UNIXサーバー構築 Ⅱ

第9章　サーバー運用・管理 Ⅱ

sudoコマンド

sudoコマンド

任意の**管理者コマンドを任意のユーザーに許可するコマンド**。su コマンドはユーザーを切り替える。Ubuntuではデフォルトで一般ユーザーでログインしてsudoコマンドでコマンドを実行（一時的に管理者になる）する。ただし、すべてユーザーが**sudoコマンドを使用する**ことはできない。sudoコマンドを実行できる**ユーザーを指定する必要**がある。

＊**sudo su –** ・・・管理者になるコマンド

sudoコマンドの設定

①sudoの設定

visudoコマンドにより、sudoが使用できるユーザー、コマンドを指定すうことができる。

sudo visudo　または　sudo su – でrootユーザーになりvisudo

・visudoの書式

ユーザー名　ホスト名=（実行ユーザー権限名）　［NOPASSWD:］コマンド

（誰）　（どのホストで）=(誰として)　（何をする）

〈例〉eccにすべての権限(root)を利用できる。

ecc　ALL=(ALL:ALL) ALL　・・・（ユーザー:グループ）　(ユーザー権限のみ)

ユーザー名

　コマンドの実行を許可するユーザー名かグループ名 もしくはALL

ホスト名

　実行を許可するホスト名かIPアドレス、もしくはALL

実行ユーザー権限名

　コマンド実行時のユーザー（省略時はroot）、もしくはALL

コマンド

　実行を許可するコマンドのパス、もしくはALL

NOPASSWD

　指定すると、コマンド実行時にパスワードを問われない

設定例

rootユーザはどこでも、誰としてでも何でも実行できる。

root ALL=(ALL:ALL) ALL

wheelグループのユーザーはどこでも、誰としてでもパスワードなしで何でも実行できる。

%wheel ALL=(ALL:ALL) NOPASSWD: ALL

#**nojima**は**taketsugu**として**閲覧コマンド(ls,cat)**を実行できる

nojima ALL=(**taketsugu**) **/bin/ls, /bin/cat**

その他の設定

エイリアスを利用した設定

使用できるコマンドをエイリアスにまとめて指定することもできる。

Cmnd_Alias エイリアス名 = コマンドパス（複数可能）

<例>

Cmnd_Alias SHUTDOWN = /sbin/halt, /sbin/shutdown, ¥

/sbin/poweroff, /sbin/reboot, /sbin/init, /bin/systemctl

nojima ALL=(ALL) **SHUTDOWN**　・・・　エイリアスで指定したコマンドを許可

sudoコマンドの利用

②sudoの利用

sudo　実行コマンド

<例>ie2a99 ユーザーでshutdown コマンドを実行する場合

sudo shutdown -h now

＊**NOPASSWD**の設定でない場合は、shutdown実行前に
パスワードを尋ねられる。入力するパスワードは、
ie2a99ユーザーのパスワード入力する。

APT（Advanced Package Tool）

APT

aptは**debian系**のディストリビューションのパッケージ管理ツール。

以前はapt-getであった。またdpkgを進化させたもの。

コマンドの例(installは除く)

apt list ・・・インストールできるパッケージのリスト

apt list --installed ・・・ インストール済みのパッケージのリスト

apt list --upgradable ・・・ アップグレードできるパッケージのリスト

apt search パッケージ名 ・・・ パッケージの検索

■コマンド例の続き

apt show パッケージ名 ・・・ パッケージの情報を表示

apt remove パッケージ名 ・・・ パッケージの削除

apt autoremove ・・・ 不要なパッケージの削除

apt clean ・・・ パッケージのキャッシュの削除

apt update ・・・ パッケージのインデックスをアップデートする

apt upgrade ・・・ パッケージをアップデートする

その他・・・ **apt moo**

システムクロックとハードウェアクロック

Linuxの時間はハードウェアクロックとシステムクロックがある。

①ハードウェアクロック

コンピュータに内蔵された時計、電源が**オフの状態**でも常に動作

②システムクロック

Linuxのカーネル内に存在する時計、電源がオフの状態では動作しない。

※システムクロックは、Liux起動時にハードウェアクロックを参照して設定される、その後は別々に動き続けるため**差が生じる。**

システムクロックの設定①

■dateコマンド

dateコマンドを使用してシステムクロックを設定する。

①システムクロックの表示　：　**date**

②システムクロックの訂正　：　**date　MMDDhhmm[CCYY][. ss]**

＊月、日、時、分、西暦年上2桁、西暦年下2桁、秒

（西暦年、秒は省略可）

<例> 12/15　10:30　に設定　→　**date 12151030**

システムクロックの設定②

表示形式を指定

date “+%Y%m%d%H%M%a%b”　年、月、日、時、分、曜日、月名

<例1> date “+%Y%m%d”　⇒　2023/12/18の場合、**20231218**と表示

<例2> tar czf　`date “+%Y%m%d”`. tar. gz　/home

※　‘（シングルクォーテーション）：囲った中身を文字列として出力

“（ダブルクォーテーション）：囲った変数の中身を文字列として出力

`（バッククォーテーション）：囲った変数の中身をコマンドとして処理し、その結果を出力

⇒　2023/04/18の場合、**20230418.tar.gz**ファイルが**/home ディレクトリにアーカイブ、更に圧縮されて保存される。**

ハードウェアクロックの設定

■hwclockコマンド

① ハードウェアクロックの表示 ： **hwclock –r**

（または、--showまたは、オプション省略 ）

② クロックの訂正 ：

・システムクロックの時刻をハードウェアクロックに設定

hwclock -w （または、--systohc ）

・ハードウェアクロックの時刻をシステムクロックに設定

hwclock -s （または、--hctosys ）

・ハードウェアクロックの時刻を指定時刻に設定

hwclock --set --date 指定時刻(2023-12-05など)

NTP（タイムサーバー）

■NTP（Network　Time　Protocol）

ネットワークに接続された機器類の時刻同期をとるためのプロトコル。

ユーザーのログインなどサーバーはログなどの参照のため正しい時刻を持っていなければならい。

① タイムサーバーの利用　**ntpdate**コマンド（**インストールが必要**）

　指定したタイムサーバーから現在時刻取得

　sudo ntpdate　タイムサーバー名

[-s]　オプションを付けると、実行結果が標準出力ではなく、syslogに出力

<例>　**sudo ntpdate　ntp.nict.jp**

NTP(タイムサーバー:ntpd)の設定

② NTP(ntpd)の運用

ここでは、ntpdをインストールしてNTPサーバーを構築する。

1.ntpdのインストール

sudo apt install –y ntp

2.設定ファイルの修正

sudo vi /etc/ntp.conf

21行目：デフォルト設定はコメントにしてタイムゾーンの NTP サーバーを追記

#pool 0.ubuntu.pool.ntp.org iburst　　・・・コメントにする（pool0～3すべて）

:

#pool 3.ubuntu.pool.ntp.org iburst

Use Ubuntu's ntp server as a fallback.

#pool ntp.ubuntu.com　　・・・　コメントにする

pool ntp.nict.jp iburst

51行目：時刻同期を許可する範囲を追記

restrict ネットワークアドレス mask サブネットマスク nomodify notrap

<例>

restrict 10.200.0.0 mask 255.255.0.0 nomodify notrap

3. NTPサーバーの再起動

sudo systemctl restart ntp

4.確認

sudo ntpq -p

<結果例>

```
     remote           refid      st t when poll reach   delay   offset  jitter
==============================================================================
 ntp.nict.jp     .POOL.          16 p    -   64    0    0.000   +0.000   0.000
-ntp-b2.nict.go. .NICT.           1 u   45   64    1   14.001   +0.904   1.566
+ntp-a3.nict.go. .NICT.           1 u   48   64    1   14.231   +1.847   1.231
-ntp-a2.nict.go. .NICT.           1 u   44   64    1   14.008   +0.710   0.499
*ntp-k1.nict.jp  .NICT.           1 u   46   64    1    8.313   +0.828   0.394
+ntp-b3.nict.go. .NICT.           1 u   45   64    1   14.044   +1.005   1.288
```

パスワード管理①

パスワードに有効期限の設定をするコマンド

※実行前に、ユーザー登録、パスワード登録が必要

chage [オプション] ユーザー名

※オプションを付けない場合は、対話モードで設定

[オプション]

-l パスワードもしくはアカウントの有効情報表示

-m 最低間隔日数　　-M 最大有効期限日数

-d 最終更新日　　-W 有効期限切れ日数

-I(アイ) 有効期限切れ後、使用不能になるまでの日数

-E アカウントを無効にする日付

パスワード管理②

<例>下記のパスワード有効期限仕様を設定する場合のオプション

・パスワードの変更には、最低5日をおかなければならない

・パスワードの最大有効期限は4週間

（28日ごとに変更しなければならない）

・パスワードが切れる3日前から警告が始まる

・パスワードが切れると即時にアカウントは停止される

・アカウントの有効期限は、2023年3月21日

chage -m 5 -M 28 -W 3 -I 0 -E 2023-03-21 ユーザー名

設定情報を表示する **chage -l**（小文字のエル）ユーザー名

セキュリティ
ログインの管理①

・すべての一般ユーザーのログインを禁止する

/etc/nologin ファイルを作成する

※全ての一般ユーザーがログインできなくなる。

ファイルの内容は一般ユーザーに表示するメッセージを書く

・個人ごとの一般ユーザーのログインを禁止する

ユーザーのログインシェルを「/bin/false」

または 「/sbin/nologin」に変更

変更方法（/sbin/nologinに変更する場合）

①**usermod -s /sbin/nologin 一般ユーザー名**

②/etc/passwdファイルを直接変更（viコマンド）

セキュリティ
ログインの管理②

・/etc/nologinの例

```
Backup now. Try later
```

一般ユーザーがログインを試みると
このメッセージが表示される

・/etc/passwdの例

```
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
   ～ 略 ～
ukad08:x:1006:1009::/home/ukad08:/bin/bash
test:x:1007:1002::/home/test:/bin/bash
ie2a99:x:1000:1000:ie2a99:/home/ie2a99:/bin/bash
```

これがログインシェル
/sbin/nologinに書き換えると
ログインできなくなる。

セキュリティ
システムリソースの管理

ユーザーが利用できるリソースを制御する　ulimit

ulimit［オプション「リミット」］

－a　制限の設定値をすべて表示

－c　生成されるコアファイルのサイズ

（プロセスが異常終了する際、メモリの内容を出力するファイル）

－f　シェルが生成できるファイルの最大サイズをブロック単位で指定

－n　同時に開くことができるファイルの最大数

－u　一人ユーザーが利用できる最大プロセス数

－v　シェルが利用できる最大仮想メモリサイズ