AMI を使った intdash サーバー構築手順
(AWS Marketplace 向け)

第 1 版

2020 年 10 月

aptpod

目次

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。

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。

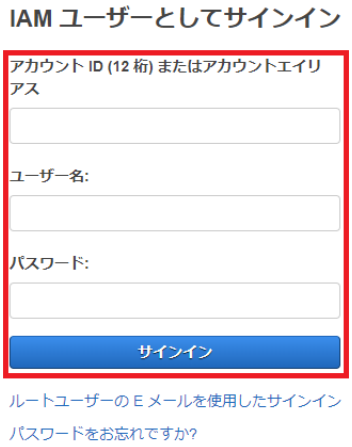


図 1　AWS へのサインイン

2. [コンピューティング] > [EC2] を選択します。

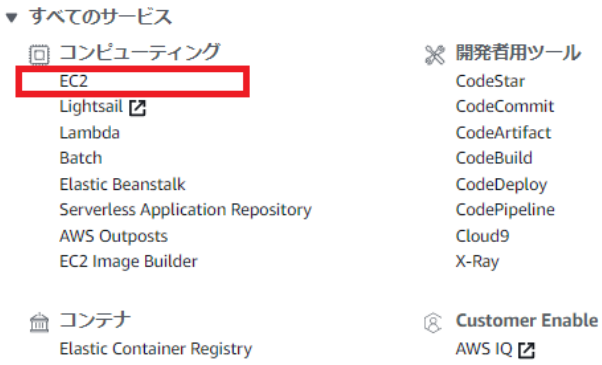


図 2　[EC2] を選択

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 LE 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。

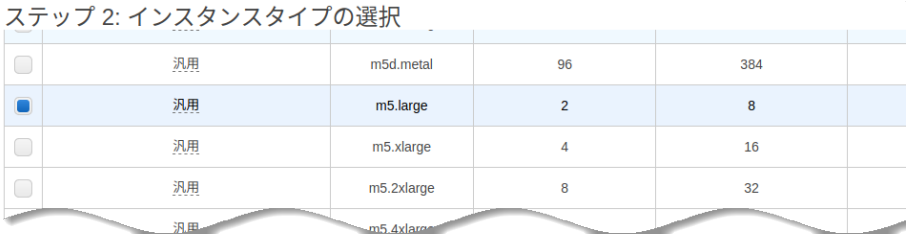


図 6　インスタンスタイプの選択

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

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証明書です。既存のキーペアを使用するか、新しいキーペア作成してください。新しいキーペアを作成する場合は、ここで必ずキーペアを自分のコンピューターにダウンロードしてください。

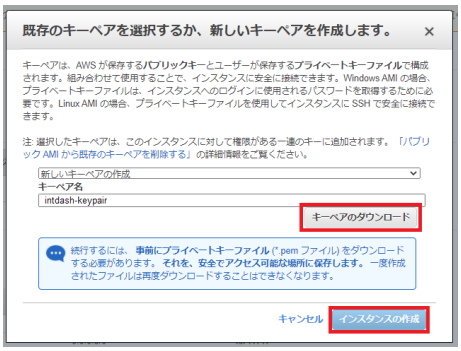


図 11　キーペアを新規作成する場合の例

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」

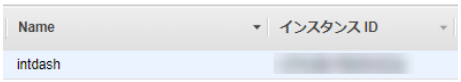


図 13　作成したインスタンスが表示されていることを確認

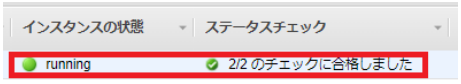


図 14　[インスタンスの状態] と [ステータスチェック] を確認

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-KEY-PAIR ec2-user@INSTANCE-IP-ADDRESS
```

PATH-TO-KEY-PAIR
: キーペアのファイルパス

INSTANCE-IP-ADDRESS
: intdash インスタンスの IP アドレス

例えば、キーペアのファイルパスが `./.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/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. 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
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、ユーザーとサーバー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。intdash に適用するサーバー証明書を取得する必要があります。サーバー証明書は、Cybertrust や DigiCert のような企業や Let's Encrypt のような団体が運営する認証局から入手することができます。

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。

```
map $http_upgrade $connection_upgrade_intdash {
    default upgrade;
    ''      close;
}

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;

    access_log  /var/log/nginx/intdash.access.log  main;
    error_log   /var/log/nginx/intdash.error.log;

    include /etc/nginx/intdash.d/*.conf;

    location / {
    }
}
```

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。

```
map $http_upgrade $connection_upgrade_intdash {
    default upgrade;
    ''      close;
}

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;      <--

    access_log  /var/log/nginx/intdash.access.log  main;
    error_log   /var/log/nginx/intdash.error.log;

    include /etc/nginx/intdash.d/*.conf;

    location / {
    }
}
```

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 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 = [
    "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"   <--
  client-driver = "static"
```

- [oauth2.client-static.authorization-code] サブセクション
  - redirect-uris

```
[oauth2.client-static.authorization-code]
  id = "533bc9ea_authorization_code.aptpod.co.jp"
  hashed-secret = ""
  redirect-uris = [
    "http://localhost:8080",
    "https://localhost:8080/oauth2/authorization/api/callback",
    "https://intdash.example.com/oauth2/authorization/api/callback",   <--
  ]
```

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。

3.1.3 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。セキュリティ上、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]
  bind-address = "127.0.0.1:8094"
  jwt-signing-method = "HS256"
```

(次のページに続く)

(前のページからの続き)

```
  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"   <--
```

3.1.4 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. 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
```

3. 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"
```

4. 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
```

5. 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	初期パスワード	用途
intdash	intdash インスタンスのインスタンス ID（インスタンス ID は EC2 コンソールで確認できます。例: `i-1234567890abcdef0`）	管理ユーザー
edge1	edge1	計測デバイス
edge2	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 パスワードを設定する

エッジアカウントのパスワードを変更します。

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、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。

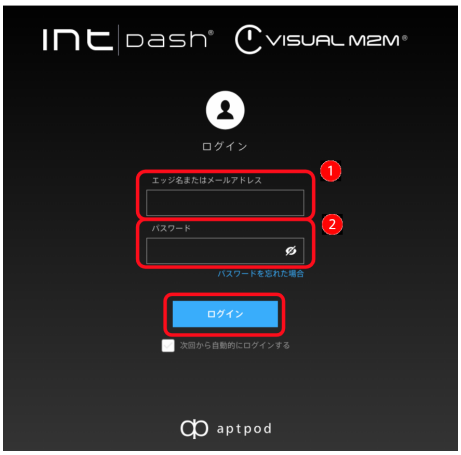


図 1　サインイン画面

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 2　パスワードの設定

4. 以上のパスワード設定手順を、すべてのエッジアカウントについて行います。

> ⚠ **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）
     - 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。

3.2.3 管理者ユーザーエッジのメールアドレスを設定する

管理者ユーザーエッジのメールアドレスを設定します。

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、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。

   ユーザー `intdash` のマイページが表示されます。

図 3　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。

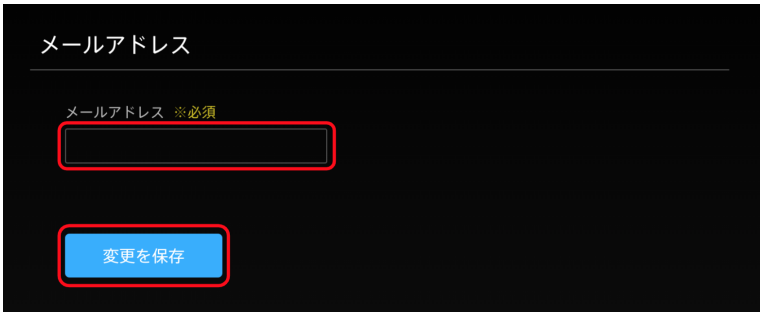


図 4　メールアドレスを設定

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。

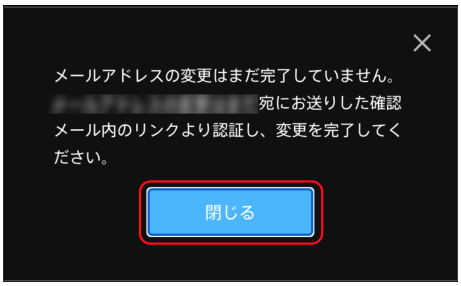
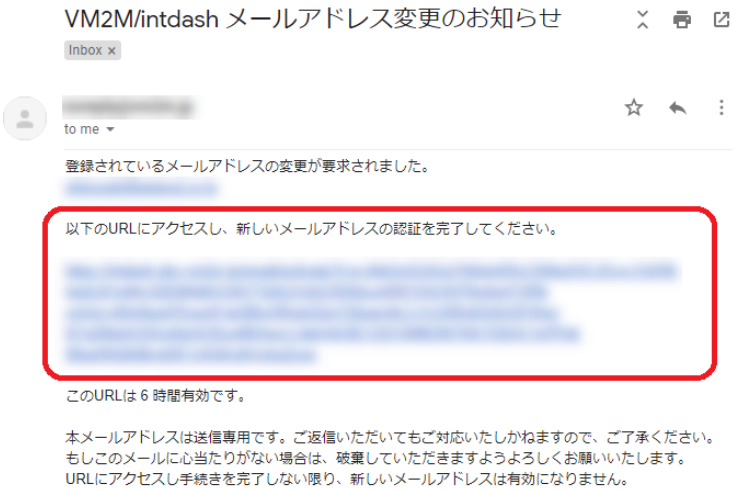


図 5　メールアドレスを確認

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。

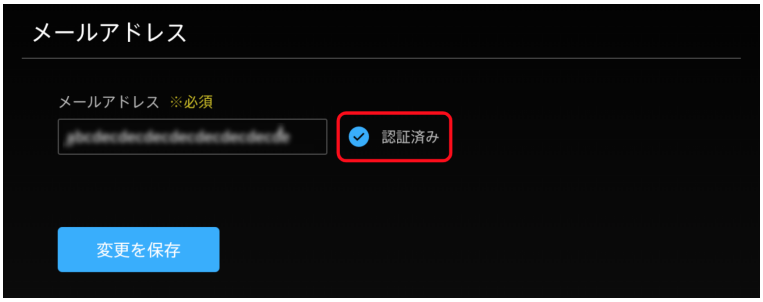


図 6　メールアドレス認証済み

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。

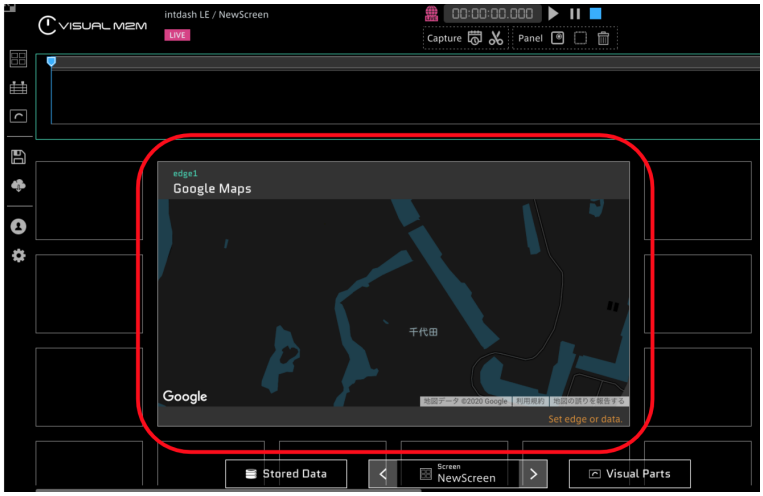


図 7　Google Maps ビジュアルパーツ

> **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。)

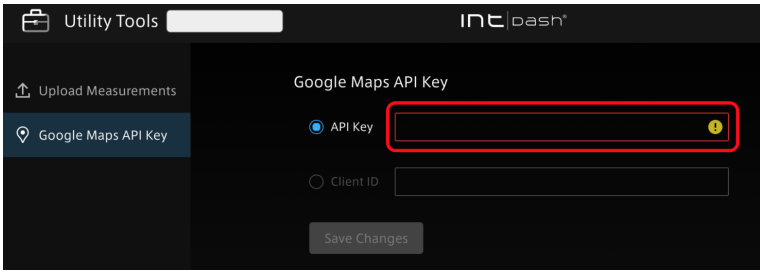


図 8　Google Maps API キーの入力

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。

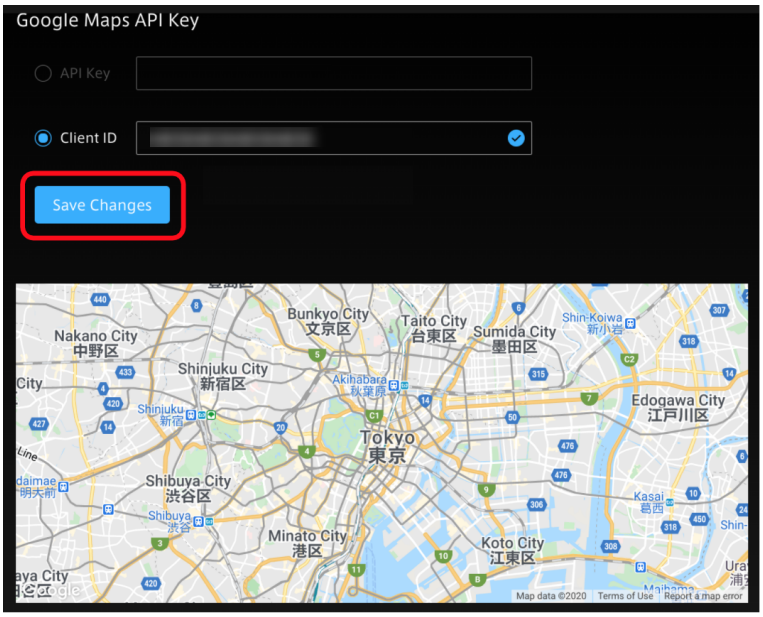


図 9　Google Maps API キーを保存

3.4 H.264 動画を扱うための設定を行う

H.264 動画を扱うために必要な設定を行います。

- 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、intdash コマンドを使って発行します。

ここでは初めて intdash コマンドを使用するため、先に intdash コマンドの設定を行い、そのあとエッジトークンを発行します。

1. 前の手順に引き続き SSH で intdash インスタンスに接続し、以下のコマンドを実行して対話的に intdash コマンドの設定を行います。

```
# intdash config
? Address? (http://localhost:8080)   --> Enter
? Username? (intdash)   --> Enter
? Password? ***********   --> Enter password and press Enter
```

2. 以下のコマンドを実行して、エッジトークンを発行します。

```
# intdash edge-token create --edge media-h264
```

3. token フィールドにエッジトークンが表示されます。エッジトークンをコピーして記録しておきます。

```
{
  "uuid": "00112233-4455-6677-8899-aabbccddeeff",
  "name": "",
  "token": "XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX",   <--
  "disabled": false,
  "protected": false,
  "last_used_at": "2020-01-23T12:34:56.123456Z",
  "expired_at": "2020-01-23T12:34:56.123456Z",
  "created_at": "2020-01-23T12:34:56.123456Z",
  "updated_at": "2020-01-23T12:34:56.123456Z"
}
```

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. 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
```

7. 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. 25) 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 VM2M 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。

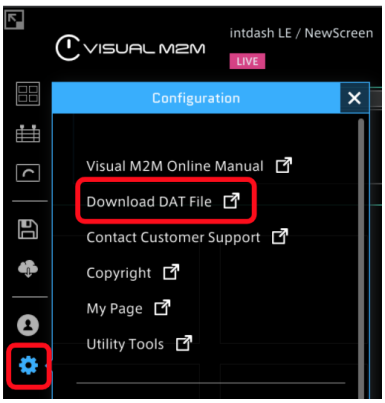


図 1　DAT ファイルダウンロードページを開く

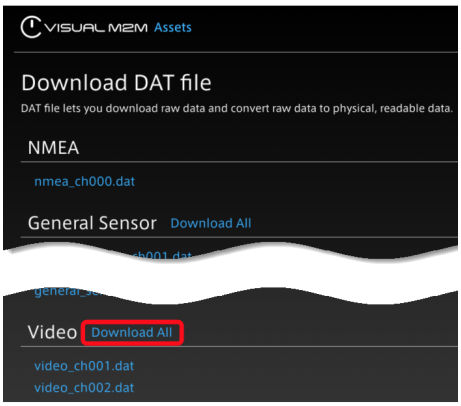


図 2　動画用の DAT ファイルをダウンロード

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。

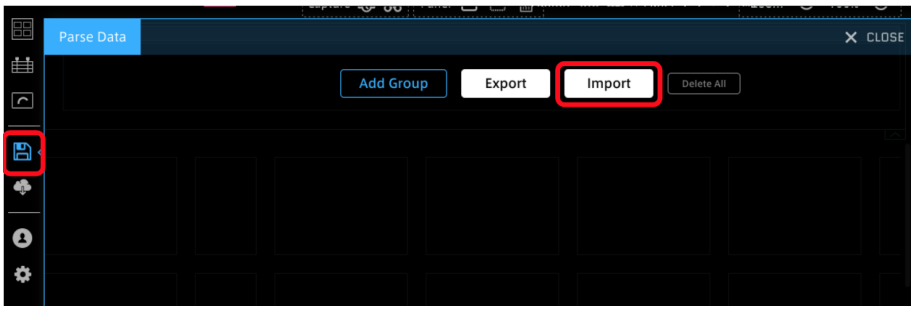


図 3　DAT ファイルのインポート

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。

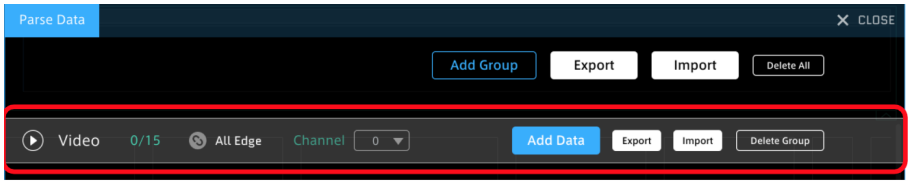


図 4　Video グループ

4. ダッシュボード上にパネルを作成します。

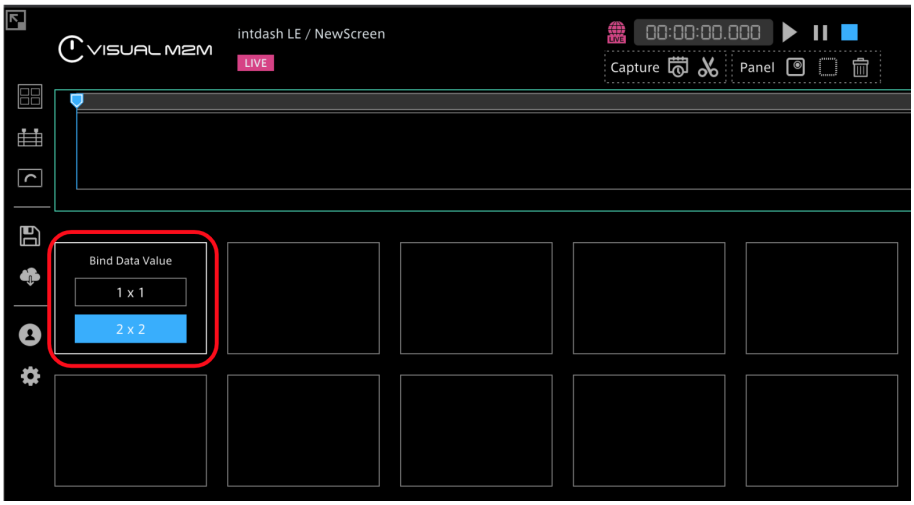


図 5　パネルを作成

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。

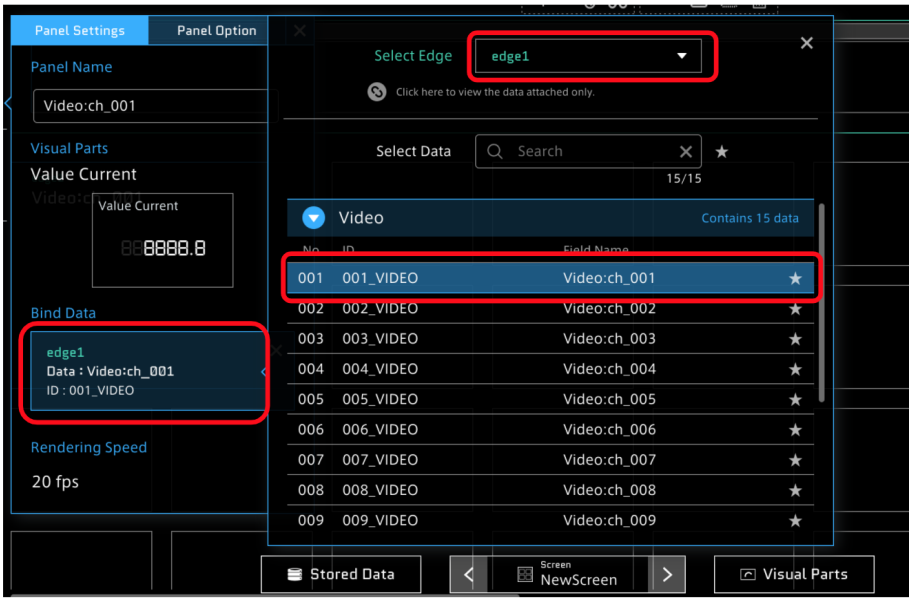


図 6　エッジを選択し、動画データをバインド

6. ビジュアルパーツは、Video Player を選択します。

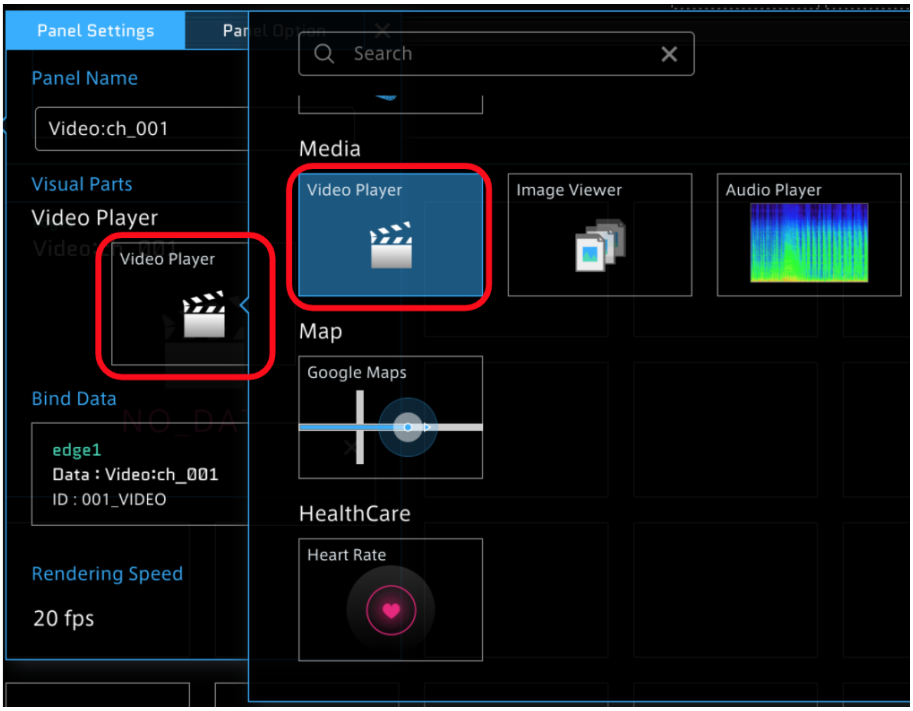


図 7　Video Player を選択

4.2 iPhone からリアルタイム動画を送信する

iPhone で VM2M Motion アプリケーションを起動し、動画を送信します。

1. iPhone で VM2M 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

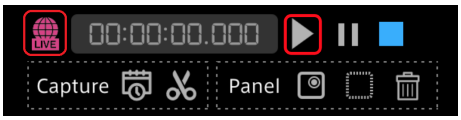
2. [Extra]>[Video for Streaming] を開いてストリーミングの設定をします。
   - [Streaming] をオンにします。
   - [Save to server] をオンにします。
   - Codec は [H.264] を選択します。

3. Main 画面に戻り、[Play] 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。

   VM2M 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、VM2M Motion で■をタップし、計測を終了します。
3. Data Visualizer で、[Stored Data]>[S to E] 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。

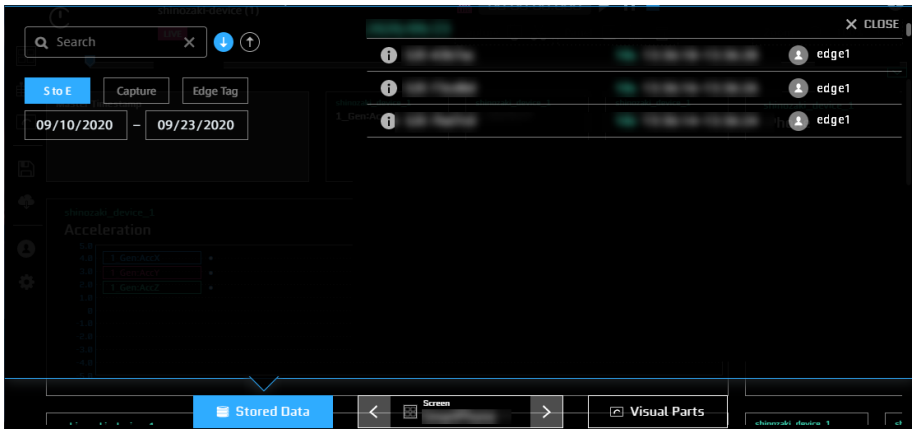


図 8　過去に取得したデータのリストから再生

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-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。サービス名と同名の CLI コマンドで設定変更が可能

- **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。サービス名と同名の CLI コマンドで設定変更が可能
- **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-me**
  - 設定ファイル: `/etc/sysconfig/intdash-web-me`
  - サービスポート: 13000
  - 用途: GUI における MyPage 機能を担うサービス
- **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

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