

Dabble in Extra Edition 3

Nextcloud 再構築したよ。

Far Northern Other World

目次

目次	7
■ はじめに	8
■ 構築環境	8
■ 構成について	8
■ NextCloud の構築	9
■ まとめ	エラー! ブックマークが定義されていません。
■ あとがき	エラー! ブックマークが定義されていません。

■はじめに

以前、Dabble in の 2 冊目で、各種リモート環境の構築的な記事を書いたのですが、その後諸々あって Nextcloud 環境が崩壊してしまいました。

(まあシンプルに言うとディスク故障によるトラブルです)

そこで、Extra Edition として、今回は Nextcloud の再構築を行ったので、簡単にまとめたいと思います。

なお、以前の号で取り扱った LDAP 連携等については今回触れていません。

あくまで Nextcloud の周辺事項のみです。

■構築環境

構築の際の各種環境について簡単に記載します。

使用ディストリビューションについて

今回は Ubuntu Linux 20.04 LTS を特に特別なことをせずにインストールを完了します。

ハードウェア環境について

今回はストレージ領域が必要なこともあり、実機で構築を行います。

CPU: Intel Pentium G620 2.60GHz (2 コア)

メモリ: 16 GB

ディスク: システム領域: 60GB SSD

データ領域用 SATA 2TB HDD x 4

構成について

今回は Nextcloud のデータ領域を、複数台のディスクをまとめて 1 つの大きな領域を確保し、その中にデータ格納用のディレクトリを作成します。

これは以前、インストール前にローカルディレクトリの適切な認識のさせ方の調査が甘く、苦労したところだったので、最初に実施します。

また、Nextcloud へのブラウザからのアクセスは Nginx によるリバースプロキシを用いて SSL 化しますが、今回はその詳細については触れていません。

SSL 化およびリバースプロキシ設定については「dabble in...」Vol.2 を参考にいただければと思います。

その他については、とりあえずは特別なことはしないで構築を進めます。

それでは次ページより順番に構築を進めていきます。

■ NextCloud の構築

複数台の HDD を 1 つのディレクトリとしてアクセスさせる

前回の原稿では行っていなかったのですが、今回は Nextcloud のデータ領域を、複数の HDD をまとめて領域を確保したディレクトリに配置します。

前述したように、今回の環境ではシステム用のドライブ以外にデータ置き場となるドライブを別途用意しています。使用する物理 HDD はすでに PC に接続されており、個別の 4 台の HDD として使用可能、マウントされた状態にしています。

当方の今回の環境では以下のように物理 HDD がマウントされています。

(df 結果から抜粋)

/dev/sda1	1.8T	7.3M	1.7T	1%	/media/disk3
/dev/sde1	1.8T	7.3M	1.7T	1%	/media/disk1
/dev/sdb1	1.8T	14M	1.7T	1%	/media/disk2
/dev/sdd1	1.8T	7.4M	1.7T	1%	/media/disk4

※マウント位置と HDD のデバイスファイル名の順序が整合していませんが諸々諸事情があるのでご了承下さい。

もちろん、LVM を使って HDD をまとめるといった対応を行っても良いのですが、HDD 故障の際のデータへの影響や交換、増設時の対応が正直言って非常に「面倒」な部分があります。

そこで今回は union Filesystem である「mergerfs」を使用します。

Mergerfs のインストールと設定

Mergerfs はいわゆる複数のファイルシステムをまとめて 1 つのファイルシステムのように見せる「union filesystem」の 1 種です。

当方は FreeBSD で geom concat を使用したことがあるのですが(似て非なる部分も多いですが)使用感というか目的としてはこれと似たようなものかと思えます。

※勿論細かい点での相違や類似のアプリケーションがあることも承知していますがここでは今回の目的の達成優先で詳細なところは端折ります。

apt で特に細かいことを気にせずインストール可能です。

```
$ sudo apt install mergerfs
```

インストールが完了したら、設定をしていきます。

/media 配下に新たに「data」のようにディレクトリを作成します。

```
$ sudo mkdir /media/data
```

OS 起動時にマージされたディレクトリを自動マウントさせたいので、`/etc/fstab` に追記します。
記述内容は以下です。改行されちゃってますが全て 1 行で記載します。

```
/media/disk* /media/data fuse.mergerfs
defaults,allow_other,use_ino,nonempty,cache.files=partial,dropcacheonclose=true
,category.create=mfs 0 0
```

上記だと見づらいうえわかりにくいので、以下に分解して記載します。

```
/media/disk*
```

<file system> カラムの内容です。前述した 4 台の HDD をマウントしたデバイスファイル名を指定しています。見ての通りワイルドカード利用が可能です。

```
/media/data
```

<mount point> カラムの内容です。インストール直後に作成したマウントポイントを記述します。

```
fuse.mergerfs
```

<type> カラムの内容です。ファイルシステムタイプを指定しています。他にも一応いくつかオプションはありますが、mergerfs を使用する場合はこうしておいたほうが無難なようです。

```
defaults,allow_other,use_ino,nonempty,cache.files=partial,dropcacheonclose=true
,category.create=mfs
```

<options> カラムの内容です。細かくオプションを指定しています。詳細は後述します。

```
0 0
```

は通常のマウントポイントへの指定と同様、<dump> <pass> のカラムです。

オプションについて

Mergerfs は多数の詳細なオプションを持ちますが、基本的な設定はオフィシャルサイトに記載されており、以下のオプションにて実行するように記されています。

```
allow_other,use_ino,cache.files=off,dropcacheonclose=true,category.create=mfs
```

なので、特に使用して不具合がなければデフォルトの設定のままです。

今回は将来的に何かしら影響が出ることを懸念して、ファイルキャッシュについての設定を追加しています。

```
defaults,allow_other,use_ino,nonempty,cache.files=partial,dropcacheonclose=true
,category.create=mfs
```

`fstab` を更新したら、`/media/data` ディレクトリが正常にマウントされるか確認します。

マウントできることが確認できたら、予め `/media/data` ディレクトリ配下にディレクトリを作成し、そこを後程 `nextcloud` のデータディレクトリに変更します。

NexrCloud のインストールと設定

以前は CentOS にインストールしていたのですが、Ubuntu Linux に現状標準で導入されている snap コマンドを用いてインストールが可能です。

(インストール時のオプションメニューにも nextcloud や rocketchat は含まれており、単に構築するだけなら本当に容易にインストール可能になっています)

Snap コマンドでインストール

Snap コマンドで nextcloud をインストールします。

```
$ sudo snap install nextcloud
```

管理者アカウントの作成

```
sudo nextcloud.manual-install [管理者アカウント] [パスワード]
```

上記の通り、管理者アカウント名とパスワードを指定します。

これで Web ブラウザ経由でアクセスした際に、最初から当該アカウントとパスワードでログイン必要な状態にセットアップされます。

ポート番号の設定

環境によりますが、今回はリバースプロキシ経由で nextcloud を利用するため、必要な設定を行います。

当方の環境ではリバースプロキシの nginx 側で SSL を終端しているので http のポート番号のみ設定します。

```
$ sudo snap set nextcloud ports.http=8080
```

信頼できるドメインの設定

Nextcloud はデフォルトではローカルホストのみが信頼できるドメインとして登録されており、このままだと外部からの接続ができません。

なので接続を許容するドメイン、IP アドレス情報などを登録します。

IP アドレスの登録時には「*」によるワイルドカード指定も可能なようなので、ローカル NW はワイルドカード指定してしまっても問題ないかもしれません。

```
sudo nextcloud.occ config:system:set trusted_domains [n] --value=[ドメイン/IP アドレスを指定]
```

[n]は複数指定する際に、1.2 のように値を指定して登録します。

value の中身は言わずもがな登録したいドメイン名や IP アドレス情報です。

ドメイン情報、URL 情報の設定

ドメインやプロトコル情報、場合によってはサブディレクトリ設定などを行います。

```
$ sudo nextcloud.occ config:system:set overwritehost --value="example.com"
```

ホスト名を指定します。

```
$ sudo nextcloud.occ config:system:set overwriteprotocol --value="https"
```

リモートからの接続時に https で接続させる場合は設定を追加します。

```
$ sudo nextcloud.occ config:system:set overwritewebroot --value="/nextcloud"
```

★今回、当方は設定していません★

nextcloud をサブディレクトリ設置する場合にはこちらを実行します。

```
$ sudo nextcloud.occ config:system:set overwrite.cli.url --value="https://example.com/"
```

リバースプロキシからの接続を行う場合に接続される URL を設定します。

```
$ sudo nextcloud.occ config:system:set trusted_proxie --value" proxie_address"
```

信頼できるプロキシアドレスとして、プロキシサーバのアドレスを指定します。

Nextcloud の設定はこれ以外にもいろいろありますが、snap でインストールした場合はだいたいはいい感じに設定してくれているので、おおよそこのあたりの設定が終わっていれば問題ないかと思います。

さらに、データディレクトリの設定を行います。

```
sudo nextcloud.occ config:system:set datadirectory --value="/media/data/nextcloud"
```

今回、Mergerfs の設定のところで作成した Nextcloud のデータ用ディレクトリを指定します。

また、元々のデータディレクトリ配下にデータがある場合は

```
/var/snap/nextcloud/common/nextcloud/data
```

配下にあるデータを全てコピーします。(.htaccess 含む「.」ファイルのコピーを忘れないようにしましょう)

インストール後の警告メッセージ

Nextcloud インストール直後にログインして、設定⇒管理⇒概要をクリックすると、いくつか警告が表示されます。

実際のところ使い方によっては放置しても問題ないものもありますが、一応消せるものは消していきます。

「Strict-Transport-Security” HTTP ヘッダが最低でも “15552000” 秒に設定されていません。セキュリティを強化するには、セキュリティ Tips で解説している HSTS を有効にすることを推奨します。」

⇒リバースプロキシの Config に

```
add_header Strict-Transport-Security 'max-age=15552000; includeSubDomains';
```

を追記します。今回はリバースプロキシ設定を後程掲載だけしますが、そこに追記する形です。

「ご使用のシステムには、デフォルトの電話地域が設定されていません。これは、国コードなしでプロフィール設定の電話番号を検証するために必要です。国コードなしで番号を許可するには、地域のそれぞれのISO3166-1コード?とともに“default_phone_region”を設定ファイルに追加してください。」
⇒以下コマンドで設定を行います。

```
nextcloud.occ config:system:set default_phone_region --value="JP"
```

「Web サーバーで "/.well-known/caldav" が解決されるように正しく設定されていません。詳細については、ドキュメント?をご覧ください。」

「Web サーバーで "/.well-known/carddav" が解決されるように正しく設定されていません。詳細については、ドキュメント?をご覧ください。」

⇒ややこしいのですが、リバースプロキシ側に以下を設定します。

※オフィシャルサイトの記載通りに、データディレクトリの.htaccess を更新すると、指定したディレクトリにちゃんとリダイレクトされるのですが、警告は消えませんでした。

```
location /.well-known/carddav {
    return 301 $scheme://$host/remote.php/dav;
}
location /.well-known/caldav {
    return 301 $scheme://$host/remote.php/dav;
}
```

これで表示されていた警告が全て消えます。

■ リバースプロキシ設定

今回はあらかじめ設定してあったリバースプロキシ設定を用いているので、nextcloud にはすぐに接続できていますが、参考までに掲載しておきます。

リバースプロキシ側の構築部分は過去の号で掲載していますのでよろしければそちらをご覧くださいと思います。

以下のリバースプロキシ用の Config には前述した 2 つの修正も含めてあります。

ホスト名や IP アドレスは適当なものに変えていますので、各自の環境に合わせたものに変更してください。

```
server {
    server_name    drv.test.com;
    listen 443 ssl;
    listen [::]:443 ssl;

    ssl_protocols TLSv1.2;
    ssl_ciphers EECDH+AESGCM:EECDH+AES;
    ssl_ecdh_curve prime256v1;
    ssl_prefer_server_ciphers on;
    ssl_session_cache shared:SSL:10m;

    # SSL 認証キーの参照先。certbot で自動的に生成されるファイルを指定する。
    ssl_certificate    /etc/letsencrypt/live/userdesk.test.com/fullchain.pem;
    ssl_certificate_key /etc/letsencrypt/live/userdesk.test.com/privkey.pem;
    ssl_dhparam        /etc/ssl/certs/dhparam.pem;

    keepalive_timeout    70;
    sendfile              on;
    client_max_body_size 0;

    proxy_set_header    Host        $host;
    proxy_set_header    X-Real-IP    $remote_addr;
    proxy_set_header    X-Forwarded-Host    $host;
    proxy_set_header    X-Forwarded-Server    $host;
    proxy_set_header    X-Forwarded-For    $remote_addr;
    proxy_set_header    X-Forwarded-Ssl    on;
    proxy_set_header    X-Forwarded-Proto    https;
    location / {
        # プロキシ先のサーバアドレスとポート番号を指定
        proxy_pass    http://192.168.100.50:8080;
    }
    location /.well-known/carddav {
        return 301 $scheme://$host/remote.php/dav;
    }
    location /.well-known/caldav {
        return 301 $scheme://$host/remote.php/dav;
    }
    add_header Strict-Transport-Security 'max-age=15552000;
includeSubDomains';
}
```

■ その他の設定

この辺りは過去の号でも載せていたのですが、今回は家族の写真格納用のサーバとして構築しているので、以前実施していたような LDAP でのアカウント管理はしていません。

ただ、クラウドストレージ的な用途として用いる際に使用すると便利そうなアプリの類を少しだけ書いておきます。

・ Group folders

グループを作成してその中で共有できるフォルダを作成できるアプリです。今回は家族の写真データ共有用のため、家族写真フォルダのようなものを作っています。

・ External storage support

外部ストレージ共有をするためのアプリです。

「設定」⇒「外部ストレージ」から、外部ストレージを追加することができます。

Amazon S3 や他の NextCloud、FTP、SMB、ローカルストレージなど様々なドライブを共有できます。

・ PDF viewer

・ Photos

・ Audio Player

・ Text

・ Video player

このあたりは今のバージョンだとだいたいデフォルトで導入済みになっているようです。

なお今回は扱いませんが、OpenOffice などを利用して完全に Google Apps みたいなこともできますし、SNAP で容易に導入可能そうなので、機会があれば機能追加してみようと思います。

・ File sharing

URL を発行してファイル共有をできるアプリです。GropoFolder を使わずこちらを用いてファイルの相互やり取りを行うなどの使い方ができます。

他にメールで共有するアプリもあります。

あとがき

あんまりにも忙しくて直近で必要に迫られて再構築した Nextcloud の記事しか作成できませんでした。ただそれでも今年はマイナーOS 本の商業出版リリースができましたので、その点はとてもうれしく思います。ご興味あれば是非こちらも見てくださいと思います。

<https://nextpublishing.jp/book/15260.html>



「dabble in...」 Extra Edition3

発行日：2022/09/12 技術書典 13 発行

発行：Far Northern Other World(Fnow)

〒107-0052

東京都港区赤坂1-4-4 富士野ビル 501

辻瀬蒼伊 aka 佐藤誠之

URL：https://www.fnwo.org

メール：makoyuki@fnwo.org

Copyright © 2022 Aoi.t All Rights Reserved

