おくやみコーナー設置自治体支援ナビ

インストールマニュアル

第 1 版

令和 2 年 5 月 15 日

内閣官房情報通信技術（IT）総合戦略室

改訂履歴

日付	改訂内容
令和2年5月	初版作成

目次

1　はじめに

当資料は、Linux 環境または Windows 環境において、支援ナビシステム(以下、支援ナビ)の稼働環境構築手順を記述しています。

支援ナビ導入のシステム要件は以下のとおりです。

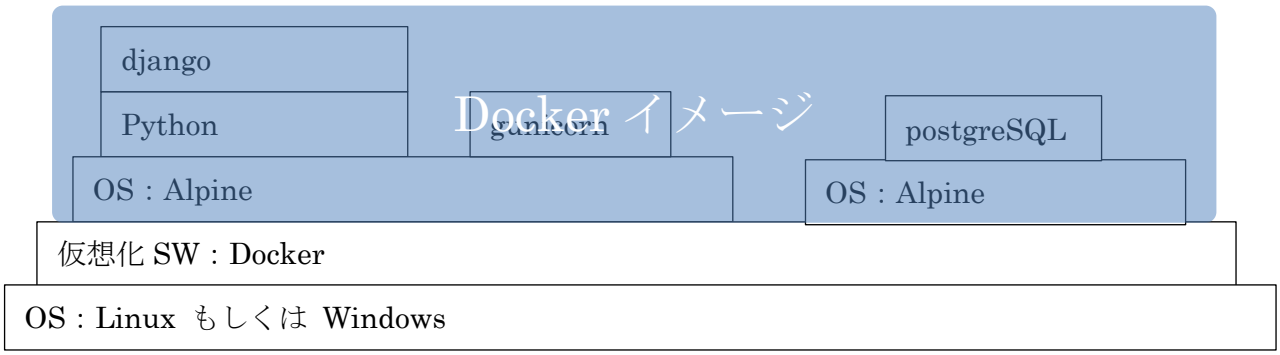
表 1：支援ナビのシステム要件

No.	項目	要件	備考
1	OS	Linux or Windows	・Docker をサポートする OS であること
2	CPU	1 コア以上	
3	メモリ	4GB 以上	
4	ディスク	20GB 以上	

2　コンポーネント概念図

(1)　アプリケーション資産の移植容易性を確保する為に OS の仮想化技術として Docker を採用しています。

(2)　OS 上に Docker 稼働環境を用意し、データベースやアプリケーションを含む Docker イメージを配置して稼働させます。



参考）Docker イメージ内のソフトウェア構成

No.	レイヤ	ソフトウェア	バージョン
1	OS	Alpine	3.10
2	言語	Python	3.8.0
3	Web アプリケーションフレームワーク	django	2.2.7
4	WSGI サーバー	gunicorn	20.0.0
5	データベース	postgreSQL	12.1

3　サーバの構築【Linux 編】

【前提】

・Linux 環境が構築済みであること。

・インストールユーザは、sudo 可能なユーザであること。

・インターネットに接続できる環境であること。

・当資料は AmazonLinux2 で下記ユーザを利用した表記となっております。環境に応じて適宜読み替えてください。

ユーザ ID：ec2-user

ホームディレクトリ：/home/ec2-user　作業ディレクトリとして利用

【手順】

（1） docker のインストール

下記のコマンドで docker のインストールを行います。

```
$ sudo yum install -y docker
Loaded plugins: priorities, update-motd, upgrade-helper
amzn-main                                                | 2.1 kB     00:00
amzn-updates                                             | 2.5 kB     00:00
Resolving Dependencies
--> Running transaction check


（中略）
Dependency Installed:
  containerd.x86_64 0:1.2.6-1.2.amzn1
  libseccomp.x86_64 0:2.3.1-2.4.amzn1
  pigz.x86_64 0:2.3.3-1.6.amzn1
  runc.x86_64 0:1.0.0-0.1.20190510.git2b18fe1.0.amzn1
  xfsprogs.x86_64 0:4.5.0-18.23.amzn1


Complete!
```

（２） docker サービスの起動

```
$ sudo service docker start

Redirecting to /bin/systemctl start docker.service
```

（３） docker-compose のインストール

下記コマンドで資源を取得します。

```
$ sudo curl -L https://github.com/docker/compose/releases/download/1.24.1/docker-compose-`uname -s`-`uname -m` -o /usr/local/bin/docker-compose
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100   617    0   617    0     0    461      0 --:--:--  0:00:01 --:--:--   461
100 15.4M  100 15.4M    0     0  3191k      0  0:00:04  0:00:04 --:--:-- 5080k
```

下記コマンドで権限を付与します。

```
$ sudo chmod +x /usr/local/bin/docker-compose
```

下記のようにバージョンの確認ができれば、インストール完了です。

```
$ docker-compose -version

docker-compose version 1.24.1, build 4667896b
```

4　支援ナビソフトウェアの導入・設定【Linux 編】

4.1　支援ナビソフトウェアの構成

＜支援ナビソフトウェアの提供物の構成＞

支援ナビソフトウェアの提供物は以下のとおり。

snavi.tar　　：　支援ナビ docker イメージ

dbdata.tar.gz　　：　支援ナビデータ（PostgreSQL データフォルダーを圧縮したもの）

docker-compose.yml：docker イメージ起動用設定ファイル

※以下の説明では、これらファイルが*/home/ec2-user* フォルダに存在すると仮定します。

＜支援ナビソフトウェアのフォルダ構成＞

支援ナビソフトウェアは以下のように、インストールフォルダ（任意）に、snavi 、dbdata の 2 つのフォルダが配置されます。

＜インストールフォルダ＞

├snavi　　：docker-compose.yml が配置される。支援ナビ作業用フォルダ。

└dbdata　　：dbdata.tar.gz が展開され配置される。支援ナビデータフォルダ。

※以下の説明では、インストールフォルダを*/home/ec2-user* と仮定します。

4.2　支援ナビソフトウェアのインストール

支援ナビのインストールは以下の手順で行います。

①　支援ナビ docker イメージのロード

docker load コマンドで snavi.tar をロードします。

```
$ sudo docker load < /home/ec2-user/snavi.tar
```

②　支援ナビデータの配置

dbdata.tar.gz を解凍し、支援ナビデータフォルダに配置する。またオーナーグループも変更します。

```
$ cd /home/ec2-user

$ ll dbdata.tar.gz

 dbdata.tar.gz
```

```
$ mkdir wrk

$ mv dbdata.tar.gz wrk

$ cd wrk

$ gunzip dbdata.tar.gz

$ tar -xvf dbdata.tar

$ mv dbdata ..

$ sudo chown 999:root dbdata

$ sudo su -

$ cd /home/ec2-user/dbdata

$ sudo chown 999:999 *
```

③　docker イメージ起動用設定ファイルの配置

```
$ cd /home/ec2-user

$ ll docker-compose.yml

 docker-compose.yml

$ mkdir snavi

$ mv docker-compose.yml snavi
```

5　支援ナビソフトウェアの運用【Linux 編】

5.1　支援ナビの起動

docker-compose コマンドで起動します。

```
$sudo　/usr/local/bin/docker-compose　-f /home/ec2-user/snavi/docker-compose.yml up -d
```

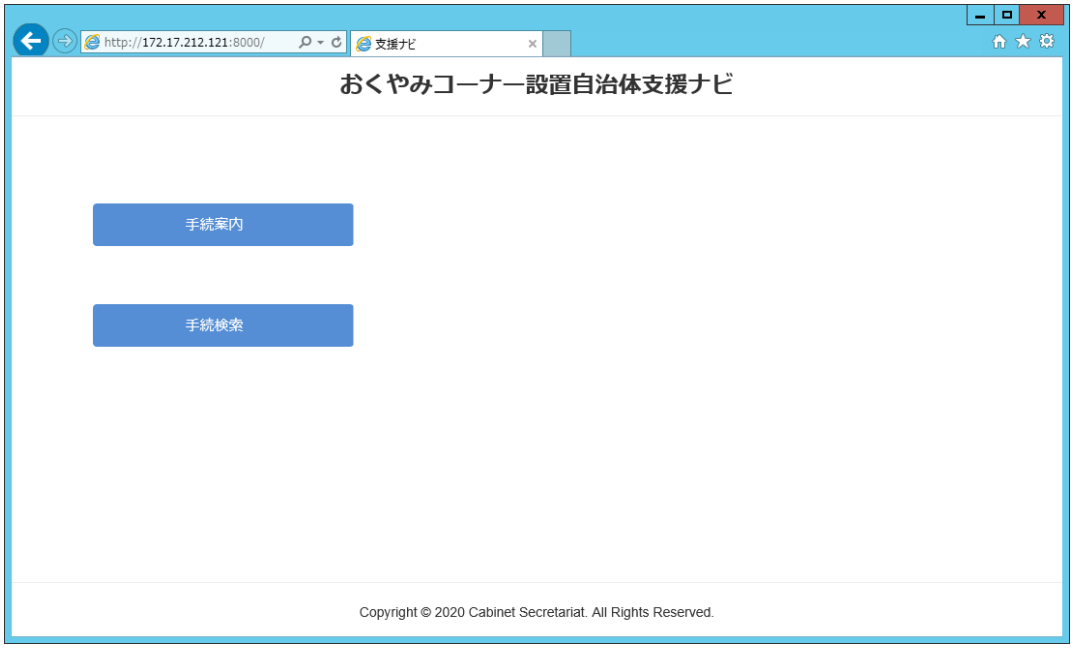
以下のコマンドで動作確認を行います。HTML 文章が表示されれば OK です。

```
$curl localhost:8000
<!DOCTYPE html>
        <html lang="ja">
        <head>
            <meta charset="utf-8">
            <meta name="format-detection" content="telephone=no">
            <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
            <title>支援ナビ</title>
              （後略）
```

ブラウザから支援ナビサーバの IP アドレスとポート(8000)を指定して画面表示を確認します。

http://<IPアドレス>:8000/

図 1：支援ナビ TOP 画面



5.2　支援ナビの停止

docker-compose コマンドで停止します。

例）

```
$ sudo /usr/local/bin/docker-compose -f /home/ec2-user/snavi/docker-compose.yml down
```

5.3　支援ナビのログ確認

docker　ps でコンテナ名を確認後、docker　logs コマンドでログを確認します。

下記コマンドでプロセス名を確認してください。

```
$ sudo docker ps
        CONTAINER  ID                  IMAGE                                COMMAND
CREATED                STATUS                PORTS                          NAMES
        868b68cfcf49           snavi:latest          "bash -c 'python man…"    3 days ago
Up 3 days              0.0.0.0:8000->8000/tcp    snavi_web_1
        44e131b68611           postgres:12.1         "docker-entrypoint.s…"    3 days ago
Up 3 days              0.0.0.0:5432->5432/tcp    snavi_postgres_1
```

下記コマンドでログを確認します。

＜支援ナビ（Web アプリ）のログを見る場合＞

```
$ sudo docker logs snavi_web_1
            System check identified some issues:


WARNINGS:


（中略）
[2020-02-03 10:36:50 +0000] [18] [INFO] Booting worker with pid: 18
[2020-02-03 10:36:50 +0000] [19] [INFO] Booting worker with pid: 19
[2020-02-03 10:36:50 +0000] [20] [INFO] Booting worker with pid: 20
[2020-02-03 10:36:51 +0000] [21] [INFO] Booting worker with pid: 21
```

＜支援ナビ（データベース）のログを見る場合＞

```
$ sudo docker logs snavi_postgres_1

PostgreSQL Database directory appears to contain a database; Skipping initialization


2020-02-03 10:36:42.754 UTC [1] LOG:  starting PostgreSQL 12.1 (Debian 12.1-1.pgdg100+1) on x86_64-pc-linux-gnu, compiled by gcc (Debian 8.3.0-6) 8.3.0, 64-bit
2020-02-03 10:36:42.754 UTC [1] LOG:  listening on IPv4 address "0.0.0.0", port 5432
2020-02-03 10:36:42.754 UTC [1] LOG:  listening on IPv6 address "::", port 5432
2020-02-03 10:36:42.772 UTC [1] LOG:  listening on Unix socket "/var/run/postgresql/.s.PGSQL.5432"
2020-02-03 10:36:42.829 UTC [27] LOG:  database system was interrupted; last known up at 2020-01-24 07:10:32 UTC
2020-02-03 10:36:46.478 UTC [27] LOG:  database system was not properly shut down; automatic recovery in progress
2020-02-03 10:36:46.501 UTC [27] LOG:  redo starts at 0/2E5C7B0
2020-02-03 10:36:46.518 UTC [27] LOG:  invalid record length at 0/2E68530: wanted 24, got 0
2020-02-03 10:36:46.518 UTC [27] LOG:  redo done at 0/2E684F8
2020-02-03 0:36:46.619 UTC [1] LOG:  database system is ready to accept connections
```

5.4 その他

(1) データバックアップ

必要に応じて DB データバックアップを取得してください。

DB データは、データフォルダ（支援ナビインストールフォルダ/../dbdata）に存在します。

同フォルダをバックアップすることでデータ断面を確保することが可能です。

(2) URL 監視

必要に応じて URL 監視を実装してください。URL は以下のとおりです。

支援ナビ： http://<サーバの IP アドレス>:8000

支援ナビ管理サイト： http://<サーバの IP アドレス>:8000/admin

(3) ログ

ログは標準出力に出力されます。必要に応じて永続化の手段を講じてください。

また必要に応じてログ監視も実装してください。

6　サーバの構築【WindowsServer 編】

【前提】

・Hyper-V がサポートされた WindowsServer 環境が構築済みであること。

・仮想化技術が有効化されていること。

有効されているかどうかは、systeminfo コマンドを実行することで確認できます。

有効化されている場合、最終行に以下の行が表示されます。

```
Hyper-V の要件:          ハイパーバイザーが検出されました。Hyper-V に必要な機能は表示されません。
```

・インターネットに接続できる環境であること。

・当資料は WindowsServer2019 で下記ユーザを利用した表記となっております。

環境に応じて適宜読み替えてください。

ユーザ ID：Administrator

ホームディレクトリ：C:¥oss　作業ディレクトリとして利用

【手順】

(1) Hyper-V のインストール

PowerShell コマンドから Hyper-V をインストールします。

インストールに成功すると Success が True となります。

```
> PowerShell Install-WindowsFeature hyper-v


Success Restart Needed Exit Code      Feature Result
------- -------------- ---------      --------------
True    Yes            SuccessRest... {Hyper-V}
警告: インストール処理を完了するには、このサーバーを再起動する必要があります。
```

(2) コンテナーのインストール

Windows のコンテナー機能をインストールします。

Hyper-V と同じようにインストールに成功すると Success が True となります。

```
> PowerShell Install-WindowsFeature containers

Success Restart Needed Exit Code      Feature Result
------- -------------- ---------      --------------
True    Yes            SuccessRest... {コンテナー}
警告: インストール処理を完了するには、このサーバーを再起動する必要があります。
```

(3) マシンの再起動

インストール処理を完了させるため PowerShell コマンドからマシンを再起動します。

```
> PowerShell Restart-Computer –Force
```

(4) プロバイダーのインストール

Docker EE をインストールするためのプロバイダーをインストールします。

インストールの確認が表示されたら y キーを入力します。

```
> PowerShell Install-Module -Name DockerMsftProvider -Repository PSGallery -Force

続行するには NuGet プロバイダーが必要です
PowerShellGet で NuGet ベースのリポジトリを操作するには、'2.8.5.201' 以降のバージョンの NuGet プロバイダーが必要です。 NuGet プロバイダーは 'C:¥Program Files¥PackageManagement¥ProviderAssemblies' または
'C:¥Users¥Administrator¥AppData¥Local¥PackageManagement¥ProviderAssemblies'
に配置する必要があります。'Install-PackageProvider -Name NuGet -MinimumVersion 2.8.5.201 -Force' を実行して NuGet プロバイダーをインストールすることもできます。今すぐ PowerShellGet で NuGet プロバイダーをインストールしてインポートしますか?
[Y] はい(Y)  [N] いいえ(N)  [S] 中断(S)  [?] ヘルプ (既定値は "Y"): y
```

(5) Docker EE のインストール

Docker EE をインストールします。

インストールの確認が表示されたら y キーを入力します。

```
> PowerShell Install-Package -Name Docker -ProviderName DockerMsftProvider -Force


パッケージは、信頼済みとマークされていないパッケージ ソースから取得されています。

'DockerDefault' からソフトウェアをアンインストールしますか?

[Y] はい(Y)  [A] すべて続行(A)  [N] いいえ(N)  [L] すべて無視(L)  [S] 中断(S)  [?] ヘルプ
(既定値は "N"): y


Name                           Version          Source           Summary

----                           -------          ------           -------

Docker                         18.09.0          DockerDefault    Contains Docker EE
for use with Windows Server
```

(6) LinuxKit のインストール

Docker EE のインストールは完了していますが docker で Linux コンテナを動かしたい場合は LinuxKit のインストールも行う必要があります。

提供物 release.zip を、以下のフォルダを作成して中身を展開します。

```
C:¥Program Files¥Linux Containers
```

(7) Docker Compose のインストール

Docker CE には Docker Compose が付属していましたが、Docker EE には付属していないので手動でインストールします。

提供物 docker-compose.exe を以下のように配置します。

```
C:¥Program Files¥Docker¥docker-compose.exe
```

(8) マシンの再起動

PowerShell コマンドからマシンを再起動します。

```
> PowerShell Restart-Computer –Force
```

(9) Experimental 機能の有効化

Linux コンテナのサポートは現時点では Experimental 扱いなので、デーモン設定ファイルから Experimental 機能の有効化を行います。

以下のファイルを新規作成します。

```
C:¥ProgramData¥docker¥config¥daemon.json
```

ファイルの内容は以下のように記載して上書き保存します。

```
{
  "experimental": true
}
```

(10) マシンの再起動

デーモン設定ファイルを読み込ませるため PowerShell コマンドからマシンを再起動します。

```
> PowerShell Restart-Computer -Force
```

以上で Docker EE が利用可能となります。

(11) Docker EE のバージョンの表示

docker version コマンドを実行してバージョンが表示されることを確認します。

Server 側の Experimental が true となっていることも確認します。

```
> docker version
Client: Docker Engine - Enterprise
 Version:           19.03.5
 API version:       1.40
 Go version:        go1.12.12
 Git commit:        2ee0c57608
 Built:             11/13/2019 08:00:16
 OS/Arch:           windows/amd64
 Experimental:      false

Server: Docker Engine - Enterprise
 Engine:
  Version:          19.03.5
  API version:      1.40 (minimum version 1.24)
  Go version:       go1.12.12
  Git commit:       2ee0c57608
  Built:            11/13/2019 07:58:51
  OS/Arch:          windows/amd64
  Experimental:     true
```

(12) Docker Compose のバージョンの表示

docker-compose version コマンドを実行してバージョンが表示されることを確認します。

```
> docker-compose version
docker-compose version 1.25.3, build d4d1b42b
docker-py version: 4.1.0
CPython version: 3.7.4
OpenSSL version: OpenSSL 1.1.1c  28 May 2019
```

7 支援ナビソフトウェアの導入・設定【WindowsServer 編】

7.1 支援ナビソフトウェアの構成

＜支援ナビソフトウェアの提供物の構成＞

支援ナビソフトウェアの提供物は以下のとおり。

snavi.tar　　：　支援ナビ docker イメージ

dbdata.zip　　：　支援ナビデータ（PostgreSQL データフォルダーを圧縮したもの）

docker-compose.yml：docker イメージ起動用設定ファイル

release.zip　　：LinuxKit

docker-compose.exe：docker-compose

※以下の説明では、これらファイルが *C:¥oss* フォルダに存在すると仮定します。

＜支援ナビソフトウェアのフォルダ構成＞

支援ナビソフトウェアは以下のように、インストールフォルダ（任意）に、snavi 、dbdata の 2 つのフォルダが配置されます。

＜インストールフォルダ＞

├snavi　　：docker-compose.yml が配置される。支援ナビ作業用フォルダ。

└dbdata　　：dbdata.tar.gz が展開され配置される。支援ナビデータフォルダ。

※以下の説明では、インストールフォルダを *C:¥oss* と仮定します。

7.2 支援ナビソフトウェアのインストール

支援ナビのインストールは以下の手順で行います。

① 支援ナビ docker イメージのロード

docker load コマンドで snavi.tar をロードしてください。

```
>cd C:¥oss

>docker load  -i  snavi.tar
```

② 支援ナビデータの配置

支援ナビデータフォルダを作成します。

```
> mkdir dbdata
```

dbdata.zip を解凍し、dbdata フォルダ配下の全ファイルを支援ナビデータフォルダに配置してください。

③ docker イメージ起動用設定ファイルの配置

```
    > mkdir snavi

    > mv .¥docker-compose.yml snavi

    > ls snavi

    Directory: C:¥oss¥snavi


Mode                 LastWriteTime         Length Name

----                 -------------         ------ ----

-a----         1/23/2020   10:09 AM            424 docker-compose.yml
```

8　支援ナビソフトウェアの運用【WindowsServer 編】

8.1　支援ナビの起動

docker-compose コマンドで起動する。

```
>cd C:¥oss

>docker-compose -f ./snavi/docker-compose.yml up –d

            Creating network "snavi_default" with the default driver

            Pulling postgres (postgres:12.1)...

            12.1: Pulling from library/postgres

            bc51dd8edc1b: Pull complete

（中略）



            Digest:
          sha256:5181eccc7c903e4f1beffa89a735cb7ed72e0c81d6c34c471552c3fa8bff0858

            Status: Downloaded newer image for postgres:12.1

            Creating snavi_postgres_1 ... done

Creating snavi_web_1          ... done
```

以下のコマンドでプロセスが 2 つ存在することを確認する。

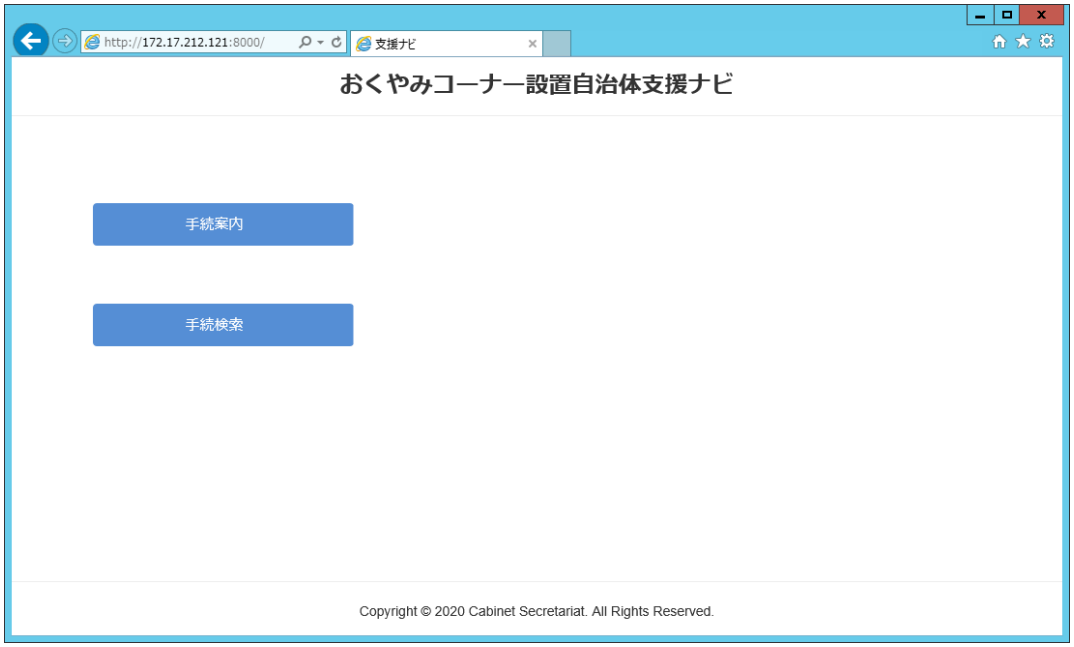
```
> docker ps

      CONTAINER  ID                IMAGE                              COMMAND
CREATED              STATUS              PORTS                        NAMES

      868b68cfcf49          snavi:latest          "bash -c 'python man…"    3 days ago
Up 3 days            0.0.0.0:8000->8000/tcp    snavi_web_1

      44e131b68611          postgres:12.1         "docker-entrypoint.s…"    3 days ago
Up 3 days            0.0.0.0:5432->5432/tcp    snavi_postgres_1
```

プロセスが 2 つ存在しない場合、一度支援ナビを停止し再度起動してください。
停止…8.2 支援ナビ停止の手順を実施
起動…8.1 支援ナビ起動の手順を実施

ブラウザから支援ナビサーバの IP アドレスとポート(8000)を指定して画面表示を確認します。

http://<IPアドレス>:8000/

図 2：支援ナビ TOP 画面



8.2 支援ナビの停止

docker-compose コマンドで停止します。

例）

```
> cd C:¥oss

> docker-compose -f ./snavi/docker-compose.yml down
```

8.3 支援ナビのログ確認

docker　ps コマンドでコンテナ名を確認後、docker　logs コマンドでログを確認します。

下記コマンドでプロセス名を確認する。

```
> docker ps
CONTAINER ID        IMAGE                  COMMAND                  CREATED             STATUS              PORTS                    NAMES
868b68cfcf49        snavi:latest           "bash -c 'python man…"   3 days ago          Up 3 days           0.0.0.0:8000->8000/tcp   snavi_web_1
44e131b68611        postgres:12.1          "docker-entrypoint.s…"   3 days ago          Up 3 days           0.0.0.0:5432->5432/tcp   snavi_postgres_1
```

下記コマンドでログを確認します。

支援ナビ（Web アプリ）のログを見る場合。

```
> docker logs snavi_web_1
        System check identified some issues:

WARNINGS:

（中略）
[2020-02-03 10:36:50 +0000] [18] [INFO] Booting worker with pid: 18
[2020-02-03 10:36:50 +0000] [19] [INFO] Booting worker with pid: 19
[2020-02-03 10:36:50 +0000] [20] [INFO] Booting worker with pid: 20
[2020-02-03 10:36:51 +0000] [21] [INFO] Booting worker with pid: 21
```

支援ナビ（データベース）のログを見る場合。

```
>docker logs snavi_postgres_1

PostgreSQL Database directory appears to contain a database; Skipping initialization


2020-02-03 10:36:42.754 UTC [1] LOG:  starting PostgreSQL 12.1 (Debian 12.1-1.pgdg100+1) on x86_64-pc-linux-gnu, compiled by gcc (Debian 8.3.0-6) 8.3.0, 64-bit
2020-02-03 10:36:42.754 UTC [1] LOG:  listening on IPv4 address "0.0.0.0", port 5432
2020-02-03 10:36:42.754 UTC [1] LOG:  listening on IPv6 address "::", port 5432
2020-02-03 10:36:42.772 UTC [1] LOG:  listening on Unix socket "/var/run/postgresql/.s.PGSQL.5432"
2020-02-03 10:36:42.829 UTC [27] LOG:  database system was interrupted; last known up at 2020-01-24 07:10:32 UTC
2020-02-03 10:36:46.478 UTC [27] LOG:  database system was not properly shut down; automatic recovery in progress
2020-02-03 10:36:46.501 UTC [27] LOG:  redo starts at 0/2E5C7B0
2020-02-03 10:36:46.518 UTC [27] LOG:  invalid record length at 0/2E68530: wanted 24, got 0
2020-02-03 10:36:46.518 UTC [27] LOG:  redo done at 0/2E684F8
2020-02-03 10:36:46.619 UTC [1] LOG:  database system is ready to accept connections
```

8.4 その他

(1) データバックアップ

必要に応じて DB データバックアップを取得してください。

DB データは、データフォルダ（支援ナビインストールフォルダ/../dbdata）に存在します。

同フォルダをバックアップすることでデータ断面を確保することが可能です。

(2) URL 監視

必要に応じて URL 監視を実装してください。URL は以下のとおりです。

支援ナビ： http://<サーバの IP アドレス>:8000

支援ナビ管理サイト： http://<サーバの IP アドレス>:8000/admin

(3) ログ

ログは標準出力に出力されます。必要に応じて永続化の手段を講じてください。

また必要に応じてログ監視も実装してください。