Sanitizer 技術文書 Ver.4.2対応版

Hiro KAWAGUCHI Laboratory

2023 年 12 月 28 日

目次

第1章　サニタイザーのシステム構成

1.1 サニタイザーのモジュール関係図

サニタイザーを構成するモジュールの関係を示します。

1.1.1 サニタイザー単体版の場合

サニタイザー単体版の構成は 図 1.1 のとおり。

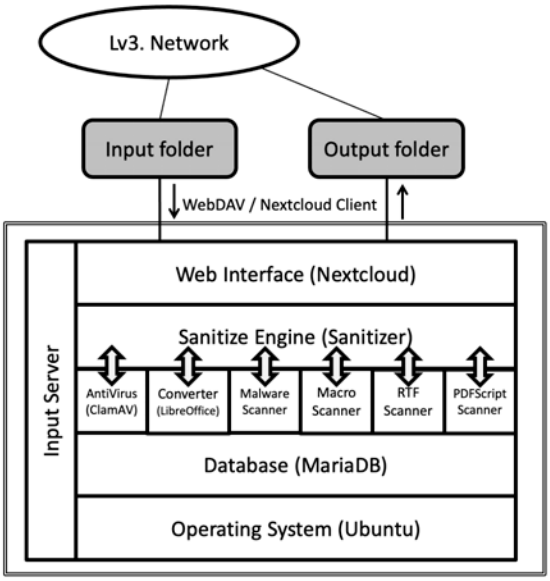

図 1.1: サニタイザー単体版の構成

サニタイザーと他のソリューション（メール無害化、ファイル受け渡し）を組み合わせる場合もこの構成です。

サニタイズのためのファイル受け渡しの接点として、input フォルダ、output フォルダへのアクセスを提供します。具体的な方法は後述します。

1.1.2 ネットワーク間でファイル受け渡しをする場合（サニタイザープロ）

サニタイザーをネットワーク間のファイル受け渡しに用いる場合の構成は 図 1.2 のとおり。

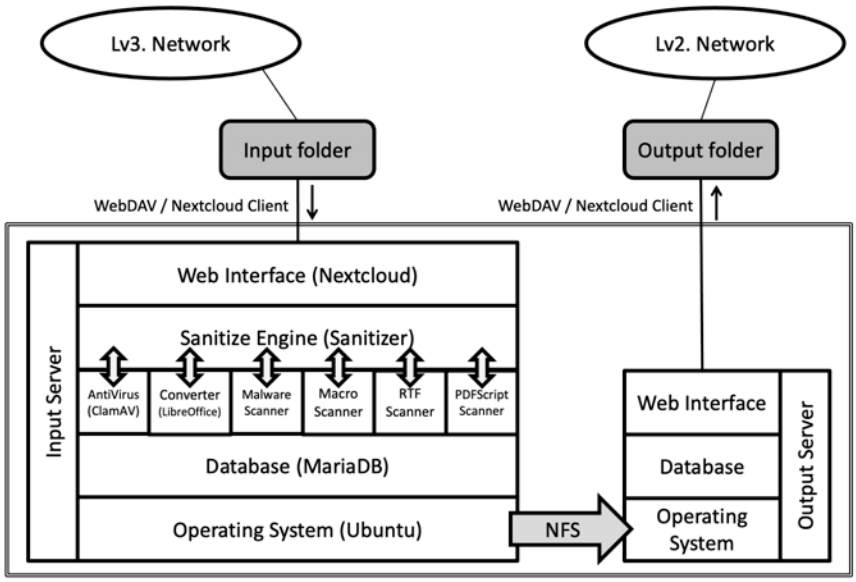

図 1.2: サニタイザープロの構成

サニタイズのためのファイル受け渡しの接点として、input フォルダ、output フォルダへのアクセスを提供するところなど、標準機能の場合と基本的な構成は同じです。ただし、内部で input 処理を行うサーバと output 処理を行うサーバを別々に起動することになります。これらのサーバは別のネットワークセグメントへのインターフェースを持ちます。

さらにこれらのサーバ間で一方通行のファイル受け渡しを実現するために、別途設定した内部ネットワーク経由でフォルダマウントを行います。

1.2 サニタイザーを構成するモジュール

サニタイザーはいくつかの機能を担うモジュール群で構成されています。ただし、この技術文書における「モジュール」とは、一般的なソフトウェアモジュールとは異なり、もっと広義で扱っています。

1.2.1 オペレーティングシステム

サニタイザー Ver.3 系は、Ubuntu 18.04 ESM を使用しています。サニタイザー Ver.4 系は、Ubuntu 22.04 LTS を使用しています。サーバ向けの最小構成セットアップを行っており、必要に応じて apt でソフトウェアをセットアップしています。

1.2.2 Web インターフェース

サニタイザー Ver.3 系、Ver.4 系とも Nextcloud を使用しています。

1.2.3 データベース

サニタイザー Ver.3 系は、MariaDB 10.3 を、サニタイザー Ver.4 系は、MariaDB 10.6 を使用しています。apt でセットアップをしています。

1.2.4 サニタイズエンジン

サニタイズエンジンは独自に開発しています。このプロセスを動作する過程で次のモジュールを呼び出しています。

アンチウイルス
ClamAV を使用しています。apt でセットアップしています。

ファイルコンバート
一部の Office 系ファイル、ODF ファイル、PDF ファイルについて、LibreOffice を使用しています。

Office ファイルマルウェア検知
OfficeMalScanner を使用しています。また、このソフトウェアを動作させるために、Wine を使用しています。

Office ファイルマクロ検知
MacroRaptor を使用しています。MacroRaptor を動作させるために、Python 3.10 を使用しています。

RTF ファイルマルウェア検知
RTFScan を使用しています。また、このソフトウェアを動作させるために、Wine を使用しています。

PDF マクロ検知
PDFMiner を使用しています。これらを動作させるために、Python 2.7 を使用しています。

ステガノグラフィ検知
StegExpose と stegdetect を使用しています。StegExpose を動作させるために、OpenJRE 11 を使用しています。StegExpose では PNG と BMP 形式を、stegdetect では JPEG 形式の画像ファイルを処理しています。

ファイル構造分析
o-checker を使用しています。o-checker を動作させるために、Python 2.7 を使用しています。

1.3 サニタイザー単体版のセットアップ

サニタイザー単体版のセットアップは、サニタイザーの構成を理解する意味でも必要なスキルです。

その手順を示します。

1.3.1 動作環境

64 ビット OS が動作するパソコンあるいはサーバが必要です。ここでは、WindowsPC（64 ビット版）あるいは Mac 上で動作させる手順を示しています。

1.3.2 セットアップの手順

Oracle VirtualBox のインストール

サニタイザーは仮想アプライアンスとして提供していますので、動作させるためには仮想化環境が必要です。ここでは、Oracle VirtualBox（無償で利用できます）を使って動作させることにします。

VMWare ESXi や Hyper-V、Nutanix（Prism）でのデプロイも実績があります。OVA ファイルを vmdk や vhd にコンバートすることでデプロイ可能となるようです。

Virtualbox のサイトに行き、VirtualBox と ExtensionPack をダウンロードする。

https://www.virtualbox.org/wiki/Downloads

ダウンロードしたものをインストールする。（インストールの手順は省略します）

Sanitizer 仮想アプライアンス（OVA）のダウンロード

下記の URL にアクセスし、Sanitizer の OVA ファイルをダウンロードする。（最新版のダウンロード URL は個別にお知らせします）

サニタイザー Ver.3 系 のダウンロード URL

https://storage.googleapis.com/sanitizerovafiles/sanitizer/Sanitizer_3.7.0.ova

サニタイザー Ver.4 系 のダウンロード URL

https://storage.googleapis.com/sanitizerovafiles/sanitizer/Sanitizer_4.2.0.ova

OVA ファイルのインポート

VirtualBox を起動し、メニューから「仮想アプライアンスのインポート」を選びます。（図 1.3）（下図は Mac の旧版ですが、Windows 版でも同様のメニューがあります）

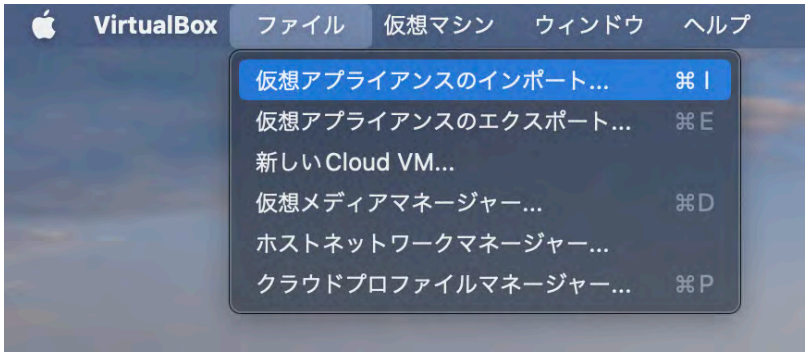

図 1.3: OVA ファイルのインポート

手順に従ってインポート作業を続けます。設定はデフォルトのままで結構です。

VirtualBox の設定

インポートされた Sanitizer サーバのネットワーク設定を確認します。

メイン画面の「設定」をクリック。（図 1.4）

図 1.4: 「設定」メニュー

ネットワーク設定（アダプター１）　→　NAT であることを確認します。（図 1.5）

図 1.5: ネットワーク設定（アダプター１）

ネットワーク設定（アダプター２）　→　ホストオンリーアダプターで、vboxnet0 であることを確認します。（図 1.6）

図 1.6: ネットワーク設定（アダプター２）

もし vboxnet0 が選択できない場合は、メインメニューの「ホストネットワークマネージャー」からネットワークアダプターの設定を行います。（図 1.7）

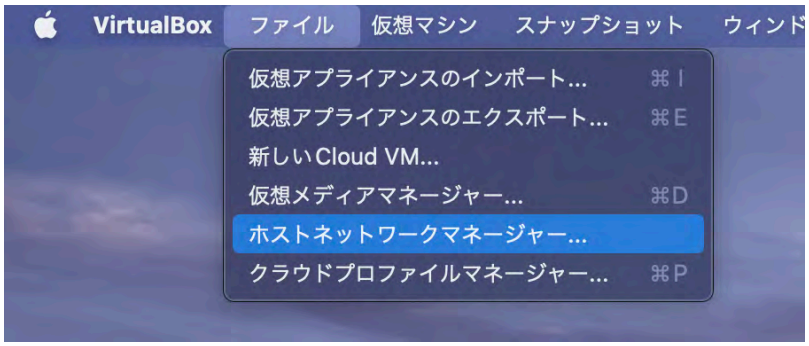

図 1.7: メインメニュー「ホストネットワークマネージャー」

ネットワーク　→　ホストオンリーネットワークで vboxnet0 を追加設定（図 1.8）

IPv4 アドレスは必ず `192.168.56.1` IPv4 ネットマスクは必ず `255.255.255.0` とします。

DHCP サーバ設定画面は 図 1.9 のとおり。

上記の設定で、これまで VirtualBox 上で IP アドレスの自動割当を行っていない場合は、192.168.56.101 が割り振られると思います。（これまで他の仮想マシンをデプロイし、DHCP でアドレスを割り振られている場合は、別の IP アドレスになる場合もあります）

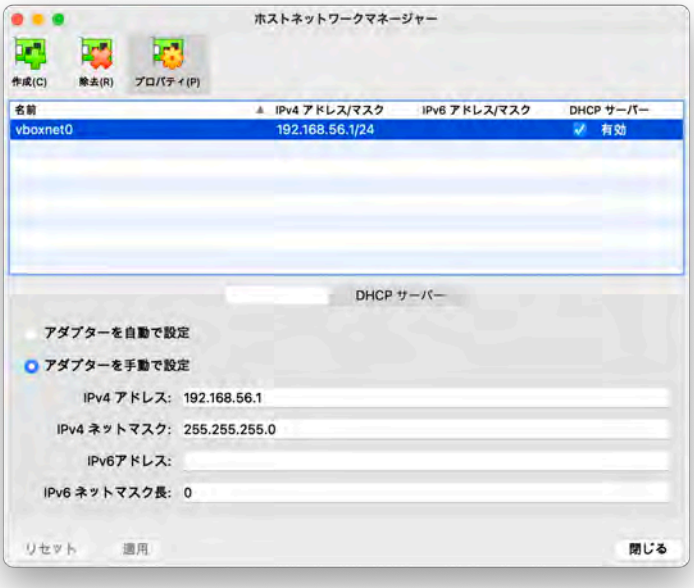

図 1.8: vboxnet0 を追加設定

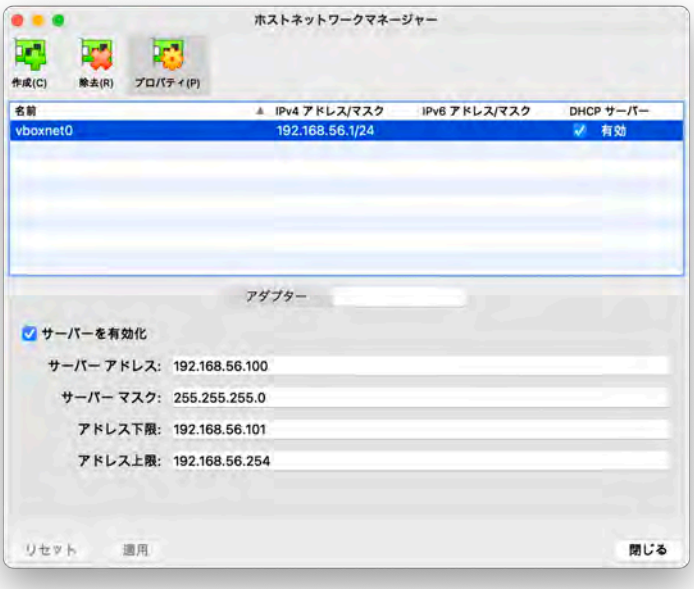

図 1.9: DHCP サーバの設定

Sanitizer サーバの起動

Sanitizer の仮想サーバの電顕を入れます。（起動ボタンをクリックする）（図 1.10）

図 1.10: 「起動」メニュー

Sanitizer の管理コンソール画面が開き、ログイン状態となります。これはそのまま放置（最小化）して構いません。

Sanitizer へのアクセス

ここまでの状態で、サニタイザー Ver.3 の場合は同じパソコンのブラウザから、`http://192.168.56.101/sanitizer/` にアクセスします。

Sanitizer のログイン画面が起動します。

ログイン名	パスワード
sanitizer	sanitizer

Sanitizer のポリシー設定

設定のためのログイン名とパスワードは次のとおり。

ログイン名	パスワード
admin	@dm!n

policy フォルダの中に、`sanitizer.conf` というファイルがあり、このファイルの内容を修正します。`sanitizer.conf` ファイルそのものに詳細なコメントを加えてありますので、設定の詳細はそちらをご覧ください。

あるいは、`sanitizer.conf` ファイルの実体は `/var/local/sanitizer/policy/sanitizer.conf` にありますので、直接エディタ等で修正することも可能です。修正した内容は即時反映されます。

1.4 サニタイザープロのセットアップ

導入の構成として最もポピュラーなサニタイザープロのセットアップ手順です。その手順を示します。

併せて、第７章「ネットワーク間のファイル受け渡し」もご覧ください。

1.4.1 動作環境

ハードウェア

64 ビット OS が動作するパソコンあるいはサーバが必要です。また、これらのパソコン、サーバは異なるネットワークに接続するために NIC（ネットワークポート）が２つ必要です。

ここでは、WindowsPC（64 ビット版）あるいは Mac 上で動作させる手順を示しています。

ネットワーク環境

ネットワーク上のサニタイザーの配置は 図 1.11 のようになります。

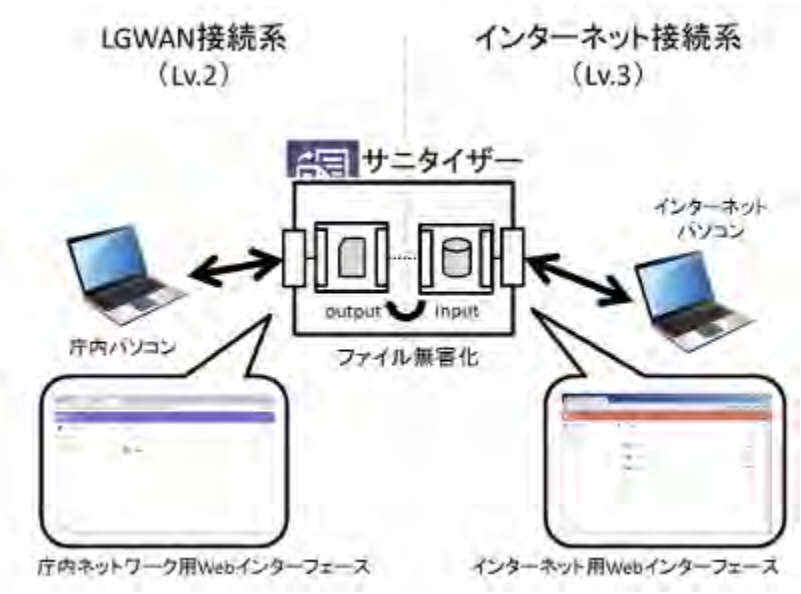

図 1.11: サニタイザープロのネットワーク上の配置

ここでは、

インターネット接続系（Level3）の IP アドレスを `192.168.100.200/255.255.255.0`

LGWAN 接続系（Level2）の IP アドレスを `10.0.100.200/255.255.255.0`

として設定する手順を示しています。

パソコン、サーバの NIC の設定が事前に必要となります。実際に評価版を設定する際には、導入先のネットワーク環境を確認して下さい。

1.4.2 セットアップの手順

Oracle VirtualBox のインストール

サニタイザーは仮想アプライアンスとして提供していますので、動作させるためには仮想化環境が必要です。ここでは、Oracle VirtualBox（無償で利用できます）を使って動作させることにします。

VMWare ESXi や Hyper-V、Nutanix（Prism）でのデプロイも実績があります。OVA ファイルを vmdk や vhd にコンバートすることでデプロイ可能となるようです。

Oracle のサイトに行き、VirtualBox と ExtensionPack をダウンロードする。

https://www.virtualbox.org/wiki/Downloads

ダウンロードしたものをインストールする。（インストールの手順は省略します）

Sanitizer 仮想アプライアンス（OVA）のダウンロード

下記の URL にアクセスし、Sanitizer の OVA ファイルをダウンロードします。

サニタイザープロ Ver.3 系 のダウンロード URL

Input Server
: https://storage.googleapis.com/sanitizerovafiles/sanitizer/SanitizerPro_input_3.7.0.ova

Output Server
: https://storage.googleapis.com/sanitizerovafiles/sanitizer/SanitizerPro_output_3.7.0.ova

サニタイザープロ Ver.4 系 のダウンロード URL

Input Server
: https://storage.googleapis.com/sanitizerovafiles/sanitizer/SanitizerPro_input_4.2.0.ova

Output Server
: https://storage.googleapis.com/sanitizerovafiles/sanitizer/SanitizerPro_output_4.2.0.ova

OVA ファイルのインポート

VirtualBox を起動し、メニューから「仮想アプラインアンスのインポート」を選びます。（図 1.12）（下図は Mac 版ですが、Windows 版でも同様のメニューがあります）

手順に従ってインポート作業を続ける。設定はデフォルトのままで結構です。

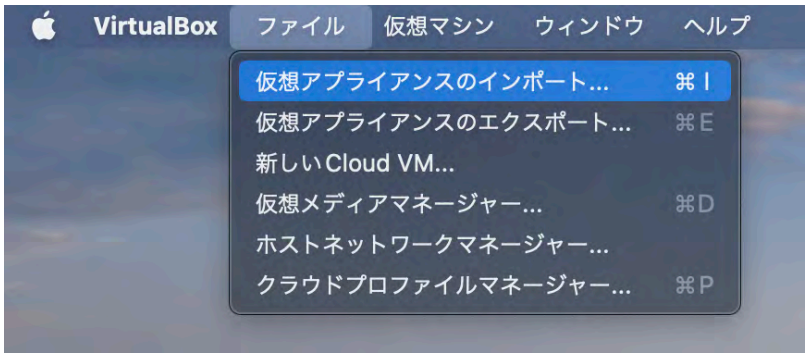

図 1.12: OVA ファイルのインポート

接続端末の hosts ファイルの設定

サニタイザー Ver.3 では、Input サーバを ”sanitizer3public” 、Output サーバを ”sanitizer3private” というサーバ名に割り当てています。Web インターフェースやフォルダ同期を行う際に、IP アドレスではなくサーバ名で接続する必要があるため、サニタイザーに接続する端末の hosts ファイルに次の設定を加えてください。(hosts ファイルの設定方法についてはここでは説明しません)

hosts ファイルの設定（抜粋）

```
192.168.100.200          sanitizer3public
10.0.100.200             sanitizer3private
```

なお、このサーバ名は Nextcloud の `config.php` ファイル中の `trusted_domains` にも設定してください。

サニタイザー Ver.4 では、Input サーバを ”sanitizer4public” 、Output サーバを ”sanitizer4private” というサーバ名に割り当てています。Web インターフェースやフォルダ同期を行う際に、IP アドレスではなくサーバ名で接続する必要があるため、サニタイザーに接続する端末の hosts ファイルに次の設定を加えてください。(hosts ファイルの設定方法についてはここでは説明しません)

hosts ファイルの設定（抜粋）

```
192.168.100.200          sanitizer4public
10.0.100.200             sanitizer4private
```

なお、このサーバ名は Nextcloud の `config.php` ファイル中の `trusted_domains` にも設定してください。

VirtualBox の設定

Input Server

インポートされた Sanitizer サーバのネットワーク設定を確認します。

メイン画面の「設定」をクリック。(図 1.13)

ネットワーク設定（アダプター１） → NAT であることを確認します。(図 1.14)

アダプター１の「高度」をクリックし、ポートフォワーディングの設定を行います。(図 1.15)

図 1.13: 「設定」メニュー

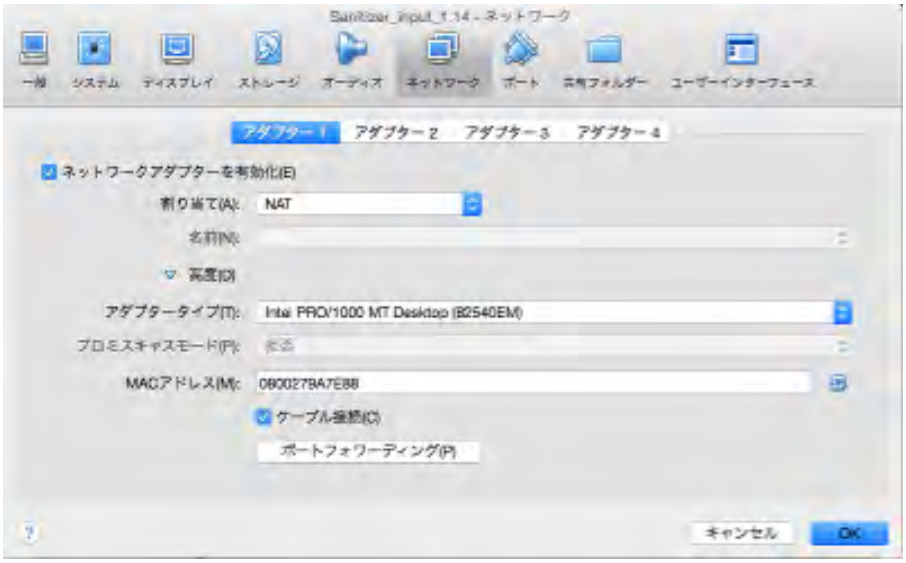

図 1.14: ネットワーク設定（アダプター１）

ここでは、前提条件として示した `192.168.100.200` ポート `8080` を設定します。（ネットワークの設定は環境に応じて適宜変更してください）

図 1.15: ポートフォワーディング

ネットワーク設定（アダプター２）　→　内部ネットワークで、intnet であることを確認します。（図 1.16）

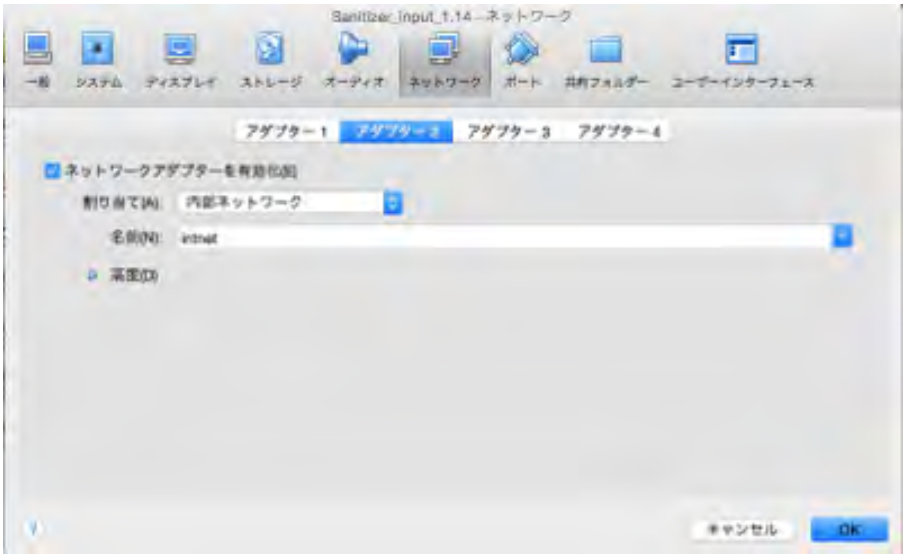

図 1.16: ネットワーク設定（アダプター２）

Output Server

インポートされた Sanitizer サーバのネットワーク設定を確認します。

メイン画面の「設定」をクリック。（図 1.17）

図 1.17: 「設定」メニュー

ネットワーク設定（アダプター１）　→　NAT であることを確認します。（図 1.18）

アダプター１の「高度」をクリックし、ポートフォワーディングの設定を行います。（図 1.19）

ここでは、前提条件として示した `10.0.100.200` ポート `8080` を設定します。（ネットワークの設定は環境に応じて適宜変更してください）

ネットワーク設定（アダプター２）　→　内部ネットワークで、intnet であることを確認します。（図 1.20）

Sanitizer サーバの起動

Sanitizer の仮想サーバの電顕を入れます。（起動ボタンをクリックする）（図 1.21）

Sanitizer の管理コンソール画面が開き、ログイン状態となります。これはそのまま放置（最小化）して構いません。

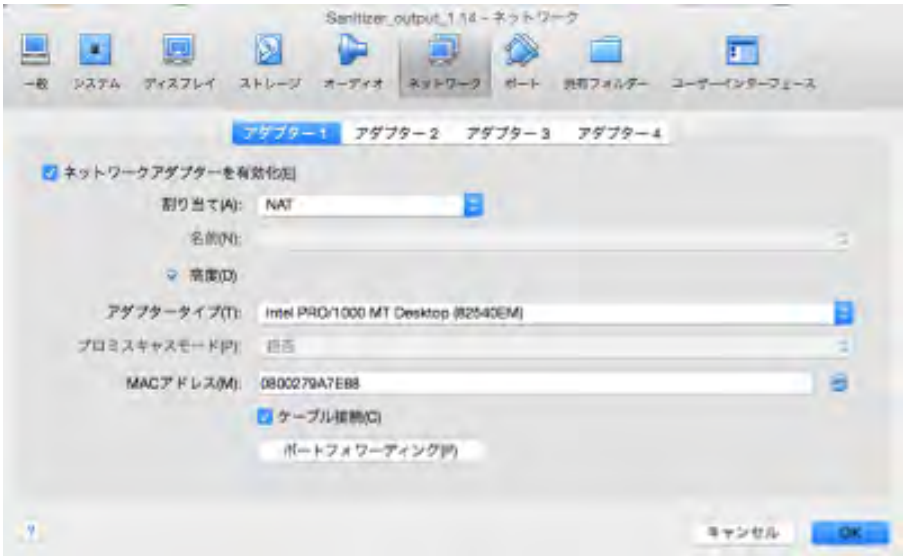

図 1.18: ネットワーク設定（アダプター１）

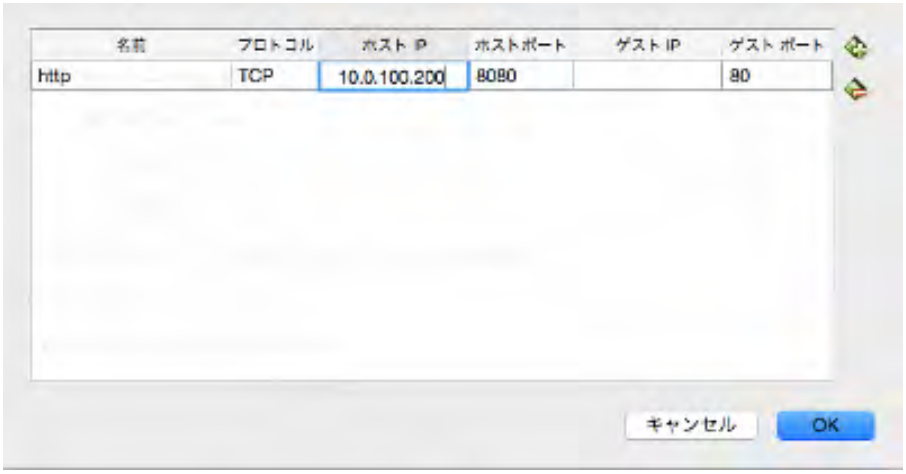

図 1.19: ポートフォワーディング

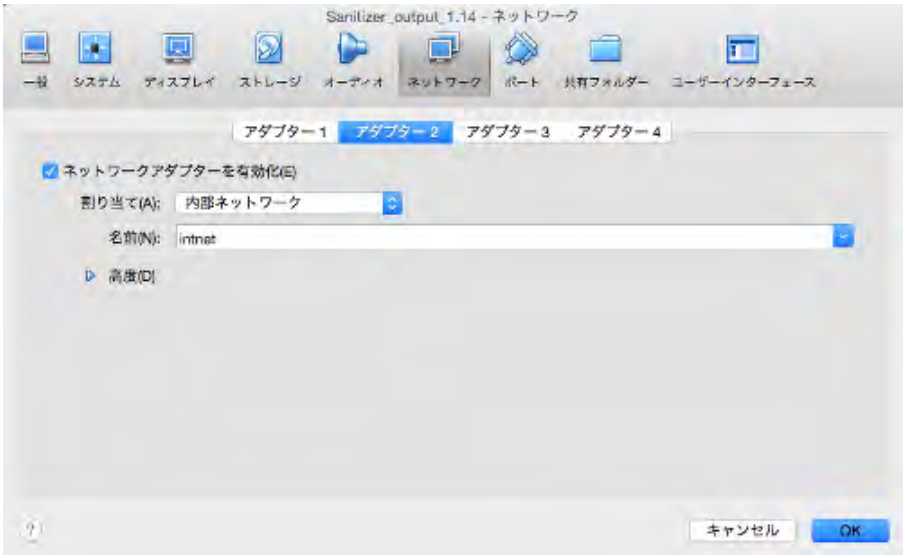

図 1.20: ネットワーク設定（アダプター２）

図 1.21: 「起動」メニュー

Sanitizer へのアクセス

Input サーバ（sanitizer3public / sanitizer4public）

インターネット接続系上のパソコンのブラウザから、http://sanitizer3public:8080/sanitizer/ あるいは http://sanitizer4public:8080/sanitizer/ にアクセスします。

Sanitizer のログイン画面が起動します。

ログイン名	パスワード
sanitizer	sanitizer

Output サーバ（sanitizer3private / sanitizer4private）

インターネット接続系上のパソコンのブラウザから、http://sanitizer4private:8080/sanitizer/ あるいは http://sanitizer3private:8080/sanitizer/ にアクセスします。

Sanitizer のログイン画面が起動します。

ログイン名	パスワード
sanitizer	sanitizer

Sanitizer のポリシー設定

設定のためのログイン名とパスワードは次のとおり。

ログイン名	パスワード
admin	@dm!n

policy フォルダの中に、sanitizer.conf というファイルがあり、このファイルの内容を修正します。sanitizer.conf ファイルそのものに詳細なコメントを加えてありますので、設定の詳細はそちらをご覧ください。

あるいは、sanitizer.conf ファイルの実体は /var/local/sanitizer/policy/sanitizer.conf にありますので、直接エディタ等で修正することも可能です。修正した内容は即時反映されます。

第2章　オペレーティングシステム

2.1 概要

サニタイザー Ver.3 系は、Ubuntu 18.04 ESM を使用しています。サニタイザー Ver.4 系は、Ubuntu 22.04 LTS を使用しています。サーバ向けの最小構成セットアップを行っており、必要に応じて apt でソフトウェアをセットアップしています。

2.1.1 アカウント情報

サニタイザーの保守に必要なアカウント情報は下記のとおり。

アカウント名	初期パスワード	説明
root		直接 root ではログインできません
sanitizer	S@n!tizerJapan#2016	sudo を実行する際もこのパスワードです

注釈: 初期パスワードは必ず変更してください

この技術文書は多くのユーザに流通していますので、初期パスワードを変更せずに放置しておくと、不正アクセスやサイバー攻撃のリスクを高めることになります。

他のシステムで使われていない、推測されにくいパスワードを設定してください。パスワードの設定方法は Ubuntu の解説書やサイト等をご覧ください。

2.1.2 ディレクトリ情報

サニタイザーで主に用いられるディレクトリ情報を示します。下記はサニタイザーで処理されるファイルの実体があるディレクトリ　/var/local/sanitizer 配下の様子です。:

```
sanitizer@sanitizer4:/var/local/sanitizer$ ls -al
合計 40
drwxr-sr-x 10 www-data www-data 4096  6月 25 04:06 .
drwxrwsr-x  3 root     staff    4096  6月 25 00:34 ..
drwxrwx--- 10 www-data www-data 4096  7月 23 13:11 data
drwxr-sr-x  6 www-data www-data 4096  6月 25 04:06 hiro
```

(次のページに続く)

(前のページからの続き)

```
drwxr-srwx  2 www-data www-data 4096  9月  1 12:55 hold
drwxr-sr-x  2 www-data www-data 4096  9月  1 13:07 input
drwxr-srwx  2 www-data www-data 4096  9月  1 12:55 output
drwxr-srwx  3 www-data www-data 4096  7月 11 18:16 policy
drwxr-sr-x  2 www-data www-data 4096  6月 25 04:05 talk
drwxr-srwx  2 www-data www-data 4096  9月  1 13:03 work
```

data
: Nextcloud（サニタイザーの Web インターフェース）で使用するディレクトリです。特に何もする必要はありません。

hiro
: 利用者独自にフォルダを分ける際に設定する実体ディレクトリの例です。（詳細は後述します）

hold
: Web インターフェースにおいて、hold フォルダにマウントされるディレクトリです。受け渡し時に一時的に保留するファイルはここに保存されます。

  サニタイザーではホールド＆パス機能を使うために、hold ディレクトリが必要ですが、利用ユーザ毎に hold ディレクトリの設定を行う必要はありません。hold ディレクトリは `/var/local/sanitizer/hold` を承認ユーザ全てが共有する仕組みです。

input
: Web インターフェースにおいて、標準ユーザ（sanitizer）が使用する input フォルダにマウントされるディレクトリです。このフォルダを監視してサニタイズ処理が行われます。

output
: Web インターフェースにおいて、標準ユーザ（sanitizer）が使用する output フォルダにマウントされるディレクトリです。サニタイズ処理後のファイルはここに保存されます。

policy
: Web インターフェースにおいて、サニタイズの挙動を制御するための設定ファイルがあるディレクトリです。また、スケルトンファイル（新規にユーザを生成した際に自動的に配置されるファイル）もここで管理しています。

talk
: Web インターフェースにおいて、標準ユーザ（sanitizer）が Talk 機能（チャットや Web 会議機能）を使う場合に使用するディレクトリです（サニタイザー Ver.4 のみ）。詳細は後述します。

work
: Web インターフェースにおいて、標準ユーザ（sanitizer）が使用する work フォルダにマウントされるディレクトリです。アーカイブファイルの解凍先はこのディレクトリです。

また、利用者独自にフォルダを分ける際に設定する、実体ディレクトリの様子は次のとおりです。ここでは `hiro` ユーザを例にしています。:

```
sanitizer@sanitizer4:/var/local/sanitizer/hiro$ ls -al
合計 24
drwxr-sr-x  6 www-data www-data 4096  6月 25 04:06 .
drwxr-sr-x 10 www-data www-data 4096  6月 25 04:06 ..
drwxr-sr-x  2 www-data www-data 4096  6月 25 04:06 input
drwxr-srwx  2 www-data www-data 4096  6月 25 04:06 output
drwxr-sr-x  2 www-data www-data 4096  6月 25 04:06 talk
drwxr-srwx  2 www-data www-data 4096  6月 25 04:06 work
```

input
: Web インターフェースにおいて、hiro が使用する input フォルダにマウントされるディレクトリです。このフォルダを監視してサニタイズ処理が行われます。

output
: Web インターフェースにおいて、hiro が使用する output フォルダにマウントされるディレクトリです。サニタイズ処理後のファイルはここに保存されます。

talk
: Web インターフェースにおいて、hiro が Talk 機能（チャットや Web 会議機能）を使う場合に使用するディレクトリです（サニタイザー Ver.4 のみ）。詳細は後述します。

work
: Web インターフェースにおいて、hiro が使用する work フォルダにマウントされるディレクトリです。アーカイブファイルの解凍先はこのディレクトリです。

2.2 サニタイザーのログ

サニタイザーのサニタイズ処理に関するログは、`/var/log/sanitizer` 配下にあります。

logrotate を使って、標準では月次ログローテーションをしており、12 世代の履歴が保存されます。したがって、１年間分のログが記録されることになります。

2.3 NTP の設定

サニタイザーはファイル受け渡し時にフォルダ間の同期を取る場面があるため、timesyncd による時刻設定を行っています。

設定ファイルは、`/etc/systemd/timesyncd.conf` にあり、NTP サーバは NICT あるいは ubuntu.com にしてあります。導入する際の庁内環境にあわせて設定してください。

2.4 ファイアーウォール設定

サニタイザーが稼働する Ubuntu では初期状態でポートのクローズを行っていません。

導入する環境にあわせて、ufw 等で不要なポートを閉じるようにしてください。

第3章　Webインターフェース

3.1 概要

サニタイザーは、Web インターフェースに Nextcloud を利用しています。

3.2 アカウント情報

サニタイザーの保守に必要なアカウント情報は下記のとおり。

アカウント名	パスワード	説明
admin	@dm!n	Nextcloud の設定変更、サニタイザーの設定変更、ログ閲覧が可能な権限
sanitizer	sanitizer	標準機能で利用可能な一般ユーザ権限
hiro	hiro	個別登録されたユーザ（例示としておいてあります）

3.3 ネットワーク設定

Web インターフェースの設定の中で、導入先の環境にあわせて変更しなければならない部分を示します。

3.3.1 trusted_domains の変更

/var/www/nextcloud/config/config.php を編集し、trusted_domain の記述を、参加しているネットワークアドレスに変更します。

例えば、サニタイザーを http://192.168.56.101/sanitizer でアクセスさせたい場合は、この値を 192.168.56.101 とします。

あるいは、この値を * にすることで、どのようなドメイン（IP アドレス）でもアクセスできるようにできます。

3.4 フォルダマウントの考え方

サニタイザーは、OS レベルのディレクトリ監視を行うことで、サニタイズの処理を自動化しています。そのため、Web インターフェース標準のファイルシステムを利用せずに、OS レベルで生成したディレクトリ（第 2 章にて説明）を外部ストレージ連携機能を使って個別にマウントしています。

標準機能では sanitizer ユーザが利用するフォルダとして、次のディレクトリをマウントしています。

input フォルダ
`/var/local/sanitizer/input`

output フォルダ
`/var/local/sanitizer/output` （サニタイザー単体版の場合のみ）

work フォルダ
`/var/local/sanitizer/work`

さらに、個別ユーザ（この場合はユーザ hiro）が利用するフォルダとして、次のディレクトリをマウントしています。

input フォルダ
`/var/local/sanitizer/hiro/input`

output フォルダ
`/var/local/sanitizer/hiro/output` （サニタイザー単体版の場合のみ）

work フォルダ
`/var/local/sanitizer/hiro/work`

また、admin ユーザが利用するフォルダとして、次のディレクトリをマウントしています。

input フォルダ
`/var/local/sanitizer/input` （sanitizer ユーザと共通）

output フォルダ
`/var/local/sanitizer/output` （sanitizer ユーザと共通）

work フォルダ
`/var/local/sanitizer/work` （sanitizer ユーザと共通）

hold フォルダ
`/var/local/sanitizer/hold` （ホールド＆パス用。詳細は後述します）

policy フォルダ
`/var/local/sanitizer/policy`

log フォルダ
`/var/log/sanitizer`

3.5 アカウントの追加

標準機能では、sanitizer ユーザを用いますが、導入環境によっては利用者独自にサニタイズ用のフォルダを設定しなければならない場合があります。

その場合、OS のコマンドで実体ディレクトリの作成をした後、Web インターフェース上でアカウントを追加する必要があります。

3.5.1 実体ディレクトリの作成

アカウント追加に先立って、OS 上で実体となるディレクトリを作成しておく必要があります。下記はユーザ hiro のディレクトリを作成する場合の例です。

```
$ sudo mkdir -p /var/local/sanitizer/hiro/input /var/local/sanitizer/hiro/output /var/
↪local/sanitizer/hiro/work
$ sudo chown -R www-data:www-data /var/local/sanitizer/hiro
$ sudo chmod o+w /var/local/sanitizer/hiro/output /var/local/sanitizer/hiro/work
```

3.5.2 Web インターフェースによるユーザ追加

admin ユーザでログイン後、管理メニューからユーザを追加します。（図 3.1）

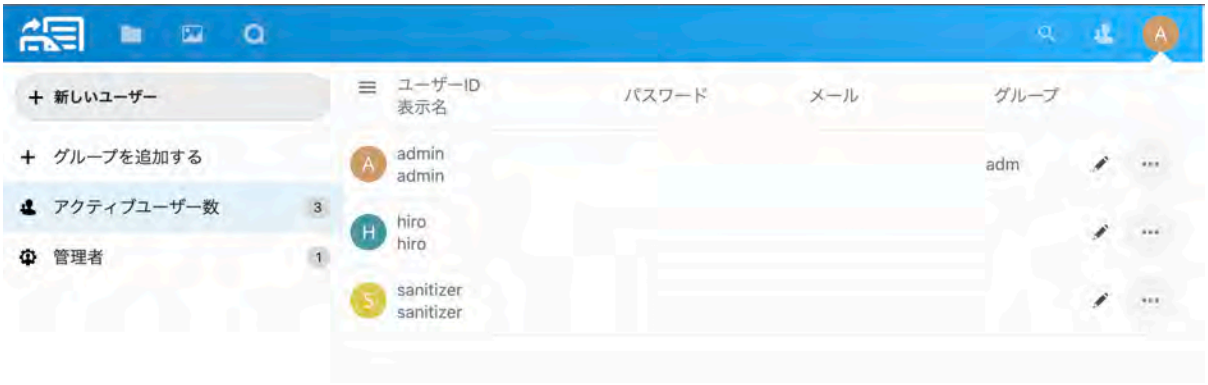

図 3.1: Web インターフェース上のユーザ追加

グループについては、特に設定の必要がなければ、sanitizer ユーザと同じ user を設定してください。

3.5.3 コマンドラインによるユーザ追加

上記と同じ作業はコマンドラインからも行うことができます。（root ユーザでのコマンド実行になります）

```
$ sudo su
# export OC_PASS=hiro; su -s /bin/sh www-data -c 'php /var/www/nextcloud/occ user:add␣
↪--password-from-env --display-name="hiro" --group="user" hiro'
```

複数ユーザを登録する場合は、あらかじめバッチファイルを作成して一括登録することも可能です。

3.5.4 Web インターフェースによる実体ディレクトリのマウント

Web インターフェースの外部ストレージ設定にて、次の実体ディレクトリをマウントします。

図 3.2 はサニタイザー単体版で初期登録されている外部ストレージ設定一覧です。

図 3.2: 実体ディレクトリのマウント

ユーザ hiro の実体ディレクトリをマウントする場合は、

input フォルダ
: ローカルディスクで /var/local/sanitizer/hiro/input をマウントします。対象ユーザは、 当該ログインユーザのみとしてください。

work フォルダ
: ローカルディスクで /var/local/sanitizer/hiro/work をマウントします。対象ユーザは、当該ログインユーザのみとしてください。

output フォルダ（サニタイザー単体版の場合のみ）
: ローカルディスクで /var/local/sanitizer/hiro/output をマウントします。対象ユーザは、 当該ログインユーザのみとしてください。

3.5.5 コマンドラインによる実体ディレクトリのマウント

上記と同じ作業はコマンドラインからも行うことができます。

あらかじめ、マウントさせたい実体ディレクトリの情報を記述した JSON 形式の外部ストレージ設定ファイルを作成しておき、この設定ファイルをインポートする手順となります。

外部ストレージ設定ファイル

設定ファイルは次のような形式となります。

```
[
   {
       "mount_id": 1,
       "mount_point": "\/input",
       "storage": "\\OC\\Files\\Storage\\Local",
       "authentication_type": "null::null",
       "configuration": {
           "datadir": "\/var\/local\/sanitizer\/hiro\/input"
       },
       "options": {
           "encrypt": true,
           "previews": true,
           "enable_sharing": false,
           "filesystem_check_changes": 1,
           "encoding_compatibility": false,
           "readonly": false
       },
       "applicable_users": [
           "hiro"
       ],
       "applicable_groups": []
   },
   {
       "mount_id": 2,
       "mount_point": "\/output",
       "storage": "\\OC\\Files\\Storage\\Local",
       "authentication_type": "null::null",
       "configuration": {
           "datadir": "\/var\/local\/sanitizer\/hiro\/output"
       },
       "options": {
           "encrypt": true,
           "previews": true,
           "enable_sharing": false,
           "filesystem_check_changes": 1,
```

(次のページに続く)

(前のページからの続き)


```
            "encoding_compatibility": false,
            "readonly": false
        },
        "applicable_users": [
            "hiro"
        ],
        "applicable_groups": []
    },
    {
        "mount_id": 3,
        "mount_point": "\/work",
        "storage": "\\OC\\Files\\Storage\\Local",
        "authentication_type": "null::null",
        "configuration": {
            "datadir": "\/var\/local\/sanitizer\/hiro\/work"
        },
        "options": {
            "encrypt": true,
            "previews": true,
            "enable_sharing": false,
            "filesystem_check_changes": 1,
            "encoding_compatibility": false,
            "readonly": false
        },
        "applicable_users": [
            "hiro"
        ],
        "applicable_groups": []
    }
]
```


設定ファイルの中の設定値は次のとおりです。

mount_id
: 任意の値（数値。詳細は後述）

mount_point
: ユーザ毎に表示させるフォルダ名

data_dir
: サニタイザー内の実体ディレクトリ

applicable_users
: 対象となるユーザ名（applicable_groups と OR 条件）

applicable_groups
: 対象となるグループ名（applicable_users と OR 条件）

外部ストレージ設定ファイルの出力（エクスポート）

上記のファイルを最初から手作業で作成するとミスしやすいため、既存の外部ストレージ設定内容を JSON ファイルに出力し、そのファイルを編集することをおすすめします。

既存の外部ストレージ設定内容の出力は次のコマンドで行います。

```
$ sudo -u www-data php /var/www/nextcloud/occ files_external:export > /home/sanitizer/
↪export.json
```

これで、/home/sanitizer ディレクトリ配下に export.json ファイルが出力されます。

なお、設定ファイル中の mount_id は任意の値で差し支え有りません。設定ファイルをインポートする際に、改めて値が設定されます。

外部ストレージ設定ファイルのインポート

設定ファイル（/home/sanitizer/export.json とした場合）をインポートするには次のコマンドで行います。

```
$ sudo -u www-data php /var/www/nextcloud/occ files_external:import /home/sanitizer/
↪export.json
```

3.6 スケルトンフォルダ

Web インターフェースにおいて新たにユーザを作成した場合、各ユーザフォルダに初期設定されるフォルダ、ファイルを定義することができます。

スケルトンフォルダは /var/local/sanitizer/policy/skeleton です。

標準では do-not-delete-this-file.txt ファイル （第 8 章参照）が配置されています。（含まれていない場合には、ユーザ追加前に配置していただきますようお願いします）

3.7 アクティビティログの自動削除

Web インターフェース上の作業はアクティビティログとして記録されます。

このアクティビティログを一定期間経過後に自動削除するには /var/www/nextcloud/config/config.php を編集し、次の記述を加えます。

```
'activity_expire_days' => 365,
```

これで、アクティビティログは 365 日間保持され、それ以前のログは削除されます。

第4章　データベース

4.1 概要

サニタイザーの Web インターフェースのバックエンド DB として MariaDB を利用しています。

4.2 アカウント情報

サニタイザー Ver.3 の保守に必要なアカウント情報は下記のとおり。

アカウント名	パスワード	説明
root	mariadbforsanitizer	データベースの管理者
nextcloud	nextcloudforsanitizer	サニタイザーの Web インターフェースユーザ

サニタイザー Ver.4 の保守に必要なアカウント情報は下記のとおり。

アカウント名	パスワード	説明
root	（なし）	データベースの管理者
nextcloud	nextcloudforsanitizer	サニタイザーの Web インターフェースユーザ

第5章　サニタイズエンジン

5.1 概要

サニタイズエンジンとして、Linux バイナリの実行ファイルを用いています。

5.2 アカウント情報

サニタイザーのサニタイズエンジンを動作させている OS 上のアカウントは次のとおりです。

アカウント名	初期パスワード	説明
sanitizer	S@n!tizerJapan#2016	初期パスワードは必ず変更してください

注釈: 初期パスワードは必ず変更してください

この技術文書は多くのユーザに流通していますので、初期パスワードを変更せずに放置しておくと、不正アクセスやサイバー攻撃のリスクを高めることになります。

他のシステムで使われていない、推測されにくいパスワードを設定してください。パスワードの設定方法は Ubuntu の解説書やサイト等をご覧ください。

5.3 実行ファイルの配置

サニタイズエンジンは次の場所で動作するように設定しています。

/usr/local/bin/saniconv.x
: サニタイズエンジンの実行ファイル

/usr/local/bin/sanichkrep.x
: サニタイザーのライブラリファイル

/usr/local/bin/sanikey.x
: サニタイザーのライセンスファイル

/usr/local/bin/sanirescan.x
: サニタイズ後の処理を行う実行ファイル。`crontab -e` で毎分実行します。

/usr/local/bin/saniclear.x
: input、output、work フォルダとゴミ箱をクリアする実行ファイル。crontab -e で毎日 4:00am に実行します。

5.4 サニタイズエンジンの呼び出し

5.4.1 概要

標準構成でのサニタイザーは特定のフォルダを監視し、そのフォルダ内にファイルが書き込まれたことをトリガーにして処理を開始します。

初期設定では、/var/local/sanitizer/input ディレクトリを監視対象とし、このディレクトリの中のファイルが、

IN_CREATE
: （Web インターフェース画面でドラッグ＆ドロップされた場合）

IN_MOVED_TO
: （ファイル同期、WebDAV 経由のアップロードがされた場合）

のイベントを取得して、当該ファイルを処理対象とします。

5.4.2 設定情報

ディレクトリ監視は、Ver.3 系及び Ver.4.0 では incron を使っています。Ver.4.2 以降は watchdog を使っています。

incron の場合

incron の設定は、sanitizer ユーザで incrontab -e をすることで、設定ファイルを編集できます。

初期設定の incrontab の内容は次のとおり。順に監視ディレクトリ、イベント、実行プログラム、引数が設定されています。引数の $@ はファイルのパス、$# はファイル名、$% はイベント名を示す符号です。

```
# Ver.3 の場合
/var/local/sanitizer/input IN_CREATE,IN_MOVED_TO /usr/local/bin/saniconv.x $@/$# $%

# Ver.4 の場合
/var/local/sanitizer/input IN_CREATE,IN_MOVED_TO /usr/local/bin/saniconv.x "$@/$#" $%
```

導入後、ログインユーザを追加した場合、監視対象ディレクトリも併せて追加する必要があります。（ログインユーザ <UserID> は追加したユーザ名に置き換えてください）

```
# Ver.3 の場合
/var/local/sanitizer/<UserID>/input IN_CREATE,IN_MOVED_TO /usr/local/bin/saniconv.x
↪$@/$# $%

# Ver.4 の場合
/var/local/sanitizer/<UserID>/input IN_CREATE,IN_MOVED_TO /usr/local/bin/saniconv.x "
↪$@/$#" $%
```

注釈: Ver.4 からの注意事項

Ver.4 の incron において、ファイル名に半角カッコ () が含まれているファイルが反応しない事象が確認されました。

これを回避するには、引数となるファイルパス及びファイル名をダブルクォートで囲む必要があります。

すでにリリースしている Ver.4.0.2 では incrontab の設定が古いまま（ダブルクォートで囲っていない）なので、上記の例のように修正し、サニタイズエンジン一式を最新版にアップデートしてご使用ください。（Ver.4.0.5 以降は incrontab の設定を修正済みです）

---

watchdog の場合

watchdog の設定は、policy ディレクトリ（/var/local/sanitizer/policy）配下の `watchdog.json` ファイルを修正することで行います。

初期設定の watchdog.json の内容は次のとおり。

```
[
    {
        "directory": "/var/local/sanitizer/input",
        "exec_program": "/usr/local/bin/saniconv.x",
        "args": []
    },
    {
        "directory": "/var/local/sanitizer/hiro/input",
        "exec_program": "/usr/local/bin/saniconv.x",
        "args": []
    }
]
```

directory が監視するディレクトリ、exec_program が 監視ディレクトリにファイルが配置された場合の実行プログラム、args は実行プログラムを実行する際のオプションです。

args の設定はリストで表記します。例えば、コマンドラインでは `-p /var/local/sanitizer/policy/foo.conf` と設定している場合は、

```
"args": [ "-p", "/var/local/sanitizer/policy/foo.conf" ]
```

と設定します。

5.4.3 サニタイズ処理後のファイルの書き出し（output）

サニタイザーは指定されたディレクトリに処理後のファイルを書き出します。

書き出し先のディレクトリは、次の規則で決定されます。

Web インターフェースでのログインユーザが admin / sanitizer の場合
`/var/local/sanitizer/output`

Web インターフェースでのログインユーザがそれ以外の場合
`/var/local/sanitizer/<UserID>/output`

5.4.4 アーカイブファイル解凍処理後のファイルの書き出し（work）

サニタイザーは指定されたディレクトリにアーカイブファイル解凍後のファイルを書き出します。

書き出し先のディレクトリは、次の規則で決定されます。

Web インターフェースでのログインユーザが admin / sanitizer の場合
`/var/local/sanitizer/work`

Web インターフェースでのログインユーザがそれ以外の場合
`/var/local/sanitizer/<UserID>/work`

なお、サニタイザーのデフォルト設定では、`work` ディレクトリに書き出されたファイルを引き続きサニタイズ処理します。サニタイズ前のファイルは `work` ディレクトリに蔵置されたままとなりますので注意してください。

5.5 サニタイズエンジンの直接実行

5.5.1 概要

コマンドラインから直接サニタイズエンジンを実行することもできます。

利用の場面としてはかなり特殊になりますが、例えば外部のサーバから ssh 経由で直接コマンドを実行し、サニタイザーを自社のサービスに組み込む等の連携方法が考えられます。

5.5.2 実行方法

コマンドラインから次のように実行します。

```
$ /usr/local/bin/saniconv.x -o <出力先 (Output) ディレクトリ> -w <作業用 (Work) ディレクトリ> -p <ポリシーファイルのフルパス> <無害化処理させたいファイルのフルパス>
```

オプションは省略可能です。

なお、標準構成で設定された場所以外の無害化処理の前後のファイル、作業ファイルは saniclear.x による自動削除の対象にはなりませんのでご注意ください。

5.6 ログファイル

サニタイザーの動作ログは、/var/log/sanitizer ディレクトリ配下にあります。

sanitizer.log
: サニタイザーの動作ログ

sanitizer_error.log
: サニタイザーのエラーログ

ログファイルはサニタイザーの Web インターフェース経由でダウンロードできます。

```
[2022-12-01 16:58:23]  2388 [BEGIN] Excel ファイル（外部ファイル生成）.xlsm
[2022-12-01 16:58:23]  2388 Filename is OK
[2022-12-01 16:58:23]  2388 [INPUT] Analyze File Type and Sanitize.
[2022-12-01 16:58:23]  2388 Excel Files is
[2022-12-01 16:58:24]  2388 Check External site seeker in OpenXML File
[2022-12-01 16:58:24]  2388 External site seeker was not found.
[2022-12-01 16:58:24]  2388 Check doubtful strings in this file
[2022-12-01 16:58:24]  2388 Doubtful strings were found.
[2022-12-01 16:58:24]  2388 Check External site seeker 2 in OpenXML File
[2022-12-01 16:58:24]  2388 External site seeker 2 was not found.
[2022-12-01 16:58:24]  2388 Check bad behavior macro code
[2022-12-01 16:58:24]  2388 High risk macro code was found. Delete it.
[2022-12-01 16:58:24]  2388 Delete Macro or Embedded Files.
[2022-12-01 16:58:24]  2388 [OUTPUT] Excel ファイル（外部ファイル生成）_2388_sanitized.
↪xlsm
[2022-12-01 16:58:24]  2388 Post sanitize and ReScan Nextcloud database
```

記録は、日時、プロセス ID、稼働記録の順で行われます。

サニタイザーは複数プロセスが同時に動くことがあり、その場合には複数のプロセス ID が混在する場合もあります。サニタイズの挙動を追う場合には、プロセス ID でソートしてください。

5.7 アンチウイルス

サニタイザーでは、サニタイズ処理の中でアンチウイルスモジュールを呼び出して処理させています。

アンチウイルスエンジンには、ClamAV を利用しています。

5.7.1 パターンファイルの更新

インターネットに接続できる環境では、定期的にパターンファイルを取得し、更新しています。

パターンファイルは、/var/lib/clamav に配置されています。ClamAV のサイトからパターンファイルをダウンロードし、手作業でファイルを差し替えることも可能です。

パターンファイルのダウンロード URL は /etc/clamav/freshclam.conf の中に、DatabaseMirror という設定値があり、そこで指定しているサーバがパターンファイルのダウンロードサーバとなります。

5.7.2 アンチウイルスの無効化

外部のシステム等によりアンチウイルス処理が行われている場合、サニタイザー側のアンチウイルス機能を無効化することができます。

ClamAV は非常にメモリを消費するため、安定稼働のためには無効にすることをおすすめします。

サニタイザーのポリシーファイル /var/local/sanitizer/policy/sanitizer.conf で設定してください。

5.7.3 ログファイル

サニタイズエンジンとは別に、アンチウイルスの動作についてログが保存されます。

保存先のディレクトリは、/var/log/clamav です。

clamav.log
: ウイルススキャンのログ

freshclam.log
: パターンファイルダウンロードのログ

5.8 ファイルコンバート

サニタイザーは一部のファイルのサニタイズ処理に、LibreOffice を用いたファイルコンバートを行っています。

ファイルコンバートは、処理対象のファイルの内容に応じて自動的に呼び出されるため、特に保守作業の中で意識する必要はありません。

5.9 マルウェア検知

サニタイザーは、Office ファイル（Word Excel Powerpoint）に含まれる既知のマルウェアの検知に Office-MalScanner を用いています。

OfficeMalScanner は Windows バイナリのため、実行させるためにサニタイザーの中に Windows ソフトウェア実行環境である Wine を経由しています。

このマルウェア検知は、処理対象のファイルの内容に応じて自動的に呼び出されるため、特に保守作業の中で意識する必要はありません。

マルウェアを検知すると、サニタイズ処理を行う前にそのファイルをブロックします。

5.10 マクロ検知

サニタイザーは、Office ファイル（Word Excel Powerpoint）に含まれる VBA マクロプログラムの検知に MacroRaptor を用いています。

このマクロ検知は、処理対象のファイルの内容に応じて自動的に呼び出されるため、特に保守作業の中で意識する必要はありません。

このマクロ検知は、サニタイズの要不要を判別するために用いられます。

5.11 RTF マルウェア検知

サニタイザーは、RTF ファイルに含まれる既知のマルウェアの検知に RTFScan を用いています。

RTFScan は Windows バイナリのため、実行させるためにサニタイザーの中に Windows ソフトウェア実行環境である Wine を経由しています。

このマルウェア検知は、処理対象のファイルの内容に応じて自動的に呼び出されるため、特に保守作業の中で意識する必要はありません。

マルウェアを検知すると、サニタイズ処理を行う前にそのファイルをブロックします。

5.12 PDF スクリプト検知

サニタイザーは、PDF ファイルに含まれる JavaScript プログラムの検知に PDFMiner を用いています。併せて、パスワード付きの PDF ファイルの識別も行います。

このスクリプト検知は、処理対象のファイルの内容に応じて自動的に呼び出されるため、特に保守作業の中で意識する必要はありません。

このスクリプト検知は、サニタイズの要不要を判別するために用いられます。

5.13 サニタイザーのライセンスファイルの更新

5.13.1 ライセンスファイルの配置

ライセンスファイルの配置先は次のフォルダです。

サニタイザープロをお使いの場合には、 Input サーバ、Output サーバの両方に配置してください。

/usr/local/bin/sanikey.x
　サニタイザーのライセンスファイル

5.13.2 ポリシーファイルの編集

当社から、導入先団体の団体コード（自治体以外の教育機関、事務組合、民間企業の場合には当社が独自にコードを付与します）をお知らせしますので、ポリシーファイルの次の箇所（LGCODE）に追記してください。

ポリシーファイル（抜粋）

```
##############################################################################
# サニタイザーの導入組織に関する設定
# サニタイザーをご利用いただいている組織に関する情報を設定します。
#
# 自治体コード
# （5 桁数字）
LGCODE=
```

5.13.3 バージョンアップに伴うライセンスキーの更新について

サニタイザー Ver.3 と Ver.4 は異なる Linux ライブラリを使っているため、そのままではライセンスキーを再適用することができません。

Ver.4 へのバージョンアップに伴いライセンスキーを更新する必要がありますので、下記メールアドレスに次の内容を記載の上、**Ver.3 のライセンスキーを添付して** ライセンスキーの再発行を依頼してください。

```
送信先メールアドレス sales@sanitizer.jp

　　サニタイザーライセンスキー更新依頼

 本メールにて現在使用中のサニタイザー Ver.3 のライセンスキーをお送りしますので、
 Ver.4 対応のライセンスキーへの更新を依頼します。
```

5.14 サニタイズエンジンのバージョンアップ

5.14.1 最新版のサニタイズエンジンのダウンロード URL

最新版のサニタイズエンジンは次の URL からダウンロード可能です。（個別対応モジュールやオプションモジュールについては個別にお問い合わせください）

サニタイザー Ver.3

`https://prague.sanitizer.online/sanitizer/index.php/s/Jdy8BoJPeTiHMMq`

サニタイザー Ver.4

`https://prague.sanitizer.online/sanitizer/index.php/s/MqKERzYcp6ZQ5xc`

5.14.2 サニタイズエンジンのバージョンアップ方法

サニタイズエンジンは Linux バイナリの実行ファイルのため、モジュールファイルを差し替えていただくだけでバージョンアップは完了します。

サーバの再起動は不要です。つまりモジュールファイルの差し替え直後から、新たなサニタイズエンジンで処理を行います。

第6章　ネットワーク間のファイル受け渡し（サニタイザープロ）

6.1 概要

サニタイザーをネットワーク間のファイル受け渡しのために用いる場合、input サーバと output サーバの 2 台のサーバを稼働させる必要があります。（図 6.1）

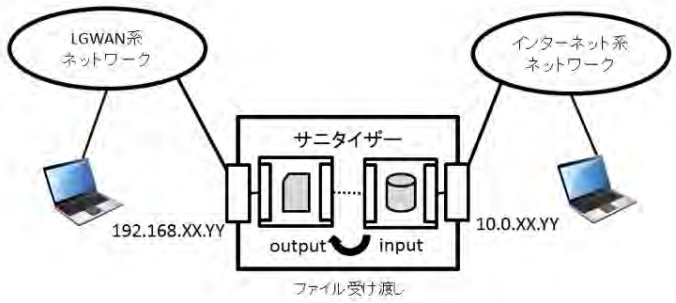

図 6.1: ネットワーク間受け渡しの構成

6.2 ネットワーク構成

6.2.1 input サーバ

ネットワークインターフェースを二つ設定します。

ネットワークインターフェース	説明
eth0	インターネット系ネットワークに接続
eth1	内部ネットワークとして output サーバに接続

6.2.2 output サーバ

ネットワークインターフェースを二つ設定します。

ネットワークインターフェース	説明
eth0	LGWAN 系ネットワークに接続
eth1	内部ネットワークとして input サーバに接続

6.2.3 trusted_domains の変更

それぞれのサーバの Web インターフェース（Nextcloud）の `/var/www/nextcloud/config/config.php` を編集し、trusted_domain の記述を参加しているネットワークアドレスに変更します。

6.3 ディレクトリマウント設定

6.3.1 概要

それぞれのサーバの output ディレクトリを NFS でマウントしています。

マウントは、autofs で設定しています。

6.3.2 設定ファイルの編集

input サーバ（NFS クライアント）の output フォルダ → output サーバ（NFS サーバ）の output フォルダへのマウントを行います。

`/etc/auto.master` ファイル（input サーバ）

input サーバの `/etc/auto.master` ファイルを編集します。

```
# NOTE: mounts done from a hosts map will be mounted with the
#       "nosuid" and "nodev" options unless the "suid" and "dev"
#       options are explicitly given.
#
#/net   -hosts
#
# Include /etc/auto.master.d/*.autofs
# The included files must conform to the format of this file.
#
+dir:/etc/auto.master.d
```

(次のページに続く)

(前のページからの続き)

```
#
# Include central master map if it can be found using
# nsswitch sources.
#
# Note that if there are entries for /net or /misc (as
# above) in the included master map any keys that are the
# same will not be seen as the first read key seen takes
# precedence.
#
#+auto.master
#
#/- /etc/auto.mount
```

ファイルの末尾にある、 /- /etc/auto.mount の行のコメントが外れていて、有効であることを確認します。コメントアウトされている場合はコメントを外します。(上記の例ではまだコメントが入ったままです)

/etc/auto.mount ファイル（input サーバ）

input サーバの /etc/auto.mount ファイルを編集します。

```
#/var/local/sanitizer/output -fstype=nfs,rw 192.168.0.2:/var/local/sanitizer/output
```

この行（1 行しかありません）のコメントが外れていて、有効であることを確認します。コメントアウトされている場合はコメントを外します。(上記の例ではまだコメントが入ったままです)

なお、今回は例示として、input サーバと output サーバをつなぐ内部ネットワークが、192.168.0.0/24 であり、input サーバに割り振られている IP アドレスは 192.168.0.3 とします。

/etc/exports　ファイル

output サーバの /etc/exports ファイルを編集します。

```
/var/local/sanitizer/output 192.168.0.0/24(rw,no_root_squash,no_subtree_check)
```

output サーバには最初からフォルダ同期用の設定がされています。新たに同期用のフォルダを追加する際には同様の設定によりマウントするディレクトリを指定する必要があります。

なお、今回は例示として、output サーバに割り振られている IP アドレスは 192.168.0.2 とします。

この状態で二つのサーバを再起動することで、双方の /var/local/sanitizer/output ディレクトリ NFS (autofs) でマウントされます。配置したファイルが互いのサーバから共有されていることを確認してください。

6.4 仮想化ソフトウェア設定（Oracle VirtualBox の例）

サニタイザーは評価用の仮想アプライアンスの動作プラットフォームとして、Oracle VirtualBox を使用しています。サニタイザーをセットアップする際に、導入先のネットワーク環境に応じた設定が必要です。

6.4.1 ネットワーク設定（ポートフォワーディング）

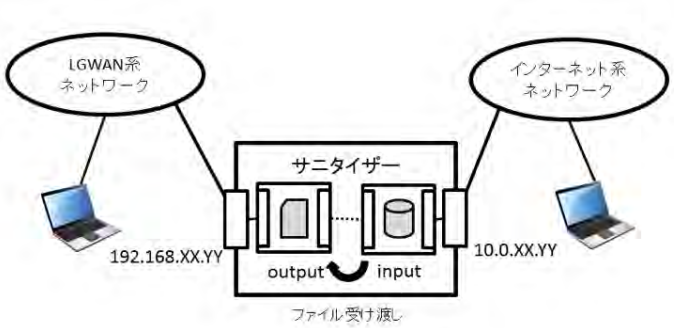

図 6.2: ネットワーク間受け渡しの構成（再掲）

図 6.2 の構成のように、1 台の物理サーバに VirtualBox をセットアップし、さらにその中で二つの仮想サーバ（input サーバ、output サーバ）を稼働させる場合、ネットワーク設定は次のようになります。

input サーバ　アダプター 1　 NAT

input サーバのアダプター 1 の設定は 図 6.3 のとおりです。

図 6.3: input サーバ　アダプター 1　 NAT

ポートフォワーディング

アダプター 1 にはポートフォワーディングの設定をしてください。（図 6.4）

この例では、物理サーバの 1 つめのネットワークポートの IP アドレスを `127.0.0.1` にしている場合です。（例はあまり適切ではありませんが）物理サーバの物理ポートと input サーバのアダプター 1 （enp0s3）が NAT 変換されてつながっているという状態です。

名前	プロトコル	ホスト IP	ホストポート	ゲスト IP	ゲスト ポート
http	TCP	127.0.0.1	8080		80
ssh	TCP	127.0.0.1	2222		22

図 6.4: ポートフォワーディング

input サーバ　アダプター 2　内部ネットワーク

input サーバのアダプター 2 の設定は 図 6.5 のとおりです。

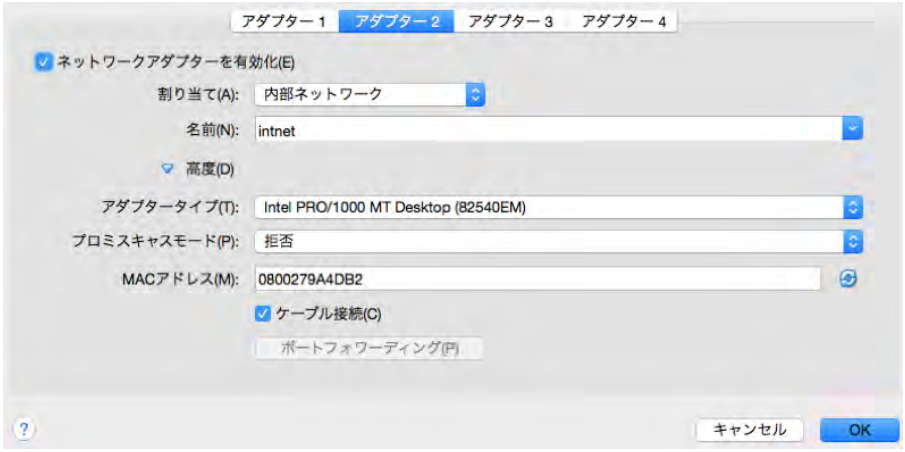

図 6.5: input サーバ　アダプター 2　内部ネットワーク

output サーバ　アダプター 1　NAT

output サーバのアダプター 1 の設定は 図 6.6 のとおりです。

図 6.6: output サーバ　アダプター 1　NAT

ポートフォワーディング

アダプター 1 にはポートフォワーディングの設定をしてください。（図 6.7）

名前	プロトコル	ホスト IP	ホストポート	ゲスト IP	ゲスト ポート
http	TCP	192.168.1.3	8080		80
ssh	TCP	192.168.1.3	2222		22

図 6.7: ポートフォワーディング

この例では、物理サーバの 2 つめのネットワークポートの IP アドレスを `192.168.1.3` にしている場合です。（例はあまり適切ではありませんが）物理サーバの物理ポートと output サーバのアダプター 1（enp0s3）が NAT 変換されてつながっているという状態です。

output サーバ　アダプター 2　内部ネットワーク

output サーバのアダプター 2 の設定は 図 6.8 のとおりです。

図 6.8: output サーバ　アダプター 2　内部ネットワーク

6.4.2 ネットワーク設定（ブリッジアダプタ）

VirtualBox を動作させている機器に割り振られた IP アドレスの他に、サニタイザーの input サーバ、output サーバ用に個別の IP アドレスを割り振れるのならば、NAT ではなくブリッジアダプタを使ってネットワーク設定を行ったほうがよいでしょう。

この場合、input サーバ、output サーバに固定 IP アドレスを割り振り、ブリッジアダプタで直接 VirtualBox を動作させている機器の NIC を紐付ければ OK です。

6.5 他のハイパーバイザーの場合（ESXi や Hyper-V など）

本番導入のために、ESXi や Hyper-V などのハイパーバイザー上でサニタイザーを稼働させる場合には、稼働環境でのネットワーク設定に従ってください。

基本的に上述したブリッジアダプタでの設定と同様に、input サーバ、output サーバに固定 IP アドレスを割り振り、ハイパーバイザー上で仮想サーバに割り当てられた仮想 NIC とサーバ機器に割り当てられる物理 NIC を紐付けすることで、固定 IP アドレスで直接 input サーバ、output サーバにアクセスできるようにすれば OK です。

第7章　ユーザー括登録

7.1 概要

サニタイザーを複数のユーザで使用する場合、そしてそのユーザ数が大量の場合において、一括してユーザを登録し、利用可能にするための手順を示します。

7.2 準備するもの

次の情報（データ）を準備しておきます。

 # ユーザリスト（ユーザ ID、ユーザ名、パスワード）
以降、ユーザ ID を <UserID> として示します。

7.3 作業手順

次の手順でユーザを登録します。

7.3.1 ユーザ登録バッチの作成と実行

第 3 章「アカウントの追加」の手順に沿ったバッチファイルを作成して実行します。

実体ディレクトリの作成
/var/local/sanitizer 配下にユーザごとのディレクトリを作成します。

コマンドラインによるユーザ追加
Nextcloud の occ コマンドを使い、ユーザを登録します。

（LDAP でユーザ情報を連携している場合（第 8 章参照）には、ユーザ登録作業は不要です）

コマンドラインによる実体ディレクトリのマウント
外部ストレージ設定ファイル（JSON）を作成し、インポートします。

7.3.2 input – output サーバ間のフォルダマウント（サニタイザープロの場合のみ）

第６章の「ディレクトリマウント設定」の手順に沿った設定情報を設定ファイル中に追記します。

/etc/auto.mount ファイル （input サーバ）

```
/var/local/sanitizer/<UserID>/output -fstype=nfs,rw 192.168.0.2:/var/local/sanitizer/
↪<UserID>/output
（以下、ユーザ分繰り返し）
```

/etc/exports ファイル　（output サーバ）

```
/var/local/sanitizer/<UserID>/output 192.168.0.0./24(rw,no_root_squash,no_subtree_
↪check)
（以下、ユーザ分繰り返し）
```

7.3.3 サニタイズエンジンの呼び出し設定

第５章の「サニタイズエンジンの呼び出し」の手順に沿った設定情報を設定ファイル中に追記します。

incrontab -e コマンドによる編集

```
/var/local/sanitizer/<UserID>/input IN_CREATE,IN_MOVED_TO /usr/local/bin/saniconv.x
↪$@/$# $%
（以下、ユーザ分繰り返し）
```

第8章　外部システム連携（フォルダ同期）

8.1 概要

サニタイザーの Web インターフェースである Nextcloud は、専用のクライアントソフトを使って、フォルダの同期を行うことができます。

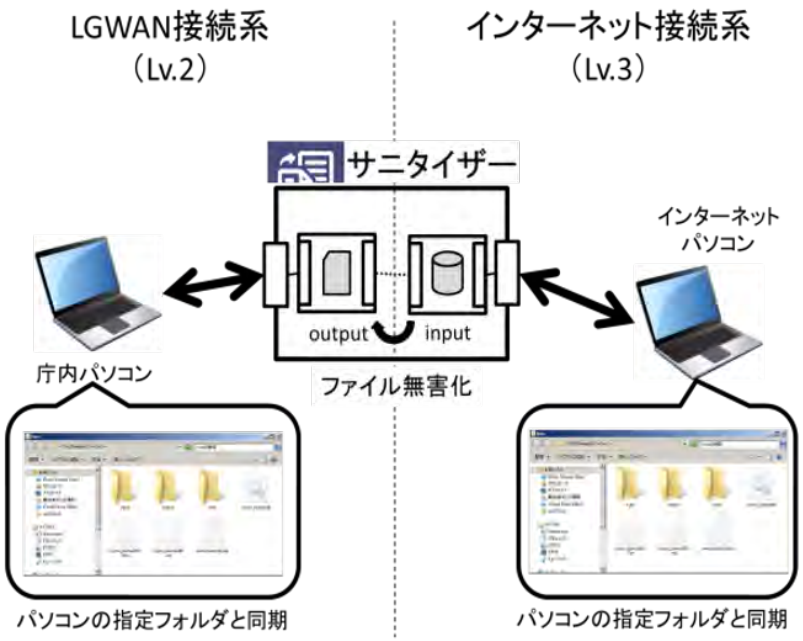

図 8.1: フォルダ同期

専用のクライアントソフトは、Windows、MacOS、Linux 版があり、基本的な機能は同じです。

8.2 Nextcloud クライアントソフトウェア

WebDAV プロトコルを用いて、特定のフォルダとサニタイザーの input、output、work フォルダを同期するソフトウェアです。

ソフトウェアは下記のサイトからダンロードできます。

https://nextcloud.com/install/#install-clients

8.3 外部システム連携のイメージ

Nextcloud クライアントソフトを用いた連携のイメージを示します。

外部システム（Windows サーバ、Linux サーバ）に Nextcloud クライアントソフトをインストールし、フォルダ同期をする先のサーバ（サニタイザーサーバ）の情報を設定すると、外部システム側に同期先のフォルダを設定することができます。（図 8.2）

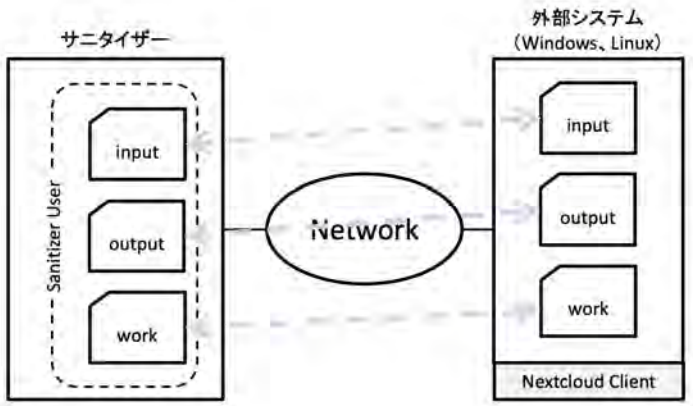

図 8.2: 連携のイメージ

8.4 設定方法（外部システム側）

サニタイザーを連携する外部システムでの設定画面は次のとおり。

8.4.1 接続先サーバの設定

画面は Windows10 のものです。

クライアントソフトウェア（Nextcloud）を起動すると、最初に 図 8.3 のような画面が表示されます。

Login ボタンをクリックして、接続先サーバの設定を行います。

接続するサニタイザーの URL を設定します。

画面例では、`http://192.168.56.104/sanitizer` となっていますが、サニタイザーの Web インターフェースの接続 URL をここに入力してください。（図 8.4）

次へ ボタンをクリックすると、別ウインドウでサニタイザーの Web インターフェースへの接続確認画面が表示されます。この画面で ログイン ボタンをクリックします。（図 8.5）

サニタイザーの Web インターフェースのログイン画面が表示されるので、ここで連携したいアカウントのユーザ名、パスワードを入力してログインします。（図 8.6）

図 8.3: 起動時の画面

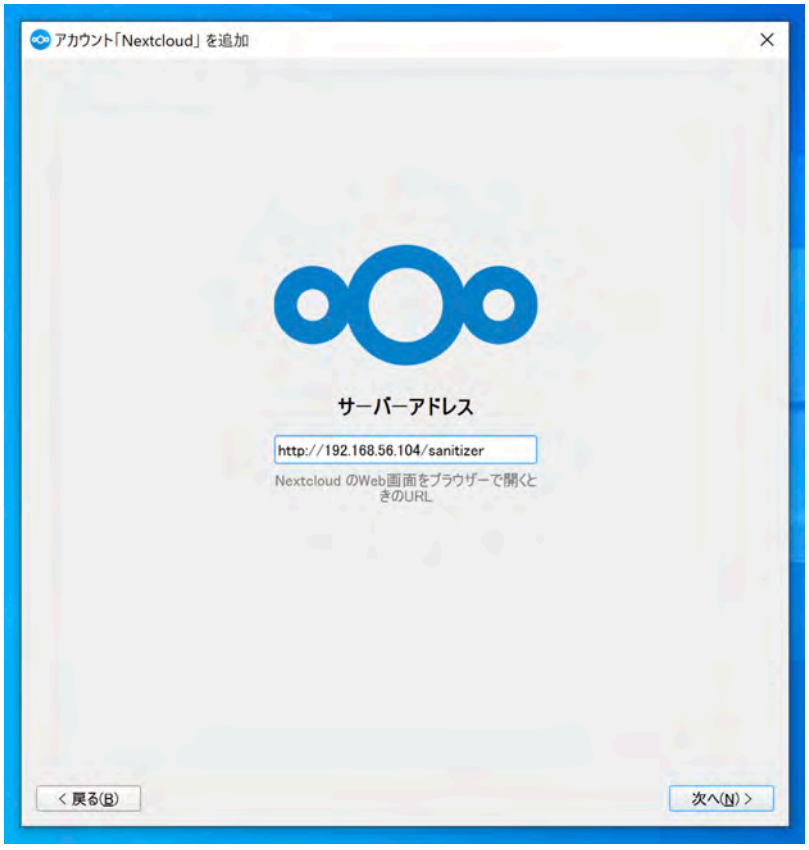

図 8.4: 接続先サーバの設定

図 8.5: Web インターフェース確認画面

図 8.6: Web インターフェースログイン画面

ログインに成功すると、アクセス確認画面が表示されます。ここで アクセスを許可 ボタンをクリックします。（図 8.7）

図 8.7: Web インターフェースアクセス確認画面

8.4.2 同期するフォルダの設定

アクセス許可をすると、同期するフォルダの設定画面が表示されます。（図 8.8）

画面上部の Choose different folder ボタンをクリックすると、外部システム（今回は Windows10 のパソコン）のどのフォルダに同期したファイルを保存するかを指定することができます。

また、画面中部のラジオボタンでは 同期フォルダを選択 ボタンをくりっくし、同期させたいサニタイザー側のディレクトリを指定することができます。

外部システム側でサニタイザーの input、output、work フォルダと同期するフォルダを指定します。（図 8.9）

同期するフォルダは、外部システム側の設定（サニタイズ対象のファイルをどのフォルダに保存するか）によって変わります。

フォルダの同期を行うため、既存のシステムであらかじめ使用するフォルダを同期対象とすることで、既存システムにサニタイズ機能を組み合わせることができます。

なお、Windows10 の場合、図 8.10 のようなフォルダができます。

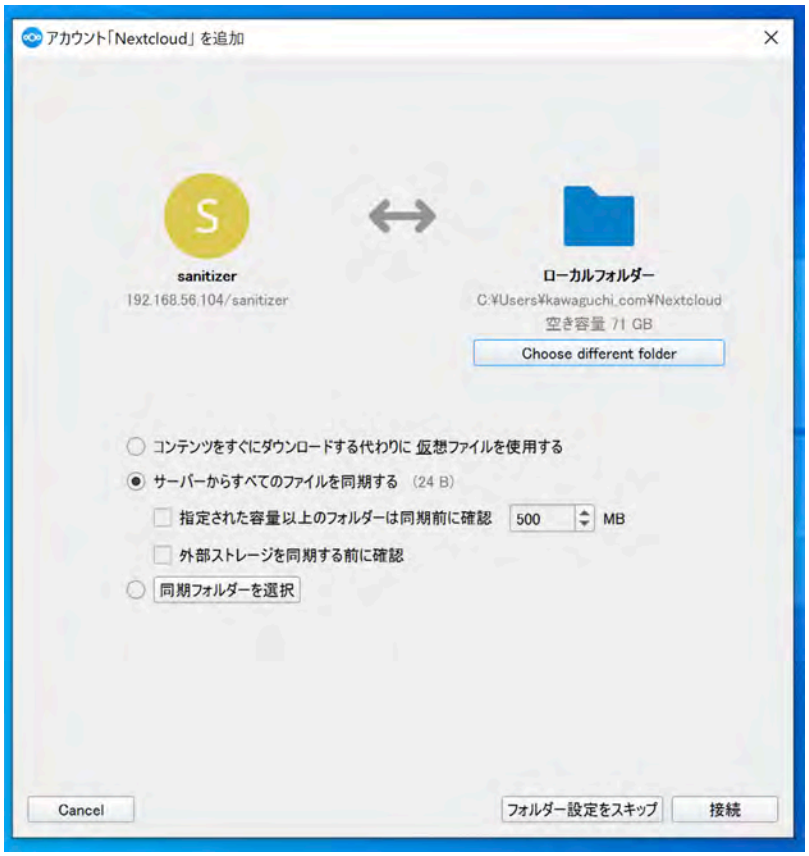

図 8.8: 同期フォルダ設定画面

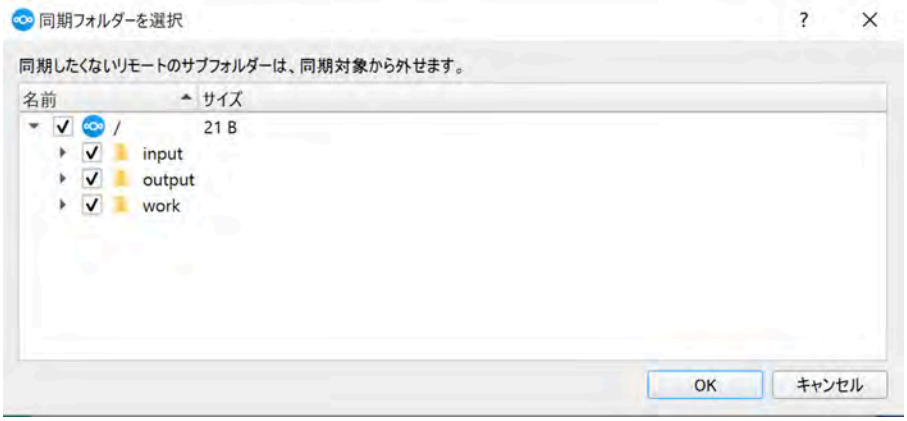

図 8.9: 同期するフォルダの指定

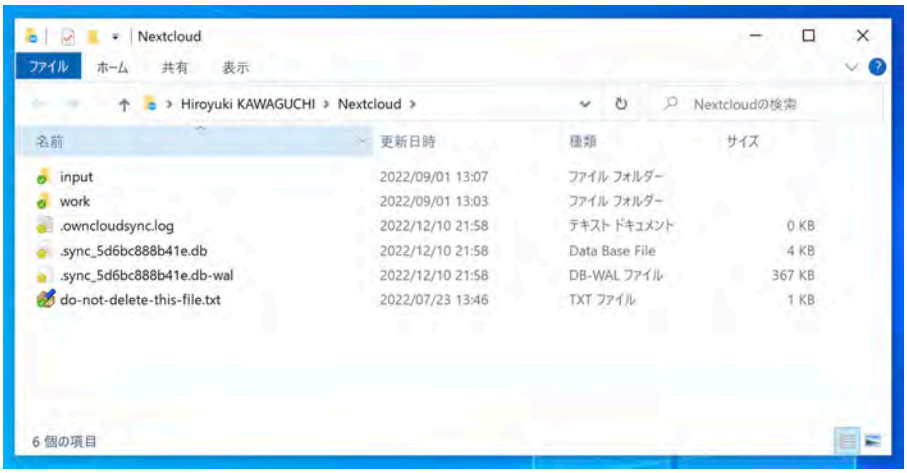

図 8.10: 同期フォルダの様子

8.4.3 同期設定の変更

同期設定を変更したり、新たに同期ユーザを追加する場合は、Nextcloud クライアントソフトの 設定 メニューから 図 8.11 のような画面を表示させ、個別に設定します。

8.5 ログファイル

Nextcloud クライアントソフトにおけるフォルダ同期ログは、同期対象フォルダと同じフォルダ内に保存されています。

owncloudsync.log
: フォルダ間同期のログ

8.6 do-not-delete-this-file.txt ファイル

Nextcloud クライアントソフトは、フォルダ同期時の事故を防ぐために、サーバ、クライアントいずれかのフォルダの内容が空になった場合に、空の状態でフォルダを同期してよいか（ファイル、フォルダを全削除してよいか）の確認メッセージが表示されます。

この確認メッセージを表示させないための手段として、フォルダ内に少なくともひとつのファイルを残すような設定をしています。このファイルが、`do-not-delete-this-file.txt` です。

図 8.11: Nextcloud クライアント 設定 画面

8.7 ファイルサーバを用いた多数ユーザのフォルダ同期の運用について（設定のヒント）

一般ユーザに Web インターフェースを使わせず、多数のユーザにフォルダ同期の機能を使わせてサニタイザーを運用する場合は、職員のパソコンに個別に同期設定を行うのではなく、職員がアクセスできるファイルサーバを設置して、サニタイザーとファイルサーバとの間でフォルダ同期をするほうが運用として安定します。（同期設定を複数行うと、同期のための通信セッションが単純に増加し、通信が安定しません）

その場合、同期設定をユーザごとに行うのではなく、フォルダ同期のための架空のユーザ（例えば sync という名前のユーザとします）を一つ作成し、そのユーザに /var/local/sanitizer を実体ディレクトリとしたマウント設定を行います（第 3 章参照）。

その後、sync ユーザでフォルダ同期設定を行うと、個別ユーザのディレクトリを含む /var/local/sanitizer/ 配下のディレクトリがすべて同期対象となります。この中で同期させたいユーザのフォルダ（例えば ユーザ hiro の input フォルダと work フォルダだけ、など）を指定して同期設定をすることで、単一の同期セッションで複数のユーザのファイルを同期させることが可能となります。

設定でお困りの場合は、個別に導入支援（有償）しますので、お気軽にお声掛けください。

第9章　LDAP/AD連携

9.1 概要

サニタイザーの Web インターフェースは、 LDAP によりユーザ情報を連携することができます。

庁内の LADP サーバや Active Directory と連携することで、ユーザ情報を個別に管理する必要がなくなります。

9.2 注意事項

LDAP 連携は Nextcloud のユーザ情報の連携のみを行います。ユーザが使用するフォルダ、連携するフォルダの設定は個別に行うことになります。（第 3 章参照）

9.3 設定方法

Web インターフェースでの設定は次のとおり。

9.3.1 LDAP サーバ情報の登録

Web インターフェースの管理画面から LDAP/AD 統合 を選択し、設定画面からサーバ情報を登録します。（図 9.1）

図 9.1: LDAP 設定画面

1 行目

サーバの IP アドレスを入力し、ポート検出をクリックします。正しい場合は検出に成功します。

2 行目

管理権限の DN を記載します。特に制約がないため、Administrator としています。

3 行目

2 行目で記載したユーザのパスワードを入力します。

4 行目

ベース DN を記載します。

入力後、ベース DN をテストのボタンをクリックし、「設定 OK」と出ることを確認します。

9.3.2 ユーザ情報抽出設定

ユーザータブを設定します。以下は一例ですが、環境や絞り込み条件によって読み替えてください。(図 9.2)

図 9.2: ユーザ情報抽出設定

このオブジェクトクラスからのみ には、person を設定します。さらにグループで特定するため、グループを設定します。

9.3.3 ログイン属性の設定

ログイン属性を設定します。デフォルトのままで良いと思います。(図 9.3)

9.3.4 詳細設定

ここでディレクトリ設定を行います。(図 9.4)

「接続設定」で「設定は有効です」をチェックし、「ディレクトリ設定」を開きます。

「ベースユーザーツリー」、「ベースグループツリー」を環境に合わせて設定します。

「グループとメンバーの関連付け」は「member(AD)」に設定されていると思いますのでこれを確認します。
「ユーザーごとに LDAP パスワードの変更を有効にする」をチェックします。

図 9.3: ログイン属性設定

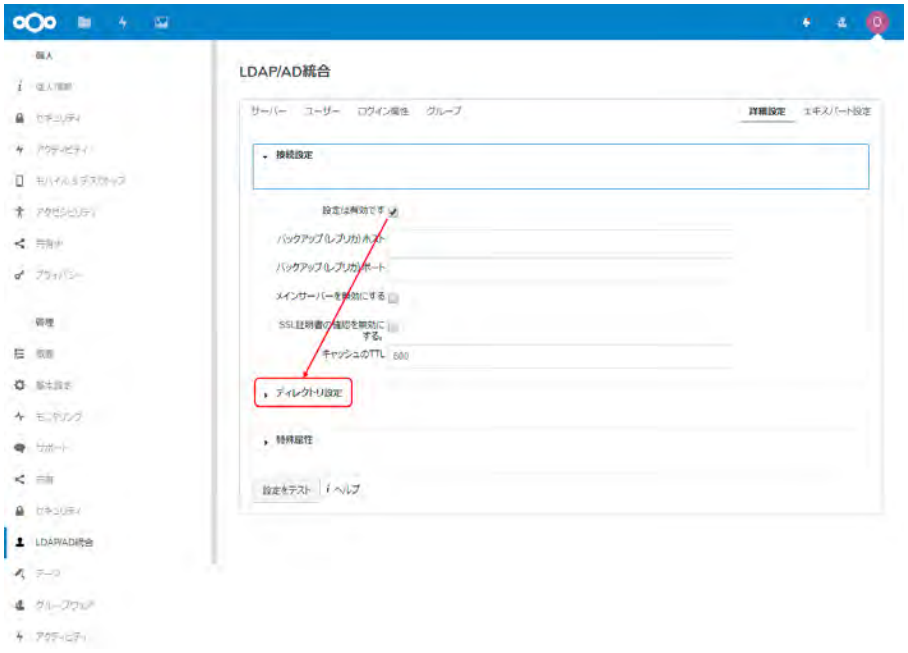

図 9.4: ディレクトリ設定

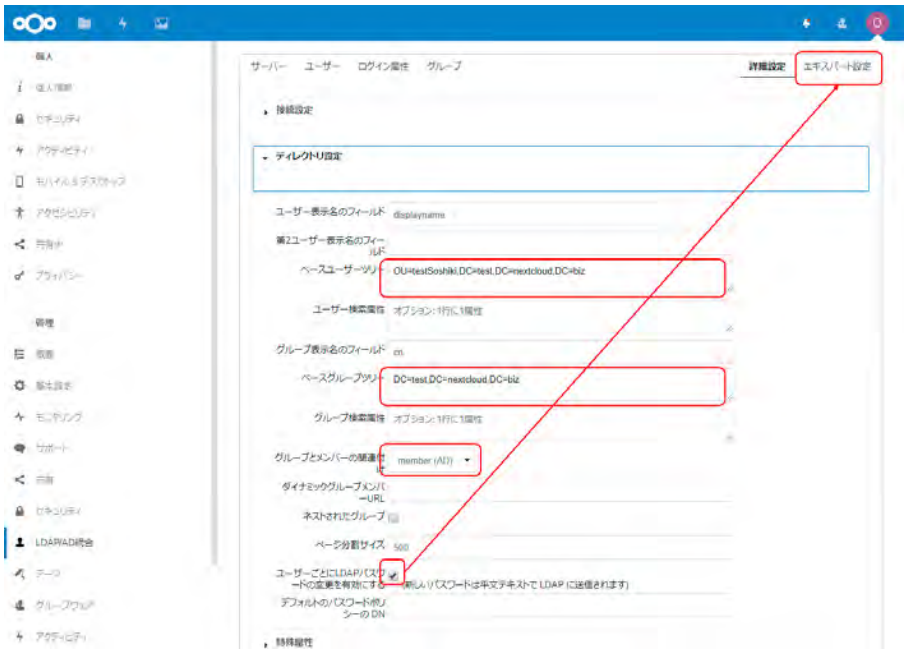

図 9.5: 詳細設定

9.3.5 エキスパート設定

ここまでの設定で、前項で検索されたユーザが登録されますが、現状ではユーザ ID が UUID という長い文字列となるため、続けてエキスパート設定を行います。（図 9.6）

エキスパート設定を選択し、内部ユーザー名属性に sAMAccountName を入力して「設定をテスト」をクリック。「正しい設定です。接続されました。」が出力されれば OK です。

ページの下部に進み、「ユーザー名と LDAP ユーザーのマッピングをクリアする」と「グループ名と LDAP グループのマッピングをクリアする」をクリックし、続けて「設定をテスト」をクリックし成功することを確認します。

これで Nextcloud と Active Directory が連携されました。

図 9.6: 内部ユーザ名属性

第10章　リンクシェア機能とオープンドロップ機能

10.1 概要

サニタイザーのフォルダに共有設定を行うことで、サニタイザーのユーザではない利用者（庁外の方など）との間で、ファイルのダウンロード、アップロードをさせることができます。

10.2 リンクシェア機能

サニタイザーにあるファイルに対して一時的な公開 URL を発行し、その URL を使ってファイルをダウンロードさせる機能です。

この機能はサニタイザーの Web インターフェースにブラウザアクセスできる状態であることが前提となります。庁外の方向けにファイルをダウンロードさせる場合は、サニタイザーを外部のネットワーク（インターネット）に公開することになりますので、不要なポートを閉じる、リバースプロキシや WAF を設置するなどのセキュリティ対策が必要となります。（第２章で説明しているとおり、初期状態ではポートのクローズは行っていません）

10.2.1 設定方法

リンクシェア機能を使うための設定は次のとおりです。

注釈: サニタイザーサーバを外部公開する場合の注意

サニタイザーは内部ネットワークに設置して運用することを想定している製品です。しかし、リンクシェア機能、オープンドロップ機能を使用する場合は、サニタイザーサーバを外部（インターネット）から見える場所に設置しなければならない場面が想定されます。

その場合は、不要なポートを閉じる、アクセス元の IP アドレスを制限する、WAF（Web Application Firewall）を手前に配置する等の対策を施してから導入してください。

外部ストレージ設定の変更

Web インターフェースに admin でログインし、設定メニュー → 外部ストレージ設定から、リンクシェアをさせたいフォルダに「共有の有効化」の設定を加えます。(図 10.1)

この例では、ユーザ sanitizer の work フォルダに共有の有効化を加えています。あらたに実体ディレクトリを作成し、リンクシェア用のフォルダを設定することもできます。

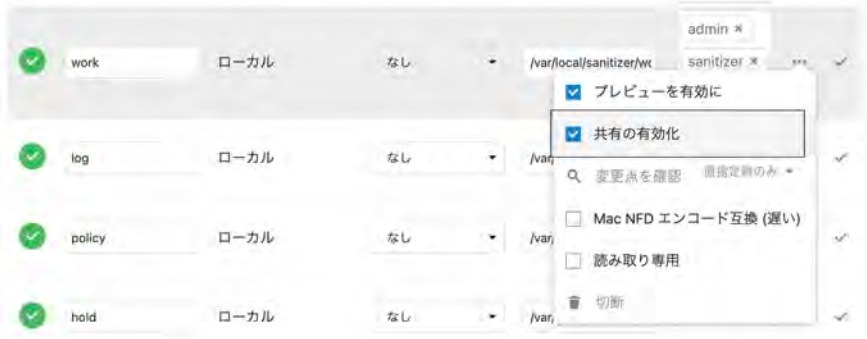

図 10.1: 外部ストレージ設定の変更

共有させたいファイルの配置

リンクシェアさせるフォルダに、共有させたいファイルを置きます。

公開 URL の発行

共有させたいファイルの共有アイコンをクリックし、共有のメニューを開きます。(図 10.2)

共有メニューの中の「URL で共有」のアイコンをクリックすると、公開 URL がクリップボードにコピーされます。この公開 URL を相手に伝えます。

ファイルのダウンロード

公開 URL にアクセスすると、次のような画面が表示され、ファイルをダウンロードすることができます。(図 10.3)

リンクシェアの詳細設定

公開 URL の有効期限を設定したり、ダウンロードに際してパスワードを設定するなど、詳細の設定を加えることもできます。

「URL で共有」の横にある詳細アイコンをクリックすると、詳細設定画面が表示されます。

図 10.2: 共有メニュー

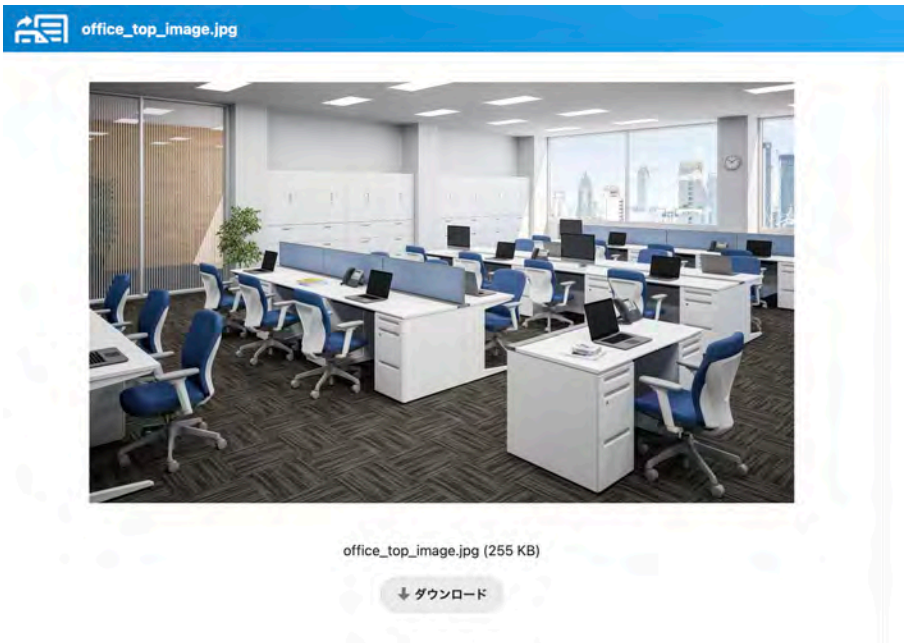

図 10.3: ファイルダウンロード

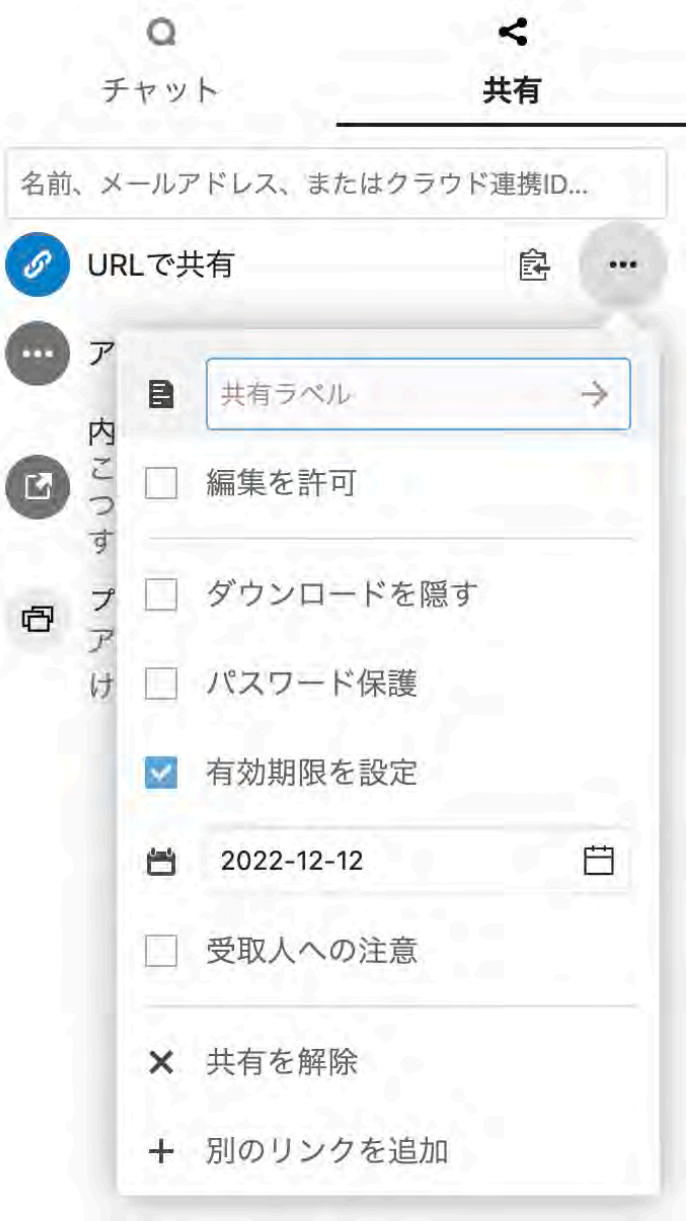
チャット
共有
名前、メールアドレス、またはクラウド連携ID...
URLで共有
共有ラベル
編集を許可
ダウンロードを隠す
パスワード保護
有効期限を設定
2022-12-12
受取人への注意
共有を解除
別のリンクを追加

図 10.4: 詳細設定画面

フォルダ自体をリンクシェアで公開する場合（設定のためのヒント）

上記の例はファイルのシェアでしたが、フォルダを配置してフォルダそのものに公開 URL を発行し、フォルダ内のファイルをダウンロードさせることも可能です。

ただし、現時点ではフォルダの階層は一階層のみです。（サブフォルダはダウンロードさせることができません）

10.3 オープンドロップ機能

リンクシェア機能とは逆向きに、サニタイザーにあるファイルに対して一時的な公開 URL を発行し、その URL を使ってファイルをアップロードさせる機能です。

この機能はサニタイザーの Web インターフェースにブラウザアクセスできる状態であることが前提となります。庁外の方向けにファイルをアップロードさせる場合は、サニタイザーを外部のネットワーク（インターネット）に公開することになりますので、不要なポートを閉じる、リバースプロキシや WAF を設置するなどのセキュリティ対策が必要となります。（第 2 章で説明しているとおり、初期状態ではポートのクローズは行っていません）

10.3.1 設定方法

オープンドロップ機能を使うための設定は次のとおりです。

外部ストレージ設定の変更

リンクシェア機能の場合と同様です。

実体ディレクトリを作成し、新たに外部ストレージ設定をすることもできます。

ファイルアップロード先フォルダの配置

外部ストレージあるいは外部ストレージ内にフォルダを作成し、そのフォルダをファイルアップロード先フォルダとします。

公開 URL の発行

リンクシェアの場合と同様に、アップロード先フォルダに共有設定を行います。

下記の例では hold フォルダに オープンドロップのための公開 URL を設定しています。（図 10.5）

詳細設定画面で「ファイルドロップ」を選択しています。

図 10.5: オープンドロップ用公開 URL 設定

ファイルアップロード画面

公開 URL にアクセスすると、次のような画面が表示され、ファイルをダウンロードすることができます。（図 10.6）

図 10.6: ファイルアップロード

第11章　ホールド＆パス機能

11.1 概要

ホールド＆パスという機能を使い、いわゆる上長承認機能に似た機能を実現することができます。

無害化処理の過程でどうしても庁内に取り込みたいファイルや、外に持ち出したいファイルを個別の承認を経て移動させることができます。

疑義のあるファイルや個別の事情で受け渡ししたいファイルは、一旦管理者（承認者）のみがアクセスできる HOLD フォルダに置かれ、管理者はこの HOLD フォルダにあるファイルを受け渡したい相手を指定して閲覧させることができます。

11.2 ホールド＆パスの動作イメージ

ホールド＆パスは次のような流れで動作します。

サニタイザーは無害化処理の過程で保留にするべきファイルを検知した場合、このファイルを管理者（承認者）のみがアクセスできる「HOLD フォルダ」に配置します。（図 11.1）

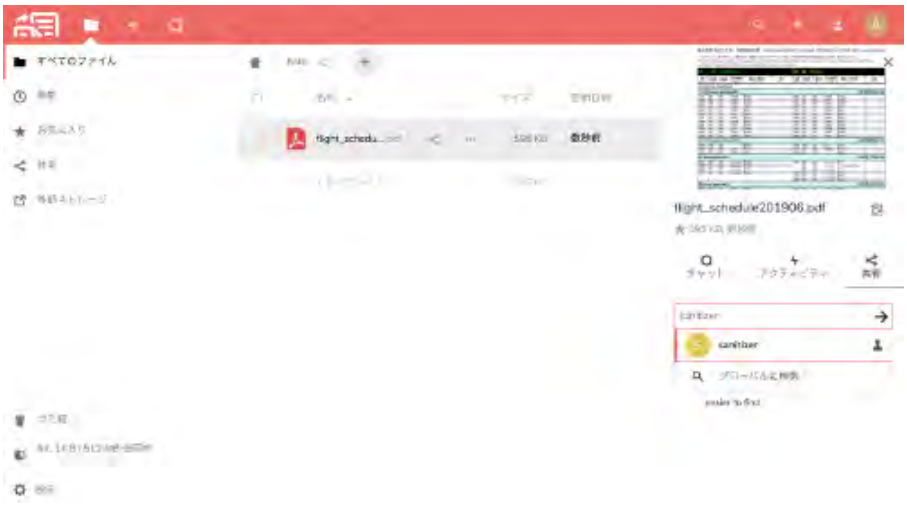

図 11.1: HOLD フォルダの様子

管理者（承認者）は HOLD フォルダにあるファイルを、個別の依頼に基づき受け渡しするかを判断します。

判断の結果、受け渡しする場合は、このファイルに対して個別の参照権限を利用者に対して付与します。共有ボタンをクリックすると共有メニューが開きますので、共有させたい利用者を指定します。

ファイルが共有されると、利用者のフォルダ一覧画面の中に、目的のファイルが追加されたことがわかります。（図 11.2）

図 11.2: 共有設定され、受け渡しができたファイルの様子

11.3 オープンドロップ機能との組み合わせによる任意のファイルの受け渡し承認（設定のヒント）

サニタイザーの HOLD ファイルは、無害化処理の過程で受け渡しの疑義が生じた場合に、そのファイルを一時保留する目的で用意されています。

この HOLD フォルダをさらに活用して、一般ユーザが自身の任意のファイルを承認を経て受け渡しすることができます。

HOLD フォルダにオープンドロップ機能をつける（第１０章）

第１０章を参考にして、HOLD フォルダにオープンドロップ機能をつけ、公開用 URL を発行します。

一般ユーザに公開用 URL を周知

公開用 URL をあらかじめ一般ユーザに周知しておき、受け渡し承認を希望する場合は、この公開用 URL にアクセスさせます。

オープンドロップで HOLD フォルダにファイルをアップロードさせる

オープンドロップで HOLD フォルダにフォルダにファイルをアップロードさせます。ファイルはアップロードしかできないので、他人がアップしたファイルをダウンロードすることはできません。

承認依頼を何らかの手段で行う

口頭、メール、チャット等、庁内の連絡手段は何でもいいので、当該ファイルを受け渡したい旨の依頼を管理者（承認者）に行います。

管理者（承認者）によるファイルの受け渡し

上記のとおり、ファイルの共有機能を使って、ファイルを依頼してきたユーザに共有して受け渡します。

第12章　メールクライアント機能

12.1 概要

サニタイザーには IMAP メールクライアントアプリがあり、メールサーバから受信したメールの添付ファイルをサニタイザーで無害化処理し、庁内に取り込むことができます。

なお、メールクライアント機能は標準では有効になっていません。ユーザ数に応じたストレージ容量や運用の設計が必要なため、導入を検討される場合は個別対応（有償）とさせていただきます。

12.2 メールクライアント機能の動作イメージ

メールクライアント機能は次のような構成で動作します。（図 12.1）

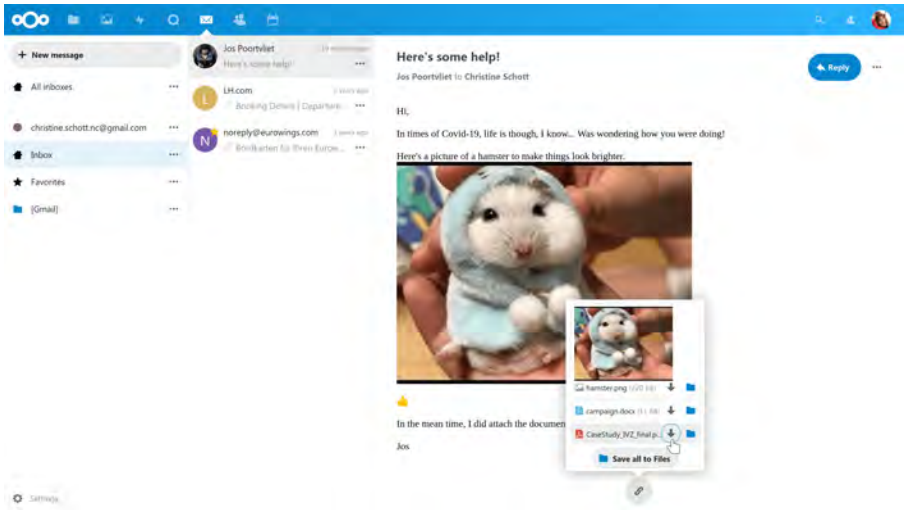

図 12.1: メールクライアント機能の動作イメージ

メールクライアント機能は主にインターネット接続系で稼働しているサニタイザーで使用します。したがって、送信サーバを適切に設定すればメールの送信もできます。

また、添付ファイル付きのメールを受信した場合、サニタイザーの Web インターフェース画面中でメール内容の確認ができ、添付ファイルをサニタイザー内に保存することができます。

サニタイザー内に保存した添付ファイルはそのまま無害化処理をさせ、庁内に取り込むことが可能となります。

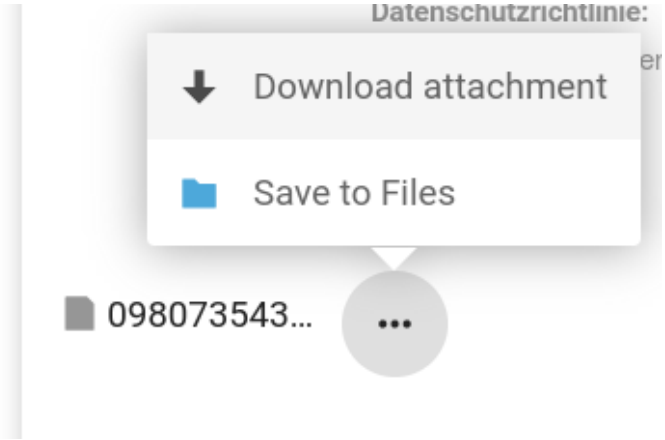

図 12.2: メールの添付ファイルの受信（ダウンロードあるいはサニタイザー内への保存）

第13章　Officeファイル同時編集機能

13.1 概要

サニタイザーのWebインタフェース上でOfficeファイルを閲覧、編集できる機能です。ファイルをユーザ間で共有すると、同じファイルを複数のユーザで同時に編集することができます。

またこの画面上ではOfficeファイルのマクロは動作しませんので、疑義のあるファイルについても安全に内容を確認することができます。

13.2 Officeファイル同時編集機能の動作イメージ

この機能を使用するために特別な準備は不要です。

サニタイザー上にOfficeファイルが置かれ、そのファイルをクリックするとブラウザ内でOfficeファイルが開きます。（図 13.1）（図 13.2）

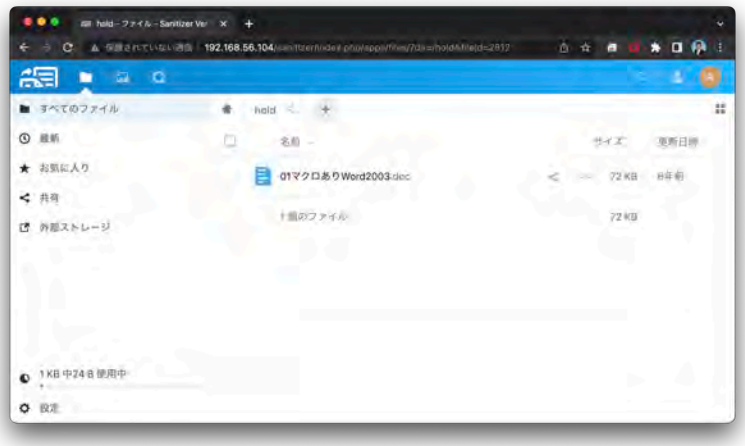

図 13.1: Officeファイルをクリックするだけ

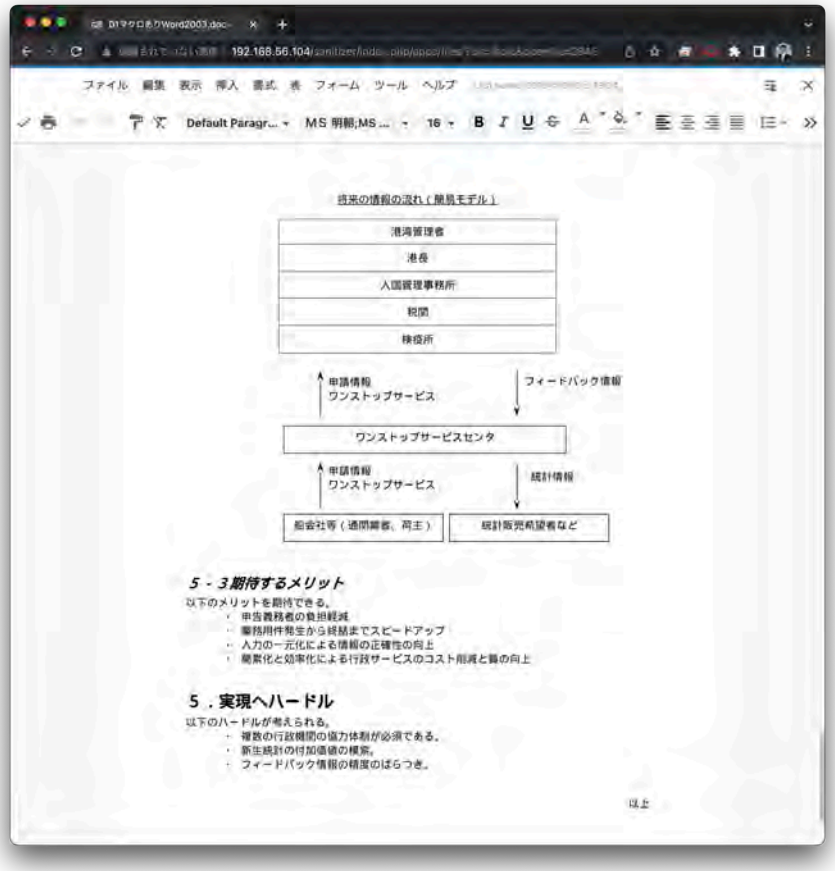

図 13.2: Word ファイルがブラウザ内で表示される

13.3 初期状態での制限事項

初期状態では、一般ユーザには Office ファイルの閲覧権限のみ、admin に閲覧権限、編集権限を付与しています。

一般ユーザにも編集権限を付与する場合は、設定メニュー → オフィスの「詳細設定」より、「特定のグループに編集を制限する」の欄に一般ユーザのグループ（user）を追記してください。

図 13.3: 詳細設定画面

13.4 性能面での制限事項

初期状態ではサニタイザーサーバの中に Office ファイルの編集を処理する機能（CODE サーバ）を稼働させています。

性能上の問題により、Office ファイルの閲覧に支障が出てくるような場合は、外部の CODE サーバを立てて運用することをおすすめします。

当社で導入支援（有償）を行っておりますので、ご連絡ください。

第14章　チャットとWeb会議機能

14.1 概要

サニタイザーの Web インターフェース（Nextcloud）では、チャットと Web 会議機能（Talk 機能）があります。

ファイルの受け渡し承認や、職員間のやり取りなどでご活用ください。

14.2 Web 会議機能を使う場合の前提条件

Web 会議機能においてカメラ、マイクを使う場合は、サニタイザーを HTTPS で通信させなければなりません。（チャット機能だけであれば HTTPS 通信は不要です）

したがって、サニタイザーにサーバ証明書を適用する作業が必要となります。

サニタイザーを HTTPS で通信させる作業支援（有償）も可能ですので、お声掛けください。

14.3 チャットと Web 会議機能の動作イメージ

チャット機能の動作のイメージは次のとおりです。（図 14.1）

チャットルームを作成し、メンバーを招待します。

図 14.1: チャットの動作イメージ

Web 会議機能の動作のイメージは次のとおりです。（図 14.2）

チャットルームの中で「通話を開始」ボタンをクリックすることで Web 会議を始めることができます。

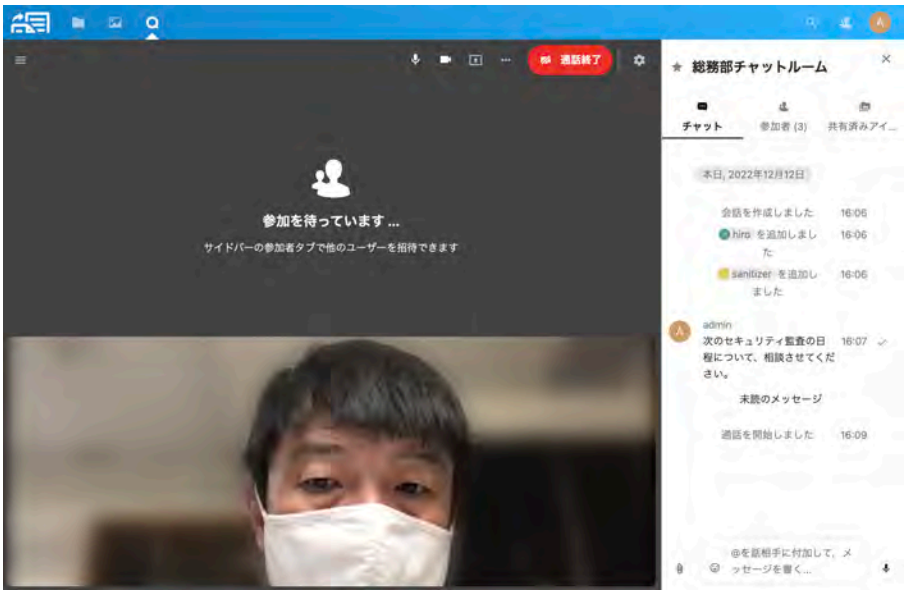

図 14.2: Web 会議機能の動作イメージ

14.4 チャットの中でのファイルの取扱いについて

チャットの中でファイルを添付したり他のユーザと共有することができます。この共有ファイルの保存先は、標準では Talk という名前のフォルダになっています。

しかし、標準状態のサニタイザーは Talk フォルダに容量制限をかけており、そのままではファイルを保存することができません。（図 14.3）

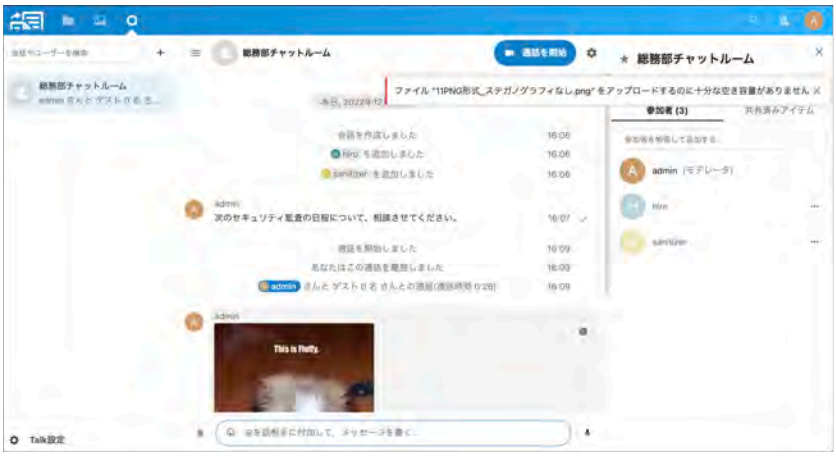

図 14.3: ファイルが保存できないエラーメッセージ

そこで Talk フォルダにファイルを保存できるようにするためには次の 2 つの方法があります。

Talk フォルダを外部ストレージとして設定する（こちらを推奨）

第 3 章を参考に、実体ディレクトリを `/var/local/sanitizer/hiro/talk` のように作成し、そのディレクトリに外部ストレージ設定を行います。

その際のフォルダ名を Talk とすると、保存したファイルは /var/local/sanitizer/hiro/talk 配下に配置されます。

ユーザごとの容量制限を緩和する

第 3 章「Web インターフェースにおけるユーザの追加」において、ユーザを設定する画面があります。この画面の中でユーザごとにデータの保存容量（クォーター）を設定することができます。

図 14.4: ユーザのクォーター設定

初期設定では 1KB のサイズとなっています。これを増やすことで Talk フォルダにファイルを保存できるようになります。ただし、この場合、ユーザが任意にフォルダを生成し、ファイルを置くことができるようになるため、無用なファイルがサニタイザー上に残りやすくなります。

14.5 ネットワークをまたいだチャットメッセージのやり取りについて（サニタイザープロの場合）

初期状態では、チャットのメッセージはそれぞれのネットワーク内に閉じた状態で保存されます。

例えば、インターネット側と庁内側との間でメッセージをやり取りしたい場合、少し設定が必要ですが、MatterBridge という機能を使うことで実現することができます。（図 14.5）

導入支援（有償）で対応させていただきます。

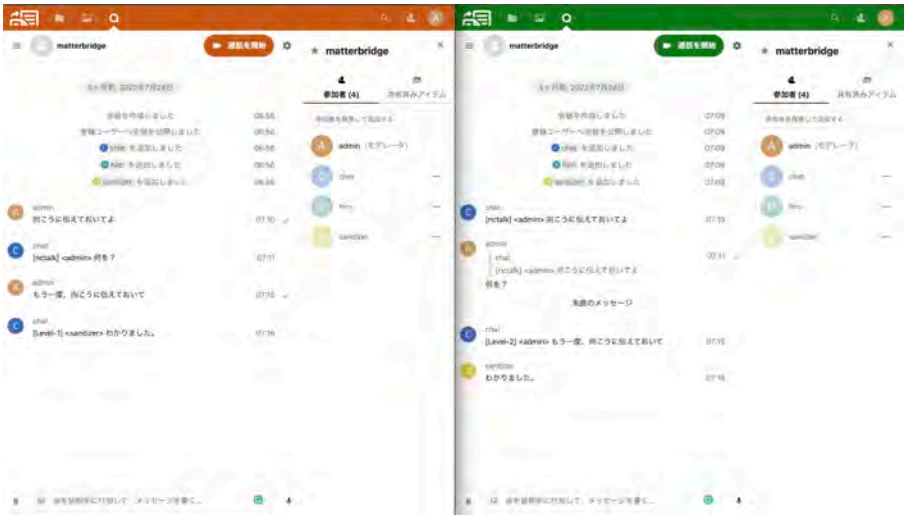

図 14.5: ユーザのクォーター設定

第15章　逆向きモジュールオプション（sanipass.x）

15.1 概要

サニタイザーの有償オプションとして、無害化処理の方向と逆向きにファイルの受け渡しを行うモジュール（sanipass.x）を用意しています。

15.2 逆向きモジュール（sanipass.x）の動作イメージ

逆向きモジュール（sanipass.x）は次のような構成で動作します。（図 15.1）

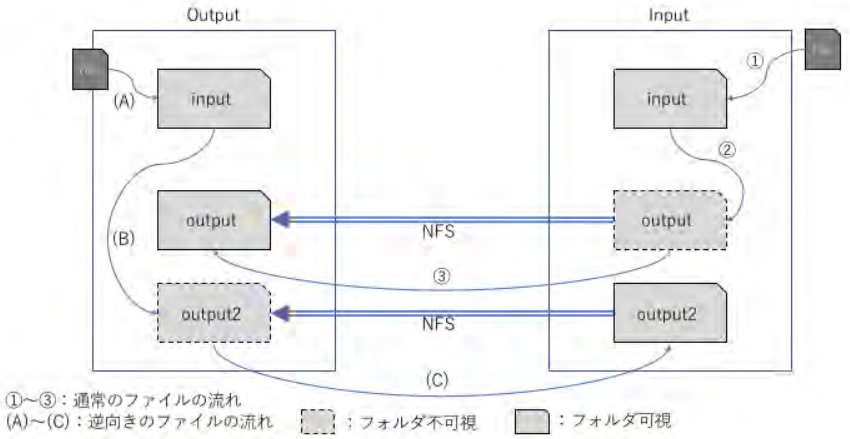

図 15.1: 逆向きモジュール（sanipass.x）の動作イメージ

15.3 逆向きモジュール（sanipass.x）の配置

逆向きモジュール（sanipass.x）は次の場所で動作するように設定しています。

/usr/local/bin/sanipass.x
: 逆向きモジュールの実行ファイル

/usr/local/bin/sanichkrep.x
: サニタイザーのライブラリファイル（saniconv と共通）

/usr/local/bin/sanikey.x
: サニタイザーのライセンスファイル（saniconv と共通）

15.4 incrontab の設定

Output サーバ にて incrontab -e で設定ファイルを編集します。

Output サーバ の input フォルダを監視し、sanipass.x が動作するようにします。

```
/var/local/sanitizer/input IN_CREATE,IN_MOVED_TO /usr/local/bin/sanipass.x  $@/$# $%
```

15.5 ポリシーファイルの編集

Output サーバ の /var/local/sanitizer/policy/sanitizer.conf を編集します。

設定箇所（抜粋）は次のとおり。

```
OUTPUTFOLDER="output2"  ←  このフォルダ名は任意に設定
SERVERTYPE=OUTPUT  ←  OUTPUT に設定されていることを確認
```

15.6 フォルダの作成と NFS マウント

第 6 章「ネットワーク間のファイル受け渡し」と同様に、フォルダの作成と NFS マウントを行います。

15.6.1 フォルダの作成

逆向き受け渡しの出力先となるフォルダを作成します。この例では OUTPUTFOLDER で設定した "output2" という名称にしています。

```
$ sudo mkdir /var/local/sanitizer/output2
```

15.6.2 NFS マウント

ファイルの受け渡しは逆向きですが、NFS マウントは第 6 章と同様に input→output で設定してください。

/etc/auto.mount ファイル （input サーバ）

```
/var/local/sanitizer/output2 -fstype=nfs,rw 192.168.0.2:/var/local/sanitizer/output2
```

/etc/exports ファイル （output サーバ）

```
/var/local/sanitizer/output2 192.168.0.0./24(rw,no_root_squash,no_subtree_check)
```

15.7 実体ディレクトリのマウント

第 3 章「Web インターフェース」を参考に、Nextcloud の設定画面で実体ディレクトリのマウントを行います。

その際、Output サーバ側の output2 フォルダはマウント設定が不要です。（不可視）

一方、Input サーバ上で output2 フォルダのマウント設定が必要となります。

15.8 sanipass の順方向への利用（設定のヒント）

元々、sanipass は庁内 → 庁外へのファイルの持ち出しを想定して「無害化処理をせずに最低限度のファイルチェックのみを行って受け渡す」モジュールとして開発されました。

ただ、このモジュールは庁外 → 庁内の場合でも使用することができます。設定のイメージは、saniconv.x の代わりに sanipass.x を動作するように incrontab の設定を行います。その際に、出力先ディレクトリ (Output) を　-o オプションで指定します。（下記の例では /var/local/sanitizer/hiro/output3 に出力するようにしています）

```
/var/local/sanitizer/hiro/input IN_CREATE,IN_MOVED_TO /usr/local/bin/sanipass.x -o /
↪var/local/sanitizer/hiro/output3  $@/$# $%
```

第16章　付録

16.1 サニタイザーポリシーファイル Ver.3.0.0

```
# This File is Sanitzer Policy for sanitize files.
# Created and ReWrited by Hiro KAWAGUCHI 20190608
# サニタイザー適用バージョン Ver.2.2.15 / Ver.3.0.0 以降

#####################################################################
# このファイルについて
# このファイルはサニタイザーの動作状況を制御するための設定ファイルです。
# それぞれ「項目=設定値」という構成になっています。
# 設定に際しては、項目名、設定値とも半角英数大文字とし、項目と設定値の間には
# 空白を入れないようにしてください。

#####################################################################
# サニタイザーのバージョンに対する考え方
# サニタイザーは主に２つのリリース体系があります。
# OVA ファイルにて提供されたもの -> 常にマイナーバージョン（最下位の数字）が
# 0 になります。
# モジュール差し替えにて提供されたもの -> マイナーバージョンが 0 以外と
# なります。
#
# 一般的にはマイナーバージョンが 0 のものをお使いいただいていますが、機能改善、
# 機能追加、個別カスタマイズにより、モジュールを差し替えたものについては、
# そのつど新たなマイナーバージョンを付与してリリースします。
# （モジュール差し替えは技術移転を受けた事業者しか対応できません）

# サニタイザーのバージョン
# (2 -> バージョン２　3 -> バージョン３)
VERSION=3

#####################################################################
# サニタイズ処理の安定性を維持するための設定
# サニタイザーに処理上の負荷がかかった場合、安定的に処理を維持するための設定
# です。
# ２コア CPU、メモリ４ GB の環境で十分に稼働する設定値としていますが、導入先の
# 機器の性能に応じてこれらの値は調整してください。

```

(次のページに続く)

(前のページからの続き)

```
# メモリの使用量が規定値を超えた場合にサニタイズ処理を中止する
# (数値はパーセント表示)
# 処理を中止した場合、「処理されなかったファイル名_resourceReport.txt」と
# いうレポートファイルを出力します。この場合、そのファイルのサニタイズ処理を
# 再度行ってください。
MEMLIMIT=70

# スワップ領域の使用量が規定値を超えた場合にサニタイズ処理を中止する
# (数値はパーセント表示)
# 処理を中止した場合、「処理されなかったファイル名_resourceReport.txt」と
# いうレポートファイルを出力します。この場合、そのファイルのサニタイズ処理を
# 再度行ってください。
SWAPLIMIT=50

# ロードアベレージが規定値を超えた場合にサニタイズ処理を中止する
# (数値はロードアベレージ)
# 処理を中止した場合、「処理されなかったファイル名_resourceReport.txt」と
# いうレポートファイルを出力します。この場合、そのファイルのサニタイズ処理を
# 再度行ってください。
LOADAVELIMIT=50

# ファイルコンバートのタイムアウト時間。これを過ぎるとサニタイズを中止する
# (数値は秒数)
# サニタイズ処理の過程でいくつかの種類のファイルは形式を変換する処理を行って
# います。
# 処理を中止した場合、「処理されなかったファイル名_errorReport.txt」という
# レポートファイルを出力します。この場合、そのファイルのサニタイズ処理を再度
# 行ってください。
CONVERTTIMEOUT=60

#########################################################################
# サニタイズ処理を行う前の処理に関する設定
# サニタイザーはファイルのサニタイズ処理を行う前に、関連するいくつかの処理を
# 行っています。
# これによりサニタイズ処理の品質を高めたり、処理速度の向上を実現しています。

# 事前にブロックする拡張子を指定する
# (ブロックする拡張子をカンマ区切りで指定。大文字小文字は無関係。
#  指定しない場合は空白)
# サニタイザーはファイル判別、無害化処理を行う前に強制的にブロックするファイルを
# 拡張子で指定できます。
# このブロックはあらゆる処理よりも先に行われます。
BLOCKEXTLIST=

# ファイル名に禁則文字が含まれている場合の挙動を設定する
```

(次のページに続く)

(前のページからの続き)

```
# (CHANGE -> ファイル名変更
#  ABORT -> ファイル名変更を行わず、レポートファイルを出力する
#  CONTINUE -> ファイル名変更を行わず、そのまま処理する（推奨）)
# サニタイザーはファイル名に禁則文字（空白や OS で使えない文字等）が含まれている場合、
# それらを _ に置き換えて処理を行います。
PROHIBITEDCHAR=CONTINUE

# 事前に対象ファイルにウイルスチェックをする
# (TRUE -> する FALSE -> しない（推奨）)
# サニタイザーは内部で ClamAV というアンチウイルスソフトを稼働させています。
# これにより既知のマルウェアについてはサニタイズ処理前に排除することができます。
# ただし、ClamAV の処理には多くのメモリを消費することと、アンチウイルスパターン
# ファイルの更新にインターネット接続を行うことの理由により、サニタイザーで
# ウイルスチェックを行わず、別のソリューションで処理させたほうがよい場合も
# あります。
# 標準では FALSE にしてありますが、導入環境に応じて設定を変更してください。
ANTIVIRUS=FALSE

# 事前に対象ファイルに不正コード混入チェックをする
# (TRUE -> する（推奨） FALSE -> しない)
# サニタイズ処理の前に対象ファイルの中身を走査し、不正なコードが含まれて
# いないかを
# チェックしています。ここでは明らかに疑わしい処理のみをチェックしています。
# 伝統的な標的型攻撃ファイルの多くはこのチェックで排除できますので、標準では
# TRUE にしてあります。
CHKMALICIOUS=TRUE

# マクロの有無を事前にチェックし、マクロが明らかに含まれていない場合は
# サニタイズしないで通す
# (TRUE -> _marked フラグをつけて通す（推奨） FALSE -> 常にサニタイズする)
# サニタイズ処理の前に対象ファイルの中身を走査し、マクロやオブジェクトの
# 埋め込みがされていないファイルについてはサニタイズ処理を行わないで通過
# させます。
# ここでのチェックの観点はマクロ、オブジェクトの埋め込みの有無であり、不正な
# ファイル形式や判別不明なバイナリコード等については、この設定に関わらず
# チェックして排除しています。
MACROAVAILFIRST=TRUE

#####################################################################
# Office ファイルのサニタイズ処理の挙動に関する設定
# サニタイズ処理には様々な手法があり、出力されるファイルの形式や態様なども
# 異なってきます。運用の場面などに応じて、最適な組み合わせで設定してください。

# パスワード付きの Office ファイルを通す
# (TRUE -> 通す FALSE -> Warning レポートを通知してファイルはブロックする（Ver.2 推奨）
```

(次のページに続く)

(前のページからの続き)

```
#  HOLD ->  保留にして受け渡し承認を受ける（Ver.3 推奨）)
# Microsoft Office のファイルのうち、パスワードが付加されたものについては、
# ファイルの内容がマクロを含めて暗号化されますので、サニタイズ処理ができません。
# ファイルの発出元が明らかである場合は、発出元を信頼してサニタイズ処理をせずに
# 受け取る運用も考えられますので、導入先のポリシーに応じて設定してください。
ALLOWMSPASS=HOLD

# RMS で暗号化された Office ファイルを通す
# (TRUE -> 通す  FALSE -> Warning レポートを通知してファイルはブロックする（Ver.2 推奨）
#  HOLD ->  保留にして受け渡し承認を受ける（Ver.3 推奨）)
# Microsoft Rights Management Server（RMS）により暗号化された Office
# ファイルについては、ファイルの内容がマクロを含めて暗号化されますので、
# サニタイズ処理ができません。
# ファイルの発出元が明らかである場合は、発出元を信頼してサニタイズ処理をせずに
# 受け取る運用も考えられますので、導入先のポリシーに応じて設定してください。
ALLOWMSRMS=HOLD

# Office2007 以降のファイルについて、PC に悪影響のあるマクロのみを削除する
# (TRUE -> 削除する（推奨）  FALSE -> 全てのマクロを削除する)
# 全てのマクロが悪影響を与えるものではないという考えから、マクロの内容を
# 精査し、PC に悪影響を与える可能性の高いマクロが含まれている場合のみ、
# マクロを削除するようにします。
DELBADCODEONLY=TRUE

# Office ファイルのサニタイズ処理基準を厳格に適用するか
# (TRUE -> 厳格に適用  FALSE -> 厳格に適用しない（推奨）)
# Office ファイルをサニタイズする前に、ファイル中でリスクになり得る情報まで厳格に
# 判断するかの基準です。
# TRUE の場合、リスク要因が含まれているだけでなく、全く別の観点からサニタイズを
# するか否かの判断を行いますが、誤検知のリスクもあります。
# FALSE の場合は、単純にリスク要因が埋め込まれているか否かだけを判断します。
EXSEEKSTRICT=FALSE

# Office ファイル、ODF 形式ファイルのサニタイズ時に PDF を同時に生成する
# (TRUE -> 生成する  FALSE -> 生成しない（推奨）)
# サニタイズ処理後の Office ファイルの内容を確認するために、同じ内容の PDF
# ファイルを同時生成することができます。ひとまず内容を確認するだけならば、
# ここで生成した PDF を取り込んで利用するという運用で安全性を高めることも
# できます。
WITHCREATEPDF=FALSE

##########################################################################
# PDF ファイルのサニタイズ処理の挙動に関する設定
# サニタイズ処理には様々な手法があり、出力されるファイルの形式や態様なども
# 異なってきます。運用の場面などに応じて、最適な組み合わせで設定してください。
```

(次のページに続く)

(前のページからの続き)

```
# パスワード付きの PDF ファイルを通す
# (TRUE -> 通す FALSE -> パスワード要求レポートを出力し再処理を促す（推奨）)
#  HOLD ->  保留にして受け渡し承認を受ける)
# PDF ファイルのうち、パスワードが付加されたものについては、ファイルの内容が
# スクリプトを含めて暗号化されますので、サニタイズ処理ができません。
# ファイルの発出元が明らかである場合は、発出元を信頼してサニタイズ処理をせずに
# 受け取る運用も考えられますので、導入先のポリシーに応じて設定してください。
ALLOWPDFPASS=FALSE

# PDF ファイルのサニタイズ処理基準を厳格に適用するか
# (TRUE -> 厳格に適用 FALSE -> JavaScript や埋め込みファイルがある場合のみ（推奨）)
# PDF ファイルをサニタイズする前に、ファイル中でリスクになり得る情報まで厳格に
# 判断するかの基準です。
# TRUE の場合、リスク要因が含まれているだけでなく、全く別の観点からサニタイズを
# するか否かの判断を行いますが、誤検知のリスクもあります。
# FALSE の場合は、単純にリスク要因が埋め込まれているか否かだけを判断します。
SANIPDFSTRICT=FALSE

# PDF のサニタイズ処理において、スクリプトの除去を行うか、画像として再作成するか
# (TRUE -> スクリプトの除去（推奨） FALSE -> 画像として再作成)
# スクリプトの除去を行うと、テキストのコピー&ペーストが可能ですが、
# 一部のオブジェクトが残る可能性があります。画像として再作成すると、オブジェクトは
# 完全に消去されますが、処理に時間が掛かり、テキストの抽出はできません。
# 導入の場面により手法を選択してください。
PDFCONVERTDOC=TRUE

# PDF ファイルを画像として再作成する場合の解像度
# (数値は解像度 (DPI). 数値が大きくなると処理時間、ファイルサイズが増加する)
# 上記設定で、画像として再作成する場合の画質を設定できます。
PDFCONVERTDENSITY=300

# PDF ファイルを画像として再作成する場合の一時ファイル領域の上限値
# (数値はディスクサイズ (GB). 余剰ディスクサイズ内の数値とすること)
PDFCONVERTTMPLIMIT=20

#########################################################################
# その他のファイルのサニタイズ処理の挙動に関する設定
# サニタイズ処理には様々な手法があり、出力されるファイルの形式や態様なども
# 異なってきます。運用の場面などに応じて、最適な組み合わせで設定してください。

# 一太郎形式ファイルをサニタイズする
# (TRUE -> サニタイズする（推奨） FALSE -> _marked フラグをつけて通す)
# 一太郎形式（その他 JustSystem 系のファイルも同様）のサニタイズ処理を
# 行います。標準では TRUE にしています。
```

(次のページに続く)

(前のページからの続き)

```
SANITIZEJUST=TRUE

# ドキュワークス形式ファイルをサニタイズする
# (TRUE -> 強制的に PDF 形式へサニタイズする（Ver.2 推奨）
#  FALSE -> _marked フラグをつけて通す
#  HOLD ->  保留にして受け渡し承認を受ける（Ver.3 推奨）)
# ドキュワークス形式（XDW) ファイルのサニタイズ処理を行います。ドキュワークス
# 形式はオブジェクトやファイルの添付が可能な形式なため、現時点では強制的に
# PDF に変換する手法を採用しています。
# 注意：この機能は外部モジュールを使っているので、Ver1.24.0 から利用可能
# です。それ以外のバージョンでは、TRUE に設定しても FALSE 相当の挙動になります。
SANITIZEXDW=HOLD

# 疑義のある画像形式ファイル（JPEG, GIF, BMP, PNG 等）を強制的にサニタイズする
# (TRUE -> 強制的にサニタイズする（推奨） FALSE -> 警告メッセージを出力する)
# 画像ファイルの中で疑義のあるものについて、強制的にサニタイズ処理を行います。
# 標準では TRUE にしています。
IMGFORCESANITIZE=TRUE

#####################################################################
# アーカイブファイルのサニタイズ処理の挙動に関する設定
# サニタイズ処理には様々な手法があり、出力されるファイルの形式や態様なども
# 異なってきます。運用の場面などに応じて、最適な組み合わせで設定してください。

# LZH ファイルの解凍処理を行う
# (TRUE -> LZH ファイルを解凍して work フォルダに保存。SANIEXTRACTED に従う。（推奨）
# FALSE -> ブロックする)
# 導入先のポリシーにより、アーカイブファイルの形式に LZH を採用していない場合
# には、この形式をブロックすることができます。
ALLOWLZH=TRUE

# CAB ファイルの解凍処理を行う
# (TRUE -> CAB ファイルを解凍して work フォルダに保存。SANIEXTRACTED に従う。（推奨）
# FALSE -> ブロックする)
# 導入先のポリシーにより、アーカイブファイルの形式に CAB を採用していない場合
# には、この形式をブロックすることができます。
ALLOWCAB=TRUE

# 7z(7-Zip) ファイルの解凍処理を行う
# (TRUE -> 7z ファイルを解凍して work フォルダに保存。SANIEXTRACTED に従う。（推奨）
# FALSE -> ブロックする)
# 導入先のポリシーにより、アーカイブファイルの形式に 7z を採用していない場合
# には、この形式をブロックすることができます。
ALLOW7Z=TRUE
```

(次のページに続く)

(前のページからの続き)

```
# RAR ファイルの解凍処理を行う
# (TRUE -> RAR ファイルを解凍して work フォルダに保存。SANIEXTRACTED に従う。(推奨)
# FALSE -> ブロックする)
# 導入先のポリシーにより、アーカイブファイルの形式に RAR を採用していない場合
# には、この形式をブロックすることができます。
ALLOWRAR=TRUE

# tar ファイルの解凍処理を行う
# (TRUE -> tar ファイルを解凍して work フォルダに保存。SANIEXTRACTED に従う。(推奨)
# FALSE -> ブロックする)
# 導入先のポリシーにより、アーカイブファイルの形式に tar を採用していない場合
# には、この形式をブロックすることができます。
ALLOWTAR=TRUE

# gzip/gz ファイルの解凍処理を行う
# (TRUE -> gzip/gz ファイルを解凍して work フォルダに保存。SANIEXTRACTED に従う。(推奨)
# FALSE -> ブロックする)
# 導入先のポリシーにより、アーカイブファイルの形式に gzip/gz を採用していない場合
# には、この形式をブロックすることができます。
ALLOWGZ=TRUE

# パスワード付き Zip ファイルの解凍を output 側から処理する
# (TRUE -> output 側からの処理で解凍する(推奨)　FALSE -> output 側から処理
# しない)
# パスワード付き Zip ファイルは解凍処理ができないため、通常は
# 「ファイル名_encryptReport.txt」というレポートファイルを出力します。
# このレポートファイルにパスワードを追記して保存することで、パスワードを
# 使った解凍処理をする機能を有効にするかを設定できます。
DECRYPTZIP=TRUE

# アーカイブファイルを解凍後、個別にサニタイズする
# (TRUE -> サニタイズする(推奨)　FALSE -> サニタイズしない)
# アーカイブファイルは、解凍処理されると、work フォルダに個別のフォルダを
# 生成してその中に解凍後のファイルが保存されます。これら解凍後のファイルを
# 自動的にサニタイズ処理させることができます。
# 注意：サニタイズ処理に失敗した場合には、work フォルダ中のファイルを手動で
# サニタイズ処理してください。
SANIEXTRACTED=TRUE

# アーカイブファイル解凍後のファイルサイズの上限値
# (数値はバイト数)
# 圧縮率の高いアーカイブファイルを解凍した際に、解凍後のファイルが
# ディスクの容量を超過する可能性があるため、ここで上限値を設定します。
EXTRACTEDSIZE=512000000
```

(次のページに続く)

(前のページからの続き)

```
# アーカイブファイルの再圧縮処理を行うか（オプション機能）
# （TRUE -> 行う FALSE -> 行わない）
# アーカイブファイルは解凍後、個別に無害化処理を経て、Output フォルダに出力する仕様と
# なっていますが、これらのファイルを再び Zip に再圧縮する機能を有効にするかを
# 設定します。この機能はオプションのため、設定値は導入先により異なります。
XXXXXXXXXXXX=TRUE

# Zip ファイル再圧縮オプションにおけるタイムアウト時間
# (数値は分)
# 上記オプションにおける無害化処理のタイムアウト時間を設定します。
# この時間を経過すると、その段階で揃っているファイルだけを Zip 再圧縮します。
ZIPTIMEOUT=10

# 使用する端末の OS
# Windows7 は Zip ファイルの文字コードが ShiftJIS として処理されるバグがあります。
# Windows7 環境で Zip ファイル解凍中に文字化けが発生する場合には、使用する OS を
# 明示的に設定してください。
# （WINDOWS7 -> Windows7 OTHER -> それ以外（推奨））
CLIENTOS=OTHER

#########################################################################
# メールファイルのサニタイズ処理の挙動に関する設定
# サニタイズ処理には様々な手法があり、出力されるファイルの形式や態様なども
# 異なってきます。運用の場面などに応じて、最適な組み合わせで設定してください。

# メール添付ファイルのサニタイズをする（メール無害化）
# (TRUE -> サニタイズする（推奨） FALSE -> サニタイズしない)
# メール無害化を目的としてサニタイザーを使う場合には、この設定を TRUE にして
# ください。
# MailDir 形式のメールファイルを読み取り、その中に含まれている添付ファイルを
# 抽出し、work フォルダに保存、サニタイズ処理までを一連で行います。
# 注意：サニタイズ処理に失敗した場合には、work フォルダ中のファイルを手動で
# サニタイズ処理してください。
# 注意：メール無害化の導入を行う場合には、連携するシステムとの調整が必要
# ですので、個別にお問い合わせください。
SANIMAIL=TRUE

#########################################################################
# サニタイズ処理全般に関わる設定
# サニタイザー全体に関わる設定です。導入先のポリシーや連携するシステムの
# 制約で設定を変更する場面があるものをまとめています。

# サニタイズ後のファイルを区別するためにプロセス ID を付記する
# (TRUE -> プロセス ID を付記する（推奨） FALSE -> プロセス ID を付記しない)
# TRUE にすることで、output フォルダへ出力するファイル名にプロセス ID を付記
```

(次のページに続く)

(前のページからの続き)

```
# するようになります。これにより、(おそらく) 出力ファイル名が重複することが
# なくなります。
# また、サニタイザーのログから、処理対象のファイルの特定も速やかに行えるように
# なります。
# なお、FALSE にした場合、ファイル名の重複が発生すると、後からサニタイズ処理
# されたファイルが残る仕組みとなります。
ADDPROCESSID=TRUE

# 判別不能なドキュメントファイルを拡張子で識別する
# (TRUE -> 拡張子で識別する (推奨)
#  FALSE -> 拡張子で識別せずに判別不能のまま処理する)
# サニタイザーは基本的にファイルの拡張子は見ず、ファイルの中身からファイルの
# 種別を識別します。
# しかし一部のファイルにおいては、ファイルの中身からでは識別できないものも
# あります。その場合、拡張子を手がかりにファイルの種別を推測します。
# 標準では TRUE としていますが、拡張子偽装に伴うリスクを排除するのならば、
# FALSE にしてください。(その場合、種別不明でブロックされる確率が増えます)
EXTDIST=TRUE

# ファイル処理後、Web インターフェースのデータベースを逐次更新する
# (TRUE -> 逐次データベースを更新する (推奨)
#  FALSE -> 定期的にデータベースを更新する)
# サニタイザーはサニタイズの処理の後に、Web インターフェース (ownCloud) の
# データベースを更新して、はじめて Web インターフェースの一覧にファイルが表示
# される仕組みとなっています。
# 設定を TRUE にすることで、サニタイズ処理後、直ちにデータベースの更新を行います。
# これにより画面への反映時間は短くなりますが、その代わりに大量のファイルを
# まとめて処理すると、頻繁にデータベースの書き換えが発生するので、全体的な
# パフォーマンスが低下します。
# FALSE にすると、データベースの更新をサニタイズ処理と関係なく定期的 (1分に
# 1回) に行います。パフォーマンスの低下はありませんが、その代わりに画面への
# 反映が最大1分後になります。導入先の状況にあわせて、設定してください。
RESCANDB=TRUE

# WARNING.BINARY フラグがついたファイルを通す
# (TRUE -> 通す  FALSE -> Warning レポートを通知してファイルはブロックする (推奨) )
# サニタイズ処理の過程で、最終的に判別不能なファイル (バイナリファイル) が
# 残った場合、このファイルをどのように処理するかを設定します。
# TRUE にすることで、該当ファイルは、WARNING.BINARY という識別子が付いた
# 状態で output フォルダに出力されます。ただし、サニタイズ処理もされない状態の
# ファイルのため相応のリスクを負うことになります。リスクについて十分理解した
# 上で設定してください。標準では FALSE にしています。
THROUGHOUT=FALSE

# WARNING.BINARY フラグがついた実行形式ファイルを通す
```

(次のページに続く)

(前のページからの続き)

```
# (TRUE -> 通す FALSE -> Warning レポートを通知してファイルはブロックする（推奨）)
# サニタイズ処理の過程で、最終的に判別不能な実行ファイル（バイナリファイル）
# が残った場合、このファイルをどのように処理するかを設定します。
# TRUE にすることで、該当ファイルは、WARNING.BINARY という識別子が付いた
# 状態で output フォルダに出力されます。ただし、サニタイズ処理もされない状態の
# ファイルであり、実行可能である分、上述のバイナリファイルよりも高いリスクを
# 負うことになります。
# リスクについて十分理解した上で設定してください。標準では FALSE にしています。
EXECTHROUGHOUT=FALSE

#########################################################################
# トラストパスに関わる設定
# サニタイザーのオプションである、トラストパスに関わる設定です。
# トラストパスの導入を検討される場合には、個別にお問い合わせください。

# トラストパスによる復号ファイルを信頼し、そのまま取り込む
# (TRUE -> 取り込む（推奨） FALSE -> 取り込まずに work フォルダに保存する)
# ファイルの送信者があらかじめファイルに電子署名を付与し、その電子署名が
# 検証されたファイルは、サニタイズ処理を行うことなく、Output フォルダに出力
# する機能です。
# なお、この設定を FALSE にすると、署名検証後のファイルは work フォルダに
# 保存され、別途サニタイズ処理を手動で行ってもらうことになります。
# また署名検証に失敗したり、トラストパスの設定がない場合には、
# 「ファイル名_errorReport.txt」というレポートファイルを出力して、
# 処理は終了します。
TRUSTPASS=TRUE

# トラストパスクライアントを有効化する
# (SANITIZE -> 無害化して有効化  THROUGHT -> 無害化せずに有効化 FALSE -> 有効化しない)
# トラストパスクライアントは、ファイルをトラストパスのみで受け取ることができるよう、
# 署名付き暗号化ファイルとして出力する機能です。
# 媒体経由でファイルを受け渡す際に、この機能を使ってファイルを書き出す運用を想定しています。
# トラストパスクライアントは、output フォルダを監視して動作します。
# このパラメータで FALSE を設定した場合は、output フォルダの監視を行いません。
# SANITIZE を設定した場合は、output フォルダに出力された無害化処理後のファイルを
# 署名付き暗号化ファイルとして signed フォルダに出力します。
# また、THROUGHT を設定した場合は、無害化処理自体をスキップして output にファイルを出力し、
# それらのファイルをそのまま署名付き暗号化ファイルとして signed フォルダに出力します。
TRUSTCLIENT=FALSE

# トラストパスで使用する自分の鍵 ID
# (値は GPG で生成された鍵 ID)
# トラストパスクライアントはファイルを出力する際に、間違いなく自分から生成・送信され、
# 改ざんされていないことを示すために、電子署名を付加します。
# この電子署名を行うための鍵 ID をここで登録します。
```

(次のページに続く)

(前のページからの続き)

```
TRUSTUSER=

# トラストパスで使用する相手の鍵 ID
# (値は GPG で生成された鍵 ID)
# トラストパスクライアントはファイルを出力する際に、第三者にファイルの中身を見られないように
# するために暗号化を施します。暗号化はファイルを受け渡す相手を指定して行います。
# この暗号化を行うための相手の鍵 ID をここで登録します。
# （通常はファイルを受け取るサニタイザーに登録している鍵 ID となります）
TRUSTADMIN=

########################################################################
# sanipass/sanicrypt に関わる設定
# サニタイザーのオプションである、sanipass 及び sanicrypt に関わる設定です。
# sanipass/sanicrypt の導入を検討される場合には、個別にお問い合わせください。
# 標準構成ではこれらの機能がは正常に稼働しません。

# sanipass/sanicrypt の出力先フォルダ
# (値はフォルダ名称)
# sanipass/sanicrypt により出力されるフォルダを設定します。これらの機能が
# 有効でない場合には、この値を参照することはありませんので、どのような値でも
# 差し支えありません。(念のためコメントアウトしてあります)
# OUTPUTFOLDER=""

# sanicrypt におけるパスワード付与形態
# (FIXED -> 固定設定　RULED -> ルールに基づく設定　LIST -> パスコードリスト
#  REPORT -> パスワードをレポートファイルで出力)
# sanicrypt により、パスワード付き Zip ファイルの出力をする際、パスワードの
# 付与形態を選択することができます。
CRYPTPASSCODE="REPORT"

########################################################################
# フォルダ内のファイル削除に関わる設定
# サニタイザーで処理した input、output、work フォルダ、また上述の
# スルーパスで設定した出力先フォルダについて、定期的にフォルダ内のファイルを
# 削除するための設定です。

# 毎日午前 4 時に各フォルダの古いファイルとゴミ箱の中身をクリーンアップする
# (TRUE -> クリーンアップする（推奨）　FALSE -> クリーンアップしない)
# TRUE にすることで、毎日午前 4 時にフォルダ内の古いファイルを削除します。
# FALSE にすると、ファイルの削除は行われませんので、利用者が手作業で
# ファイルを削除していただくことになります。
CLEARFOLDER=TRUE

# 古いファイルとゴミ箱の中身をクリーンアップする期間
# (数値はクリーンアップせずに保持しておく日数)
```

(次のページに続く)

(前のページからの続き)

```
# 標準では 2 日間（48 時間）経過したファイルから順に削除されます。この値を
# 変更することで、フォルダ内に保持しておく期間を設定することができます。
STOREDAYS=2

######################################################################
# サニタイザーの導入組織に関する設定
# サニタイザーをご利用いただいている組織に関する情報を設定します。

# 自治体コード
# （5 桁数字あるいは 5 桁数字と 1 桁英文字）
LGCODE=

######################################################################
# サニタイザーの構成に関わる設定
# ネットーワーク間の受け渡し（サニタイザープロ）、負荷分散のためのクラスタ
# 構成など、サニタイザーを複数台で構成する場合に、それぞれのサニタイザーが
# どのような役割を持つのかを設定します。
# 注意：サニタイザーを複数台で構成する場合には、複数台の間の調整が必要です
# ので、個別にお問い合わせください。

# サニタイザーのサーバ種別
# (INPUT -> input サーバ OUTPUT -> output サーバ
# STANDALONE -> 一台構成
# ３セグメント間の受け渡しの場合は個別に設定要)
# サニタイザーをどのような役割で稼働させるのかを設定します。
# 標準では STANDALONE となります。
# サニタイザープロの場合、input サーバの役割となるサーバには INPUT、
# output サーバの役割となるサーバには OUTPUT を設定してください。
SERVERTYPE=STANDALONE

# サニタイザープロをクラスタ構成で稼働する
# (TRUE -> クラスタ構成で稼働する FALSE -> シングル構成で稼働する)
# サニタイザープロをクラスタ構成で稼働させる場合、input サーバが複数台と
# なる構成を取りますが、その際には、この値を TRUE にしてください。
# TRUE にすることで、サニタイズ処理後のファイルが output フォルダに出力した
# 場合、どのサーバから送られてきたものかをファイル名に識別子として付記します。
CLUSTER=FALSE

# input フォルダのファイルをアーカイブ用に別途保管する
# (TRUE -> アーカイブフォルダに保管する FALSE -> 保管しない)
# サニタイザーをクラスタ構成で稼働させている際に、input フォルダに保存されて
# いるファイルをアーカイブ用に保管するための設定です。
# 注意：アーカイブ用のフォルダを別途設定する必要がありますので、
# クラスタ構成を構築する場合以外はこの設定を TRUE にしないでください。
ALLOWARCHIVE=FALSE
```

16.2 サニタイザーポリシーファイル Ver.3.7.0

```
# This File is Sanitzer Policy for sanitize files.
# Created and ReWrited by Hiro KAWAGUCHI 20231201
# サニタイザー適用バージョン Ver.3.7.0 以降

#########################################################################
# このファイルについて
# このファイルはサニタイザーの動作状況を制御するための設定ファイルです。
# それぞれ「項目=設定値」という構成になっています。
# 設定に際しては、項目名、設定値とも半角英数大文字とし、項目と設定値の間には
# 空白を入れないようにしてください。

#########################################################################
# サニタイザーのバージョンに対する考え方
# サニタイザーは主に２つのリリース体系があります。
# OVA ファイルにて提供されたもの -> 常にマイナーバージョン（最下位の数字）が
# 0 になります。
# モジュール差し替えにて提供されたもの -> マイナーバージョンが 0 以外と
# なります。
#
# 一般的にはマイナーバージョンが 0 のものをお使いいただいていますが、機能改善、
# 機能追加、個別カスタマイズにより、モジュールを差し替えたものについては、
# そのつど新たなマイナーバージョンを付与してリリースします。
# （モジュール差し替えは技術移転を受けた事業者しか対応できません）

# サニタイザーのバージョン
# (2 -> バージョン２  3 -> バージョン３)
VERSION=3

#########################################################################
# サニタイズ処理の安定性を維持するための設定
# サニタイザーに処理上の負荷がかかった場合、安定的に処理を維持するための設定
# です。
# ２コア CPU、メモリ４ GB の環境で十分に稼働する設定値としていますが、導入先の
# 機器の性能に応じてこれらの値は調整してください。

# メモリの使用量が規定値を超えた場合にサニタイズ処理を中止する
# (数値はパーセント表示)
# 処理を中止した場合、「処理されなかったファイル名_resourceReport.txt」と
# いうレポートファイルを出力します。この場合、そのファイルのサニタイズ処理を
# 再度行ってください。
MEMLIMIT=70

# スワップ領域の使用量が規定値を超えた場合にサニタイズ処理を中止する
```

(次のページに続く)

(前のページからの続き)

```
# (数値はパーセント表示)
# 処理を中止した場合、「処理されなかったファイル名_resourceReport.txt」と
# いうレポートファイルを出力します。この場合、そのファイルのサニタイズ処理を
# 再度行ってください。
SWAPLIMIT=50

# ロードアベレージが規定値を超えた場合にサニタイズ処理を中止する
# (数値はロードアベレージ)
# 処理を中止した場合、「処理されなかったファイル名_resourceReport.txt」と
# いうレポートファイルを出力します。この場合、そのファイルのサニタイズ処理を
# 再度行ってください。
LOADAVELIMIT=50

# ファイルコンバートのタイムアウト時間。これを過ぎるとサニタイズを中止する
# (数値は秒数)
# サニタイズ処理の過程でいくつかの種類のファイルは形式を変換する処理を行って
# います。
# 処理を中止した場合、「処理されなかったファイル名_errorReport.txt」という
# レポートファイルを出力します。この場合、そのファイルのサニタイズ処理を再度
# 行ってください。
CONVERTTIMEOUT=60

#########################################################################
# サニタイズ処理を行う前の処理に関する設定
# サニタイザーはファイルのサニタイズ処理を行う前に、関連するいくつかの処理を
# 行っています。
# これによりサニタイズ処理の品質を高めたり、処理速度の向上を実現しています。

# 事前にブロックする拡張子を指定する
# (ブロックする拡張子をカンマ区切りで指定。大文字小文字は無関係。
#  指定しない場合は空白)
# サニタイザーはファイル判別、無害化処理を行う前に強制的にブロックするファイルを
# 拡張子で指定できます。
# このブロックはあらゆる処理よりも先に行われます。
BLOCKEXTLIST=

# ファイル名に禁則文字が含まれている場合の挙動を設定する
# (CHANGE -> ファイル名変更
#  ABORT -> ファイル名変更を行わず、レポートファイルを出力する
#  CONTINUE -> ファイル名変更を行わず、そのまま処理する（推奨）)
# サニタイザーはファイル名に禁則文字（空白や OS で使えない文字等）が含まれている場合、
# それらを _ に置き換えて処理を行います。
PROHIBITEDCHAR=CONTINUE

# 事前に対象ファイルにウイルスチェックをする
```

(次のページに続く)

(前のページからの続き)

```
# (TRUE -> する FALSE -> しない（推奨）)
# サニタイザーは内部で ClamAV というアンチウイルスソフトを稼働させています。
# これにより既知のマルウェアについてはサニタイズ処理前に排除することができます。
# ただし、ClamAV の処理には多くのメモリを消費することと、アンチウイルスパターン
# ファイルの更新にインターネット接続を行うことの理由により、サニタイザーで
# ウイルスチェックを行わず、別のソリューションで処理させたほうがよい場合も
# あります。
# 標準では FALSE にしてありますが、導入環境に応じて設定を変更してください。
ANTIVIRUS=FALSE

# 事前に対象ファイルに不正コード混入チェックをする
# (TRUE -> する（推奨） FALSE -> しない)
# サニタイズ処理の前に対象ファイルの中身を走査し、不正なコードが含まれて
# いないかを
# チェックしています。ここでは明らかに疑わしい処理のみをチェックしています。
# 伝統的な標的型攻撃ファイルの多くはこのチェックで排除できますので、標準では
# TRUE にしてあります。
CHKMALICIOUS=TRUE

# マクロの有無を事前にチェックし、マクロが明らかに含まれていない場合は
# サニタイズしないで通す
# (TRUE -> _marked フラグをつけて通す（推奨） FALSE -> 常にサニタイズする)
# サニタイズ処理の前に対象ファイルの中身を走査し、マクロやオブジェクトの
# 埋め込みがされていないファイルについてはサニタイズ処理を行わないで通過
# させます。
# ここでのチェックの観点はマクロ、オブジェクトの埋め込みの有無であり、不正な
# ファイル形式や判別不明なバイナリコード等については、この設定に関わらず
# チェックして排除しています。
MACROAVAILFIRST=TRUE

#########################################################################
# Office ファイルのサニタイズ処理の挙動に関する設定
# サニタイズ処理には様々な手法があり、出力されるファイルの形式や態様なども
# 異なってきます。運用の場面などに応じて、最適な組み合わせで設定してください。

# パスワード付きの Office ファイルのパスワードを解除する（Ver.3.7 及び Ver.4 より）
# (TRUE -> パスワード解除する（Ver.4 のみ）
#  FALSE -> パスワード解除せずに ALLOWMSPASS の設定に従う（Ver.3 推奨）)
# Microsoft Office のファイルのうち、パスワードが付加されたものについては、
# パスワードを解除したうえで、無害化処理を行うことになります。
DECRYPTMS=TRUE

# パスワード付きの Office ファイルを通す
# (TRUE -> 通す FALSE -> Warning レポートを通知してファイルはブロックする（Ver.2 推奨）
#  HOLD ->  保留にして受け渡し承認を受ける（Ver.3 推奨）)
```

(次のページに続く)

(前のページからの続き)

```
# Microsoft Office のファイルのうち、パスワードが付加されたものについては、
# ファイルの内容がマクロを含めて暗号化されますので、サニタイズ処理ができません。
# ファイルの発出元が明らかである場合は、発出元を信頼してサニタイズ処理をせずに
# 受け取る運用も考えられますので、導入先のポリシーに応じて設定してください。
ALLOWMSPASS=HOLD

# RMS で暗号化された Office ファイルを通す
# (TRUE -> 通す  FALSE -> Warning レポートを通知してファイルはブロックする（Ver.2 推奨）
#  HOLD ->  保留にして受け渡し承認を受ける（Ver.3 推奨）)
# Microsoft Rights Management Server（RMS）により暗号化された Office
# ファイルについては、ファイルの内容がマクロを含めて暗号化されますので、
# サニタイズ処理ができません。
# ファイルの発出元が明らかである場合は、発出元を信頼してサニタイズ処理をせずに
# 受け取る運用も考えられますので、導入先のポリシーに応じて設定してください。
ALLOWMSRMS=HOLD

# Office2007 以降のファイルについて、PC に悪影響のあるマクロのみを削除する
# (TRUE -> 削除する（推奨）  FALSE -> 全てのマクロを削除する)
# 全てのマクロが悪影響を与えるものではないという考えから、マクロの内容を
# 精査し、PC に悪影響を与える可能性の高いマクロが含まれている場合のみ、
# マクロを削除するようにします。
DELBADCODEONLY=TRUE

# Office ファイルのサニタイズ処理基準を厳格に適用するか
# (TRUE -> 厳格に適用  FALSE -> 厳格に適用しない（推奨）)
# Office ファイルをサニタイズする前に、ファイル中でリスクになり得る情報まで厳格に
# 判断するかの基準です。
# TRUE の場合、リスク要因が含まれているだけでなく、全く別の観点からサニタイズを
# するか否かの判断を行いますが、誤検知のリスクもあります。
# FALSE の場合は、単純にリスク要因が埋め込まれているか否かだけを判断します。
EXSEEKSTRICT=FALSE

# Office ファイル、ODF 形式ファイルのサニタイズ時に PDF を同時に生成する
# (TRUE -> 生成する  FALSE -> 生成しない（推奨）)
# サニタイズ処理後の Office ファイルの内容を確認するために、同じ内容の PDF
# ファイルを同時生成することができます。ひとまず内容を確認するだけならば、
# ここで生成した PDF を取り込んで利用するという運用で安全性を高めることも
# できます。
WITHCREATEPDF=FALSE

#########################################################################
# PDF ファイルのサニタイズ処理の挙動に関する設定
# サニタイズ処理には様々な手法があり、出力されるファイルの形式や態様なども
# 異なってきます。運用の場面などに応じて、最適な組み合わせで設定してください。
```

(次のページに続く)

(前のページからの続き)

```
# パスワード付きの PDF ファイルを通す
# (TRUE -> 通す FALSE -> パスワード要求レポートを出力し再処理を促す（推奨）)
#  HOLD ->  保留にして受け渡し承認を受ける)
# PDF ファイルのうち、パスワードが付加されたものについては、ファイルの内容が
# スクリプトを含めて暗号化されますので、サニタイズ処理ができません。
# ファイルの発出元が明らかである場合は、発出元を信頼してサニタイズ処理をせずに
# 受け取る運用も考えられますので、導入先のポリシーに応じて設定してください。
ALLOWPDFPASS=FALSE

# PDF ファイルのサニタイズ処理基準を厳格に適用するか
# (TRUE -> 厳格に適用 FALSE -> JavaScript や埋め込みファイルがある場合のみ（推奨）)
# PDF ファイルをサニタイズする前に、ファイル中でリスクになり得る情報まで厳格に
# 判断するかの基準です。
# TRUE の場合、リスク要因が含まれているだけでなく、全く別の観点からサニタイズを
# するか否かの判断を行いますが、誤検知のリスクもあります。
# FALSE の場合は、単純にリスク要因が埋め込まれているか否かだけを判断します。
SANIPDFSTRICT=FALSE

# PDF のサニタイズ処理において、スクリプトの除去を行うか、画像として再作成するか
# (TRUE -> スクリプトの除去（推奨） FALSE -> 画像として再作成)
# スクリプトの除去を行うと、テキストのコピー&ペーストが可能ですが、
# 一部のオブジェクトが残る可能性があります。画像として再作成すると、オブジェクトは
# 完全に消去されますが、処理に時間が掛かり、テキストの抽出はできません。
# 導入の場面により手法を選択してください。
PDFCONVERTDOC=TRUE

# PDF ファイルを画像として再作成する場合の解像度
# (数値は解像度 (DPI). 数値が大きくなると処理時間、ファイルサイズが増加する)
# 上記設定で、画像として再作成する場合の画質を設定できます。
PDFCONVERTDENSITY=300

# PDF ファイルを画像として再作成する場合の一時ファイル領域の上限値
# (数値はディスクサイズ (GB). 余剰ディスクサイズ内の数値とすること)
PDFCONVERTTMPLIMIT=20

##########################################################################
# その他のファイルのサニタイズ処理の挙動に関する設定
# サニタイズ処理には様々な手法があり、出力されるファイルの形式や態様なども
# 異なってきます。運用の場面などに応じて、最適な組み合わせで設定してください。

# 一太郎形式ファイルをサニタイズする
# (TRUE -> サニタイズする（推奨） FALSE -> _marked フラグをつけて通す)
# 一太郎形式（その他 JustSystem 系のファイルも同様）のサニタイズ処理を
# 行います。標準では TRUE にしています。
SANITIZEJUST=TRUE
```

(次のページに続く)

(前のページからの続き)

```
# ドキュワークス形式ファイルをサニタイズする
# (TRUE -> 強制的に PDF 形式へサニタイズする（Ver.2 推奨）
#  FALSE -> _marked フラグをつけて通す
#  HOLD ->  保留にして受け渡し承認を受ける（Ver.3 推奨）)
# ドキュワークス形式（XDW) ファイルのサニタイズ処理を行います。ドキュワークス
# 形式はオブジェクトやファイルの添付が可能な形式なため、現時点では強制的に
# PDF に変換する手法を採用しています。
# 注意：この機能は外部モジュールを使っているので、Ver1.24.0 から利用可能
# です。それ以外のバージョンでは、TRUE に設定しても FALSE 相当の挙動になります。
SANITIZEXDW=HOLD

# 疑義のある画像形式ファイル（JPEG, GIF, BMP, PNG 等）を強制的にサニタイズする
# (TRUE -> 強制的にサニタイズする（推奨） FALSE -> 警告メッセージを出力する)
# 画像ファイルの中で疑義のあるものについて、強制的にサニタイズ処理を行います。
# 標準では TRUE にしています。
IMGFORCESANITIZE=TRUE

#########################################################################
# アーカイブファイルのサニタイズ処理の挙動に関する設定
# サニタイズ処理には様々な手法があり、出力されるファイルの形式や態様なども
# 異なってきます。運用の場面などに応じて、最適な組み合わせで設定してください。

# LZH ファイルの解凍処理を行う
# (TRUE -> LZH ファイルを解凍して work フォルダに保存。SANIEXTRACTED に従う。（推奨）
# FALSE -> ブロックする)
# 導入先のポリシーにより、アーカイブファイルの形式に LZH を採用していない場合
# には、この形式をブロックすることができます。
ALLOWLZH=TRUE

# CAB ファイルの解凍処理を行う
# (TRUE -> CAB ファイルを解凍して work フォルダに保存。SANIEXTRACTED に従う。（推奨）
# FALSE -> ブロックする)
# 導入先のポリシーにより、アーカイブファイルの形式に CAB を採用していない場合
# には、この形式をブロックすることができます。
ALLOWCAB=TRUE

# 7z(7-Zip) ファイルの解凍処理を行う
# (TRUE -> 7z ファイルを解凍して work フォルダに保存。SANIEXTRACTED に従う。（推奨）
# FALSE -> ブロックする)
# 導入先のポリシーにより、アーカイブファイルの形式に 7z を採用していない場合
# には、この形式をブロックすることができます。
ALLOW7Z=TRUE

# RAR ファイルの解凍処理を行う
```

(次のページに続く)

(前のページからの続き)

```
# (TRUE -> RAR ファイルを解凍して work フォルダに保存。SANIEXTRACTED に従う。(推奨)
# FALSE -> ブロックする)
# 導入先のポリシーにより、アーカイブファイルの形式に RAR を採用していない場合
# には、この形式をブロックすることができます。
ALLOWRAR=TRUE

# tar ファイルの解凍処理を行う
# (TRUE -> tar ファイルを解凍して work フォルダに保存。SANIEXTRACTED に従う。(推奨)
# FALSE -> ブロックする)
# 導入先のポリシーにより、アーカイブファイルの形式に tar を採用していない場合
# には、この形式をブロックすることができます。
ALLOWTAR=TRUE

# gzip/gz ファイルの解凍処理を行う
# (TRUE -> gzip/gz ファイルを解凍して work フォルダに保存。SANIEXTRACTED に従う。(推奨)
# FALSE -> ブロックする)
# 導入先のポリシーにより、アーカイブファイルの形式に gzip/gz を採用していない場合
# には、この形式をブロックすることができます。
ALLOWGZ=TRUE

# パスワード付き Zip ファイルの解凍を output 側から処理する
# (TRUE -> output 側からの処理で解凍する(推奨)　FALSE -> output 側から処理
# しない)
# パスワード付き Zip ファイルは解凍処理ができないため、通常は
# 「ファイル名_encryptReport.txt」というレポートファイルを出力します。
# このレポートファイルにパスワードを追記して保存することで、パスワードを
# 使った解凍処理をする機能を有効にするかを設定できます。
DECRYPTZIP=TRUE

# アーカイブファイルを解凍後、個別にサニタイズする
# (TRUE -> サニタイズする(推奨)　FALSE -> サニタイズしない)
# アーカイブファイルは、解凍処理されると、work フォルダに個別のフォルダを
# 生成してその中に解凍後のファイルが保存されます。これら解凍後のファイルを
# 自動的にサニタイズ処理させることができます。
# 注意:サニタイズ処理に失敗した場合には、work フォルダ中のファイルを手動で
# サニタイズ処理してください。
SANIEXTRACTED=TRUE

# アーカイブファイル解凍後のファイルサイズの上限値
# (数値はバイト数)
# 圧縮率の高いアーカイブファイルを解凍した際に、解凍後のファイルが
# ディスクの容量を超過する可能性があるため、ここで上限値を設定します。
EXTRACTEDSIZE=512000000

# アーカイブファイルの再圧縮処理を行うか(オプション機能)
```

(次のページに続く)

(前のページからの続き)

```
# （TRUE -> 行う FALSE -> 行わない）
# アーカイブファイルは解凍後、個別に無害化処理を経て、Output フォルダに出力する仕様と
# なっていますが、これらのファイルを再び Zip に再圧縮する機能を有効にするかを
# 設定します。この機能はオプションのため、設定値は導入先により異なります。
XXXXXXXXXXXXXXXXXXX=TRUE

# Zip ファイル再圧縮オプションにおけるタイムアウト時間
# (数値は分)
# 上記オプションにおける無害化処理のタイムアウト時間を設定します。
# この時間を経過すると、その段階で揃っているファイルだけを Zip 再圧縮します。
ZIPTIMEOUT=10

# 使用する端末の OS
# Windows7 は Zip ファイルの文字コードが ShiftJIS として処理されるバグがあります。
# Windows7 環境で Zip ファイル解凍中に文字化けが発生する場合には、使用する OS を
# 明示的に設定してください。
# （WINDOWS7 -> Windows7 OTHER -> それ以外（推奨））
CLIENTOS=OTHER

#########################################################################
# メールファイルのサニタイズ処理の挙動に関する設定
# サニタイズ処理には様々な手法があり、出力されるファイルの形式や態様なども
# 異なってきます。運用の場面などに応じて、最適な組み合わせで設定してください。

# メール添付ファイルのサニタイズをする（メール無害化）
# (TRUE -> サニタイズする（推奨） FALSE -> サニタイズしない)
# メール無害化を目的としてサニタイザーを使う場合には、この設定を TRUE にして
# ください。
# MailDir 形式のメールファイルを読み取り、その中に含まれている添付ファイルを
# 抽出し、work フォルダに保存、サニタイズ処理までを一連で行います。
# 注意：サニタイズ処理に失敗した場合には、work フォルダ中のファイルを手動で
# サニタイズ処理してください。
# 注意：メール無害化の導入を行う場合には、連携するシステムとの調整が必要
# ですので、個別にお問い合わせください。
SANIMAIL=TRUE

#########################################################################
# サニタイズ処理全般に関わる設定
# サニタイザー全体に関わる設定です。導入先のポリシーや連携するシステムの
# 制約で設定を変更する場面があるものをまとめています。

# サニタイズ後のファイルを区別するためにプロセス ID を付記する
# (TRUE -> プロセス ID を付記する（推奨） FALSE -> プロセス ID を付記しない)
# TRUE にすることで、output フォルダへ出力するファイル名にプロセス ID を付記
# するようになります。これにより、（おそらく）出力ファイル名が重複することが
```

(次のページに続く)

(前のページからの続き)

```
# なくなります。
# また、サニタイザーのログから、処理対象のファイルの特定も速やかに行えるように
# なります。
# なお、FALSE にした場合、ファイル名の重複が発生すると、後からサニタイズ処理
# されたファイルが残る仕組みとなります。
ADDPROCESSID=TRUE

# 判別不能なドキュメントファイルを拡張子で識別する
# (TRUE -> 拡張子で識別する（推奨）
#  FALSE -> 拡張子で識別せずに判別不能のまま処理する)
# サニタイザーは基本的にファイルの拡張子は見ず、ファイルの中身からファイルの
# 種別を識別します。
# しかし一部のファイルにおいては、ファイルの中身からでは識別できないものも
# あります。その場合、拡張子を手がかりにファイルの種別を推測します。
# 標準では TRUE としていますが、拡張子偽装に伴うリスクを排除するのならば、
# FALSE にしてください。（その場合、種別不明でブロックされる確率が増えます）
EXTDIST=TRUE

# ファイル処理後、Web インターフェースのデータベースを逐次更新する
# (TRUE -> 逐次データベースを更新する（推奨）
#  FALSE -> 定期的にデータベースを更新する)
# サニタイザーはサニタイズの処理の後に、Web インターフェース（ownCloud）の
# データベースを更新して、はじめて Web インターフェースの一覧にファイルが表示
# される仕組みとなっています。
# 設定を TRUE にすることで、サニタイズ処理後、直ちにデータベースの更新を行います。
# これにより画面への反映時間は短くなりますが、その代わりに大量のファイルを
# まとめて処理すると、頻繁にデータベースの書き換えが発生するので、全体的な
# パフォーマンスが低下します。
# FALSE にすると、データベースの更新をサニタイズ処理と関係なく定期的（１分に
# １回）に行います。パフォーマンスの低下はありませんが、その代わりに画面への
# 反映が最大１分後になります。導入先の状況にあわせて、設定してください。
RESCANDB=TRUE

# WARNING.BINARY フラグがついたファイルを通す
# (TRUE -> 通す FALSE -> Warning レポートを通知してファイルはブロックする（推奨）)
# サニタイズ処理の過程で、最終的に判別不能なファイル（バイナリファイル）が
# 残った場合、このファイルをどのように処理するかを設定します。
# TRUE にすることで、該当ファイルは、WARNING.BINARY という識別子が付いた
# 状態で output フォルダに出力されます。ただし、サニタイズ処理もされない状態の
# ファイルのため相応のリスクを負うことになります。リスクについて十分理解した
# 上で設定してください。標準では FALSE にしています。
THROUGHOUT=FALSE

# WARNING.BINARY フラグがついた実行形式ファイルを通す
# (TRUE -> 通す FALSE -> Warning レポートを通知してファイルはブロックする（推奨）)
```

(次のページに続く)

(前のページからの続き)

```
# サニタイズ処理の過程で、最終的に判別不能な実行ファイル（バイナリファイル）
# が残った場合、このファイルをどのように処理するかを設定します。
# TRUE にすることで、該当ファイルは、WARNING.BINARY という識別子が付いた
# 状態で output フォルダに出力されます。ただし、サニタイズ処理もされない状態の
# ファイルであり、実行可能である分、上述のバイナリファイルよりも高いリスクを
# 負うことになります。
# リスクについて十分理解した上で設定してください。標準では FALSE にしています。
EXECTHROUGHOUT=FALSE

#########################################################################
# トラストパスに関わる設定
# サニタイザーのオプションである、トラストパスに関わる設定です。
# トラストパスの導入を検討される場合には、個別にお問い合わせください。

# トラストパスによる復号ファイルを信頼し、そのまま取り込む
# (TRUE -> 取り込む（推奨） FALSE -> 取り込まずに work フォルダに保存する)
# ファイルの送信者があらかじめファイルに電子署名を付与し、その電子署名が
# 検証されたファイルは、サニタイズ処理を行うことなく、Output フォルダに出力
# する機能です。
# なお、この設定を FALSE にすると、署名検証後のファイルは work フォルダに
# 保存され、別途サニタイズ処理を手動で行ってもらうことになります。
# また署名検証に失敗したり、トラストパスの設定がない場合には、
# 「ファイル名_errorReport.txt」というレポートファイルを出力して、
# 処理は終了します。
TRUSTPASS=TRUE

# トラストパスクライアントを有効化する
# (SANITIZE -> 無害化して有効化  THROUGHT -> 無害化せずに有効化 FALSE -> 有効化しない)
# トラストパスクライアントは、ファイルをトラストパスのみで受け取ることができるよう、
# 署名付き暗号化ファイルとして出力する機能です。
# 媒体経由でファイルを受け渡す際に、この機能を使ってファイルを書き出す運用を想定しています。
# トラストパスクライアントは、output フォルダを監視して動作します。
# このパラメータで FALSE を設定した場合は、output フォルダの監視を行いません。
# SANITIZE を設定した場合は、output フォルダに出力された無害化処理後のファイルを
# 署名付き暗号化ファイルとして signed フォルダに出力します。
# また、THROUGHT を設定した場合は、無害化処理自体をスキップして output にファイルを出力し、
# それらのファイルをそのまま署名付き暗号化ファイルとして signed フォルダに出力します。
TRUSTCLIENT=FALSE

# トラストパスで使用する自分の鍵 ID
# (値は GPG で生成された鍵 ID)
# トラストパスクライアントはファイルを出力する際に、間違いなく自分から生成・送信され、
# 改ざんされていないことを示すために、電子署名を付加します。
# この電子署名を行うための鍵 ID をここで登録します。
TRUSTUSER=
```

(次のページに続く)

(前のページからの続き)

```
# トラストパスで使用する相手の鍵 ID
# (値は GPG で生成された鍵 ID)
# トラストパスクライアントはファイルを出力する際に、第三者にファイルの中身を見られないように
# するために暗号化を施します。暗号化はファイルを受け渡す相手を指定して行います。
# この暗号化を行うための相手の鍵 ID をここで登録します。
# （通常はファイルを受け取るサニタイザーに登録している鍵 ID となります）
TRUSTADMIN=

#########################################################################
# sanipass/sanicrypt に関わる設定
# サニタイザーのオプションである、sanipass 及び sanicrypt に関わる設定です。
# sanipass/sanicrypt の導入を検討される場合には、個別にお問い合わせください。
# 標準構成ではこれらの機能がは正常に稼働しません。

# sanipass/sanicrypt の出力先フォルダ
# (値はフォルダ名称)
# sanipass/sanicrypt により出力されるフォルダを設定します。これらの機能が
# 有効でない場合には、この値を参照することはありませんので、どのような値でも
# 差し支えありません。(念のためコメントアウトしてあります)
# OUTPUTFOLDER=""

# sanicrypt におけるパスワード付与形態
# (FIXED -> 固定設定　RULED -> ルールに基づく設定　LIST -> パスコードリスト
#  REPORT -> パスワードをレポートファイルで出力)
# sanicrypt により、パスワード付き Zip ファイルの出力をする際、パスワードの
# 付与形態を選択することができます。
CRYPTPASSCODE="REPORT"

#########################################################################
# フォルダ内のファイル削除に関わる設定
# サニタイザーで処理した input、output、work フォルダ、また上述の
# スルーパスで設定した出力先フォルダについて、定期的にフォルダ内のファイルを
# 削除するための設定です。

# 毎日午前 4 時に各フォルダの古いファイルとゴミ箱の中身をクリーンアップする
# (TRUE -> クリーンアップする（推奨）　FALSE -> クリーンアップしない)
# TRUE にすることで、毎日午前 4 時にフォルダ内の古いファイルを削除します。
# FALSE にすると、ファイルの削除は行われませんので、利用者が手作業で
# ファイルを削除していただくことになります。
CLEARFOLDER=TRUE

# 古いファイルとゴミ箱の中身をクリーンアップする期間
# (数値はクリーンアップせずに保持しておく日数)
# 標準では 2 日間（48 時間）経過したファイルから順に削除されます。この値を
```

(次のページに続く)

(前のページからの続き)

```
# 変更することで、フォルダ内に保持しておく期間を設定することができます。
STOREDAYS=2

#########################################################################
# サニタイザーの導入組織に関する設定
# サニタイザーをご利用いただいている組織に関する情報を設定します。

# 自治体コード
# （5桁数字あるいは5桁数字と1桁英文字）
LGCODE=

#########################################################################
# サニタイザーの構成に関わる設定
# ネットーワーク間の受け渡し（サニタイザープロ）、負荷分散のためのクラスタ
# 構成など、サニタイザーを複数台で構成する場合に、それぞれのサニタイザーが
# どのような役割を持つのかを設定します。
# 注意：サニタイザーを複数台で構成する場合には、複数台の間の調整が必要です
# ので、個別にお問い合わせください。

# サニタイザーのサーバ種別
# (INPUT -> input サーバ OUTPUT -> output サーバ
# STANDALONE -> 一台構成
# ３セグメント間の受け渡しの場合は個別に設定要)
# サニタイザーをどのような役割で稼働させるのかを設定します。
# 標準では STANDALONE となります。
# サニタイザープロの場合、input サーバの役割となるサーバには INPUT、
# output サーバの役割となるサーバには OUTPUT を設定してください。
SERVERTYPE=STANDALONE

# サニタイザープロをクラスタ構成で稼働する
# (TRUE -> クラスタ構成で稼働する FALSE -> シングル構成で稼働する)
# サニタイザープロをクラスタ構成で稼働させる場合、input サーバが複数台と
# なる構成を取りますが、その際には、この値を TRUE にしてください。
# TRUE にすることで、サニタイズ処理後のファイルが output フォルダに出力した
# 場合、どのサーバから送られてきたものかをファイル名に識別子として付記します。
CLUSTER=FALSE

# input フォルダのファイルをアーカイブ用に別途保管する
# (TRUE -> アーカイブフォルダに保管する FALSE -> 保管しない)
# サニタイザーをクラスタ構成で稼働させている際に、input フォルダに保存されて
# いるファイルをアーカイブ用に保管するための設定です。
# 注意：アーカイブ用のフォルダを別途設定する必要がありますので、
# クラスタ構成を構築する場合以外はこの設定を TRUE にしないでください。
ALLOWARCHIVE=FALSE
```

16.3 /var/www/nextcloud/config/config.php ファイル（サニタイザープロ Ver.3.0.0 input サーバ）

設定内容の解説は、 https://docs.nextcloud.com/server/16/admin_manual/configuration_server/config_sample_php_parameters.html にあります。

```
<?php
$CONFIG = array (
  'instanceid' => 'ocho4pgugo9w',
  'passwordsalt' => '+3LZBMrs9fswkhH+IpGsvCKQVPcbNr',
  'secret' => 'LbUAbTGjmoUTVlmsB5FVVFugYTEFHhPqNrXjFQ1Iuxiqo8cp',
  'trusted_domains' =>
  array (
    0 => '*',
  ),
  'datadirectory' => '/var/local/sanitizer/data',
  'dbtype' => 'mysql',
  'version' => '16.0.1.1',
  'overwrite.cli.url' => 'http://192.168.0.3/sanitizer',
  'dbname' => 'nextcloud',
  'dbhost' => 'localhost',
  'dbport' => '',
  'dbtableprefix' => 'oc_',
  'mysql.utf8mb4' => true,
  'dbuser' => 'nextcloud',
  'dbpassword' => 'nextcloudforsanitizer',
  'default_language' => 'ja',
  'force_language' => 'ja',
  'default_locale' => 'ja_JP',
  'force_locale' => 'ja_JP',
  'defaultapp' => 'files',
  'knowledgebaseenabled' => false,
  'allow_user_to_change_display_name' => false,
  'skeletondirectory' => '/var/local/sanitizer/policy/skeleton',
  'log_type' => 'file',
  'logfile' => '/var/log/nextcloud/nextcloud.log',
  'loglevel' => 1,
  'logdateformat' => 'Y-m-d H:i:s',
  'logtimezone' => 'Asia/Tokyo',
  'installed' => true,
  'memcache.locking' => '\\OC\\Memcache\\Redis',
  'memcache.distributed' => '\\OC\\Memcache\\Redis',
  'memcache.local' => '\\OC\\Memcache\\Redis',
  'redis' =>
  array (
```

(次のページに続く)

(前のページからの続き)

```
    'host' => 'localhost',
    'port' => 6379,
    'timeout' => 3,
  ),
  'ldapIgnoreNamingRules' => false,
  'ldapProviderFactory' => 'OCA\\User_LDAP\\LDAPProviderFactory',
);
```

16.4 サニタイザーポリシーファイル Ver.4.0.0

```
# This File is Sanitzer Policy for sanitize files.
# Created and ReWrited by Hiro KAWAGUCHI 20220711
# サニタイザー適用バージョン Ver.3.0.0 / Ver.4.0.1 以降

#########################################################################
# このファイルについて
# このファイルはサニタイザーの動作状況を制御するための設定ファイルです。
# それぞれ「項目=設定値」という構成になっています。
# 設定に際しては、項目名、設定値とも半角英数大文字とし、項目と設定値の間には
# 空白を入れないようにしてください。

#########################################################################
# サニタイザーのバージョンに対する考え方
# サニタイザーは主に２つのリリース体系があります。
# OVA ファイルにて提供されたもの -> 常にマイナーバージョン（最下位の数字）が
# 0 になります。
# モジュール差し替えにて提供されたもの -> マイナーバージョンが 0 以外と
# なります。
#
# 一般的にはマイナーバージョンが 0 のものをお使いいただいていますが、機能改善、
# 機能追加、個別カスタマイズにより、モジュールを差し替えたものについては、
# そのつど新たなマイナーバージョンを付与してリリースします。
# （モジュール差し替えは技術移転を受けた事業者しか対応できません）

# サニタイザーのバージョン
# (3 -> バージョン３  4 -> バージョン 4)
VERSION=4

#########################################################################
# サニタイズ処理の安定性を維持するための設定
# サニタイザーに処理上の負荷がかかった場合、安定的に処理を維持するための設定
# です。
# ２コア CPU、メモリ４ GB の環境で十分に稼働する設定値としていますが、導入先の
```

(次のページに続く)

(前のページからの続き)

```
# 機器の性能に応じてこれらの値は調整してください。

# メモリの使用量が規定値を超えた場合にサニタイズ処理を中止する
# (数値はパーセント表示)
# 処理を中止した場合、「処理されなかったファイル名_resourceReport.txt」と
# いうレポートファイルを出力します。この場合、そのファイルのサニタイズ処理を
# 再度行ってください。
MEMLIMIT=70

# スワップ領域の使用量が規定値を超えた場合にサニタイズ処理を中止する
# (数値はパーセント表示)
# 処理を中止した場合、「処理されなかったファイル名_resourceReport.txt」と
# いうレポートファイルを出力します。この場合、そのファイルのサニタイズ処理を
# 再度行ってください。
SWAPLIMIT=50

# ロードアベレージが規定値を超えた場合にサニタイズ処理を中止する
# (数値はロードアベレージ)
# 処理を中止した場合、「処理されなかったファイル名_resourceReport.txt」と
# いうレポートファイルを出力します。この場合、そのファイルのサニタイズ処理を
# 再度行ってください。
LOADAVELIMIT=50

# ファイルコンバートのタイムアウト時間。これを過ぎるとサニタイズを中止する
# (数値は秒数)
# サニタイズ処理の過程でいくつかの種類のファイルは形式を変換する処理を行って
# います。
# 処理を中止した場合、「処理されなかったファイル名_errorReport.txt」という
# レポートファイルを出力します。この場合、そのファイルのサニタイズ処理を再度
# 行ってください。
CONVERTTIMEOUT=60

#####################################################################
# サニタイズ処理を行う前の処理に関する設定
# サニタイザーはファイルのサニタイズ処理を行う前に、関連するいくつかの処理を
# 行っています。
# これによりサニタイズ処理の品質を高めたり、処理速度の向上を実現しています。

# 事前にブロックする拡張子を指定する
# (ブロックする拡張子をカンマ区切りで指定。大文字小文字は無関係。
#  指定しない場合は空白)
# サニタイザーはファイル判別、無害化処理を行う前に強制的にブロックするファイルを
# 拡張子で指定できます。
# このブロックはあらゆる処理よりも先に行われます。
BLOCKEXTLIST=
```

(次のページに続く)

(前のページからの続き)

```
# ファイル名に禁則文字が含まれている場合の挙動を設定する
# (CHANGE -> ファイル名変更
#  ABORT -> ファイル名変更を行わず、レポートファイルを出力する
#  CONTINUE -> ファイル名変更を行わず、そのまま処理する（推奨）)
# サニタイザーはファイル名に禁則文字（空白や OS で使えない文字等）が含まれている場合、
# それらを _ に置き換えて処理を行います。
PROHIBITEDCHAR=CONTINUE

# 事前に対象ファイルにウイルスチェックをする
# (TRUE -> する FALSE -> しない（推奨）)
# サニタイザーは内部で ClamAV というアンチウイルスソフトを稼働させています。
# これにより既知のマルウェアについてはサニタイズ処理前に排除することができます。
# ただし、ClamAV の処理には多くのメモリを消費することと、アンチウイルスパターン
# ファイルの更新にインターネット接続を行うことの理由により、サニタイザーで
# ウイルスチェックを行わず、別のソリューションで処理させたほうがよい場合も
# あります。
# 標準では FALSE にしてありますが、導入環境に応じて設定を変更してください。
ANTIVIRUS=FALSE

# 事前に対象ファイルに不正コード混入チェックをする
# (TRUE -> する（推奨） FALSE -> しない)
# サニタイズ処理の前に対象ファイルの中身を走査し、不正なコードが含まれて
# いないかを
# チェックしています。ここでは明らかに疑わしい処理のみをチェックしています。
# 伝統的な標的型攻撃ファイルの多くはこのチェックで排除できますので、標準では
# TRUE にしてあります。
CHKMALICIOUS=TRUE

# マクロの有無を事前にチェックし、マクロが明らかに含まれていない場合は
# サニタイズしないで通す
# (TRUE -> _marked フラグをつけて通す（推奨） FALSE -> 常にサニタイズする)
# サニタイズ処理の前に対象ファイルの中身を走査し、マクロやオブジェクトの
# 埋め込みがされていないファイルについてはサニタイズ処理を行わないで通過
# させます。
# ここでのチェックの観点はマクロ、オブジェクトの埋め込みの有無であり、不正な
# ファイル形式や判別不明なバイナリコード等については、この設定に関わらず
# チェックして排除しています。
MACROAVAILFIRST=TRUE

#########################################################################
# Office ファイルのサニタイズ処理の挙動に関する設定
# サニタイズ処理には様々な手法があり、出力されるファイルの形式や態様なども
# 異なってきます。運用の場面などに応じて、最適な組み合わせで設定してください。
```

(次のページに続く)

(前のページからの続き)

```
# パスワード付きの Office ファイルのパスワードを解除する（Ver.4 より）
# (TRUE -> パスワード解除する（Ver.4 のみ）
#  FALSE -> パスワード解除せずに ALLOWMSPASS の設定に従う（Ver.3 推奨）)
# Microsoft Office のファイルのうち、パスワードが付加されたものについては、
# パスワードを解除したうえで、無害化処理を行うことになります。
DECRYPTMS=TRUE

# パスワード付きの Office ファイルを通す
# (TRUE -> 通す FALSE -> Warning レポートを通知してファイルはブロックする
#  HOLD ->  保留にして受け渡し承認を受ける（Ver.3 推奨）)
# Microsoft Office のファイルのうち、パスワードが付加されたものについては、
# ファイルの内容がマクロを含めて暗号化されますので、Ver.3 では無害化できません。
# ファイルの発出元が明らかである場合は、発出元を信頼してサニタイズ処理をせずに
# 受け取る運用も考えられますので、導入先のポリシーに応じて設定してください。
ALLOWMSPASS=HOLD

# RMS で暗号化された Office ファイルを通す
# (TRUE -> 通す FALSE -> Warning レポートを通知してファイルはブロックする（Ver.2 推奨）
#  HOLD ->  保留にして受け渡し承認を受ける（Ver.3 推奨）)
# Microsoft Rights Management Server（RMS）により暗号化された Office
# ファイルについては、ファイルの内容がマクロを含めて暗号化されますので、
# サニタイズ処理ができません。
# ファイルの発出元が明らかである場合は、発出元を信頼してサニタイズ処理をせずに
# 受け取る運用も考えられますので、導入先のポリシーに応じて設定してください。
ALLOWMSRMS=HOLD

# Office2007 以降のファイルについて、PC に悪影響のあるマクロのみを削除する
# (TRUE -> 削除する（推奨） FALSE -> 全てのマクロを削除する)
# 全てのマクロが悪影響を与えるものではないという考えから、マクロの内容を
# 精査し、PC に悪影響を与える可能性の高いマクロが含まれている場合のみ、
# マクロを削除するようにします。
DELBADCODEONLY=TRUE

# Office ファイルのサニタイズ処理基準を厳格に適用するか
# (TRUE -> 厳格に適用 FALSE -> 厳格に適用しない（推奨）)
# Office ファイルをサニタイズする前に、ファイル中でリスクになり得る情報まで厳格に
# 判断するかの基準です。
# TRUE の場合、リスク要因が含まれているだけでなく、全く別の観点からサニタイズを
# するか否かの判断を行いますが、誤検知のリスクもあります。
# FALSE の場合は、単純にリスク要因が埋め込まれているか否かだけを判断します。
EXSEEKSTRICT=FALSE

# Office ファイル、ODF 形式ファイルのサニタイズ時に PDF を同時に生成する（Ver.4 で廃止）
# WITHCREATEPDF=FALSE
```

(次のページに続く)

(前のページからの続き)

```
#########################################################################
# PDF ファイルのサニタイズ処理の挙動に関する設定
# サニタイズ処理には様々な手法があり、出力されるファイルの形式や態様なども
# 異なってきます。運用の場面などに応じて、最適な組み合わせで設定してください。

# パスワード付きの PDF ファイルを通す
# (TRUE -> 通す FALSE -> パスワード要求レポートを出力し再処理を促す（推奨）)
#  HOLD ->  保留にして受け渡し承認を受ける)
# PDF ファイルのうち、パスワードが付加されたものについては、ファイルの内容が
# スクリプトを含めて暗号化されますので、サニタイズ処理ができません。
# ファイルの発出元が明らかである場合は、発出元を信頼してサニタイズ処理をせずに
# 受け取る運用も考えられますので、導入先のポリシーに応じて設定してください。
ALLOWPDFPASS=FALSE

# PDF ファイルのサニタイズ処理基準を厳格に適用するか
# (TRUE -> 厳格に適用 FALSE -> JavaScript や埋め込みファイルがある場合のみ（推奨）)
# PDF ファイルをサニタイズする前に、ファイル中でリスクになり得る情報まで厳格に
# 判断するかの基準です。
# TRUE の場合、リスク要因が含まれているだけでなく、全く別の観点からサニタイズを
# するか否かの判断を行いますが、誤検知のリスクもあります。
# FALSE の場合は、単純にリスク要因が埋め込まれているか否かだけを判断します。
SANIPDFSTRICT=FALSE

# PDF のサニタイズ処理において、スクリプトの除去を行うか、画像として再作成するか
# (TRUE -> スクリプトの除去（推奨） FALSE -> 画像として再作成)
# スクリプトの除去を行うと、テキストのコピー&ペーストが可能ですが、
# 一部のオブジェクトが残る可能性があります。画像として再作成すると、オブジェクトは
# 完全に消去されますが、処理に時間が掛かり、テキストの抽出はできません。
# 導入の場面により手法を選択してください。
PDFCONVERTDOC=TRUE

# PDF ファイルを画像として再作成する場合の解像度
# (数値は解像度 (DPI). 数値が大きくなると処理時間、ファイルサイズが増加する)
# 上記設定で、画像として再作成する場合の画質を設定できます。
PDFCONVERTDENSITY=300

# PDF ファイルを画像として再作成する場合の一時ファイル領域の上限値
# (数値はディスクサイズ (GB). 余剰ディスクサイズ内の数値とすること)
PDFCONVERTTMPLIMIT=20

#########################################################################
# その他のファイルのサニタイズ処理の挙動に関する設定
# サニタイズ処理には様々な手法があり、出力されるファイルの形式や態様なども
# 異なってきます。運用の場面などに応じて、最適な組み合わせで設定してください。
```

(次のページに続く)

(前のページからの続き)

```
# 一太郎形式ファイルをサニタイズする
# (TRUE -> サニタイズする（推奨） FALSE -> _marked フラグをつけて通す)
# 一太郎形式（その他 JustSystem 系のファイルも同様）のサニタイズ処理を
# 行います。標準では TRUE にしています。
SANITIZEJUST=TRUE

# ドキュワークス形式ファイルをサニタイズする
# (TRUE -> 強制的に PDF 形式へサニタイズする（Ver.2 推奨）
#  FALSE -> _marked フラグをつけて通す
#  HOLD ->  保留にして受け渡し承認を受ける（Ver.3 推奨）)
# ドキュワークス形式（XDW) ファイルのサニタイズ処理を行います。ドキュワークス
# 形式はオブジェクトやファイルの添付が可能な形式なため、現時点では強制的に
# PDF に変換する手法を採用しています。
# 注意：この機能は外部モジュールを使っているので、Ver1.24.0 から利用可能
# です。それ以外のバージョンでは、TRUE に設定しても FALSE 相当の挙動になります。
SANITIZEXDW=HOLD

# 疑義のある画像形式ファイル（JPEG, GIF, BMP, PNG 等）を強制的にサニタイズする
# (TRUE -> 強制的にサニタイズする（推奨） FALSE -> 警告メッセージを出力する)
# 画像ファイルの中で疑義のあるものについて、強制的にサニタイズ処理を行います。
# 標準では TRUE にしています。
IMGFORCESANITIZE=TRUE

#########################################################################
# アーカイブファイルのサニタイズ処理の挙動に関する設定
# サニタイズ処理には様々な手法があり、出力されるファイルの形式や態様なども
# 異なってきます。運用の場面などに応じて、最適な組み合わせで設定してください。

# LZH ファイルの解凍処理を行う
# (TRUE -> LZH ファイルを解凍して work フォルダに保存。SANIEXTRACTED に従う。（推奨）
# FALSE -> ブロックする)
# 導入先のポリシーにより、アーカイブファイルの形式に LZH を採用していない場合
# には、この形式をブロックすることができます。
ALLOWLZH=TRUE

# CAB ファイルの解凍処理を行う
# (TRUE -> CAB ファイルを解凍して work フォルダに保存。SANIEXTRACTED に従う。（推奨）
# FALSE -> ブロックする)
# 導入先のポリシーにより、アーカイブファイルの形式に CAB を採用していない場合
# には、この形式をブロックすることができます。
ALLOWCAB=TRUE

# 7z(7-Zip) ファイルの解凍処理を行う
# (TRUE -> 7z ファイルを解凍して work フォルダに保存。SANIEXTRACTED に従う。（推奨）
# FALSE -> ブロックする)
```

(次のページに続く)

(前のページからの続き)

```
# 導入先のポリシーにより、アーカイブファイルの形式に 7z を採用していない場合
# には、この形式をブロックすることができます。
ALLOW7Z=TRUE

# RAR ファイルの解凍処理を行う
# (TRUE -> RAR ファイルを解凍して work フォルダに保存。SANIEXTRACTED に従う。(推奨)
# FALSE -> ブロックする)
# 導入先のポリシーにより、アーカイブファイルの形式に RAR を採用していない場合
# には、この形式をブロックすることができます。
ALLOWRAR=TRUE

# tar ファイルの解凍処理を行う
# (TRUE -> tar ファイルを解凍して work フォルダに保存。SANIEXTRACTED に従う。(推奨)
# FALSE -> ブロックする)
# 導入先のポリシーにより、アーカイブファイルの形式に tar を採用していない場合
# には、この形式をブロックすることができます。
ALLOWTAR=TRUE

# gzip/gz ファイルの解凍処理を行う
# (TRUE -> gzip/gz ファイルを解凍して work フォルダに保存。SANIEXTRACTED に従う。(推奨)
# FALSE -> ブロックする)
# 導入先のポリシーにより、アーカイブファイルの形式に gzip/gz を採用していない場合
# には、この形式をブロックすることができます。
ALLOWGZ=TRUE

# パスワード付き Zip ファイルの解凍を output 側から処理する
# (TRUE -> output 側からの処理で解凍する(推奨)  FALSE -> output 側から処理
# しない)
# パスワード付き Zip ファイルは解凍処理ができないため、通常は
# 「ファイル名_encryptReport.txt」というレポートファイルを出力します。
# このレポートファイルにパスワードを追記して保存することで、パスワードを
# 使った解凍処理をする機能を有効にするかを設定できます。
DECRYPTZIP=TRUE

# アーカイブファイルを解凍後、個別にサニタイズする
# (TRUE -> サニタイズする(推奨)  FALSE -> サニタイズしない)
# アーカイブファイルは、解凍処理されると、work フォルダに個別のフォルダを
# 生成してその中に解凍後のファイルが保存されます。これら解凍後のファイルを
# 自動的にサニタイズ処理させることができます。
# 注意:サニタイズ処理に失敗した場合には、work フォルダ中のファイルを手動で
# サニタイズ処理してください。
SANIEXTRACTED=TRUE

# アーカイブファイル解凍後のファイルサイズの上限値
# (数値はバイト数)
```

(次のページに続く)

(前のページからの続き)

```
# 圧縮率の高いアーカイブファイルを解凍した際に、解凍後のファイルが
# ディスクの容量を超過する可能性があるため、ここで上限値を設定します。
EXTRACTEDSIZE=5120000000

# アーカイブファイルの再圧縮処理を行うか（オプション機能）
# （TRUE -> 行う FALSE -> 行わない）
# アーカイブファイルは解凍後、個別に無害化処理を経て、Output フォルダに出力する仕様と
# なっていますが、これらのファイルを再び Zip に再圧縮する機能を有効にするかを
# 設定します。この機能はオプションのため、設定値は導入先により異なります。
XXXXXXXXXXXXXXXXXX=TRUE

# Zip ファイル再圧縮オプションにおけるタイムアウト時間
# (数値は分)
# 上記オプションにおける無害化処理のタイムアウト時間を設定します。
# この時間を経過すると、その段階で揃っているファイルだけを Zip 再圧縮します。
ZIPTIMEOUT=10

# 使用する端末の OS
# Windows7 は Zip ファイルの文字コードが ShiftJIS として処理されるバグがあります。
# Windows7 環境で Zip ファイル解凍中に文字化けが発生する場合には、使用する OS を
# 明示的に設定してください。
# （WINDOWS7 -> Windows7 OTHER -> それ以外（推奨））
CLIENTOS=OTHER

#########################################################################
# フォルダのサニタイズ処理の挙動に関する設定
# なお、これらの設定は Zip ファイル再圧縮オプションを有効にした場合のみ機能します。

# フォルダのサニタイズ処理を行う
# (TRUE -> サニタイズする（推奨） FALSE -> サニタイズしない)
# サニタイザーは input フォルダにフォルダそのものを投入すると、そのフォルダ配下の
# ファイルやサブフォルダを個別に無害化処理し、Zip ファイルに圧縮した状態で
# output フォルダに出力する機能があります。
# この機能の有効／無効を設定することができます。
ALLOWFOLDER=TRUE

# フォルダを投入した場合のタイムアウト時間。
# (数値は秒数)
# サニタイザーの Zip ファイル再圧縮オプションを有効にすると、input フォルダに
# ファイルだけでなくフォルダをまとめて投入することができます。
# これを過ぎるとその時点で転送済みのフォルダ、ファイルのみを無害化処理します。
# 大容量のファイルを含むフォルダを投入する場合はこの値を調整してください。
FOLDERTIMEOUT=120

#########################################################################
```

(次のページに続く)

(前のページからの続き)

```
# メールファイルのサニタイズ処理の挙動に関する設定
# サニタイズ処理には様々な手法があり、出力されるファイルの形式や態様なども
# 異なってきます。運用の場面などに応じて、最適な組み合わせで設定してください。

# メール添付ファイルのサニタイズをする（メール無害化）
# (TRUE -> サニタイズする（推奨） FALSE -> サニタイズしない)
# メール無害化を目的としてサニタイザーを使う場合には、この設定を TRUE にして
# ください。
# MailDir 形式のメールファイルを読み取り、その中に含まれている添付ファイルを
# 抽出し、work フォルダに保存、サニタイズ処理までを一連で行います。
# 注意：サニタイズ処理に失敗した場合には、work フォルダ中のファイルを手動で
# サニタイズ処理してください。
# 注意：メール無害化の導入を行う場合には、連携するシステムとの調整が必要
# ですので、個別にお問い合わせください。
SANIMAIL=TRUE

########################################################################
# サニタイズ処理全般に関わる設定
# サニタイザー全体に関わる設定です。導入先のポリシーや連携するシステムの
# 制約で設定を変更する場面があるものをまとめています。

# サニタイズ後のファイルを区別するためにプロセス ID を付記する
# (TRUE -> プロセス ID を付記する（推奨） FALSE -> プロセス ID を付記しない)
# TRUE にすることで、output フォルダへ出力するファイル名にプロセス ID を付記
# するようになります。これにより、（おそらく）出力ファイル名が重複することが
# なくなります。
# また、サニタイザーのログから、処理対象のファイルの特定も速やかに行えるように
# なります。
# なお、FALSE にした場合、ファイル名の重複が発生すると、後からサニタイズ処理
# されたファイルが残る仕組みとなります。
ADDPROCESSID=TRUE

# 判別不能なドキュメントファイルを拡張子で識別する
# (TRUE -> 拡張子で識別する（推奨）
#  FALSE -> 拡張子で識別せずに判別不能のまま処理する)
# サニタイザーは基本的にファイルの拡張子は見ず、ファイルの中身からファイルの
# 種別を識別します。
# しかし一部のファイルにおいては、ファイルの中身からでは識別できないものも
# あります。その場合、拡張子を手がかりにファイルの種別を推測します。
# 標準では TRUE としていますが、拡張子偽装に伴うリスクを排除するのならば、
# FALSE にしてください。（その場合、種別不明でブロックされる確率が増えます）
EXTDIST=TRUE

# ファイル処理後、Web インターフェースのデータベースを逐次更新する
# (TRUE -> 逐次データベースを更新する（推奨）
```

(次のページに続く)

(前のページからの続き)

```
#  FALSE -> 定期的にデータベースを更新する)
# サニタイザーはサニタイズの処理の後に、Web インターフェース（ownCloud）の
# データベースを更新して、はじめて Web インターフェースの一覧にファイルが表示
# される仕組みとなっています。
# 設定を TRUE にすることで、サニタイズ処理後、直ちにデータベースの更新を行います。
# これにより画面への反映時間は短くなりますが、その代わりに大量のファイルを
# まとめて処理すると、頻繁にデータベースの書き換えが発生するので、全体的な
# パフォーマンスが低下します。
# FALSE にすると、データベースの更新をサニタイズ処理と関係なく定期的（１分に
# １回）に行います。パフォーマンスの低下はありませんが、その代わりに画面への
# 反映が最大１分後になります。導入先の状況にあわせて、設定してください。
RESCANDB=TRUE

# WARNING.BINARY フラグがついたファイルを通す
# (TRUE -> 通す  FALSE -> Warning レポートを通知してファイルはブロックする（推奨）)
# サニタイズ処理の過程で、最終的に判別不能なファイル（バイナリファイル）が
# 残った場合、このファイルをどのように処理するかを設定します。
# TRUE にすることで、該当ファイルは、WARNING.BINARY という識別子が付いた
# 状態で output フォルダに出力されます。ただし、サニタイズ処理もされない状態の
# ファイルのため相応のリスクを負うことになります。リスクについて十分理解した
# 上で設定してください。標準では FALSE にしています。
THROUGHOUT=FALSE

# WARNING.BINARY フラグがついた実行形式ファイルを通す
# （Ver.4 で廃止。THROUGHOUT に一本化）
# EXECTHROUGHOUT=FALSE

#########################################################################
# トラストパスに関わる設定
# サニタイザーのオプションである、トラストパスに関わる設定です。
# トラストパスの導入を検討される場合には、個別にお問い合わせください。

# トラストパスによる復号ファイルを信頼し、そのまま取り込む
# (TRUE -> 取り込む（推奨）  FALSE -> 取り込まずに work フォルダに保存する)
# ファイルの送信者があらかじめファイルに電子署名を付与し、その電子署名が
# 検証されたファイルは、サニタイズ処理を行うことなく、Output フォルダに出力
# する機能です。
# なお、この設定を FALSE にすると、署名検証後のファイルは work フォルダに
# 保存され、別途サニタイズ処理を手動で行ってもらうことになります。
# また署名検証に失敗したり、トラストパスの設定がない場合には、
# 「ファイル名_errorReport.txt」というレポートファイルを出力して、
# 処理は終了します。
TRUSTPASS=TRUE

# トラストパスクライアントを有効化する
```

(次のページに続く)

(前のページからの続き)

```
# (SANITIZE -> 無害化して有効化　THROUGHT -> 無害化せずに有効化 FALSE -> 有効化しない)
# トラストパスクライアントは、ファイルをトラストパスのみで受け取ることができるよう、
# 署名付き暗号化ファイルとして出力する機能です。
# 媒体経由でファイルを受け渡す際に、この機能を使ってファイルを書き出す運用を想定しています。
# トラストパスクライアントは、output フォルダを監視して動作します。
# このパラメータで FALSE を設定した場合は、output フォルダの監視を行いません。
# SANITIZE を設定した場合は、output フォルダに出力された無害化処理後のファイルを
# 署名付き暗号化ファイルとして signed フォルダに出力します。
# また、THROUGHT を設定した場合は、無害化処理自体をスキップして output にファイルを出力し、
# それらのファイルをそのまま署名付き暗号化ファイルとして signed フォルダに出力します。
TRUSTCLIENT=FALSE

# トラストパスで使用する自分の鍵 ID
# (値は GPG で生成された鍵 ID)
# トラストパスクライアントはファイルを出力する際に、間違いなく自分から生成・送信され、
# 改ざんされていないことを示すために、電子署名を付加します。
# この電子署名を行うための鍵 ID をここで登録します。
TRUSTUSER=

# トラストパスで使用する相手の鍵 ID
# (値は GPG で生成された鍵 ID)
# トラストパスクライアントはファイルを出力する際に、第三者にファイルの中身を見られないように
# するために暗号化を施します。暗号化はファイルを受け渡す相手を指定して行います。
# この暗号化を行うための相手の鍵 ID をここで登録します。
# （通常はファイルを受け取るサニタイザーに登録している鍵 ID となります）
TRUSTADMIN=

#########################################################################
# sanipass/sanicrypt に関わる設定
# サニタイザーのオプションである、sanipass 及び sanicrypt に関わる設定です。
# sanipass/sanicrypt の導入を検討される場合には、個別にお問い合わせください。
# 標準構成ではこれらの機能がは正常に稼働しません。

# sanipass/sanicrypt の出力先フォルダ
# (値はフォルダ名称)
# sanipass/sanicrypt により出力されるフォルダを設定します。これらの機能が
# 有効でない場合には、この値を参照することはありませんので、どのような値でも
# 差し支えありません。(念のためコメントアウトしてあります)
# OUTPUTFOLDER=""

# sanicrypt におけるパスワード付与形態
# (FIXED -> 固定設定　RULED -> ルールに基づく設定　LIST -> パスコードリスト
#  REPORT -> パスワードをレポートファイルで出力)
# sanicrypt により、パスワード付き Zip ファイルの出力をする際、パスワードの
# 付与形態を選択することができます。
```

(次のページに続く)

(前のページからの続き)

```
CRYPTPASSCODE="REPORT"

#########################################################################
# フォルダ内のファイル削除に関わる設定
# サニタイザーで処理した input、output、work フォルダ、また上述の
# スルーパスで設定した出力先フォルダについて、定期的にフォルダ内のファイルを
# 削除するための設定です。

# 毎日午前 4 時に各フォルダの古いファイルとゴミ箱の中身をクリーンアップする
# (TRUE -> クリーンアップする（推奨） FALSE -> クリーンアップしない)
# TRUE にすることで、毎日午前 4 時にフォルダ内の古いファイルを削除します。
# FALSE にすると、ファイルの削除は行われませんので、利用者が手作業で
# ファイルを削除していただくことになります。
CLEARFOLDER=TRUE

# 古いファイルとゴミ箱の中身をクリーンアップする期間
# (数値はクリーンアップせずに保持しておく日数)
# 標準では 2 日間（48 時間）経過したファイルから順に削除されます。この値を
# 変更することで、フォルダ内に保持しておく期間を設定することができます。
STOREDAYS=2

#########################################################################
# サニタイザーの導入組織に関する設定
# サニタイザーをご利用いただいている組織に関する情報を設定します。

# 自治体コード
# （5 桁数字あるいは 5 桁数字と 1 桁英文字）
LGCODE=

#########################################################################
# サニタイザーの構成に関わる設定
# ネットワーク間の受け渡し（サニタイザープロ）、負荷分散のためのクラスタ
# 構成など、サニタイザーを複数台で構成する場合に、それぞれのサニタイザーが
# どのような役割を持つのかを設定します。
# 注意：サニタイザーを複数台で構成する場合には、複数台の間の調整が必要です
# ので、個別にお問い合わせください。

# サニタイザーのサーバ種別
# (INPUT -> input サーバ OUTPUT -> output サーバ
# STANDALONE -> 一台構成
# ３セグメント間の受け渡しの場合は個別に設定要)
# サニタイザーをどのような役割で稼働させるのかを設定します。
# 標準では STANDALONE となります。
# サニタイザープロの場合、input サーバの役割となるサーバには INPUT、
# output サーバの役割となるサーバには OUTPUT を設定してください。
```

(次のページに続く)

(前のページからの続き)

```
SERVERTYPE=STANDALONE

# サニタイザープロをクラスタ構成で稼働する（Ver.4 で廃止）
# CLUSTER=FALSE

# input フォルダのファイルをアーカイブ用に別途保管する（Ver.4 で廃止）
# ALLOWARCHIVE=FALSE
```

16.5 サニタイザーポリシーファイル Ver.4.2.0

```
# This File is Sanitzer Policy for sanitize files.
# Created and ReWrited by Hiro KAWAGUCHI 20231201
# サニタイザー適用バージョン Ver.4.2.0 以降

#########################################################################
# このファイルについて
# このファイルはサニタイザーの動作状況を制御するための設定ファイルです。
# それぞれ「項目=設定値」という構成になっています。
# 設定に際しては、項目名、設定値とも半角英数大文字とし、項目と設定値の間には
# 空白を入れないようにしてください。

#########################################################################
# サニタイザーのバージョンに対する考え方
# サニタイザーは主に２つのリリース体系があります。
# OVA ファイルにて提供されたもの -> 常にマイナーバージョン（最下位の数字）が
# 0 になります。
# モジュール差し替えにて提供されたもの -> マイナーバージョンが 0 以外と
# なります。
#
# 一般的にはマイナーバージョンが 0 のものをお使いいただいていますが、機能改善、
# 機能追加、個別カスタマイズにより、モジュールを差し替えたものについては、
# そのつど新たなマイナーバージョンを付与してリリースします。
# （モジュール差し替えは技術移転を受けた事業者しか対応できません）

# サニタイザーのバージョン
# (3 -> バージョン３  4 -> バージョン 4)
VERSION=4

#########################################################################
# サニタイズ処理の安定性を維持するための設定
# サニタイザーに処理上の負荷がかかった場合、安定的に処理を維持するための設定
# です。
# ２コア CPU、メモリ４ GB の環境で十分に稼働する設定値としていますが、導入先の
```

(次のページに続く)

(前のページからの続き)

```
# 機器の性能に応じてこれらの値は調整してください。

# メモリの使用量が規定値を超えた場合にサニタイズ処理を中止する
# (数値はパーセント表示)
# 処理を中止した場合、「処理されなかったファイル名_resourceReport.txt」と
# いうレポートファイルを出力します。この場合、そのファイルのサニタイズ処理を
# 再度行ってください。
MEMLIMIT=70

# スワップ領域の使用量が規定値を超えた場合にサニタイズ処理を中止する
# (数値はパーセント表示)
# 処理を中止した場合、「処理されなかったファイル名_resourceReport.txt」と
# いうレポートファイルを出力します。この場合、そのファイルのサニタイズ処理を
# 再度行ってください。
SWAPLIMIT=50

# ロードアベレージが規定値を超えた場合にサニタイズ処理を中止する
# (数値はロードアベレージ)
# 処理を中止した場合、「処理されなかったファイル名_resourceReport.txt」と
# いうレポートファイルを出力します。この場合、そのファイルのサニタイズ処理を
# 再度行ってください。
LOADAVELIMIT=50

# ファイルコンバートのタイムアウト時間。これを過ぎるとサニタイズを中止する
# (数値は秒数)
# サニタイズ処理の過程でいくつかの種類のファイルは形式を変換する処理を行って
# います。
# 処理を中止した場合、「処理されなかったファイル名_errorReport.txt」という
# レポートファイルを出力します。この場合、そのファイルのサニタイズ処理を再度
# 行ってください。
CONVERTTIMEOUT=60

#####################################################################
# サニタイズ処理を行う前の処理に関する設定
# サニタイザーはファイルのサニタイズ処理を行う前に、関連するいくつかの処理を
# 行っています。
# これによりサニタイズ処理の品質を高めたり、処理速度の向上を実現しています。

# 事前にブロックする拡張子を指定する
# (ブロックする拡張子をカンマ区切りで指定。大文字小文字は無関係。
#  指定しない場合は空白)
# サニタイザーはファイル判別、無害化処理を行う前に強制的にブロックするファイルを
# 拡張子で指定できます。
# このブロックはあらゆる処理よりも先に行われます。
BLOCKEXTLIST=
```

(次のページに続く)

(前のページからの続き)

```
# ファイル名に禁則文字が含まれている場合の挙動を設定する
# (CHANGE -> ファイル名変更
#  ABORT -> ファイル名変更を行わず、レポートファイルを出力する
#  CONTINUE -> ファイル名変更を行わず、そのまま処理する（推奨）)
# サニタイザーはファイル名に禁則文字（空白や OS で使えない文字等）が含まれている場合、
# それらを _ に置き換えて処理を行います。
PROHIBITEDCHAR=CONTINUE

# 事前に対象ファイルにウイルスチェックをする
# (TRUE -> する FALSE -> しない（推奨）)
# サニタイザーは内部で ClamAV というアンチウイルスソフトを稼働させています。
# これにより既知のマルウェアについてはサニタイズ処理前に排除することができます。
# ただし、ClamAV の処理には多くのメモリを消費することと、アンチウイルスパターン
# ファイルの更新にインターネット接続を行うことの理由により、サニタイザーで
# ウイルスチェックを行わず、別のソリューションで処理させたほうがよい場合も
# あります。
# 標準では FALSE にしてありますが、導入環境に応じて設定を変更してください。
ANTIVIRUS=FALSE

# 事前に対象ファイルに不正コード混入チェックをする
# (TRUE -> する（推奨） FALSE -> しない)
# サニタイズ処理の前に対象ファイルの中身を走査し、不正なコードが含まれて
# いないかを
# チェックしています。ここでは明らかに疑わしい処理のみをチェックしています。
# 伝統的な標的型攻撃ファイルの多くはこのチェックで排除できますので、標準では
# TRUE にしてあります。
CHKMALICIOUS=TRUE

# マクロの有無を事前にチェックし、マクロが明らかに含まれていない場合は
# サニタイズしないで通す
# (TRUE -> _marked フラグをつけて通す（推奨） FALSE -> 常にサニタイズする)
# サニタイズ処理の前に対象ファイルの中身を走査し、マクロやオブジェクトの
# 埋め込みがされていないファイルについてはサニタイズ処理を行わないで通過
# させます。
# ここでのチェックの観点はマクロ、オブジェクトの埋め込みの有無であり、不正な
# ファイル形式や判別不明なバイナリコード等については、この設定に関わらず
# チェックして排除しています。
MACROAVAILFIRST=TRUE

#########################################################################
# Office ファイルのサニタイズ処理の挙動に関する設定
# サニタイズ処理には様々な手法があり、出力されるファイルの形式や態様なども
# 異なってきます。運用の場面などに応じて、最適な組み合わせで設定してください。
```

(次のページに続く)

(前のページからの続き)

```
# パスワード付きの Office ファイルのパスワードを解除する（Ver.4 より）
# (TRUE -> パスワード解除する（Ver.4 のみ）
#  FALSE -> パスワード解除せずに ALLOWMSPASS の設定に従う（Ver.3 推奨）)
# Microsoft Office のファイルのうち、パスワードが付加されたものについては、
# パスワードを解除したうえで、無害化処理を行うことになります。
DECRYPTMS=TRUE

# パスワード付きの Office ファイルを通す
# (TRUE -> 通す FALSE -> Warning レポートを通知してファイルはブロックする
#  HOLD ->  保留にして受け渡し承認を受ける（Ver.3 推奨）)
# Microsoft Office のファイルのうち、パスワードが付加されたものについては、
# ファイルの内容がマクロを含めて暗号化されますので、Ver.3 では無害化できません。
# ファイルの発出元が明らかである場合は、発出元を信頼してサニタイズ処理をせずに
# 受け取る運用も考えられますので、導入先のポリシーに応じて設定してください。
ALLOWMSPASS=HOLD

# RMS で暗号化された Office ファイルを通す
# (TRUE -> 通す FALSE -> Warning レポートを通知してファイルはブロックする（Ver.2 推奨）
#  HOLD ->  保留にして受け渡し承認を受ける（Ver.3 推奨）)
# Microsoft Rights Management Server（RMS）により暗号化された Office
# ファイルについては、ファイルの内容がマクロを含めて暗号化されますので、
# サニタイズ処理ができません。
# ファイルの発出元が明らかである場合は、発出元を信頼してサニタイズ処理をせずに
# 受け取る運用も考えられますので、導入先のポリシーに応じて設定してください。
ALLOWMSRMS=HOLD

# Office2007 以降のファイルについて、PC に悪影響のあるマクロのみを削除する
# (TRUE -> 削除する（推奨） FALSE -> 全てのマクロを削除する)
# 全てのマクロが悪影響を与えるものではないという考えから、マクロの内容を
# 精査し、PC に悪影響を与える可能性の高いマクロが含まれている場合のみ、
# マクロを削除するようにします。
DELBADCODEONLY=TRUE

# Office ファイルのサニタイズ処理基準を厳格に適用するか
# (TRUE -> 厳格に適用 FALSE -> 厳格に適用しない（推奨）)
# Office ファイルをサニタイズする前に、ファイル中でリスクになり得る情報まで厳格に
# 判断するかの基準です。
# TRUE の場合、リスク要因が含まれているだけでなく、全く別の観点からサニタイズを
# するか否かの判断を行いますが、誤検知のリスクもあります。
# FALSE の場合は、単純にリスク要因が埋め込まれているか否かだけを判断します。
EXSEEKSTRICT=FALSE

# Office ファイル、ODF 形式ファイルのサニタイズ時に PDF を同時に生成する（Ver.4 で廃止）
# WITHCREATEPDF=FALSE
```

(次のページに続く)

(前のページからの続き)

```
########################################################################
# PDF ファイルのサニタイズ処理の挙動に関する設定
# サニタイズ処理には様々な手法があり、出力されるファイルの形式や態様なども
# 異なってきます。運用の場面などに応じて、最適な組み合わせで設定してください。

# パスワード付きの PDF ファイルを通す
# (TRUE -> 通す FALSE -> パスワード要求レポートを出力し再処理を促す（推奨）)
#  HOLD ->  保留にして受け渡し承認を受ける)
# PDF ファイルのうち、パスワードが付加されたものについては、ファイルの内容が
# スクリプトを含めて暗号化されますので、サニタイズ処理ができません。
# ファイルの発出元が明らかである場合は、発出元を信頼してサニタイズ処理をせずに
# 受け取る運用も考えられますので、導入先のポリシーに応じて設定してください。
ALLOWPDFPASS=FALSE

# PDF ファイルのサニタイズ処理基準を厳格に適用するか
# (TRUE -> 厳格に適用 FALSE -> JavaScript や埋め込みファイルがある場合のみ（推奨）)
# PDF ファイルをサニタイズする前に、ファイル中でリスクになり得る情報まで厳格に
# 判断するかの基準です。
# TRUE の場合、リスク要因が含まれているだけでなく、全く別の観点からサニタイズを
# するか否かの判断を行いますが、誤検知のリスクもあります。
# FALSE の場合は、単純にリスク要因が埋め込まれているか否かだけを判断します。
SANIPDFSTRICT=FALSE

# PDF のサニタイズ処理において、スクリプトの除去を行うか、画像として再作成するか
# (TRUE -> スクリプトの除去（推奨） FALSE -> 画像として再作成)
# スクリプトの除去を行うと、テキストのコピー&ペーストが可能ですが、
# 一部のオブジェクトが残る可能性があります。画像として再作成すると、オブジェクトは
# 完全に消去されますが、処理に時間が掛かり、テキストの抽出はできません。
# 導入の場面により手法を選択してください。
PDFCONVERTDOC=TRUE

# PDF ファイルを画像として再作成する場合の解像度
# (数値は解像度 (DPI). 数値が大きくなると処理時間、ファイルサイズが増加する)
# 上記設定で、画像として再作成する場合の画質を設定できます。
PDFCONVERTDENSITY=300

# PDF ファイルを画像として再作成する場合の一時ファイル領域の上限値
# (数値はディスクサイズ (GB). 余剰ディスクサイズ内の数値とすること)
PDFCONVERTTMPLIMIT=20

########################################################################
# その他のファイルのサニタイズ処理の挙動に関する設定
# サニタイズ処理には様々な手法があり、出力されるファイルの形式や態様なども
# 異なってきます。運用の場面などに応じて、最適な組み合わせで設定してください。
```

(次のページに続く)

(前のページからの続き)

```
# 一太郎形式ファイルをサニタイズする
# (TRUE -> サニタイズする（推奨） FALSE -> _marked フラグをつけて通す)
# 一太郎形式（その他 JustSystem 系のファイルも同様）のサニタイズ処理を
# 行います。標準では TRUE にしています。
SANITIZEJUST=TRUE

# ドキュワークス形式ファイルをサニタイズする
# (TRUE -> 強制的に PDF 形式へサニタイズする（Ver.2 推奨）
#  FALSE -> _marked フラグをつけて通す
#  HOLD ->  保留にして受け渡し承認を受ける（Ver.3 推奨）)
# ドキュワークス形式（XDW) ファイルのサニタイズ処理を行います。ドキュワークス
# 形式はオブジェクトやファイルの添付が可能な形式なため、現時点では強制的に
# PDF に変換する手法を採用しています。
# 注意：この機能は外部モジュールを使っているので、Ver1.24.0 から利用可能
# です。それ以外のバージョンでは、TRUE に設定しても FALSE 相当の挙動になります。
SANITIZEXDW=TRUE

# 疑義のある画像形式ファイル（JPEG, GIF, BMP, PNG 等）を強制的にサニタイズする
# (TRUE -> 強制的にサニタイズする（推奨） FALSE -> 警告メッセージを出力する)
# 画像ファイルの中で疑義のあるものについて、強制的にサニタイズ処理を行います。
# 標準では TRUE にしています。
IMGFORCESANITIZE=TRUE

#########################################################################
# アーカイブファイルのサニタイズ処理の挙動に関する設定
# サニタイズ処理には様々な手法があり、出力されるファイルの形式や態様なども
# 異なってきます。運用の場面などに応じて、最適な組み合わせで設定してください。

# LZH ファイルの解凍処理を行う
# (TRUE -> LZH ファイルを解凍して work フォルダに保存。SANIEXTRACTED に従う。（推奨）
# FALSE -> ブロックする)
# 導入先のポリシーにより、アーカイブファイルの形式に LZH を採用していない場合
# には、この形式をブロックすることができます。
ALLOWLZH=TRUE

# CAB ファイルの解凍処理を行う
# (TRUE -> CAB ファイルを解凍して work フォルダに保存。SANIEXTRACTED に従う。（推奨）
# FALSE -> ブロックする)
# 導入先のポリシーにより、アーカイブファイルの形式に CAB を採用していない場合
# には、この形式をブロックすることができます。
ALLOWCAB=TRUE

# 7z(7-Zip) ファイルの解凍処理を行う
# (TRUE -> 7z ファイルを解凍して work フォルダに保存。SANIEXTRACTED に従う。（推奨）
# FALSE -> ブロックする)
```

(次のページに続く)

(前のページからの続き)

```
# 導入先のポリシーにより、アーカイブファイルの形式に 7z を採用していない場合
# には、この形式をブロックすることができます。
ALLOW7Z=TRUE

# RAR ファイルの解凍処理を行う
# (TRUE -> RAR ファイルを解凍して work フォルダに保存。SANIEXTRACTED に従う。(推奨)
# FALSE -> ブロックする)
# 導入先のポリシーにより、アーカイブファイルの形式に RAR を採用していない場合
# には、この形式をブロックすることができます。
ALLOWRAR=TRUE

# tar ファイルの解凍処理を行う
# (TRUE -> tar ファイルを解凍して work フォルダに保存。SANIEXTRACTED に従う。(推奨)
# FALSE -> ブロックする)
# 導入先のポリシーにより、アーカイブファイルの形式に tar を採用していない場合
# には、この形式をブロックすることができます。
ALLOWTAR=TRUE

# gzip/gz ファイルの解凍処理を行う
# (TRUE -> gzip/gz ファイルを解凍して work フォルダに保存。SANIEXTRACTED に従う。(推奨)
# FALSE -> ブロックする)
# 導入先のポリシーにより、アーカイブファイルの形式に gzip/gz を採用していない場合
# には、この形式をブロックすることができます。
ALLOWGZ=TRUE

# パスワード付き Zip ファイルの解凍を output 側から処理する
# (TRUE -> output 側からの処理で解凍する(推奨)  FALSE -> output 側から処理
# しない)
# パスワード付き Zip ファイルは解凍処理ができないため、通常は
# 「ファイル名_encryptReport.txt」というレポートファイルを出力します。
# このレポートファイルにパスワードを追記して保存することで、パスワードを
# 使った解凍処理をする機能を有効にするかを設定できます。
DECRYPTZIP=TRUE

# アーカイブファイルを解凍後、個別にサニタイズする
# (TRUE -> サニタイズする(推奨)  FALSE -> サニタイズしない)
# アーカイブファイルは、解凍処理されると、work フォルダに個別のフォルダを
# 生成してその中に解凍後のファイルが保存されます。これら解凍後のファイルを
# 自動的にサニタイズ処理させることができます。
# 注意:サニタイズ処理に失敗した場合には、work フォルダ中のファイルを手動で
# サニタイズ処理してください。
SANIEXTRACTED=TRUE

# アーカイブファイル解凍後のファイルサイズの上限値
# (数値はバイト数)
```

(次のページに続く)

(前のページからの続き)

```
# 圧縮率の高いアーカイブファイルを解凍した際に、解凍後のファイルが
# ディスクの容量を超過する可能性があるため、ここで上限値を設定します。
EXTRACTEDSIZE=5120000000

# アーカイブファイルの再圧縮処理を行うか（オプション機能）
# （TRUE -> 行う FALSE -> 行わない）
# アーカイブファイルは解凍後、個別に無害化処理を経て、Output フォルダに出力する仕様と
# なっていますが、これらのファイルを再び Zip に再圧縮する機能を有効にするかを
# 設定します。この機能はオプションのため、設定値は導入先により異なります。
XXXXXXXXXXXXXXXXXX=TRUE

# Zip ファイル再圧縮オプションにおけるタイムアウト時間
# (数値は分)
# 上記オプションにおける無害化処理のタイムアウト時間を設定します。
# この時間を経過すると、その段階で揃っているファイルだけを Zip 再圧縮します。
ZIPTIMEOUT=10

# 使用する端末の OS
# Windows7 は Zip ファイルの文字コードが ShiftJIS として処理されるバグがあります。
# Windows7 環境で Zip ファイル解凍中に文字化けが発生する場合には、使用する OS を
# 明示的に設定してください。
# （WINDOWS7 -> Windows7 OTHER -> それ以外（推奨））
CLIENTOS=OTHER

#########################################################################
# フォルダのサニタイズ処理の挙動に関する設定
# なお、これらの設定は Zip ファイル再圧縮オプションを有効にした場合のみ機能します。

# フォルダのサニタイズ処理を行う
# (TRUE -> サニタイズする（推奨） FALSE -> サニタイズしない)
# サニタイザーは input フォルダにフォルダそのものを投入すると、そのフォルダ配下の
# ファイルやサブフォルダを個別に無害化処理し、Zip ファイルに圧縮した状態で
# output フォルダに出力する機能があります。
# この機能の有効／無効を設定することができます。
ALLOWFOLDER=TRUE

# フォルダを投入した場合のタイムアウト時間。
# (数値は秒数)
# サニタイザーの Zip ファイル再圧縮オプションを有効にすると、input フォルダに
# ファイルだけでなくフォルダをまとめて投入することができます。
# これを過ぎるとその時点で転送済みのフォルダ、ファイルのみを無害化処理します。
# 大容量のファイルを含むフォルダを投入する場合はこの値を調整してください。
FOLDERTIMEOUT=120

#########################################################################
```

(次のページに続く)

(前のページからの続き)

```
# メールファイルのサニタイズ処理の挙動に関する設定
# サニタイズ処理には様々な手法があり、出力されるファイルの形式や態様なども
# 異なってきます。運用の場面などに応じて、最適な組み合わせで設定してください。

# メール添付ファイルのサニタイズをする（メール無害化）
# (TRUE -> サニタイズする（推奨） FALSE -> サニタイズしない)
# メール無害化を目的としてサニタイザーを使う場合には、この設定を TRUE にして
# ください。
# MailDir 形式のメールファイルを読み取り、その中に含まれている添付ファイルを
# 抽出し、work フォルダに保存、サニタイズ処理までを一連で行います。
# 注意：サニタイズ処理に失敗した場合には、work フォルダ中のファイルを手動で
# サニタイズ処理してください。
# 注意：メール無害化の導入を行う場合には、連携するシステムとの調整が必要
# ですので、個別にお問い合わせください。
SANIMAIL=TRUE

########################################################################
# サニタイズ処理全般に関わる設定
# サニタイザー全体に関わる設定です。導入先のポリシーや連携するシステムの
# 制約で設定を変更する場面があるものをまとめています。

# フォルダ監視用のサービスプログラムを選択する（Ver.4.2 以降）
# (INCRON -> incrond を使用する（Ver.4.0 まで）
#  WATCHDOG -> watchdog を使用する（Ver.4.2 以降推奨））
#  サニタイザーは input フォルダにファイルが書き込まれたことを監視して、
#  無害化処理を行う仕組みです。このフォルダ監視のサービスプログラムを
#  選択できます。
TRIGGER=WATCHDOG

# サニタイズ後のファイルを区別するためにプロセス ID を付記する
# (TRUE -> プロセス ID を付記する（推奨） FALSE -> プロセス ID を付記しない)
# TRUE にすることで、output フォルダへ出力するファイル名にプロセス ID を付記
# するようになります。これにより、（おそらく）出力ファイル名が重複することが
# なくなります。
# また、サニタイザーのログから、処理対象のファイルの特定も速やかに行えるように
# なります。
# なお、FALSE にした場合、ファイル名の重複が発生すると、後からサニタイズ処理
# されたファイルが残る仕組みとなります。
ADDPROCESSID=TRUE

# 判別不能なドキュメントファイルを拡張子で識別する
# (TRUE -> 拡張子で識別する（推奨）
#  FALSE -> 拡張子で識別せずに判別不能のまま処理する)
# サニタイザーは基本的にファイルの拡張子は見ず、ファイルの中身からファイルの
# 種別を識別します。
```

(次のページに続く)

(前のページからの続き)

```
# しかし一部のファイルにおいては、ファイルの中身からでは識別できないものも
# あります。その場合、拡張子を手がかりにファイルの種別を推測します。
# 標準では TRUE としていますが、拡張子偽装に伴うリスクを排除するのならば、
# FALSE にしてください。(その場合、種別不明でブロックされる確率が増えます)
EXTDIST=TRUE

# ファイル処理後、Web インターフェースのデータベースを逐次更新する
# (TRUE -> 逐次データベースを更新する(推奨)
#  FALSE -> 定期的にデータベースを更新する)
# サニタイザーはサニタイズの処理の後に、Web インターフェース(ownCloud)の
# データベースを更新して、はじめて Web インターフェースの一覧にファイルが表示
# される仕組みとなっています。
# 設定を TRUE にすることで、サニタイズ処理後、直ちにデータベースの更新を行います。
# これにより画面への反映時間は短くなりますが、その代わりに大量のファイルを
# まとめて処理すると、頻繁にデータベースの書き換えが発生するので、全体的な
# パフォーマンスが低下します。
# FALSE にすると、データベースの更新をサニタイズ処理と関係なく定期的(1分に
# 1回)に行います。パフォーマンスの低下はありませんが、その代わりに画面への
# 反映が最大1分後になります。導入先の状況にあわせて、設定してください。
RESCANDB=TRUE

# WARNING.BINARY フラグがついたファイルを通す
# (TRUE -> 通す  FALSE -> Warning レポートを通知してファイルはブロックする(推奨))
# サニタイズ処理の過程で、最終的に判別不能なファイル(バイナリファイル)が
# 残った場合、このファイルをどのように処理するかを設定します。
# TRUE にすることで、該当ファイルは、WARNING.BINARY という識別子が付いた
# 状態で output フォルダに出力されます。ただし、サニタイズ処理もされない状態の
# ファイルのため相応のリスクを負うことになります。リスクについて十分理解した
# 上で設定してください。標準では FALSE にしています。
THROUGHOUT=FALSE

# WARNING.BINARY フラグがついた実行形式ファイルを通す
# (Ver.4 で廃止。THROUGHOUT に一本化)
# EXECTHROUGHOUT=FALSE

#########################################################################
# トラストパスに関わる設定(Ver.4.2 で廃止)
# サニタイザーのオプションである、トラストパスに関わる設定です。
# トラストパスの導入を検討される場合には、個別にお問い合わせください。

# トラストパスによる復号ファイルを信頼し、そのまま取り込む
# (TRUE -> 取り込む(推奨)  FALSE -> 取り込まずに work フォルダに保存する)
# ファイルの送信者があらかじめファイルに電子署名を付与し、その電子署名が
# 検証されたファイルは、サニタイズ処理を行うことなく、Output フォルダに出力
# する機能です。
```

(次のページに続く)

(前のページからの続き)

```
# なお、この設定を FALSE にすると、署名検証後のファイルは work フォルダに
# 保存され、別途サニタイズ処理を手動で行ってもらうことになります。
# また署名検証に失敗したり、トラストパスの設定がない場合には、
# 「ファイル名_errorReport.txt」というレポートファイルを出力して、
# 処理は終了します。
# TRUSTPASS=TRUE

# トラストパスクライアントを有効化する
# (SANITIZE -> 無害化して有効化　THROUGHT -> 無害化せずに有効化 FALSE -> 有効化しない)
# トラストパスクライアントは、ファイルをトラストパスのみで受け取ることができるよう、
# 署名付き暗号化ファイルとして出力する機能です。
# 媒体経由でファイルを受け渡す際に、この機能を使ってファイルを書き出す運用を想定しています。
# トラストパスクライアントは、output フォルダを監視して動作します。
# このパラメータで FALSE を設定した場合は、output フォルダの監視を行いません。
# SANITIZE を設定した場合は、output フォルダに出力された無害化処理後のファイルを
# 署名付き暗号化ファイルとして signed フォルダに出力します。
# また、THROUGHT を設定した場合は、無害化処理自体をスキップして output にファイルを出力し、
# それらのファイルをそのまま署名付き暗号化ファイルとして signed フォルダに出力します。
# TRUSTCLIENT=FALSE

# トラストパスで使用する自分の鍵 ID
# (値は GPG で生成された鍵 ID)
# トラストパスクライアントはファイルを出力する際に、間違いなく自分から生成・送信され、
# 改ざんされていないことを示すために、電子署名を付加します。
# この電子署名を行うための鍵 ID をここで登録します。
# TRUSTUSER=

# トラストパスで使用する相手の鍵 ID
# (値は GPG で生成された鍵 ID)
# トラストパスクライアントはファイルを出力する際に、第三者にファイルの中身を見られないように
# するために暗号化を施します。暗号化はファイルを受け渡す相手を指定して行います。
# この暗号化を行うための相手の鍵 ID をここで登録します。
# （通常はファイルを受け取るサニタイザーに登録している鍵 ID となります）
# TRUSTADMIN=

#########################################################################
# sanipass/sanicrypt に関わる設定
# サニタイザーのオプションである、sanipass 及び sanicrypt に関わる設定です。
# sanipass/sanicrypt の導入を検討される場合には、個別にお問い合わせください。
# 標準構成ではこれらの機能がは正常に稼働しません。

# sanipass/sanicrypt の出力先フォルダ
# (値はフォルダ名称)
# sanipass/sanicrypt により出力されるフォルダを設定します。これらの機能が
# 有効でない場合には、この値を参照することはありませんので、どのような値でも
```

(次のページに続く)

(前のページからの続き)

```
# 差し支えありません。(念のためコメントアウトしてあります)
# OUTPUTFOLDER=""

# sanicrypt におけるパスワード付与形態
# (FIXED -> 固定設定   RULED -> ルールに基づく設定   LIST -> パスコードリスト
#  REPORT -> パスワードをレポートファイルで出力)
# sanicrypt により、パスワード付き Zip ファイルの出力をする際、パスワードの
# 付与形態を選択することができます。
CRYPTPASSCODE="REPORT"

#########################################################################
# フォルダ内のファイル削除に関わる設定
# サニタイザーで処理した input、output、work フォルダ、また上述の
# スルーパスで設定した出力先フォルダについて、定期的にフォルダ内のファイルを
# 削除するための設定です。

# 毎日午前 4 時に各フォルダの古いファイルとゴミ箱の中身をクリーンアップする
# (TRUE -> クリーンアップする(推奨)  FALSE -> クリーンアップしない)
# TRUE にすることで、毎日午前 4 時にフォルダ内の古いファイルを削除します。
# FALSE にすると、ファイルの削除は行われませんので、利用者が手作業で
# ファイルを削除していただくことになります。
CLEARFOLDER=TRUE

# 古いファイルとゴミ箱の中身をクリーンアップする期間
# (数値はクリーンアップせずに保持しておく日数)
# 標準では 2 日間(48 時間)経過したファイルから順に削除されます。この値を
# 変更することで、フォルダ内に保持しておく期間を設定することができます。
STOREDAYS=2

#########################################################################
# サニタイザーの導入組織に関する設定
# サニタイザーをご利用いただいている組織に関する情報を設定します。

# 自治体コード
# (5 桁数字あるいは 5 桁数字と 1 桁英文字)
LGCODE=

#########################################################################
# サニタイザーの構成に関わる設定
# ネットーワーク間の受け渡し(サニタイザープロ)、負荷分散のためのクラスタ
# 構成など、サニタイザーを複数台で構成する場合に、それぞれのサニタイザーが
# どのような役割を持つのかを設定します。
# 注意:サニタイザーを複数台で構成する場合には、複数台の間の調整が必要です
# ので、個別にお問い合わせください。
```

(次のページに続く)

(前のページからの続き)

```
# サニタイザーのサーバ種別
# (INPUT -> input サーバ OUTPUT -> output サーバ
# STANDALONE -> 一台構成
# ３セグメント間の受け渡しの場合は個別に設定要)
# サニタイザーをどのような役割で稼働させるのかを設定します。
# 標準では STANDALONE となります。
# サニタイザープロの場合、input サーバの役割となるサーバには INPUT、
# output サーバの役割となるサーバには OUTPUT を設定してください。
SERVERTYPE=STANDALONE

# サニタイザープロをクラスタ構成で稼働する（Ver.4 で廃止）
# CLUSTER=FALSE

# input フォルダのファイルをアーカイブ用に別途保管する（Ver.4 で廃止）
# ALLOWARCHIVE=FALSE
```

16.6 /var/www/nextcloud/config/config.php ファイル（サニタイザー Ver.4.0.0）

設定内容の解説は、 https://docs.nextcloud.com/server/24/admin_manual/configuration_server/config_sample_php_parameters.html にあります。

```
<?php
$CONFIG = array (
  'instanceid' => 'xxxxxxxxxxxx',
  'passwordsalt' => 'xxxxxxxxxxxxxxxxxxxxxxxxxxxxxx',
  'secret' => 'xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx',
  'trusted_domains' =>
  array (
    0 => '*',
  ),
  'datadirectory' => '/var/local/sanitizer/data',
  'dbtype' => 'mysql',
  'version' => '24.0.7.1',
  'overwrite.cli.url' => 'http://localhost/sanitizer',
  'dbname' => 'nextcloud',
  'dbhost' => 'localhost',
  'dbport' => '',
  'dbtableprefix' => 'oc_',
  'mysql.utf8mb4' => true,
  'dbuser' => 'nextcloud',
  'dbpassword' => 'nextcloudforsanitizer',
  'installed' => true,
```

(次のページに続く)

(前のページからの続き)

```
  'default_language' => 'ja',
  'default_locale' => 'ja_JP',
  'default_phone_region' => 'JP',
  'defaultapp' => 'files',
  'knowledgebaseenabled' => false,
  'allow_user_to_change_display_name' => false,
  'auth.webauthn.enabled' => false,
  'skeletondirectory' => '/var/local/sanitizer/policy/skeleton',
  'log_type' => 'file',
  'logfile' => '/var/log/nextcloud/nextcloud.log',
  'loglevel' => 2,
  'logdateformat' => 'F d, Y H:i:s',
  'logtimezone' => 'Asia/Tokyo',
  'log_rotate_size' => 104857600,
  'ldapProviderFactory' => 'OCA\\User_LDAP\\LDAPProviderFactory',
  'mail_smtpmode' => 'smtp',
  'mail_smtpsecure' => 'ssl',
  'mail_sendmailmode' => 'smtp',
  'mail_smtpauth' => 1,
  'mail_smtpname' => '',
  'mail_smtppassword' => '',
  'memcache.local' => '\\OC\\Memcache\\APCu',
  'memcache.distributed' => '\\OC\\Memcache\\Redis',
  'memcache.locking' => '\\OC\\Memcache\\Redis',
  'redis' =>
  array (
    'host' => 'localhost',
    'port' => 6379,
  ),
  'maintenance' => false,
  'theme' => '',
  'mail_from_address' => '',
  'mail_domain' => '',
  'mail_smtphost' => '',
  'mail_smtpport' => '',
  'updater.secret' => 'xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx',
);
```

16.7 サニタイザーメッセージ表

分類	メッセージ名（末尾）	発生場面
通過	*_marked.*	ファイルチェック後、不正な要素が検知されず、通過させるべきと判断した。
通過	*_marked.MSOfficeWithPassword	ファイルにパスワードが掛かっているため、チェックできなかった。（Office ファイルであることは判っている）
通過	*_withPassword.*	PDF ファイルにスクリプトの混入は無いものの、パスワードが掛かっているため、無害化処理は行わなかった。
通過	*_withEncrypt.*	PDF ファイルにスクリプトの混入は無いものの、電子署名が付加されているため、無害化処理は行わなかった。
通過	*_withEncryptRMS.*	ファイルに Microsoft RMS による暗号化がされているため、チェックできなかった。
無害化	*_sanitized.*	ファイルチェック後、マクロやスクリプトが検知されたため無害化して通過させた。
無害化	*_semi-sanitized.zip	Zip ファイル中のファイルを無害化し再圧縮を行ったが、タイムアウトにより一部のファイルが再圧縮に含まれていない。
報告	*_decryptReport.txt	トラストパスによる暗号化ファイルの復号に成功したが、ファイルは蔵置されている。(Ver.4.2 で廃止)
報告	*_extractedReport.txt	アーカイブファイルの解凍に成功したが、ファイルは蔵置されている。
報告	*_noticeReport.txt	アーカイブファイル内の無害化処理中に、留意すべきファイルが含まれている。（要因はレポートファイルに書かれている）
報告	*_holdReport.txt	無害化できないファイルをポリシーにより保留にし、HOLD フォルダに配置した。

分類	メッセージ名（末尾）	発生場面
エラー	*_resourceReport.txt	サニタイザーがリソース不足で無害化処理できなかった。
エラー	*_filenameReport.txt	ファイル名に不正な文字列が入っている。
エラー	*_errorReport.txt	サニタイザー側の処理に失敗した。（要因はレポートファイルに書かれている）
エラー	*_directoryReport.txt	ディレクトリ（フォルダ）が input フォルダに入ってきた。
エラー	*_encryptReport.txt	Zip ファイルにパスワードが付与されているため処理できなかった。（再処理指示可）
エラー	*_pdfConvertReport.txt	PDF ファイルのテキストコンバート形式での無害化に失敗した。（再処理指示可）
エラー	*_pdfEncryptReport.txt	PDF ファイルにパスワードが付与されているため処理できなかった。（再処理指示可）
エラー	*_officeEncryptReport.txt	Office ファイルにパスワードが付与されているため処理できなかった。（再処理指示可。Ver.3.7 及び Ver.4 以降）
脅威	*_maliciousReport.txt	ファイル中にマルウェアあるいは挙動の疑わしいコードが検知された。（再処理指示可）
脅威	*_virusReport.txt	ClamAV によるウイルスが検知された。ファイルは即時削除された。
警告	*_warningReport.txt	バイナリファイルなど、通過させるべきでないファイルである。（ブロック）
警告	*_WARNING.BINARY	バイナリファイルなど、通過させるべきでないファイルである。（通過）