ツール実行環境利用ガイド

1. はじめに

本書は、Generative AI FWにおいて、Generative AI FWサーバの利用者向けにツール実行のためのpython実行環境の利用手順について記載したものです。

1.1. 用語定義について

用語定義については「スタートアップマニュアル（概要編）」をご確認ください。

2. 前提条件

- Generative AI FWのセットアップガイドが事前に完了している必要があります。
- 本書に記載のGenerative AI FWサーバ上での手順は全てサーバの管理者ユーザなどの**管理者権限を持つユーザで行う必要**があります。一般ユーザでしかログオンできない環境の場合は以下を実行し、管理者ユーザに昇格させてください。もしくはコマンド実行時に「sudo」を付けてください。

```
sudo -i
```

3. 事前確認

本ツールは起動すると、HTTPSプロトコルにてポート8888でツールの待ち受けを開始します。セキュリティの観点からアクセス元のIPアドレスを限定することを強く推奨します。パスワード認証も必要に応じて有効にしてください。

また初回起動時はHTTPSの証明書を更新する必要があります。詳細は後述の「HTTPSの証明書設定」をご確認ください。

3.1. アクセス元のIPアドレス制限

修正物件適用後、以下の手順でアクセス元のIPアドレスを制限することができます。

1. 後述の「コンテナの停止」の手順に従い、コンテナを停止します。
2. /opt/nec/genai_tool/share/scripts/genai_tool.envを開きます。

```
vi /opt/nec/genai_tool/share/scripts/genai_tool.env
```

3. アクセスを許可するIPアドレスを「ALLOWED_IP_ADDRESSES」に記載します。例として、192.168.1.0/24に限定するには以下のように記載してください。値はシングルクォーテーション「’’」で囲う必要があります。IPv6形式の指定には対応しておりません。

```
ALLOWED_IP_ADDRESSES='192.168.1.0/24'
```

複数指定する場合は,(半角カンマ)で区切って記載します。例として、192.168.1.100/32と192.168.1.101/32に限定する記載は以下の通りです。

```
ALLOWED_IP_ADDRESSES='192.168.1.100/32,192.168.1.101/32'
```

IPアドレス制限をしない場合は空文字を指定します。

```
ALLOWED_IP_ADDRESSES=''
```

4. 後述の「サーバ起動」の手順に従い、コンテナを起動します。
5. 後述の「Jupyter Notebookの利用」の手順に従い、許可したIPアドレスは利用でき、許可していないIPアドレスは利用できないことを確認してください。

3.2. パスワード認証を有効化する場合

修正物件適用後、以下の手順でJupyter Notebookのパスワード認証を有効化することができます。パスワード変更時も同様の手順を実施してください。

- 設定したパスワードを後から確認する方法はありません。設定したパスワードは忘れないように保管することを推奨します。

1. 後述の「サーバ起動」手順実施後、jupyter_containerが起動するのを待ちます。
2. 起動中のjupyter_containerコンテナに入ります。

```
podman exec -it jupyter_container bash
```

3. パスワードを生成します。

```
jupyter notebook password
```

設定したいパスワードを入力します。

```
Enter password:
Verify password:
```

設定後、パスワードのハッシュ値がjsonファイルとして出力されます。

```
[NotebookPasswordApp] Wrote hashed password to /root/.jupyter/jupyter_server_config.jso
```

4. 出力されたjsonファイル中身を表示して、パスワードのハッシュ値をコピーします。

```
cat /root/.jupyter/jupyter_server_config.json
```

ファイルの中身を表示すると以下のように表示されます。赤字の個所をコピーしてください。

```
{
  "NotebookApp": {
    "password": "argon2:$argon2id$v=19$m=10240,t=10,p=XXXXX"
```

```
    }
}
```

5. コンテナから抜けます。

```
exit
```

6. 設定ファイル（/opt/nec/genai_tool/share/scripts/genai_tool.env）を開きます。

```
vi /opt/nec/genai_tool/share/scripts/genai_tool.env
```

7. 「JUPYTER_HASHED_PASSWORD」にコピーした文字列を記載します。

```
JUPYTER_HASHED_PASSWORD='<コピーした文字列>'
```

   上記の場合の記載例は以下になります。※特殊文字を含むためシングルクォーテーション「''」で囲う必要があります。

```
JUPYTER_HASHED_PASSWORD='argon2:$argon2id$v=19$m=10240,t=10,p=XXXXX'
```

8. jupyter_containerを再起動します。後述の「コンテナの停止」→「コンテナの開始」で実施し、再起動してください。
9. 後述の「Jupyter Notebookの利用」の手順に従い、設定したパスワードでログインできることを確認してください。

3.3. パスワード認証を無効化する場合

1. 設定ファイル（/opt/nec/genai_tool/share/scripts/genai_tool.env）を開きます。

```
vi /opt/nec/genai_tool/share/scripts/genai_tool.env
```

2. 「JUPYTER_HASHED_PASSWORD」に空文字を記載します。

```
JUPYTER_HASHED_PASSWORD=''
```

3. jupyter_containerを再起動します。後述の「コンテナの停止」→「コンテナの開始」で実施し、再起動してください。
4. 後述の「Jupyter Notebookの利用」の手順に従い、設定したパスワード認証なしでログインできることを確認してください。

3.4. HTTPSの証明書設定

HTTPSで使用する証明書を設定する必要があります。自己証明書か、正規の証明書など自前の証明書を使用するかで手順が異なります。証明書の有効期限が切れた場合も同様の手順で更新してください。

3.4.1. 自己署名証明書を使用する場合

1. コンテナが起動している場合は停止します。

```
podman stop jupyter_container
```

2. 自己署名証明書作成スクリプトを実行します。

```
cd /opt/nec/genai_tool/share/scripts
bash create_certs.sh
```

3. 確認メッセージが表示されます。「y」を入力してください。

```
Self-signed certificate will be created in the directory '/opt/nec/genai_tool/share/cer
```

4. 証明書ファイルと秘密鍵ファイルを上書きされる旨の確認メッセージが表示されます。上書きし、更新したい場合はそれぞれ「y」を入力してください。

```
Certificate file '/opt/nec/genai_tool/share/certs/self-signed-genai-tool.pem' already e
Private key file '/opt/nec/genai_tool/share/certs/self-signed-genai-tool.key' already e
```

5. 成功したら以下が表示されます。

```
Successfully issued a self-signed certificate.
```

6. /opt/nec/genai_tool/share/scripts/genai_tool.envの内容を確認します。

```
nl /opt/nec/genai_tool/share/scripts/genai_tool.env
```

7. 以下の通り設定されていれば編集不要です。

```
GENAI_TOOL_CERTIFICATE_FILE=self-signed-genai-tool.pem
GENAI_TOOL_CERTIFICATE_KEY=self-signed-genai-tool.key
```

8. 上記設定と異なる場合は以下コマンドで/opt/nec/genai_tool/share/scripts/genai_tool.envを開き、内容を編集します。

```
vi /opt/nec/genai_tool/share/scripts/genai_tool.env
```

9. コンテナを起動するスクリプトを実行します。

```
cd /opt/nec/genai_tool/share/scripts
bash start_container.sh
```

10. コンテナが正常に起動できているか確認します。

```
podman ps | grep jupyter_container
```

3.4.2. 自前の証明書を利用する場合

1. コンテナが起動している場合は停止します。

```
podman stop jupyter_container
```

2. 証明書ファイル（.crt / .pem）および秘密鍵（.pem）を、/opt/nec/genai_tool/share/certs ディレクトリ直下に配置します。
3. /opt/nec/genai_tool/share/scripts/genai_tool.env を以下のように編集します。

```
GENAI_TOOL_CERTIFICATE_FILE=<証明書ファイル名>
GENAI_TOOL_CERTIFICATE_KEY=<秘密鍵ファイル名>
```

4. コンテナを起動するスクリプトを実行します。

```
cd /opt/nec/genai_tool/share/scripts
bash start_container.sh
```

5. コンテナが正常に起動できているか確認します。

```
podman ps | grep jupyter_container
```

4. 起動・停止

4.1. サーバ起動

サーバ起動時にはコンテナが停止しているため、手動で起動させる必要があります。

1. サーバにログオンしたのち、以下を実行してください。

```
cd /opt/nec/genai_tool/share/scripts
bash start_container.sh
```

2. コンテナが正常に起動できているか確認します。

```
podman ps | grep jupyter_container
```

podman ps では以下に示すコンテナイメージ(IMAGE列、NAMES列)のSTATUSが実行中（Up XX minutesなど）になっていることを確認してください（時間表記は経過時間によって異なります）。IMAGE列の数字については動作環境によって一致しないことがあります。

状況によって結果が表示されるまで時間がかかることがあります。

IMAGE列	NAMES列
localhost/image_tuning:1.0	jupyter_container

4.2. サーバ停止

特に考慮することはありません。シャットダウン時にサービスは停止するため各コンテナが自動的に停止します。

4.3. コンテナの開始

「サーバ起動」手順と同様です。

4.4. コンテナの停止

```
podman stop jupyter_container
```

5. バックアップ・リストア

5.1. バックアップ・リストアの基本方針

バックアップをリストアする際には、サーバを工場出荷状態に戻し、そのうえで、バックアップデータを上書きしてリストアする方法を基本方針とします。

- バックアップ・リストア中はツール実行環境が停止します。停止せずに行うことには対応しておりません。

バックアップ対象は「/opt/nec/genai_tool/share」配下にあるファイル・フォルダ全てです。

- ツール実行環境で使用するデータは必ず全てバックアップ対象のフォルダに格納してください。バックアップ対象以外に置いたデータは復元されません。
- バックアップ・リストアではバックアップ対象配下のファイル・フォルダのデータを復元するのみです。先述の「事前確認」のセキュリティ対策の設定は復元されます。

5.2. バックアップ

1. バックアップディレクトリを作成します。任意の場所を指定してください。なお、バックアップではバックアップディレクトリが存在しない場合、自動で作成するため本手順は必須ではありません。

```
cd /opt/nec/genai_tool/
mkdir -p backup/2025XXXX
```

2. バックアップ用のスクリプトを実行します。スクリプト実行時の引数には作成したバックアップディレクトリを指定します。バックアップには時間を要しますので実行後しばらくお待ちください。

```
cd /opt/nec/genai_tool/operation
bash genai_tool_backup.sh ../backup/2025XXXX
```

2. backup succeeded. が表示されればバックアップ完了となります。

```
Backup /opt/nec/genai_tool/share to <バックアップファイルパス> start
Archive saved to: <バックアップファイルパス>
backup succeeded.
```

5.3. リストア

1. リストア用のスクリプトを実行します。リストアには時間を要しますので実行後しばらくお待ちください。

- リストア時には バックアップ実行時に Archive saved to: で表示されるファイルパスを指定してください。

```
cd /opt/nec/genai_tool/operation
bash genai_tool_restore.sh <バックアップファイルパス>
```

2. restore succeeded. が表示されればリストア完了となります。

```
Stop the jupyter_container container.
Restore <バックアップファイルパス> to /opt/nec/genai_tool/share start
Start the jupyter_container container.
restore succeeded.
```

リストアではバックアップファイルを削除しません。リストア後に正常な動作を確認した後、不要になったバックアップファイルの削除を検討してください。

6. Generative AI FWのアクセス先の変更した場合

Generative AI FWのアクセス先(GENAI_DOMAIN)を変更した後に、ツールを使用する場合は起動中のコンテナを一度削除し、再度開始させる必要があります。

1. サーバを再起動します。
2. 上述の「サーバ起動」の手順を実施します。

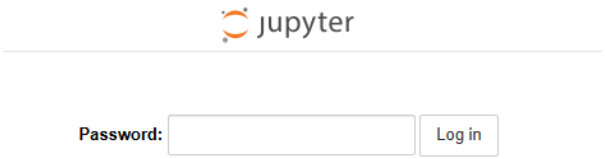
7. Jupyter Notebookの利用

Jupyter Notebookへのアクセス手順を記載します。

1. Webブラウザを起動し、以下にアクセスします。

```
https://<サーバIP>:8888/tree?
```

2. パスワード認証を有効化している場合は以下に作成したパスワードを入力し Log in ボタンを押します。有効化していない場合は本手順は不要です。



3. 成功すると以下が表示されます。

8. サンプルプログラムの実行

Jupyter Notebookからサンプルプログラムを実行します。

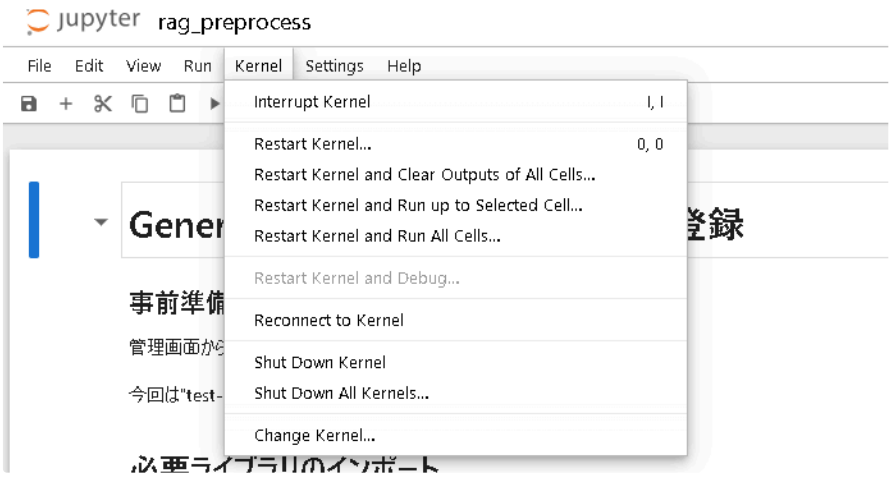
1. Generative AI FWの管理ポータル画面にアクセスしログインします。管理ポータル画面へのログイン手順は「正常性確認ガイド」の「管理ポータル画面の確認」をご確認ください。
2. 左側のメニューからインデックスを選択、インデックス一覧の追加ボタンを押します。
3. サンプルプログラム用の以下の値を設定し、追加ボタンを押します。

> インデックス名：test-rag-index
>
> 所属グループ選択：ALL USERS GROUP （システムユーザであるAPI USERが所属するグループを選択する必要があります）
>
> インデックスの説明：（任意）

4. Jupyter Notebookにアクセスします。アクセス方法は前述の「Jupyter Notebookの利用」を確認してください。
5. rag_preprocess.ipynbをクリックし、内容を編集して必要な設定を記載します。最低限変更が必要な値は以下の通りです。

変数	設定値
GEN_AI_API_BASE_URL	https://<Generative AI FWサーバのドメイン名>/
GEN_AI_API_KEY	APIキー
INDEX_NAME	上記で作成したインデックス名（test-rag-index）
MODEL	LLMのモデル名。Generative AI FWで利用可能なLLMのモデルを指定してください。例えばcotomi v3の場合はcotomi-v3.0になります。

6. rag_preprocess.ipynbの「Kernel」メニューの「Restart Kernal and Run All Cells…」を押して実行します。



8.1. proxy対応

本手順はGenerative AI FWサーバからインターネットを通じて外部通信する際にproxyや証明書が必要な環境で使用する場合の手順を記載します。本対応が必要でない場合は手順の実施は必要ありません。

proxy設定が必要な場合、Jupyter Notebookのプログラムにproxyに関する環境変数(http_proxy、https_proxy)を記載してください。

サンプルプログラム（rag_preprocess.ipynb）の場合は以下のように記載します。

9. 注意事項

実行環境でプログラムを作成、実行する時の注意事項について記載します。

- 実行環境で実行するプログラムでGenerative AI FWのAPIにアクセスする場合（サンプルプログラムを含む）、Generative AI FWサーバからGenerative AI FWで公開しているドメインに対してアクセスできるようにする必要があります。
- Generative AI FWではHTTPSに既定では自己証明書を使用しています。そのためAPI利用時に考慮が必要です。詳細は「スタートアップマニュアル（導入準備編）」をご確認ください。
- Jupyter Notebookは認証なしで使用できます。そのためセキュリティの観点から「事前確認」に示す制限を実施していただくことを強く推奨します。また、プログラムで使用するAPIキーも定期的に更新することを推奨します。
- jupyter_containerはサーバ再起動の度に削除されます。必要なデータをコンテナ内に保存しないようサンプルプログラムを書く際は注意してください。