AMI を使った intdash サーバー構築手順（AWS Marketplace での購入者様向け）

intdash All-in-One Ver. 202103

第 2 版（2021 年 3 月）

aptpod

50fda76

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 11

3.1 ドメイン名とサーバー証明書を設定する ... 11
3.2 エッジアカウントを設定する ... 18
3.3 Google Maps の API キーを設定する ... 24
3.4 H.264 動画を扱うための設定を行う ... 26

04 動作を確認する 29

4.1 ダッシュボードを準備する ... 29
4.2 iPhone からリアルタイム動画を送信する ... 32
4.3 Data Visualizer で表示を確認する ... 33

05 付録：intdash を構成するサービスと設定ファイル 34

5.1 独自アプリケーション ... 34
5.2 オープンソースアプリケーション ... 36

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 37

6.1 アップデートのための準備をする ... 37
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 ... 38

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
- ウェブブラウザー 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. 11)
起動した intdash インスタンスにおいて認証やエッジ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. 29)
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. 34)
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. 37)
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。

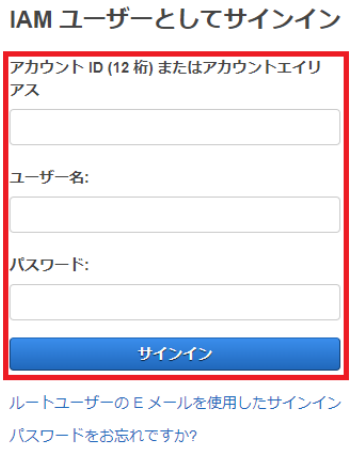


図 1　AWS へのサインイン

2. [コンピューティング] > [EC2] を選択します。

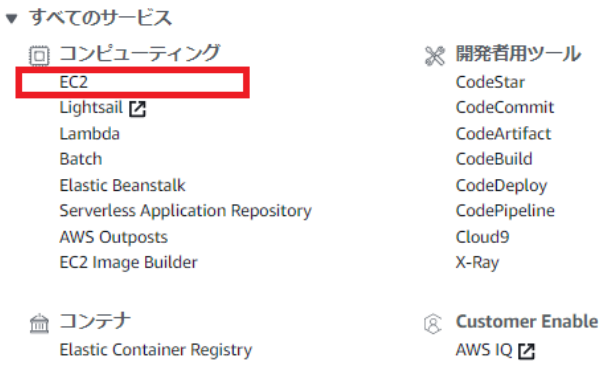


図 2　[EC2] を選択

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

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。

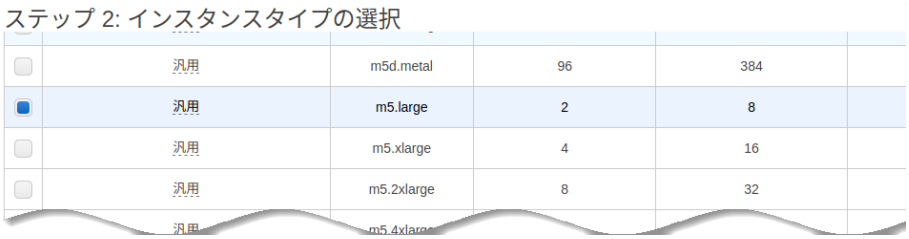


図 6　インスタンスタイプの選択

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。

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。

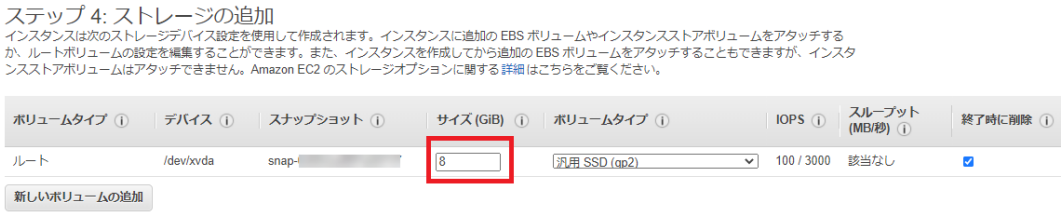


図 8　ストレージの追加

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。

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)

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ーにダウンロードしてください。

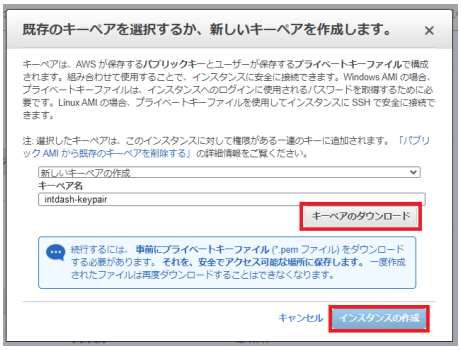


図 11　キーペアを新規作成する場合の例

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」

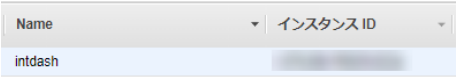


図 13　作成したインスタンスが表示されていることを確認

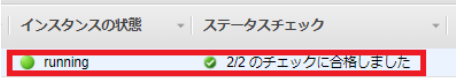


図 14　[インスタンスの状態] と [ステータスチェック] を確認

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.

    _          _      |  ____                _
   (_) _ __   | |_    |  |  _ \   __ _  ___ | |__
   | || '_ \ | __|   |  | | | | / _` |/ __|| '_ \
```

(次のページに続く)

(前のページからの続き)

```
   | || | | || |_   |  | |_| || (_| |\__ \| | | |
   |_||_| |_| \__|  |  |____/  \__,_||___/|_| |_|
                    |

https://www.aptpod.co.jp/products/middleware/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. 11) 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;
    server_name  intdash.example.com;
    access_log   off;

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

server {
```

(次のページに続く)

(前のページからの続き)

```
    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 API サービスを担う intdash-api 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 intdash-api 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 intdashd.conf intdashd.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 intdashd.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。
   - [api] セクション
     - allow-origins

```
[api]
  bind-address = "127.0.0.1:8080"
  jwk-endpoints-url = "http://127.0.0.1:8094/api/v1/oauth2/jwks"
  allow-origins = [
    "https://intdash.example.com",                          <--
    "http://intdash.example.com",                           <--
    "wss://intdash.example.com:443/api/v1/ws/measurements",   <--
  ]
```

(次のページに続く)

(前のページからの続き)

```
...
```

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

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

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

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

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
     - redirect-url-host
     - forgot-password-allowed-redirect-urls
     - activate-edges-allowed-redirect-urls

```
[email]
  from = "noreply@example.com"
  reply-to = ""
  subject-prefix = ""
  redirect-url-host = "https://intdash.example.com"   <--
  forgot-password-redirect-url-path = "/password/new"
  forgot-password-allowed-redirect-urls = [
    "http://localhost:8080",
    "https://localhost:8080/password/new",
    "https://intdash.example.com/password/new",   <--
  ]
  activate-edges-redirect-url-path = "/password/new"
  activate-edges-allowed-redirect-urls = [
```

(次のページに続く)

(前のページからの続き)

```
    "http://localhost:8080",
    "https://localhost:8080/oauth2/authorization/api/password-new",
    "https://intdash.example.com/oauth2/authorization/api/password-new",   <--
  ]
  activate-email-redirect-url-path = "/email/activate"
  jwt-signing-method = "HS256"
  jwt-private-key = ""
  ...
```

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

```
[oauth2]
  issuer = "https://intdash.example.com"   <--
  access-token-lifespan-sec = 3600
  refresh-token-lifespan-sec = 2592000
  rsa-signing-key-name = "key-1"
  hmac-key-name = "key-1"
  sign-in-page-uri = "https://intdash.example.com/signin/"   <--
  ...
```

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

```
[oauth2.default-client]
  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 psql -d intdash-micro-auth -c 'DELETE FROM oauth2_client;'
# 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 JWT(JSON Web Token) の秘密鍵を固定化する

注釈: この設定は必須ではありませんが、設定することを推奨します。

初期設定では、サービスを起動するたびに認証用の JWT 秘密鍵や RSA 秘密鍵が自動生成されます。サービスが再起動すると、既存のサインイン状態が解除されたりメール内のリンクが無効になったりするので、これらの秘密鍵は固定して管理することを推奨します。

秘密鍵を固定する場合は、intdash-micro-auth サービスの設定ファイル `/etc/intdash/authd.conf` で秘密鍵を設定します。

- [api] セクション
  - `jwt-private-key`：JWT 秘密鍵としてランダムで生成した文字列を設定します。

```
[api]
  ...
  jwt-private-key = "XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"   <--
  ...
```

- [email] セクション
  - `jwt-private-key`：JWT 秘密鍵としてランダムで生成した文字列を設定します。メールアドレスのアクティベーション時に使用されます。

```
[email]
  ...
  jwt-private-key = "XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"   <--
  ...
```

- [keys] セクション
  - `key`：PEM 形式の RSA 秘密鍵を設定します。
  - `hash-key`：HMAC 署名用の鍵としてランダムで生成した文字列を設定します。

```
[keys]
  [keys.rsas]
    [keys.rsas.key-1]
      driver = "static"
      [keys.rsas.key-1.static]
        key = "-----BEGIN RSA PRIVATE KEY-----\nXXXXXX...XXXXXX\n-----END RSA PRIVATE KEY-----"   <--
  [keys.hmacs]
    [keys.hmacs.key-1]
      hash-key = "XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"   <--
```

注釈: key に PEM 形式の文字列を直接書き込む代わりに、driver に file を指定することで 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。

```
[keys]
  [keys.rsas]
    [keys.rsas.key-1]
      driver = "file"
      [keys.rsas.key-1.file]
        path = "/etc/intdash/authd.keys/privkey.pem"   <--
```

3.1.5 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 エッジアカウントを設定する

初期状態では以下の 3 つのエッジアカウントが存在します。これらのエッジアカウントについて、必要な設定を行います。

エッジ名	用途
intdash	管理ユーザー
edge1	計測デバイス
edge2	計測デバイス

注釈: intdash では、intdash に接続されるデバイスとユーザーはいずれも「エッジ」と呼ばれます。エッジが intdash に接続するためにはエッジごとにアカウントが必要です。これを「エッジアカウント」と呼びます。エッジアカウントには、エッジの名前、メールアドレス、タイプなどを設定します。

3.2.1 管理者エッジのパスワードを設定する

管理者エッジのパスワードを設定します。

1. ウェブブラウザーで 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. ❶ にエッジ名 `intdash` と、❷ に初期パスワードを入力して、[ログイン] をクリックします。

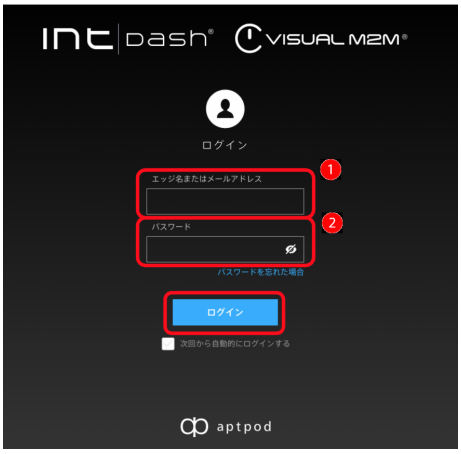


図 15　サインイン画面

エッジ `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. 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 管理者以外のエッジのパスワードを設定する

管理者以外のエッジには、初期パスワードが設定されていません。管理者以外のエッジの初期パスワードは、管理者が Edge Admin Console 上で発行します。

1. ウェブブラウザーで `https://intdash.example.com/edges/admin/`（ドメイン名はご利用の環境に合わせてください）にアクセスします。
2. サインイン画面が表示されたら、管理者エッジ名 `intdash` と、さきほど設定したパスワードでサインインします。

   エッジ一覧が表示されます。

図 17　エッジ一覧

3. 対象のエッジを検索し、[詳細] をクリックします。

図 18　エッジの詳細

4. パスワードの [編集] をクリックします。

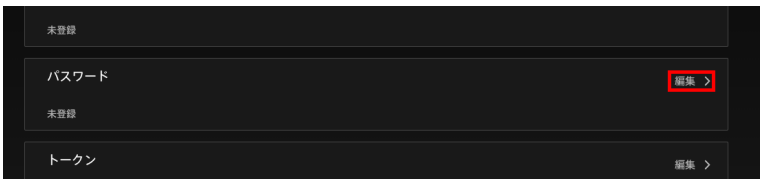


図 19　パスワードの編集

5. [発行] をクリックします。

図 20　初期パスワードの発行

6. 以下のメッセージが表示されます。必要に応じて初期パスワードをクリップボードにコピーしてください。

図 21　初期パスワードの発行完了

7. エッジアカウントの使用者に、Data Visualizer の URL、エッジ名、初期パスワードを安全な方法で伝えます。エッジアカウント使用者は、ウェブブラウザーで Data Visualizer にアクセスし、画面の指示に従ってパスワードを設定します。

3.2.3 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）
     - subject-prefix: メールの件名に付ける接頭辞（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"
  reply-to = ""
  subject-prefix = ""

  ...

  [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。

3.2.4 管理者エッジのメールアドレスを設定する

1. ウェブブラウザーで `https://intdash.example.com/edges/me/` にアクセスします。
2. サインイン画面が表示されたら、管理者エッジ名 `intdash` と、さきほど設定したパスワードでサインインします。

   エッジ `intdash` のマイページが表示されます。

図 22　マイページ

3. [メールアドレス] をクリックして、管理者エッジ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。

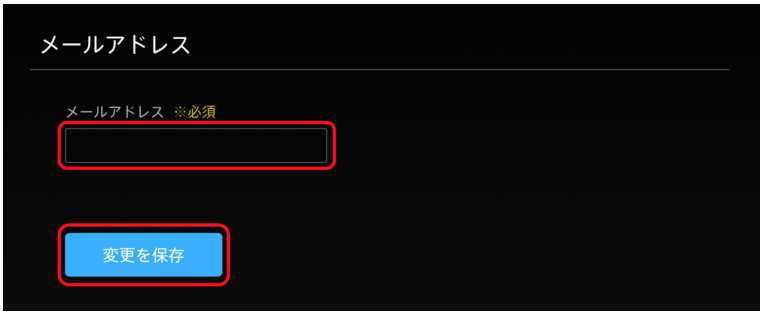


図 23　メールアドレスを設定

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。

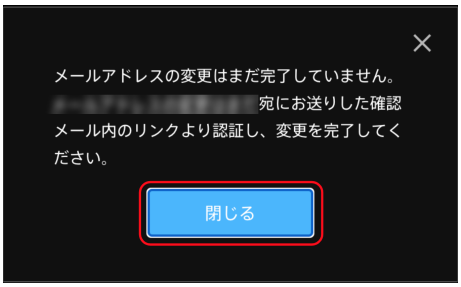
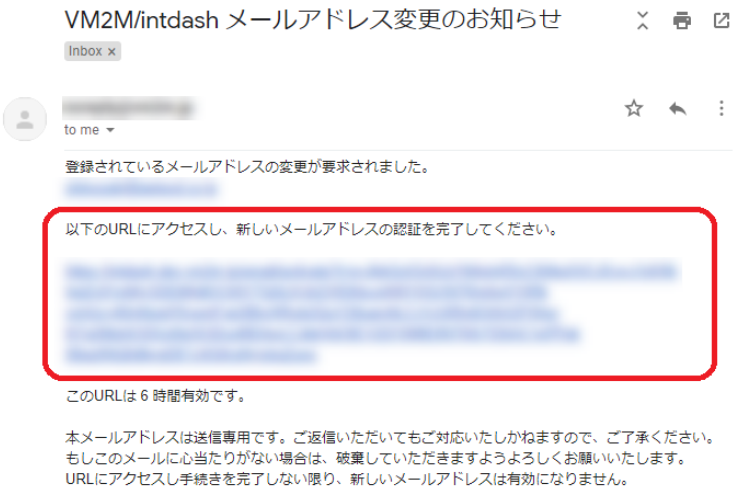


図 24　メールアドレスを確認

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。

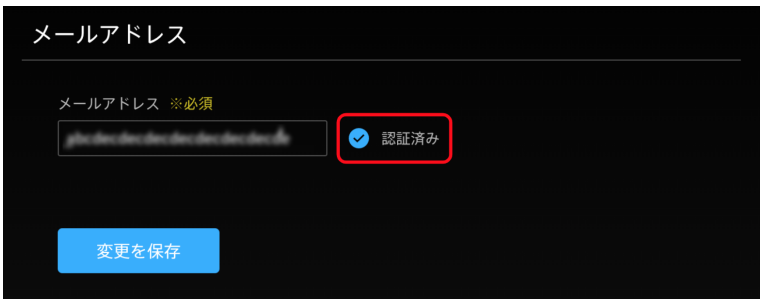


図 25　メールアドレス認証済み

3.3 Google Maps の API キーを設定する

Data Visualizer のビジュアルパーツの 1 つに、地図を表示する「Google Maps」があります。これ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。

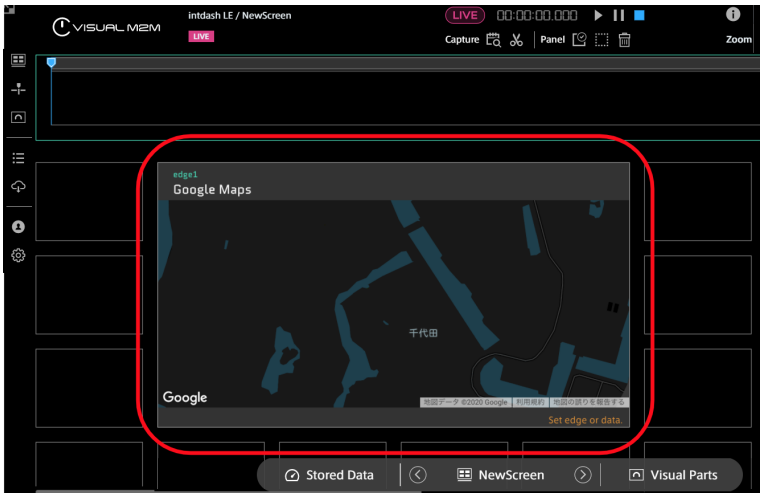


図 26　Google Maps ビジュアルパーツ

注釈: 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。

Google Maps の API キーを intdash サーバーに設定する手順は以下の通りです。

1. Utility Tools の「Google Maps API Key」ページ（`https://intdash.example.com/utility-tools/google-maps-api-key`）を開きます。
2. [API Key] の欄に、Google Maps API キーを入力します。(API キーではなく Client ID をお持ちの場合は、Client ID の欄に入力してください。)

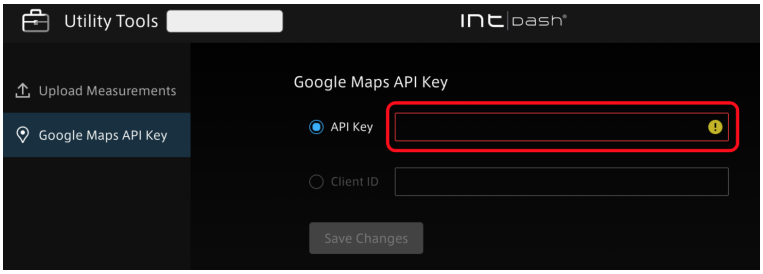


図 27　Google Maps API キーの入力

3. 入力された情報が正しければ、以下のように地図が表示されます。地図が表示されていることを確認して、[Save Changes] をクリックします。

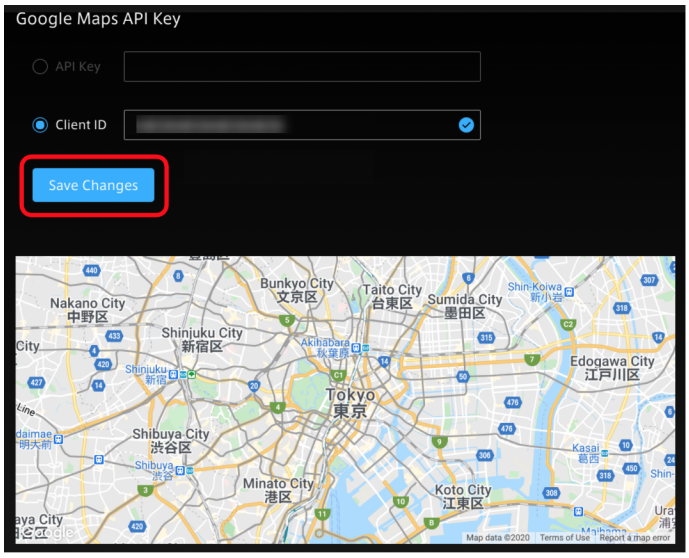


図 28　Google Maps API キーを保存

3.4 H.264 動画を扱うための設定を行う

H.264 動画を扱うために以下を行います。

- H.264 動画を fMP4 に変換するためのソフトウェア FFmpeg を intdash サーバーにインストールします。
- FFmpeg を使って動画を処理するために、intdash サーバー上では intdash-media-h264 というサービスが動作します。このサービスはエッジとして動作するため、このサービス用のエッジトークンを発行します。
- intdash-media-h264 サービスの設定を行います。

3.4.1 FFmpeg をインストールする

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 /usr/local/src/
```

2. FFmpeg の最新リリース版をダウンロードします。

```
# wget https://johnvansickle.com/ffmpeg/releases/ffmpeg-release-amd64-static.tar.xz
```

3. ダウンロードしたファイルを解凍し、バイナリを設置します。

```
# tar Jxvf ffmpeg-release-amd64-static.tar.xz
# install -m 0755 -o root -g root ffmpeg-*-amd64-static/ffmpeg /usr/local/bin/ffmpeg
```

4. ffmpeg コマンドを実行するとバージョン情報が出力されることを確認します。

```
# ffmpeg

ffmpeg version X.Y.Z-static https://johnvansickle.com/ffmpeg/  Copyright (c) 2000-20XX the FFmpeg␣
↪developers
```

3.4.2 エッジトークンを発行する

intdash-media-h264 サービスが使用するためのエッジトークンを、Edge Admin Console を使って発行します。

1. ウェブブラウザーで https://intdash.example.com/edges/admin/ にアクセスします。
2. エッジ media-h264 を検索し、[詳細] をクリックします。
3. トークンの [編集] をクリックします。

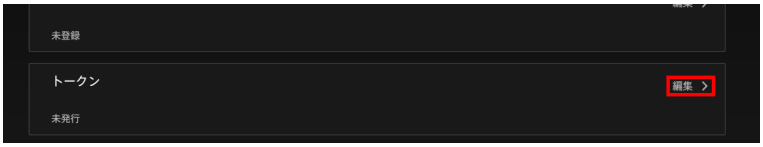


図 29　トークンの編集

4. [トークンを発行] をクリックします。

図 30　トークンを発行

5. 必要に応じてトークン名を入力し、[作成] をクリックします。

図 31　トークン名の入力と作成

6. 以下のメッセージが表示されます。エッジトークンをコピーして記録しておきます。

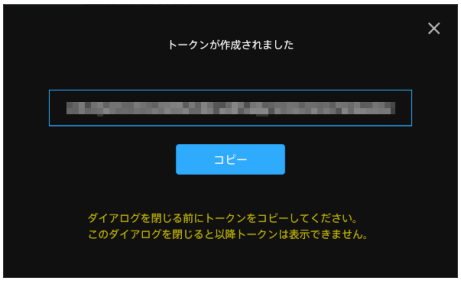


図 32　トークンの発行完了

3.4.3 エッジトークンを設定する

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 mediad-h264.toml mediad-h264.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 mediad-h264.toml
```

4. `edge-token` にエッジトークンを入力します。その後、設定ファイルを閉じます。

```
[Intdash]
  url = "http://127.0.0.1:8080"
  host = "127.0.0.1"
  port = 8080
  enable-tls = false
  edge-token = "XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"   <--
```

5. 設定を反映させるため、以下のコマンドで intdash-media-h264 を再起動します。

```
# systemctl restart intdash-media-h264
```

6. 再起動後、intdash-media-h264 が正常に起動していることを確認します。

```
# systemctl status intdash-media-h264
```

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

以上で設定は完了です。動作を確認する (p. 29) に進んでください。

04 動作を確認する

前章までの設定手順が完了したら、簡易的な動作確認として以下を行います。

- iPhone で intdash Motion アプリケーションを起動し、動画を撮影しながら intdash サーバーに送信します。
- PC で Data Visualizer アプリケーションを使って動画を確認します。
- Data Visualizer で過去のデータを再生できることを確認します。

手順は以下の通りです。

4.1 ダッシュボードを準備する

1. PC で Chrome ブラウザーで `https://intdash.example.com/vm2m/` にアクセスし、サインインします。

   Data Visualizer のダッシュボードが表示されます。

2. [Config]>[Download DAT File] から、動画データのパース定義ファイル（DAT ファイル）をダウンロードします。

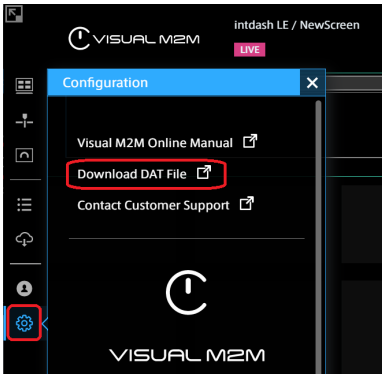


図 33　DAT ファイルダウンロードページを開く

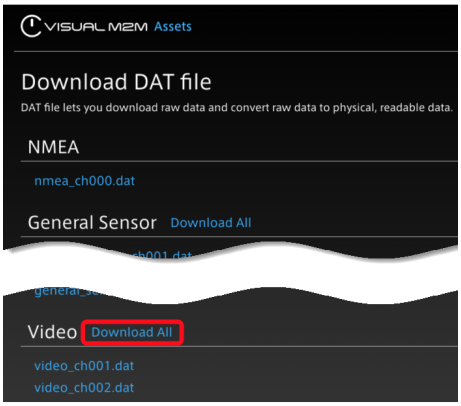


図 34　動画用の DAT ファイルをダウンロード

3. [Data Settings]>[Import] をクリックし、ダウンロードしたパース定義ファイルをインポートします。

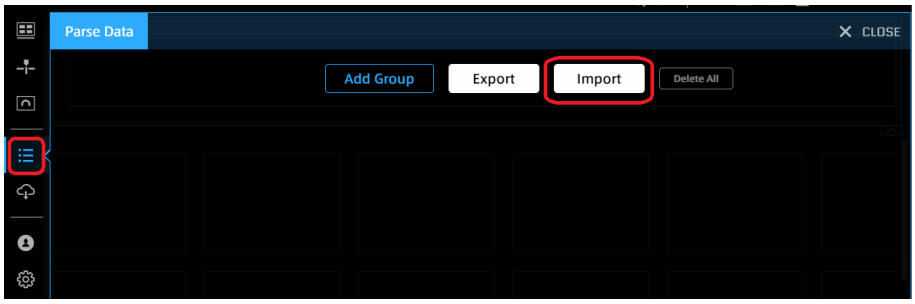


図 35　DAT ファイルのインポート

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。

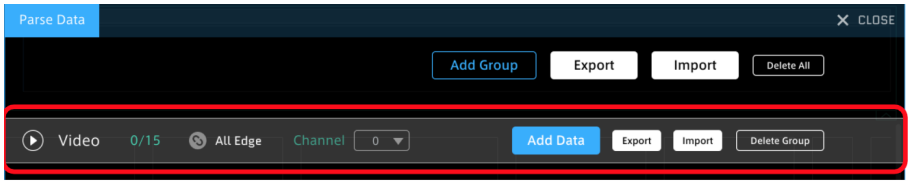


図 36　Video グループ

4. ダッシュボード上にパネルを作成します。

図 37　パネルを作成

5. パネルをクリックして動画を撮影するエッジ（iPhone）を選択し、Video:ch_001 をバインドします。

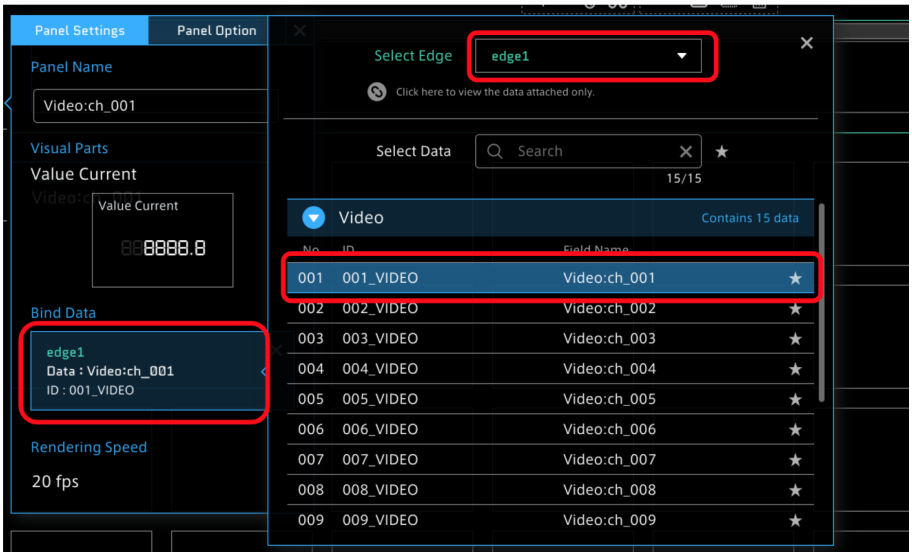


図 38　エッジを選択し、動画データをバインド

6. ビジュアルパーツは、Video Player を選択します。

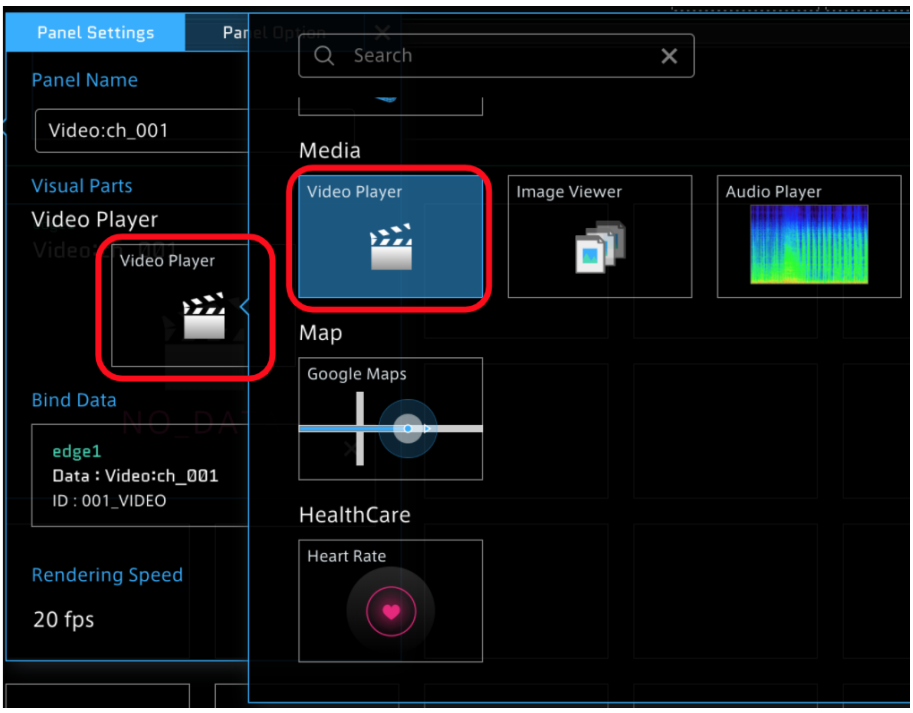


図 39　Video Player を選択

4.2 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、サインインします。

   サイン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`
   - Edge ID: `edge1` または `edge2`
   - Password: 設定したパスワード

2. [Settings] を開いて [Video] をオンにし、以下のように設定します。

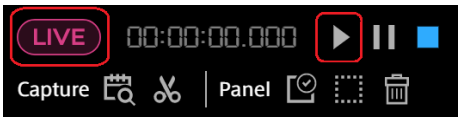
   - [Stream to Server] をオンにします。
   - [Save to Server] をオンにします。
   - [Channel] は 1 を選択します。
   - [Video Codec] は [H.264] を選択します。

3. Main 画面に戻り、▶ をタップして、撮影と送信を開始します。

4.3 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。

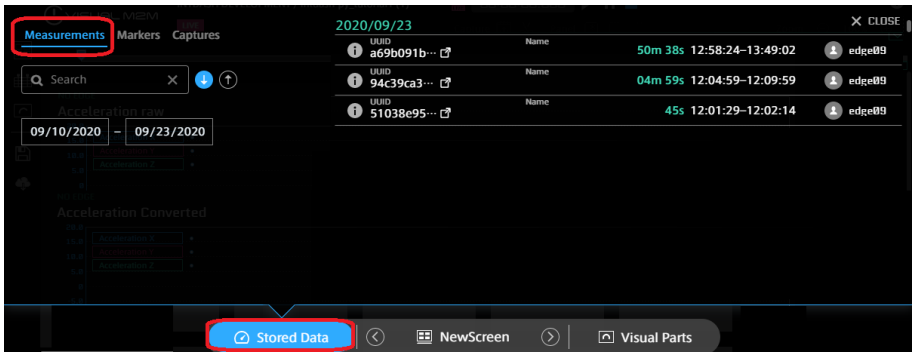


図 40　過去に取得したデータのリストから再生

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: 8080
  - 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
  - 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
  - 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
  - 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
  - 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-media-h264
  - 設定ファイル: `/etc/intdash/mediad-h264.toml`
  - サービスポート: 8084
  - 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
  - 用途: ログインしているエッジ（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
  - 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-edges-admin**
  - 設定ファイル: `/etc/sysconfig/intdash-web-edges-admin`
  - サービスポート: 13002
  - 用途: エッジ管理のためのウェブアプリケーション「Edge Admin Console」
- **intdash-web-oauth2**
  - 設定ファイル: `/etc/sysconfig/intdash-web-oauth2`
  - サービスポート: 13003
  - 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
  - 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
  - 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
- **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、443
- **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

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. ウェブブラウザーで、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=<発行されたユーザ名>
password=<発行されたパスワード>
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. 11) 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。