは自動起動しません。)

```
# systemctl enable intdash-micro-broker.service
# systemctl enable intdash-web-edges.service
# systemctl enable intdash-web-edges-admin.service
# systemctl enable intdash-web-measurements.service
# systemctl enable vm2m-dataviz-backend.service
```

7. 以下のコマンドを実行してサービスをリロード・再起動します。

```
# systemctl reload nginx.service
# systemctl restart intdash-api.service
# systemctl restart intdash-micro-auth.service
# systemctl restart intdash-micro-measurement.service
# systemctl restart intdash-micro-config.service
# systemctl restart intdash-micro-broker.service
# systemctl restart intdash-media-h264.service
# systemctl restart intdash-web-me.service
# systemctl restart intdash-web-edges.service
# systemctl restart intdash-web-edges-admin.service
# systemctl restart intdash-web-measurements.service
# systemctl restart intdash-web-widget-app-links.service
# systemctl restart intdash-web-oauth2.service
# systemctl restart intdash-web-oauth2-redirector.service
# systemctl restart intdash-web-signin-redirector.service
# systemctl restart vm2m-dataviz-backend.service
```

以上で、AMI バージョン 202009 のソフトウェアをバージョン 202103 相当にアップデートする手順は完了です。

6.3 AMI バージョン 202103 のソフトウェアをバージョン 202106 相当にアップデートする

AMI バージョン 202103 を使って作成したインスタンスを使用している方は、以下の手順によりソフトウェアをバージョン 202106 相当にアップデートすることができます。

注釈: 以下の手順を実行するとインスタンス内のソフトウェアはアップデートされますが、EC2 コンソールで表示される AMI のバージョン名は変更されません。

1. SSH で intdash インスタンスに接続します。
2. 新しいバージョンのソフトウェアをインストールします。

```
# yum install \
  intdash-api \
  intdash-micro-broker \
  intdash-micro-measurement \
  intdash-web-edges \
  intdash-web-edges-admin \
  intdash-web-email \
  intdash-web-me \
  intdash-web-measurements \
  intdash-web-oauth2 \
  intdash-web-password \
  intdash-web-shared-assets \
  intdash-web-signin \
```

(次のページに続く)

(前のページからの続き)

```
intdash-web-widget-app-links \  
vm2m-assets \  
vm2m-dataviz \  
vm2m-dataviz-backend \  
vm2m-dataviz-plugins-core
```

設定ファイルを更新する必要がある場合、このコマンドの出力に警告が表示されます。

3. 設定ファイルを更新します。

設定ファイルを更新する必要がある場合、既存の設定ファイルと同じ階層に、`.rpmnew` という拡張子が付与された新しいデフォルトの設定ファイルが作成されます。既存の設定ファイルの内容と `.rpmnew` の内容を比較し、設定ファイルに新しい設定を反映してください。設定内容については、[intdash サーバーの設定を行う](#) (p. 10) を参考にしてください。

注釈: 例えば、設定ファイル `intdash.conf` と同じ階層に、`intdash.conf.rpmnew` という新しい設定ファイルが作成された場合は以下のようにします。

1. 使用中の設定ファイル `intdash.conf` と新しい設定ファイル `intdash.conf.rpmnew` の差分を確認します。
2. `intdash.conf.rpmnew` で変更されている箇所について、内容を確認しながら `intdash.conf` に反映します。
3. `intdash.conf.rpmnew` を削除します。

4. 以下のコマンドを実行してデータベーススキーマを更新します。

```
# sudo -u intdash intdashd db migrate -c /etc/intdash/intdashd.conf  
# sudo -u intdash measurementd db migrate -c /etc/intdash/measurementd.conf  
# sudo -u intdash brokerd db migrate -c /etc/intdash/brokerd.conf
```

5. 以下のコマンドを実行してサービスをリロード・再起動します。

```
# systemctl restart intdash-api.service  
# systemctl restart intdash-micro-measurement.service  
# systemctl restart intdash-micro-broker.service  
# systemctl restart intdash-web-me.service  
# systemctl restart intdash-web-edges.service  
# systemctl restart intdash-web-edges-admin.service  
# systemctl restart intdash-web-measurements.service  
# systemctl restart intdash-web-widget-app-links.service  
# systemctl restart intdash-web-oauth2.service  
# systemctl restart intdash-web-oauth2-redirector.service  
# systemctl restart intdash-web-signin-redirector.service  
# systemctl restart vm2m-dataviz-backend.service
```

以上で、AMI バージョン 202103 のソフトウェアをバージョン 202106 相当にアップデートする手順は完了です。

6.4 AMI バージョン 202106 のソフトウェアをバージョン 202112 相当にアップデートする

AMI バージョン 202106 を使って作成したインスタンスを使用している方は、以下の手順によりソフトウェアをバージョン 202112 相当にアップデートすることができます。以下の手順は、SSH で intdash インスタンスに接続して行ってください。

注釈: 以下の手順を実行するとインスタンス内のソフトウェアはアップデートされますが、EC2 コンソールで表示される AMI のバージョン名は変更されません。

6.4.1 リポジトリ定義ファイルを更新する

1. テキストエディターでリポジトリ定義ファイルを開きます。

```
# vi /etc/yum.repos.d/intdash.repo
```

2. 以下の個所を編集します。その後、設定ファイルを閉じます。

- [intdash], [intdash-testing] セクション
- baseurl

```
[intdash]
...
baseurl=https://private-repository.aptpod.jp/intdash/202112/linux/rpm
...

[intdash-testing]
...
baseurl=https://private-repository.aptpod.jp/intdash/202112/testing/linux/rpm
...
```

3. パッケージ管理に関するメタデータを削除します。

```
# yum clean metadata
```

6.4.2 ソフトウェアを更新・削除する

1. 新しいバージョンのソフトウェアと、新規で追加するソフトウェアをインストールします。

```
# yum install \
  intdash-api \
  intdash-micro-auth \
  intdash-micro-broker \
  intdash-micro-config \
  intdash-micro-measurement \
  intdash-micro-media \
```

(次のページに続く)

(前のページからの続き)

```
intdash-nginx \
intdash-web-admin \
intdash-web-edges \
intdash-web-email \
intdash-web-me \
intdash-web-measurements \
intdash-web-oauth2 \
intdash-web-password \
intdash-web-shared-assets \
intdash-web-signin \
intdash-web-utility-tools \
intdash-web-widget-app-links \
vm2m-assets \
vm2m-dataviz \
vm2m-dataviz-backend \
vm2m-dataviz-plugins-core
```

設定ファイルを更新する必要がある場合、このコマンドの出力に警告が表示されます。

2. 不要になったソフトウェアの削除を行います。

```
# yum remove intdash-client
```

3. 設定ファイルを更新します。

設定ファイルを更新する必要がある場合、既存の設定ファイルと同じ階層に、`.rpmnew` という拡張子が付与された新しいデフォルトの設定ファイルが作成されます。既存の設定ファイルの内容と `.rpmnew` の内容を比較し、設定ファイルに新しい設定を反映してください。設定内容については、[intdash サーバーの設定を行う](#) (p. 10) を参考にしてください。

注釈: 例えば、設定ファイル `intdash.conf` と同じ階層に、`intdash.conf.rpmnew` という新しい設定ファイルが作成された場合は以下のようになります。

1. 使用中の設定ファイル `intdash.conf` と新しい設定ファイル `intdash.conf.rpmnew` の差分を確認します。
2. `intdash.conf.rpmnew` で変更されている箇所について、内容を確認しながら `intdash.conf` に反映します。
3. `intdash.conf.rpmnew` を削除します。

新しいバージョンでは一部のソフトウェアの名称が変更されました。アップデートを実行することにより旧名称のソフトウェアは削除されます。ただし場合によっては、旧名称のソフトウェアの設定ファイルが、`.rpmsave` という拡張子が追加されて残ることがあります。これら旧名称のソフトウェアの設定ファイルは、新しいソフトウェアの設定時に参考にした後、削除して構いません。

注釈: 名称が変更されたソフトウェアは以下です。

- `intdash-micro-media` (旧: `intdash-media-h264`)
- `intdash-web-admin` (旧: `intdash-web-edges-admin`)

6.4.3 Auth Service のデータベーススキーマを更新する

AMI バージョン 202106 から 202112 では、Auth Service のメジャーバージョンがアップデートされており、データベーススキーマを移行するためにマイグレーションツールの実行が必要です。

1. 以下のコマンドを実行して Auth Service を停止します。

```
# systemctl stop intdash-micro-auth.service
```

2. Auth Service の補助コマンドが配置されているディレクトリに移動します。

```
# cd /usr/libexec/intdash-micro-auth
```

3. authd-v1.9.0 を利用してデータベーススキーマをバージョン 1.9.0 の状態に更新します。

```
# sudo -u intdash ./authd-v1.9.0 db migrate -c /etc/intdash/authd.conf
```

4. 以下のコマンドを実行して、既存のデータベースの名称を変更します。また、既存のデータベースと同名の新しいデータベースを作成します。

```
# sudo -u postgres psql -d template1 -c 'ALTER DATABASE "intdash-micro-auth" RENAME TO "intdash-micro-  
↪auth-v1"'  
# sudo -u postgres psql -d template1 -c 'CREATE DATABASE "intdash-micro-auth" OWNER "intdash"'
```

5. authd-v2.0.0 を利用して新しいデータベースのデータベーススキーマをバージョン 2.0.0 の状態に更新します。

```
# sudo -u intdash ./authd-v2.0.0 db migrate -c /etc/intdash/authd.conf
```

6. テキストエディターでマイグレーションツール設定ファイルを作成します。

```
# vi /tmp/auth-mig.conf
```

7. 以下の内容で設定ファイルを作成します。その後、設定ファイルを閉じます。

```
dry-run = false  
  
[src]  
  driver = "postgres"  
  address = "/var/run/postgresql"  
  dbname = "intdash-micro-auth-v1"  
  username = "intdash"  
  password = ""  
  
[dest]  
  driver = "postgres"  
  address = "/var/run/postgresql"  
  dbname = "intdash-micro-auth"  
  username = "intdash"  
  password = ""
```

8. マイグレーションツールを実行します。

```
# sudo -u intdash ./auth-mig run -c /tmp/auth-mig.conf
```

9. データベーススキーマを最新に更新します。

```
# sudo -u intdash authd db migrate -c /etc/intdash/authd.conf
```

10. 以下のコマンドを実行して Auth Service を再開します。

```
# systemctl start intdash-micro-auth.service
```

6.4.4 ソフトウェア更新に伴う移行作業を行う

1. 以下のコマンドを実行して Auth Service 以外のデータベーススキーマを更新します。

```
# sudo -u intdash intdashd db migrate -c /etc/intdash/intdashd.conf
# sudo -u intdash brokerd db migrate -c /etc/intdash/brokerd.conf
# sudo -u intdash configd db migrate -c /etc/intdash/configd.conf
# sudo -u intdash measurementd db migrate -c /etc/intdash/measurementd.conf
# sudo -u intdash mediad db migrate -c /etc/intdash/mediad.toml
```

2. 以下のコマンドを実行して Media Service の動画ファイルを移行します。

```
# sudo -u intdash mediad migrate -c /etc/intdash/mediad.toml
```

6.4.5 （必要な場合のみ）Data Visualizer 用の独自ビジュアルパーツの設定を行う

注釈: この手順は、Data Visualizer Visual Parts SDK を使って開発された独自ビジュアルパーツを使用している場合にのみ必要です。

AMI バージョン 202112 では、Data Visualizer の設定ファイルの構成が変更されました。以前のバージョンで独自ビジュアルパーツを使用していた場合、ビジュアルパーツ自体は引き続き使用できますが、ビジュアルパーツ設定ファイルを新しく作成する必要があります。

独自ビジュアルパーツを使用するには、`/etc/vm2m-dataviz-backend/visual-parts-plugins/` ディレクトリに、以下のようなビジュアルパーツ設定ファイルを追加してください。

管理しやすいよう、設定ファイル名は `{数字}.{分かりやすい名前}.json` とすることを推奨します。

例: `/etc/vm2m-dataviz-backend/visual-parts-plugins/99.visual-parts-sample.json`

```
{
  "items": [
    {
      "url": "/vm2m/data-visualizer/plugins/visual-parts-sample/app.js",
      "avoidSendingPartSpecificMetadata": true
    }
  ]
}
```

(次のページに続く)

(前のページからの続き)

```
]
}
```

url 独自ビジュアルパーツの URL (以前のバージョンの `vm2m-2nd-visual-parts.production.json` で plugin-URLs に記載していたもの) を入力します。

avoidSendingPartSpecificMetadata

true にすると、このビジュアルパーツ固有の情報 (ビジュアルパーツ名など) が、利用情報の収集から除外されます。通常は true を指定してください。

ビジュアルパーツ設定ファイルの詳細については、「[Visual Parts SDK による Data Visualizer 用ビジュアルパーツの作成](#)」の[デプロイ手順の説明](#)を参照してください。

6.4.6 (必要な場合のみ) Data Visualizer 用の独自リンクの設定を行う

注釈: この手順は、Data Visualizer の Link 画面 () に表示されるリンクをカスタマイズしている場合にのみ必要です。

AMI バージョン 202112 では、Data Visualizer の設定ファイルの構成が変更されました。以前のバージョンで独自のリンク設定をしていた場合は、設定ファイルを新しく作成する必要があります。

独自リンクを使用するには、`/etc/vm2m-dataviz-backend/links/` ディレクトリに、以下のようなリンク設定ファイルを追加してください。

管理しやすいよう、設定ファイル名は `{数字}.{分かりやすい名前}.json` とすることを推奨します。

例: `/etc/vm2m-dataviz-backend/links/900.100.vm2m-dataviz.json`

```
{
  "items": [
    {
      "label": "",
      "href": "",
      "type": "separator"
    },
    {
      "label": "Visual M2M Online Manual",
      "href": "https://docs.intdash.jp/manual/data-visualizer-operation/ed8/ja/data-visualizer-operation-ja.
↪pdf",
      "type": "manual"
    },
    ...
  ]
}
```

label

ラベル

href リンク先の URL またはパス

type 区切り線を表示する場合や、エラー発生時にもリンクを表示させたい場合に、以下を設定します。

- **separator**: 区切り線を表示します。{ "label": "", "href": "", "type": "separator" } とすることで、区切り線を表示することができます。
- **support**: 問い合わせ先のリンクとしても使用します。エラーが発生した時に、このリンクが表示されます。

これら以外の文字列が設定された場合は、表示や機能には影響しません。

設定ファイルはファイル名の順で読み込まれ、Link 画面上のリンクはその順序で表示されます。

6.4.7 （必要な場合のみ）アプリケーションリンクの設定を行う

注釈: この手順は、他アプリケーションへのリンク（）をカスタマイズしている場合にのみ必要です。

AMI バージョン 202112 では、リンクの設定ファイルの構成が変更されました。以前のバージョンで独自のリンク設定をしていた場合は、設定ファイルを新しく作成する必要があります。

独自のリンクを追加するには、`/etc/intdash-web-widget-app-links/` ディレクトリに以下のような設定ファイルを追加してください。

例: `/etc/intdash-web-widget-app-links/100.100.vm2m-dataviz.json`

```
{
  "items": [
    {
      "iconUrl": "/share/image/icon_vm2m.svg",
      "href": "/vm2m/",
      "label_ja-JP": "Visual M2M Data Visualizer",
      "label_en-US": "Visual M2M Data Visualizer"
    }
  ]
}
```

iconUrl

アイコンの URL またはパス

href リンク先の URL またはパス

label_ja-JP

日本語表示のときに表示するラベル

label_en-US

英語表示のときに表示するラベル

設定ファイルはファイル名の順で読み込まれ、リンクはその順序で表示されます。

6.4.8 サービス関連の設定を行う

1. 以下のコマンドを実行して、追加されたサービスが自動起動するように設定します。（新規に追加されたサービスは、デフォルトでは自動起動しません。）

```
# systemctl enable intdash-api-auth.service
# systemctl enable intdash-api-gateway.service
# systemctl enable intdash-micro-media.service
# systemctl enable intdash-web-admin.service
```

2. 以下のコマンドを実行してサービスをリロード・再起動します。

```
# systemctl reload nginx.service
# systemctl restart intdash-api.service
# systemctl restart intdash-api-auth.service
# systemctl restart intdash-api-gateway.service
# systemctl restart intdash-micro-auth.service
# systemctl restart intdash-micro-broker.service
# systemctl restart intdash-micro-config.service
# systemctl restart intdash-micro-measurement.service
# systemctl restart intdash-micro-media.service
# systemctl restart intdash-web-admin.service
# systemctl restart intdash-web-edges.service
# systemctl restart intdash-web-measurements.service
# systemctl restart intdash-web-me.service
# systemctl restart intdash-web-oauth2-redirector.service
# systemctl restart intdash-web-oauth2.service
# systemctl restart intdash-web-signin-redirector.service
# systemctl restart intdash-web-widget-app-links.service
# systemctl restart vm2m-dataviz-backend.service
```

以上で、AMI バージョン 202106 のソフトウェアをバージョン 202112 相当にアップデートする手順は完了です。

6.5 AMI バージョン 202112 のソフトウェアをバージョン 202203 相当にアップデートする

AMI バージョン 202112 を使って作成したインスタンスを使用している方は、以下の手順によりソフトウェアをバージョン 202203 相当にアップデートすることができます。以下の手順は、SSH で intdash インスタンスに接続して行ってください。

注釈: 以下の手順を実行するとインスタンス内のソフトウェアはアップデートされますが、EC2 コンソールで表示される AMI のバージョン名は変更されません。

6.5.1 リポジトリ定義ファイルを更新する

1. テキストエディターでリポジトリ定義ファイルを開きます。

```
# vi /etc/yum.repos.d/intdash.repo
```

2. 以下の個所を編集します。その後、設定ファイルを閉じます。

- [intdash], [intdash-testing] セクション
- baseurl

```
[intdash]
...
baseurl=https://private-repository.aptpod.jp/intdash/202203/linux/rpm
...

[intdash-testing]
...
baseurl=https://private-repository.aptpod.jp/intdash/202203/testing/linux/rpm
...
```

3. パッケージ管理に関するメタデータを削除します。

```
# yum clean metadata
```

6.5.2 ソフトウェアを更新・削除する

1. 新しいバージョンのソフトウェアと、新規で追加するソフトウェアをインストールします。

```
# yum install \
  intdash-api \
  intdash-ffmpeg \
  intdash-micro-auth \
  intdash-micro-broker \
  intdash-micro-config \
  intdash-micro-measurement \
  intdash-micro-media \
  intdash-nginx \
  intdash-web-admin \
  intdash-web-edges \
  intdash-web-email \
  intdash-web-me \
  intdash-web-measurements \
  intdash-web-media \
  intdash-web-oauth2 \
  intdash-web-password \
  intdash-web-shared-assets \
  intdash-web-signin \
  intdash-web-widget-app-links \
```

(次のページに続く)

(前のページからの続き)

```
vm2m-assets \  
vm2m-dataviz \  
vm2m-dataviz-backend \  
vm2m-dataviz-plugins-core
```

設定ファイルを更新する必要がある場合、このコマンドの出力に警告が表示されます。

2. 不要になったソフトウェアの削除を行います。

```
# yum remove intdash-web-utility-tools
```

3. 不要になった以前の FFmpeg をアンインストールします。

AMI バージョン 202112 以前をセットアップした際にインストールした ffmpeg コマンドは、不要になったため削除します。

```
# rm -i /usr/local/bin/ffmpeg
```

4. 設定ファイルを更新します。

設定ファイルを更新する必要がある場合、既存の設定ファイルと同じ階層に、.rpmnew という拡張子が付与された新しいデフォルトの設定ファイルが作成されます。既存の設定ファイルの内容と .rpmnew の内容を比較し、設定ファイルに新しい設定を反映してください。設定内容については、[intdash サーバーの設定を行う](#) (p. 10) を参考にしてください。

注釈: 例えば、設定ファイル intdash.conf と同じ階層に、intdash.conf.rpmnew という新しい設定ファイルが作成された場合は以下のようにします。

1. 使用中の設定ファイル intdash.conf と新しい設定ファイル intdash.conf.rpmnew の差分を確認します。
2. intdash.conf.rpmnew で変更されている箇所について、内容を確認しながら intdash.conf に反映します。
3. intdash.conf.rpmnew を削除します。

6.5.3 ソフトウェア更新に伴う移行作業を行う

1. 以下のコマンドを実行してデータベーススキーマを更新します。

```
# sudo -u intdash intdashd db migrate -c /etc/intdash/intdashd.conf  
# sudo -u intdash authd db migrate -c /etc/intdash/authd.conf  
# sudo -u intdash brokerd db migrate -c /etc/intdash/brokerd.conf  
# sudo -u intdash configd db migrate -c /etc/intdash/configd.conf  
# sudo -u intdash measurementd db migrate -c /etc/intdash/measurementd.conf  
# sudo -u intdash mediad db migrate -c /etc/intdash/mediad.toml
```

2. 以下のコマンドを実行して Media Service の動画ファイルを移行します。

```
# sudo -u intdash mediad ts2fmp4 -c /etc/intdash/mediad.toml
```

6.5.4 サービス関連の設定を行う

1. 以下のコマンドを実行して、追加されたサービスが自動起動するように設定します。(新規に追加されたサービスは、デフォルトでは自動起動しません。)

```
# systemctl enable intdash-web-media.service
```

2. 以下のコマンドを実行してサービスをリロード・再起動します。

```
# systemctl reload nginx.service
# systemctl restart intdash-api.service
# systemctl restart intdash-api-auth.service
# systemctl restart intdash-api-gateway.service
# systemctl restart intdash-micro-auth.service
# systemctl restart intdash-micro-broker.service
# systemctl restart intdash-micro-config.service
# systemctl restart intdash-micro-measurement.service
# systemctl restart intdash-micro-media.service
# systemctl restart intdash-web-admin.service
# systemctl restart intdash-web-edges.service
# systemctl restart intdash-web-me.service
# systemctl restart intdash-web-measurements.service
# systemctl restart intdash-web-media.service
# systemctl restart intdash-web-oauth2-redirector.service
# systemctl restart intdash-web-oauth2.service
# systemctl restart intdash-web-signin-redirector.service
# systemctl restart intdash-web-widget-app-links.service
# systemctl restart vm2m-dataviz-backend.service
```

以上で、AMI バージョン 202112 のソフトウェアをバージョン 202203 相当にアップデートする手順は完了です。

6.6 AMI バージョン 202203 のソフトウェアをバージョン 202206 相当にアップデートする

AMI バージョン 202203 を使って作成したインスタンスを使用している方は、以下の手順によりソフトウェアをバージョン 202206 相当にアップデートすることができます。以下の手順は、SSH で intdash インスタンスに接続して行ってください。

注釈: 以下の手順を実行するとインスタンス内のソフトウェアはアップデートされますが、EC2 コンソールで表示される AMI のバージョン名は変更されません。

6.6.1 リポジトリ定義ファイルを更新する

1. テキストエディターでリポジトリ定義ファイルを開きます。

```
# vi /etc/yum.repos.d/intdash.repo
```

2. 以下の個所を編集します。その後、設定ファイルを閉じます。

- [intdash], [intdash-testing] セクション
- baseurl

```
[intdash]
...
baseurl=https://private-repository.aptpod.jp/intdash/202206/linux/rpm
...

[intdash-testing]
...
baseurl=https://private-repository.aptpod.jp/intdash/202206/testing/linux/rpm
...
```

3. パッケージ管理に関するメタデータを削除します。

```
# yum clean metadata
```

6.6.2 ソフトウェアを更新する

1. 新しいバージョンのソフトウェアをインストールします。

```
# yum install \
  intdash-api \
  intdash-ffmpeg \
  intdash-micro-auth \
  intdash-micro-broker \
  intdash-micro-config \
  intdash-micro-measurement \
  intdash-micro-media \
  intdash-nginx \
  intdash-web-admin \
  intdash-web-edges \
  intdash-web-email \
  intdash-web-me \
  intdash-web-measurements \
  intdash-web-media \
  intdash-web-oauth2 \
  intdash-web-password \
  intdash-web-shared-assets \
  intdash-web-signin \
  intdash-web-widget-app-links \
```

(次のページに続く)

(前のページからの続き)

```
vm2m-assets \
vm2m-dataviz \
vm2m-dataviz-backend \
vm2m-dataviz-plugins-core
```

設定ファイルを更新する必要がある場合、このコマンドの出力に警告が表示されます。

2. 設定ファイルを更新します。

設定ファイルを更新する必要がある場合、既存の設定ファイルと同じ階層に、`.rpmnew` という拡張子が付与された新しいデフォルトの設定ファイルが作成されます。既存の設定ファイルの内容と `.rpmnew` の内容を比較し、設定ファイルに新しい設定を反映してください。設定内容については、[intdash サーバーの設定を行う](#) (p. 10) を参考にしてください。

注釈: 例えば、設定ファイル `intdash.conf` と同じ階層に、`intdash.conf.rpmnew` という新しい設定ファイルが作成された場合は以下のようにします。

1. 使用中の設定ファイル `intdash.conf` と新しい設定ファイル `intdash.conf.rpmnew` の差分を確認します。
2. `intdash.conf.rpmnew` で変更されている箇所について、内容を確認しながら `intdash.conf` に反映します。
3. `intdash.conf.rpmnew` を削除します。

6.6.3 ソフトウェア更新に伴う移行作業を行う

1. 以下のコマンドを実行してデータベーススキーマを更新します。

```
# sudo -u intdash intdashd db migrate -c /etc/intdash/intdashd.conf
# sudo -u intdash authd db migrate -c /etc/intdash/authd.conf
# sudo -u intdash brokerd db migrate -c /etc/intdash/brokerd.conf
# sudo -u intdash configd db migrate -c /etc/intdash/configd.conf
# sudo -u intdash measurementd db migrate -c /etc/intdash/measurementd.conf
# sudo -u intdash mediad db migrate -c /etc/intdash/mediad.toml
```

6.6.4 サービス関連の設定を行う

1. 以下のコマンドを実行してサービスをリロード・再起動します。

```
# systemctl reload nginx.service
# systemctl restart intdash-api.service
# systemctl restart intdash-api-auth.service
# systemctl restart intdash-api-gateway.service
# systemctl restart intdash-micro-auth.service
# systemctl restart intdash-micro-broker.service
# systemctl restart intdash-micro-config.service
# systemctl restart intdash-micro-measurement.service
```

(次のページに続く)

(前のページからの続き)

```
# systemctl restart intdash-micro-media.service
# systemctl restart intdash-web-admin.service
# systemctl restart intdash-web-edges.service
# systemctl restart intdash-web-me.service
# systemctl restart intdash-web-measurements.service
# systemctl restart intdash-web-media.service
# systemctl restart intdash-web-oauth2-redirector.service
# systemctl restart intdash-web-oauth2.service
# systemctl restart intdash-web-signin-redirector.service
# systemctl restart intdash-web-widget-app-links.service
# systemctl restart vm2m-dataviz-backend.service
```

以上で、AMI バージョン 202203 のソフトウェアをバージョン 202206 相当にアップデートする手順は完了です。

6.7 AMI バージョン 202206 のソフトウェアをバージョン 202209 相当にアップデートする

AMI バージョン 202206 を使って作成したインスタンスを使用している方は、以下の手順によりソフトウェアをバージョン 202209 相当にアップデートすることができます。以下の手順は、SSH で intdash インスタンスに接続して行ってください。

注釈: 以下の手順を実行するとインスタンス内のソフトウェアはアップデートされますが、EC2 コンソールで表示される AMI のバージョン名は変更されません。

6.7.1 リポジトリ定義ファイルを更新する

1. テキストエディターでリポジトリ定義ファイルを開きます。

```
# vi /etc/yum.repos.d/intdash.repo
```

2. 以下の個所を編集します。その後、設定ファイルを閉じます。

- [intdash], [intdash-testing] セクション
- baseurl

```
[intdash]
...
baseurl=https://private-repository.aptpod.jp/intdash/202209/linux/rpm
...

[intdash-testing]
...
baseurl=https://private-repository.aptpod.jp/intdash/202209/testing/linux/rpm
...
```

3. パッケージ管理に関するメタデータを削除します。

```
# yum clean metadata
```

6.7.2 ソフトウェアを更新・削除する

1. 新しいバージョンのソフトウェアと、新規で追加するソフトウェアをインストールします。

```
# yum install \  
intdash-api \  
intdash-ffmpeg \  
intdash-micro-auth \  
intdash-micro-broker \  
intdash-micro-config \  
intdash-micro-measurement \  
intdash-micro-media \  
intdash-nginx \  
intdash-web-admin \  
intdash-web-edges \  
intdash-web-email \  
intdash-web-me \  
intdash-web-measurements \  
intdash-web-media \  
intdash-web-oauth2 \  
intdash-web-password \  
intdash-web-project \  
intdash-web-shared-assets \  
intdash-web-signin \  
vm2m-assets \  
vm2m-dataviz \  
vm2m-dataviz-backend \  
vm2m-dataviz-plugins-core
```

設定ファイルを更新する必要がある場合、このコマンドの出力に警告が表示されます。

2. 不要になったソフトウェアの削除を行います。

```
# yum remove intdash-web-widget-app-links
```

3. 設定ファイルを更新します。

設定ファイルを更新する必要がある場合、既存の設定ファイルと同じ階層に、`.rpmnew` という拡張子が付与された新しいデフォルトの設定ファイルが作成されます。既存の設定ファイルの内容と `.rpmnew` の内容を比較し、設定ファイルに新しい設定を反映してください。設定内容については、[intdash サーバーの設定を行う](#) (p. 10) を参考にしてください。

注釈: 例えば、設定ファイル `intdash.conf` と同じ階層に、`intdash.conf.rpmnew` という新しい設定ファイルが作成された場合は以下のようにします。

1. 使用中の設定ファイル `intdash.conf` と新しい設定ファイル `intdash.conf.rpmnew` の差分を確認します。
2. `intdash.conf.rpmnew` で変更されている個所について、内容を確認しながら `intdash.conf` に反映します。
3. `intdash.conf.rpmnew` を削除します。

6.7.3 ソフトウェア更新に伴う移行作業を行う

1. 以下のコマンドを実行してデータベーススキーマを更新します。

```
# sudo -u intdash intdashd db migrate -c /etc/intdash/intdashd.conf
# sudo -u intdash authd db migrate -c /etc/intdash/authd.conf
# sudo -u intdash brokerd db migrate -c /etc/intdash/brokerd.conf
# sudo -u intdash configd db migrate -c /etc/intdash/configd.conf
# sudo -u intdash measurementd db migrate -c /etc/intdash/measurementd.conf
# sudo -u intdash mediad db migrate -c /etc/intdash/mediad.toml
```

6.7.4 サービス関連の設定を行う

1. 以下のコマンドを実行して、追加されたサービスが自動起動するように設定します。(新規に追加されたサービスは、デフォルトでは自動起動しません。)

```
# systemctl enable intdash-web-project.service
```

2. 以下のコマンドを実行してサービスをリロード・再起動します。

```
# systemctl reload nginx.service
# systemctl restart intdash-api.service
# systemctl restart intdash-api-auth.service
# systemctl restart intdash-api-gateway.service
# systemctl restart intdash-micro-auth.service
# systemctl restart intdash-micro-broker.service
# systemctl restart intdash-micro-config.service
# systemctl restart intdash-micro-measurement.service
# systemctl restart intdash-micro-media.service
# systemctl restart intdash-web-admin.service
# systemctl restart intdash-web-edges.service
# systemctl restart intdash-web-me.service
# systemctl restart intdash-web-measurements.service
# systemctl restart intdash-web-media.service
# systemctl restart intdash-web-oauth2-redirector.service
# systemctl restart intdash-web-oauth2.service
# systemctl restart intdash-web-project.service
```

(次のページに続く)

(前のページからの続き)

```
# systemctl restart intdash-web-signin-redirector.service  
# systemctl restart vm2m-dataviz-backend.service
```

以上で、AMI バージョン 202206 のソフトウェアをバージョン 202209 相当にアップデートする手順は完了です。