



仮想サーバの使用準備

2024年4月22日

ファイルを編集する際に使用する、**viエディタ**の使い方を確認しておきましょう。

（文字の挿入、カーソル移動、ファイルの保存と基本的な使い方だけでOK！

例えば：文字を挿入するとき i キーを押す。コマンドモードへの切り替えは Esc キーを押す。）

windowsからサーバにログインする方

PuTTYやTeraTermをインストールしておきましょう。

これまでの授業等で
「webページを作成して公開してみよう！」と言われたら

s777というユーザ



実習室から

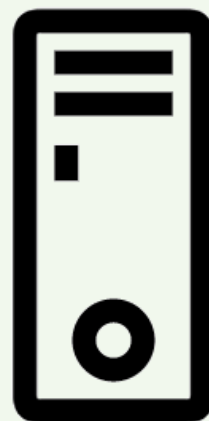
実習室の端末から**ログイン**して
ファイルを編集し、
自分の**wwwというフォルダに保存**

自宅から

WinSCPを利用して、
自分の **www** フォルダに**ファイルを送る**

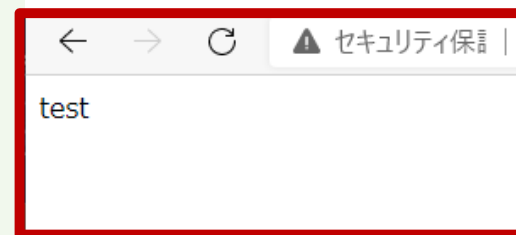
Puttyを利用して、
ssh接続でログインし
自分の**www**フォルダにて、ファイルを編集

全学計算機システム
サーバ



webページを学外から閲覧

<http://www.u.tsukuba.ac.jp/~s777/index.html>



仮想サーバ

- 1 グループに 1 台の「**仮想サーバ**」をお渡しします
 - 仮想サーバとは
 - 物理的なサーバの動作をエミュレート（模倣）するソフトウェア
 - 模倣された電源ボタンや入出力端末が操作できる
 - 家にいながら、物理的なサーバをセットアップする経験ができる
 - ~~OSのインストールからみなさんにやってもらいます~~ **今回OSインストール済みの仮想サーバを用意しました**

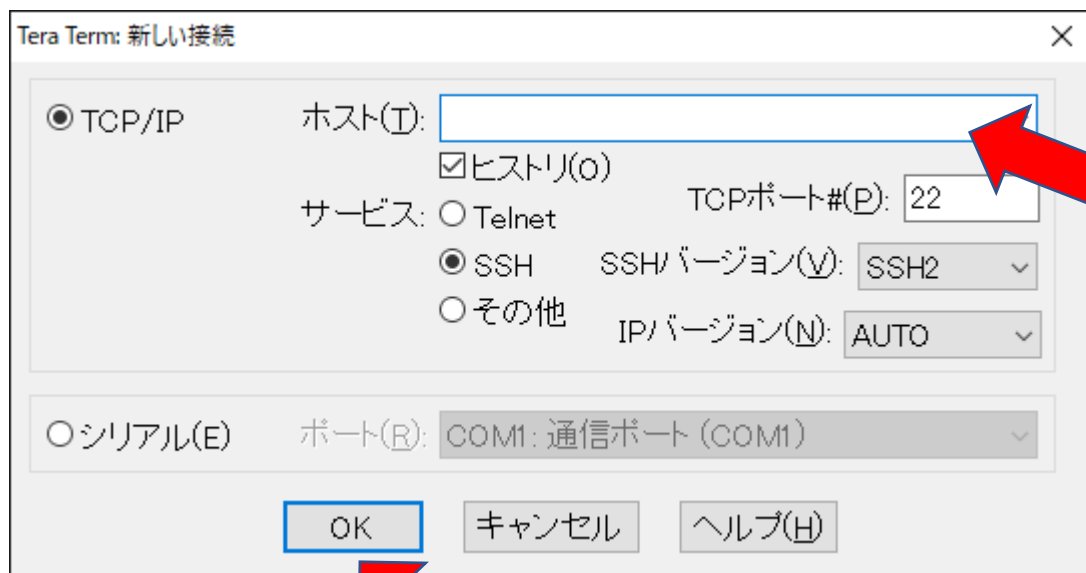
ファイルを編集する際に使用する、viエディタの使い方を確認しておきましょう。
（文字の挿入、カーソル移動、ファイルの保存と基本的な使い方だけでOK！
例えば：文字を挿入するとき i キーを押す。コマンドモードへの切り替えは Esc キーを押す。）

windowsからサーバにログインする方
PuTTYやTeraTermをインストールしておきましょう。

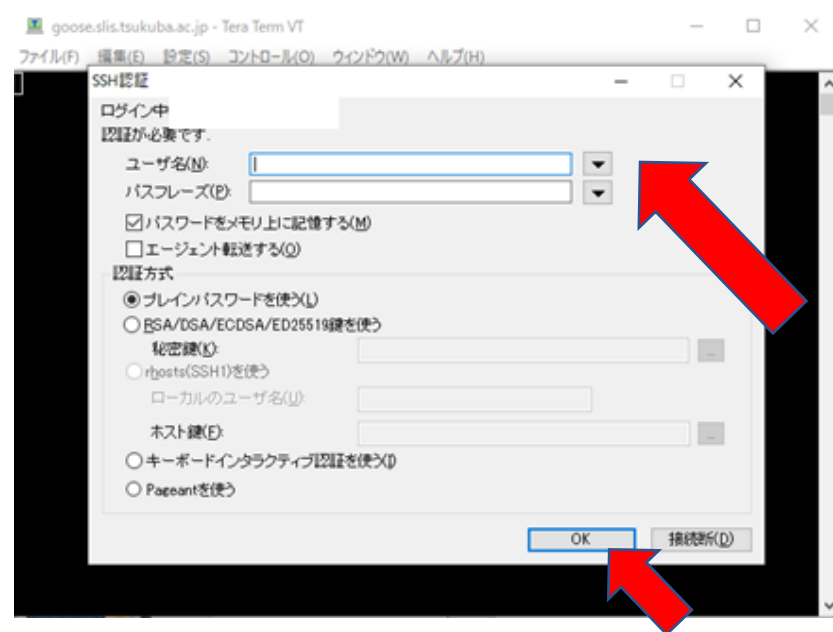
参考：TeraTerm

<https://teratermproject.github.io/>

- ①Tera Termをインストール
- ②起動
- ③ホストの欄にサーバのアドレス(**.**.**.*)を入力し、OKボタンをクリック



- ④ ユーザ名とパスワードを入力し、OKボタンをクリック



- ⑤ \$の後ろにコマンドを入力し、[Enter]キーを押す。
例えば ls (小文字のエルとエス) と入力し、[Enter]キーを押すと、ファイル一覧が表示される。

\$ **ls**

- ⑥ 操作を終えたら、ログアウトする。exit と入力し [Enter]キーを押す。または、
上部メニュー [ファイル] - [Tera Termの全終了] をクリックする。

②

ユーザA



実習室から

実習室の端末から**ログイン**して
ファイルを編集し、
自分の**public_html**というフォルダに保存

自宅から

WinSCPを利用して、
自分のpublic_htmlフォルダに**ファイルを送る**



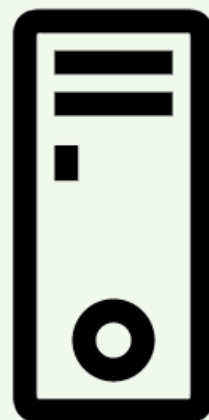
Putty等を利用して、
ssh接続でログインし

自分のpublic_htmlフォルダにて、
ファイルを編集



①

仮想サーバ



①仮想サーバ起動

webサービスを開始するための準備

②ユーザ登録

③webページを作成するための準備

セキュリティ保持のための作業

④プログラムを実行するための準備 (CGI, python, SQL)

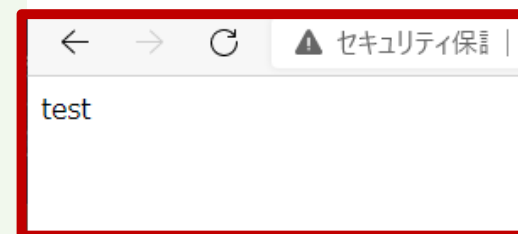
⑤テストページを作成、閲覧確認

⑤

セキュリティ保持が必要！

webページを**学外**から閲覧

http://*.***.***.***/ユーザA/index.html**



セットアップを始める前に

- サーバとは
 - 固定のIPアドレスを持ち、24時間起動しているコンピュータ
 - HTTPなどのリクエストがあった時に、即座にレスポンスを生成して返す
 - 基本的にSSHなどのリモートログインを通して遠隔で作業
 - マルチユーザ
 - サーバの構築とは
 - OSをインストールする（大前提、ここでは**Ubuntu 20.04**）
 - Webサーバソフトウェア（Apache）などのインストールと設定
 - コンテンツ（HTML, CGI）を作成し、サーバに置く
- 今日はこの流れを一通り行って、次回以降開発に取り組む準備を整えます

さくらのVPS

- さくらインターネットのレンタル仮想サーバサービス
- サーバ情報をTeams各グループプライベートチャンネルに掲載しています
- これを自由に使ってWebサービスを実装してもらいます

IPアドレス： **.*.*.*

[管理用ユーザー]

ユーザ名： ****

初期パスワード： ****

←サーバにログイン時に使用する
管理者 ユーザ名とパスワード

まずは
各グループに配布した
データの**赤***の部分を
確認してください。

[VPSコントロールパネル ログイン情報]

URL : <https://secure.sakura.ad.jp/vps/>

IPアドレス： **.*.*.*

パスワード： ****

←ブラウザでこのURLにアクセスし
サーバの起動/シャットダウンを行う

←上記のURLアクセス後に
入力するサーバのアドレスとパスワード

これ以降の作業の進め方

- Teamsでグループ作業
- これ以降のスライドを読みながら作業する
- 役割分担
 - 作業代表者（1名）：最初のサーバインストール作業をする
 - 作業チェック管理者（1名）：作業のチェックリストに記入 & 提出
 - サポート（1 or 2名）：その他のこと
- 解決できない！困った！助けて！という場合
 - 誰かがTeams一般で開いている会議室へ呼びにきてください



進捗状況を確認するために

Teams->GE70113_知識情報システム実習A_2024年度->Class Notebook

->（左上の本棚のアイコンから）_Collaboration Space ->（赤）共同作業スペース->サーバ設定進捗報告 -> サーバ設定チェックリスト
作業を終えた項目に随時チェックをいれてください。

アクティビティ

チャット

チーム

課題

成績

Insights

チャンネル

一般

01G

02G

03G

04G

05G

06G

07G

08G

09G

10G

11G

staff

アプリ

ヘルプ

Class Notebook

ホーム 挿入 描画 表示 ヘルプ クラス ノートブック

必要な作業を入力 ブラウザーで開く 編集

Calibri Light 20 B I U A A ...

第2回から サーバ設定確認用（準備中）

2022年4月14日 13:28

設定が完了した項目にチェックをいれてください。

	1G	2G	3G	4G	5G	6G
1.OSのインストール【代表】	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
2.初回ログイン【代表】	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
3.アカウント（メンバとklls）作成とsudo権限の付与【代表】	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
4.ssh接続の確認とパスワード変更【全員】	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
5.日本語化【分担】	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	1G	2G	3G	4G	5G	6G
6.Apacheのインストール【分担】	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
7.デフォルトwebページの表示	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

項目1～項目11は全グループ
項目12と13はDBを利用する場合のみ設定でよい

以後スライドにチェック
項目提示



番号



これ以降の作業 3 種類



ブラウザを利用

- さくら **VPSコンソールログイン** から **全員のユーザ登録** まで
 - 最初に決めた作業代表者の画面を全員で見る
 - 作業代表以外の人には適宜アドバイスなどをする
 - 作業代表者はインストールなどやったことない人が良い？

代表

- **個々のユーザがSSH接続**を確認後は 2 種類の作業がある
 - 分担して操作する
 - 誰か一人がやれば良い
 - なるべく皆が作業に関われるようにする
 - 全員がもれなく操作する

分担

全員

Tera Term, Putty などを利用



ログイン `https://secure.sakura.ad.jp/vps/`



サーバーのIPアドレスでログイン

IPアドレスでログインすると、対象のサーバーのみ操作が行えます。

IPアドレス

XXX.XXX.XXX.XXX

パスワード

ログイン

ご契約中のすべてのサーバーの操作を行う場合は

IPアドレス：**.**.**.**

[管理用ユーザー]

ユーザ名：****

初期パスワード：****

[VPSコントロールパネル ログイン情報]

URL：https://secure.sakura.ad.jp/vps/

IPアドレス：**.**.**.**

パスワード：****

ログインが成功すると下記の画面が表示される

代表

✓ 1

The screenshot shows the 'サーバー一覧' (Server List) page in the Sakura VPS control panel. The browser address bar shows the URL 'https://secure.sakura.ad.jp/vps/servers?search=&zone&switch&sort=-service_cd'. The left sidebar contains navigation links: 'サーバー' (Server), 'スイッチ' (Switch), 'スクリプト' (Script), 'ストレージ' (Storage), 'サーバー監視' (Server Monitoring), '新規作成' (New Creation), 'ドメイン取得' (Domain Acquisition), 'SSL証明書' (SSL Certificate), and 'DNS登録' (DNS Registration). The main content area has a search bar and filters. Below, a table lists servers. One server is highlighted with a blue arrow pointing to its status '稼働中' (Running), which is accompanied by a green dot. The server details are: Name '名称未設定', Domain 'vs.sakura.ne.jp', IP Address (redacted), Zone '東京第2', and Specs '仮想2Core, 1GB メモリ, SSD 50GB'.

名前	IPアドレス	ゾーン	スペック
<input type="checkbox"/> 名称未設定 vs.sakura.ne.jp		東京第2	仮想2Core 1GB メモリ SSD 50GB

稼働中

*** .vs.sakura.ne.jpと
記載されているところをクリックする

ブラウザのアドレスバーには `https://secure.sakura.ad.jp/vps/servers/4468162/info` が表示されている。

左側のメニューには「サーバー」「スイッチ」「スクリプト」「ストレージ」「サーバー監視」「新規作成」「ドメイン取得」「SSL証明書」「DNS登録」がある。

メインコンテンツエリアの上部には「VPS 名称未設定」と表示されている。その下には「稼動中」のステータスと、以下の操作ボタンがある：

- ▼ 電源操作
- ▼ コンソール
- OS再インストール

これらのボタンは赤い枠で囲まれ、青い矢印が「電源操作」ボタンを指している。

下部には「サーバー情報」タブと「グローバルネットワーク」タブがあり、「サーバー情報」が選択されている。

「基本情報」セクションには以下の項目がある：

項目	値	操作
名前	名称未設定	編集
説明	未設定	編集
ホスト名		
IPv4		
IPv6		
サービスコード		

左下のバナーには「さくらのVPS ならご利用可能！ / Acronis Cyber Protect Cloud バックアップサービス お申し込み」の広告がある。

電源操作、コンソール
というメニューがあることを
確認する

●稼働中

▼電源操作

起動する

シャットダウン

強制停止

強制再起動

説明

ホスト名

もし[稼働中]と表示されていない場合は
[電源操作]－[起動する]
をクリックする。

停止させる場合は、
[シャットダウン]をクリックする。

注意！ シャットダウンは、
利用中のユーザがいらないことを確認の上、
シャットダウンは行うこと。

「起動する」 ボタンをクリックすることは、
物理サーバの電源ボタンを押すことに
対応している

サーバはほとんどシャットダウンせずに使うのが普通だが、
もし今後シャットダウンした場合には「起動する」 ボタンを押して
起動する

VNCコンソールを起動するには、
[コンソール]—[VNCコンソール]をクリックする

電源操作

▼ コンソール

OS再インストール



🖥️ VNCコンソール

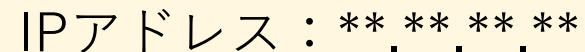
🔧 シリアルコンソール(β)

ワーク

🔍 検索

代表

-  2



ユーザ名：****

初期パスワード：****

URL : <https://secure.sakura.ad.jp/vps/>

IPアドレス： **.*.*.*

パスワード: ****

初回□グイン

代表

 2

```

Ubuntu 20.04.2 LTS tk2-127-63376 tty1

tk2-127-63376 login: ubuntu
Password: _

```

```

Ubuntu 20.04.2 LTS tk2-127-63376 tty
tk2-127-63376 login: ubuntu
Password:
Login incorrect
tk2-127-63376 login: ubuntu
Password:
Welcome to Ubuntu 20.04.2 LTS (GNU/L
* Documentation:  https://help.ubun
* Management:    https://landscape
* Support:        https://ubuntu.co

System information as of 2022-04-24
System load: 0.06           Processes: 1
Usage of /: 8.1% of 45.21GB  Memory: 100%
Memory usage: 20%           Swap usage: 0%

* Super-optimized for small spaces
  footprint of MicroK8s to make it
  https://ubuntu.com/blog/microk8s-
102 updates can be installed immediat
1 of these updates is a security upda
To see these additional updates run:

*** System restart required ***

SAKURA internet [Virtual Private Ser
Last login: Thu Apr 21 10:30:52 JST
ubuntu@tk2-127-63376:~$ _

```

Password:に後ろにパスワードを入力し、
Enterキーを押す。

[注意!]

パスワードを入力しても何も表示されません。

IPアドレス： **.**.**.*

[管理用ユーザー]

ユーザ名：****

初期パスワード：****

今日やること

**パスワードは
強固なものを設定してください！**

- 他のメンバのユーザアカウントを作成
 - 教員が内部を確認できるように教員用アカウントの作成もお願いします
- CUIに慣れる
- Webサーバ構築のためのソフトウェアのインストール
- 試しにWebページを作ってみて、ブラウザで確認する
- 個人ページの有効化やCGIの有効化などの設定

以降の説明に関する注記

- 実行するコマンドの列は以下のように示します：

```
$ pwd
$ cd /
$ ls
bin dev home lib ...
$ pwd
$ cd
$ pwd
```

- **ドルマーク(\$)** はコマンドラインの先頭を表すための記号であり，入力しないでください
- ドルマークが先頭についていない行は，実行結果を示しているのであり，コマンドではありません
- 全ての実行結果を示しているとは限りません

```
matsumur@tk2-204-11573:~$ pwd
/home/matsumur
matsumur@tk2-204-11573:~$ cd /
matsumur@tk2-204-11573:/$ ls
bin    home      lib64      opt      sbin      tmp        vmlinuz.old
boot   initrd.img  lost+found proc      srv        usr
dev    initrd.img.old media      root      swapfile  var
etc    lib         mnt       run       sys       vmlinuz
matsumur@tk2-204-11573:/$ pwd
/
matsumur@tk2-204-11573:/$ cd
matsumur@tk2-204-11573:~$ pwd
/home/matsumur
matsumur@tk2-204-11573:~$
```

sudo コマンド

★ちょっと最初に戸惑う点：

パスワードは入力しても画面上には何も表示されません。
パスワードをキーボードから入力後、
[Enter]キーを押してください。

- （主に）管理者権限でしか実行できないコマンドを使う時に利用
- sudo を最初に実行した時には，パスワードの入力を求められる

```
$ sudo xxxxx
```

```
[sudo] password for uuuuu:
```

xxxxxxは管理者権限でしか実行できないコマンド

ユーザ（uuuu）のパスワードの入力が必要

- sudo コマンドを使えるアカウントを設定可能
 - 現在ログインしている最初のアカウントはsudoが使える
 - あとで必要に応じてsudoの実行権限を与える

ユーザアカウントの作成 (注意)

- 他のメンバのアカウントを作成してください
 - パスワードは**仮想マシンの初期パスワード**として、あとで**各自変更**する

✓ 3

```
$ sudo useradd -s /bin/bash -m xxxxx
$ sudo passwd xxxxx
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
```

xxxxxxは
作成するアカウントのユーザ名

- また、教員が内部を確認するためのアカウントも作成をお願いします

```
$ sudo useradd -s /bin/bash -m klis
$ sudo passwd klis
```

パスワードは、配布した
仮想マシンの初期パスワードに
設定してください

sudo権限の付与

- サーバ内の全てのファイル・コマンドを閲覧・編集・実行できる権限
 - 無闇に与えるものではないが、今回はグループ内のメンバ全員にsudo権限を与えてもよいかもしれません（判断はお任せします）

```
$ sudo gpasswd -a xxxxx sudo
```



- 教員用アカウントには必ずsudo権限を付与しておいてください

```
$ sudo gpasswd -a klis sudo
```

SSH接続の確認とパスワード変更

- **ここからは各自がSSHによるリモートログイン環境**で作業しましょう

✓ 4

- Windowsであれば**PuTTY**や**TeraTerm**など
- Mac, LinuxなどUNIX系OSであればターミナルを開いて以下のコマンド

```
$ ssh xxxxx@aaa.bbb.ccc.ddd
```

xxxxxはユーザ名
aaa.bbb.ccc.dddは
サーバのIPアドレス

- 全員リモートログインできることを確認してください
- 仮パスワードを各自で変更
- **確認できたら、ブラウザの仮想コンソールは閉じてもいいと思います**
 - その際は以下のコマンドか、Ctrl + D でログアウトしてから、ブラウザを閉じます

```
$ exit
```


ここからssh接続

Windowsの方：

Putty や Tera Termを使ってサーバに接続してください。

Mac, LinuxなどUNIX系OSの方：

ターミナルを開いて以下のコマンド

\$ **ssh xxxxx@aaa.bbb.ccc.ddd**

xxxxxはユーザ名
aaa.bbb.ccc.dddは
サーバのIPアドレス

以後、SSHによるリモートログイン環境で作業

日本語化の設定

<https://manual.sakura.ad.jp/vps/os-packages/ubuntu20.html#os>

分担

✓ 5

- デフォルトでは英語環境になっているため、ロケールを日本語に設定する

```
$ sudo apt update
```

```
$ sudo apt -y install language-pack-ja language-pack-ja-base ibus-mozc
```

```
$ sudo localectl set-locale LANG=ja_JP.UTF-8 LANGUAGE="ja_JP:ja"
```

```
$ localectl
```

- 設定が完了したら、再起動

```
System Locale: LANG=ja_JP.UTF-8
```

```
LANGUAGE=ja_JP:ja
```

```
VC Keymap: n/a
```

```
X11 Layout: jp
```

```
X11 Model: pc105
```

```
$ sudo reboot
```

日本語化の設定後の注意

SSH接続では

ために\$のうしろに
date と入力し[Enter]キーを押してみると、
日付が日本語で表示される。

```
102 updates can be installed immediately.
1 of these updates is a security update.
To see these additional updates run: apt list --upgradable
```

```
*** System restart required ***
```

```
SAKURA internet [Virtual Private Server SERVICE]
```

```
Last login: Wed Apr 20 10:19:58 2022
klls@tk2-127-63376:~$ date
2022年 4月 21日 木曜日 10:28:30 JST
klls@tk2-127-63376:~$
```



さくらVPSブラウザから接続の VNCコンソールでは

date と入力し[Enter]キーを押すと、
日付が文字化けした状態で表示されてしまう。

```
Last login: Thu Apr 21 10:26:34 JST 2022
ubuntu@tk2-127-63376:~$ date
2022#  4#  21#  # # #  10:30:54 JST
ubuntu@tk2-127-63376:~$ LANG=C
```

\$のうしろに

LANG=C

と入力しEnterキーを押す。

英語表示に戻るので、

再度 **date** コマンドを実行すると

日付が文字化けすることなく表示される



```
Last login: Thu Apr 21 10:26:34 JST 2022 on tty1
ubuntu@tk2-127-63376:~$ date
2022#  4#  21#  # # #  10:30:54 JST
ubuntu@tk2-127-63376:~$ LANG=C
ubuntu@tk2-127-63376:~$ date
Thu Apr 21 10:32:49 JST 2022
ubuntu@tk2-127-63376:~$
```

CUIに慣れよう

- GUIで出来ることの多くはCUIでもできます
 - フォルダの中身を見たい！ -> **ls** コマンド
 - テキストファイルの中身を見たい！
 - -> **less** コマンド
 - スペースキーで順送り，bキーで逆送り，qキーで終了
 - -> **cat** コマンド
 - 全部標準出力に出すので，短いテキストファイル向け
 - フォルダを作りたい！ -> **mkdir** コマンド
 - ファイルやフォルダをコピーしたい！ -> **cp** コマンド
 - ファイルやフォルダを移動したい！ -> **mv** コマンド
 - ファイルやフォルダを削除したい！ -> **rm** コマンド
 - テキストファイルを編集したい！ -> **vim** や **emacs** （要インストール）
 - 自分がどこにいるのか分からなくなった！ -> **pwd** コマンド
 - 他に誰がログインしてるのかなあ -> **w** コマンド
- 詳しいコマンドの使い方は適宜調べること

Apacheとエディタのインストール

- Webサーバ構築に必要なソフトウェアをインストールします
 - Ubuntuではaptというパッケージ管理ツールを使うことで
コマンドラインでパッケージのインストール・削除ができます

```
$ sudo apt update
```

パッケージリストを最新に更新

```
$ sudo apt install build-essential
```

build-essential パッケージを
インストール

```
パッケージリストを読み込んでいます...
```

```
...
```

```
続行しますか? [Y/n]
```

Y とタイプしてEnter

```
$ sudo apt install apache2
```

apache2 パッケージを
インストール

```
$ sudo apt install vim emacs
```

Vim と Emacs を
インストール

今後作成したページは
`http://aaa.bbb.ccc.ddd/~ユーザ名/index.html`
で閲覧できる

**まずは、ページ公開前に、代表者による作業が必要！
さくらVPSのサーバ設定ページに再度開き、
webページを公開できるようにする。**

`https://secure.sakura.ad.jp/vps/`
サーバのIPアドレスとパスワードを入力してログイン

webページ公のための準備として、接続可能ポート設定を行う

① 「**パケットフィルターを設定**」をクリックする。

SSHだけが表示されている状態となっていることを確認する。

その後 「**パケットフィルター設定を追加する**」をクリックする。

The screenshot shows the VPS management interface for a VPS named "Ubuntu 20.04 amd64 東京第2 1G". The left sidebar contains navigation links: サーバー, スイッチ, スクリプト, ストレージ, サーバー監視, ドメイン取得, SSL証明書, and DNS登録. The main content area shows the "パケットフィルター設定" (Packet Filter Settings) tab. A red arrow points to the "OS再インストール" (Reinstall OS) button. Another red arrow points to the "+ パケットフィルター設定を追加する" (Add packet filter setting) button. The current settings show "SSH" for the protocol, "TCP 22" for the port, and "送信元IPアドレス：すべて許可する" (Allow all source IP addresses). The "設定を保存する" (Save settings) button is visible. At the bottom, there is a warning box that says "パケットフィルターを利用しない" (Do not use packet filter) with a note "(ご自身でファイアウォール等のセキュリティ設定を行う方向け)" (For those who will perform their own security settings like firewall).

VPS Ubuntu 20.04 amd64 東京第2 1G

稼動中

電源操作 コンソール OS再インストール

基本情報 **パケットフィルター設定** ネットワーク サーバー監視情報 コンソール設定

パケットフィルター ?

キャンセル 設定を保存する

SSH TCP 22 送信元IPアドレス：すべて許可する

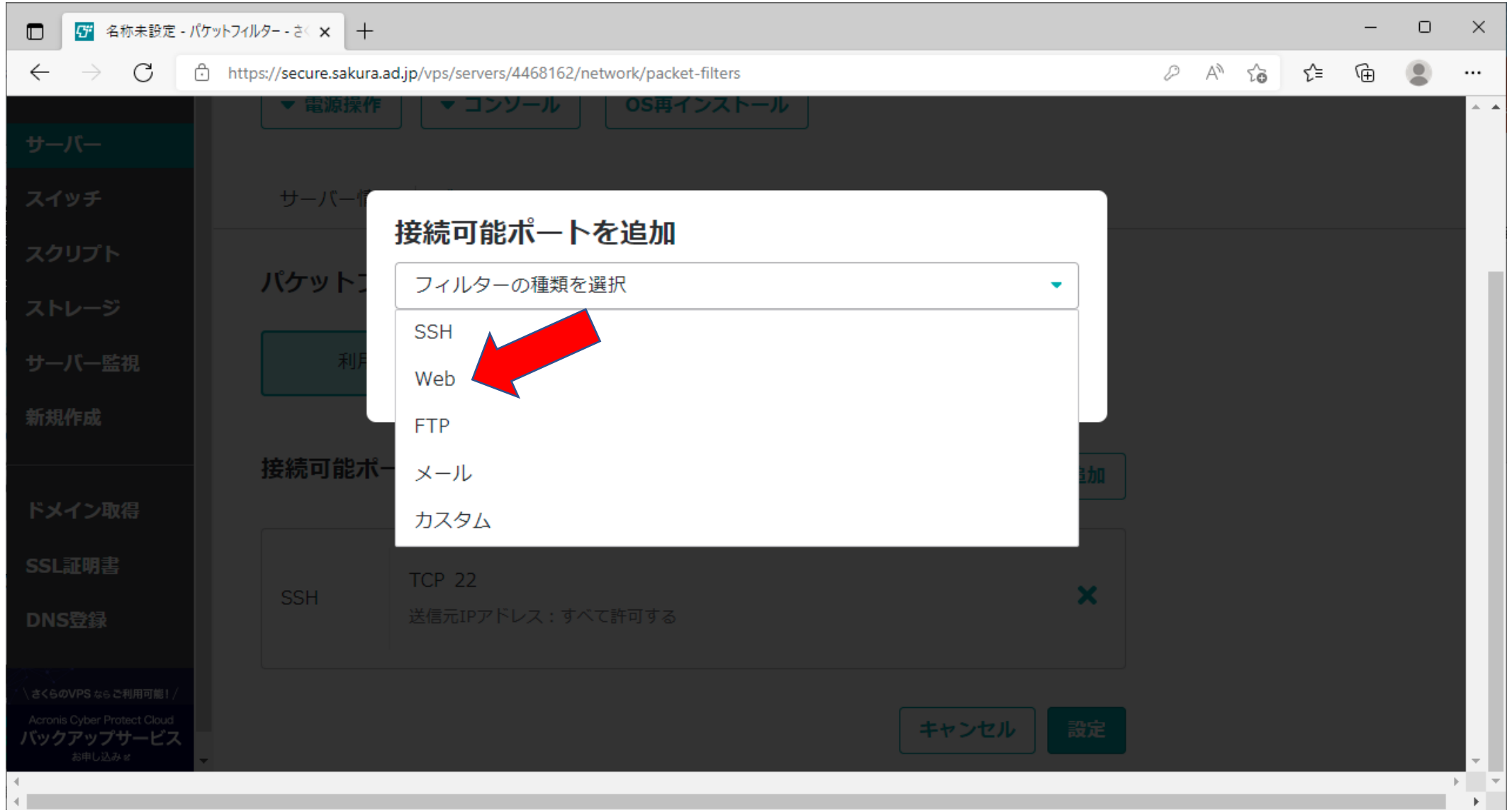
+ パケットフィルター設定を追加する

パケットフィルターをご利用の場合でも エフェメラルポート TCP/UDP 32768-65535番ポート、NTP UDP 123番ポート、IPv6通信、ICMP、フラグメントパケットは全て許可されます。詳しくは [マニュアル](#) をご確認ください。

パケットフィルターを利用しない (ご自身でファイアウォール等のセキュリティ設定を行う方向け)

webページ公開のための準備として、接続可能ポート設定を行う

②フィルターの種類を選択する項目が表示されるので、「Web」をクリックする。



webページ公開のための準備として、接続可能ポート設定を行う
③ プロトコルTCP といった情報が提示されたことを確認後、
「追加」ボタンをクリックする。



④ 「設定を保存する」 ボタンをクリックする。

▼ 電源操作 ▼ コンソール OS再インストール

基本情報 | パケットフィルター設定 | ネットワーク | サーバー監視情報 | コンソール設定

パケットフィルター ?

キャンセル

設定を保存する

SSH TCP 22 送信元IPアドレス：すべて許可する

×

Web TCP 80 / 443 送信元IPアドレス：すべて許可する

×

+ パケットフィルター設定を追加する

パケットフィルターをご利用の場合でもエフェメラルポート TCP/UDP 32768-65535番ポート、NTP UDP 123番ポート、IPv6通信、ICMP、フラグメントパケットは全て許可されます。詳しくは[マニュアル](#)をご確認ください。

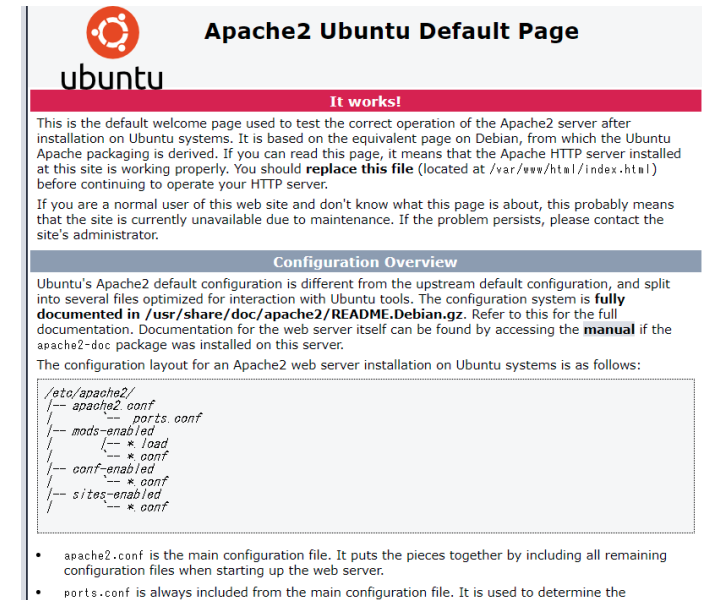
パケットフィルターを利用しない (ご自身でファイアウォール等のセキュリティー設定を行う方向け)

サーバ側の設定完了！

- Apacheをインストールした時点で、Webブラウザのアドレスバーに**IPアドレス**を入力するとデフォルトのWebページが閲覧できるはずである

下記のURLを指定し、
http://aaa.bbb.ccc.ddd/index.html

デフォルトページが
表示されることを確認しよう。



The screenshot shows the Apache2 Ubuntu Default Page. At the top, there's a header with the Ubuntu logo and the text 'Apache2 Ubuntu Default Page'. Below this, a red banner says 'It works!'. The main content area contains a message: 'This is the default welcome page used to test the correct operation of the Apache2 server after installation on Ubuntu systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is derived. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP server.' Below this, there's a section titled 'Configuration Overview' which explains that Ubuntu's Apache2 default configuration is different from the upstream default and is split into several files. It mentions that the configuration system is 'fully documented in `/usr/share/doc/apache2/README.Debian.gz`'. A code block shows the configuration layout for an Apache2 web server installation on Ubuntu systems, listing files like `/etc/apache2/`, `apache2.conf`, `ports.conf`, `mods-enabled`, `load`, `*.conf`, `conf-enabled`, `*.conf`, `sites-enabled`, and `*.conf`. At the bottom, there are two bullet points: '• `apache2.conf` is the main configuration file. It puts the pieces together by including all remaining configuration files when starting up the web server.' and '• `ports.conf` is always included from the main configuration file. It is used to determine the

Webページの作成

- 自分のオリジナルのページに差し替えてみよう

```
$ cd /var/www/html
```

```
$ ls
```

(小文字でエルエスと入力：ファイル一覧表示)

```
$ sudo mv index.html temp.html
```

```
$ sudo vi index.html
```

(emacs派の人はsudo emacs index.html)

```
<html>
```

```
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
```

このページは、○班のWebアプリ建設予定地です！

```
</html>
```

vimで日本語を表示するには...

- Vimを使う場合は，エンコードの設定が必要
- `/etc/vim/vimrc.local` を以下のようにエディタで開いて，

```
$ sudo vi /etc/vim/vimrc.local
```

- 以下の内容を1行だけ記入して，保存して終了

```
set encoding=utf-8
```

- 以降は日本語が正しく表示できるようになる

デフォルトwebページのアクセス制御

分担

- パスワードを考える (複雑なもの)
- .htpasswdファイル (ユーザ名と暗号化されたパスワードのセットを記したファイル) を作る. ユーザ名はkisl2024とする

エル

```
$ cd /etc/apache2
```

```
$ sudo htpasswd -c .htpasswd kisl2024
```

New password:

Re-type new password:

Adding password for user kisl2024

パスワードを入力する。
画面に何も表示されないが、
入力後、[Enter]キーを押す。



デフォルトwebページのアクセス制御

分担

- Basic認証用ファイルを作成する

```
$ sudo vi /etc/apache2/conf-available/basic-auth.conf
```

- ファイルの中身は右の通り

```
<Directory /var/www/html>  
    AllowOverride None  
    AuthType Basic  
    AuthName Documents  
    AuthBasicProvider file  
    AuthUserFile /etc/apache2/.htpasswd  
    Require valid-user  
</Directory>
```

- 設定の有効化

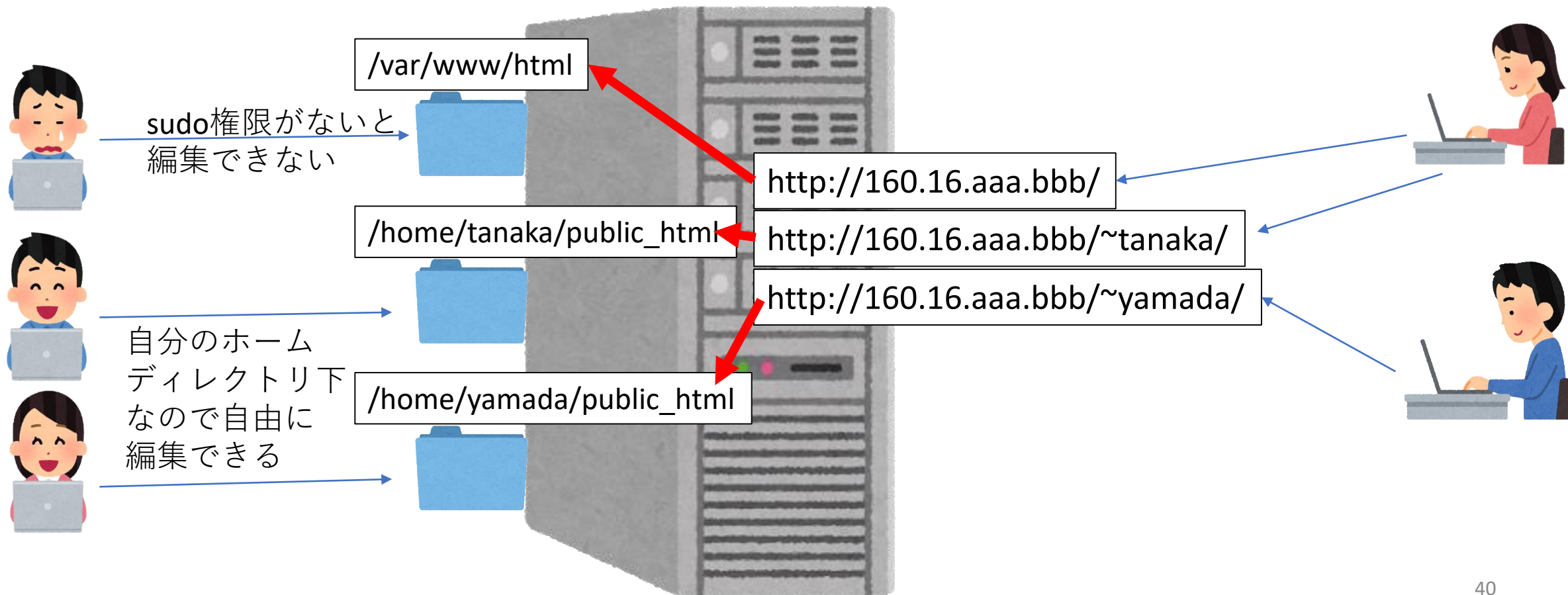
```
$ sudo a2enconf basic-auth
```

- Apacheのリロード

```
$ sudo systemctl reload apache2
```

ユーザディレクトリ

- 各自のホームディレクトリ下の「public_html」フォルダをWebサイトとして公開する機能



ユーザディレクトリの有効化

分担

- それぞれのユーザのホームディレクトリにもWebページを置けるように設定する

(小文字 エル エヌ)

```
$ sudo ln -s /etc/apache2/mods-available/userdir.conf /etc/apache2/mods-enabled/userdir.conf
$ sudo ln -s /etc/apache2/mods-available/userdir.load /etc/apache2/mods-enabled/userdir.load
$ sudo service apache2 restart
```

- 設定が完了したら、メンバー全員が自分のホームディレクトリにページを作ってみてください
- 各自sshで各自のユーザIDでログインし、下記のコマンドを実行

全員

```
$ cd ~
$ mkdir public_html
$ vi public_html/index.html
```

/var/www/html 以下はオーナーがrootなのでsudo権限がないと編集できないが、自分のホームなのでsudoが不要

(小文字 エル エヌ)

ln コマンド (ファイル間のリンク作成)

- 先のコマンドは何をやっているか？

```
$ sudo ln -s /etc/apache2/mods-available/userdir.conf /etc/apache2/mods-enabled/userdir.conf
```

このファイルに対して

この名前でリンクを作成する

- コピーではなくリンクを作成
- Macのエイリアス, Windowsのショートカットに相当するもの
- リンク先が既にあるとエラー
- mods-availableにあるファイルは実体. mods-enabledにあるファイルはリンク
- Apacheに関しては, これらを簡単に実行できる a2enconf, a2enmod, a2ensite というコマンドがある (各自調べよう)

ユーザのwebページのアクセス制御

全員

✓ 10

- 各ユーザのホームに .htpasswd を作成する

🔑 各自のアクセス制御作業の際は
まず自分ホームディレクトリに移動

```
$cd
```

.htpasswdファイル作成

```
$htpasswd -c .htpasswd kisl2024
```

パスワードを何にするか聞かれるので、
パスワードを入力する。
画面に何も表示されないが、
入力後、[Enter]キーを押す。

- 各ユーザの ~/public_html ディレクトリに .htaccess を作成する

```
$cd ~
```

```
$cd public_html
```

```
$ vi .htaccess
```

- 中身は右の通り
- apacheのリロード

```
$ sudo service apache2 restart
```

```
AuthType Basic
AuthName The-T-Directory
AuthBasicProvider file
AuthUserFile /home/xxxxxxx/.htpasswd
require valid-user
```

xxxxxxx は各自のユーザID

CGIの有効化

- 以下のように設定する

```
$ sudo ln -s /etc/apache2/mods-available/cgid.conf /etc/apache2/mods-enabled/cgid.conf
```

```
$ sudo ln -s /etc/apache2/mods-available/cgid.load /etc/apache2/mods-enabled/cgid.load
```

```
$ sudo vi /etc/apache2/mods-available/mime.conf
```

```
#AddHandler cgi-script .cgi
```

と書かれている行（かなり下の方）の#を削除して，以下のようにする

```
AddHandler cgi-script .cgi
```

.cgi という拡張子を
CGIスクリプトとして
認識させる設定

CGIの有効化（続き）

/var/www/ 以下でのCGIの有効化

\$ **sudo vi /etc/apache2/apache2.conf**

```
<Directory /var/www/>
Options Indexes FollowSymLinks
AllowOverride None
Require all granted
</Directory>
```

Options Indexes FollowSymLinks ExecCGI

この行に
右のように
「ExecCGI」と追記

ユーザディレクトリでの
CGIの有効化

\$ **sudo vi /etc/apache2/mods-available/userdir.conf**

```
Options MultiViews Indexes SymLinksIfOwnerMatch IncludesNoExec
と書かれている行に、以下のように「ExecCGI」と追記
Options ExecCGI MultiViews Indexes SymLinksIfOwnerMatch IncludesNoExec
```

\$ **sudo service apache2 restart**

CGI用にpythonのバージョンチェック

まずは、pythonのバージョンチェック

\$ **python3 --version**

Python 3.8.10

← 表示結果

CGIを作る(pythonの場合)

- 各自, 自分のpublic_html以下でCGIを作ってみる. 例えば:

```
$ cd ~/public_html
```

```
$ vi test.cgi
```

```
#!/usr/bin/python3  
# -*- coding: utf-8 -*-  
print("Content-Type: text/plain")  
print()  
print( "hello world.")
```

```
$ chmod a+x test.cgi
```

パーミッションの設定で
「実行権限 (x)」を付与しないと
CGIとして実行できないので注意!

- http://aaa.bbb.ccc.ddd/~ユーザ名/test.cgi を見てみよう

SQLite3のインストール

- SQLite3本体をインストール

```
$ sudo apt install sqlite3 libsqlite3-dev
```



SQLite3でデータベース連携

- 各自DB連携するCGIを作ってみる

```
$ cd ~/public_html  
$ sqlite3 test.db
```

```
CREATE TABLE test (a int, b text);  
INSERT INTO test VALUES (1000, 'abc');  
INSERT INTO test VALUES (2000, 'def');
```

sqlite3補足：

```
$ sqlite3 test.db
```

SQLite version 3.22.0 2018-01-22 18:45:57

Enter ".help" for usage hints.

```
sqlite> ここに命令文を入力;
```

```
sqlite> .exit
```

```
$ vi dbtest.cgi
```

```
#!/usr/bin/python3  
# -*- coding: utf-8 -*-  
print("Content-Type: text/plain¥n¥n")  
import sqlite3  
con = sqlite3.connect('test.db')  
cur = con.cursor()  
  
cur.execute('select * from test')  
for row in cur.fetchall():  
    print(row)  
  
con.close()
```

```
$ chmod a+x dbtest.cgi
```

手元のPCからサーバにファイルを送る

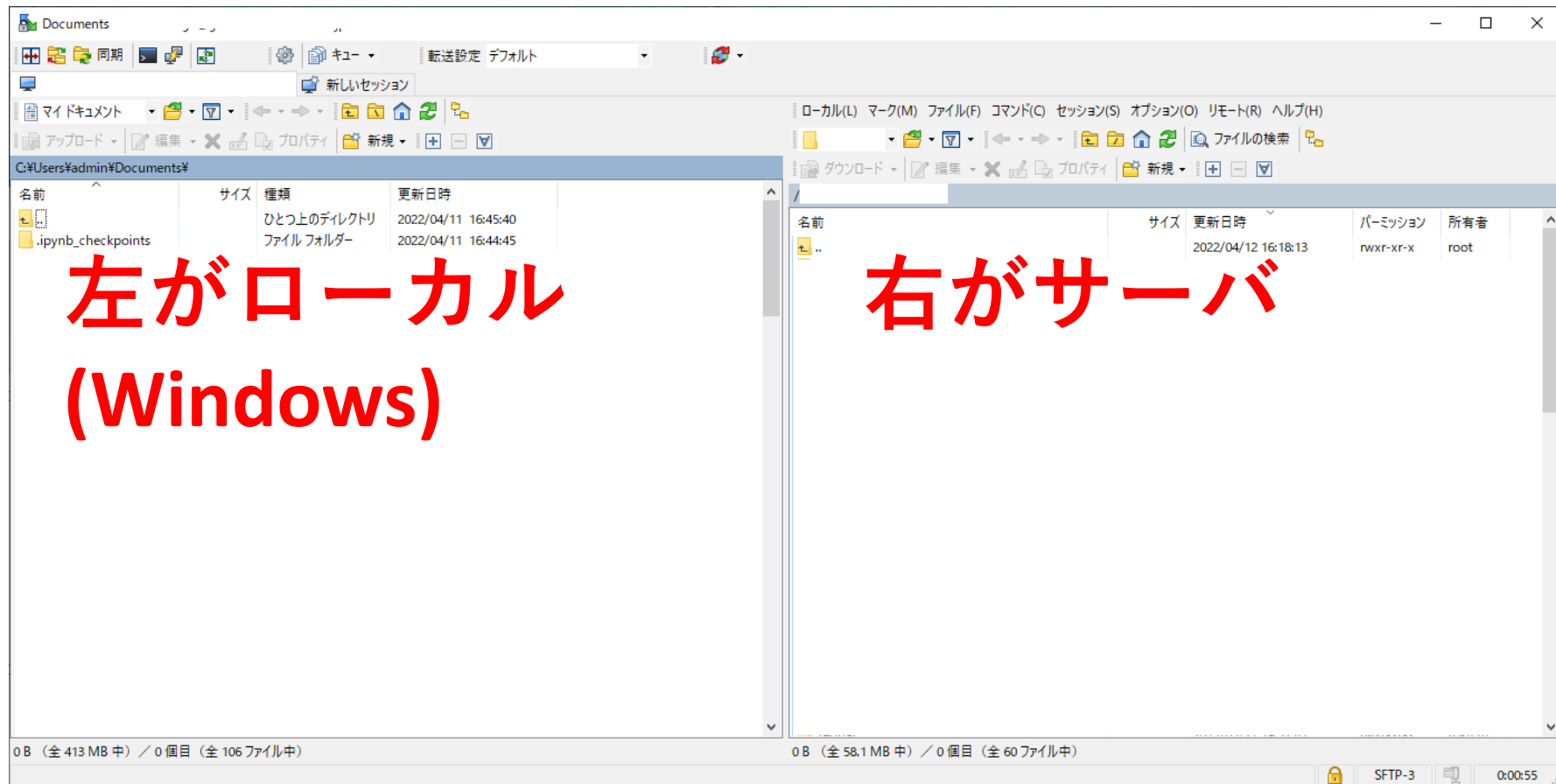
- ファイル転送の方法は色々ある
 - SCPを使う方法
 - WindowsであればWinSCPなどのアプリケーションを使うのがよい
 - MacなどのUNIX系OSの場合はターミナルの**scp**コマンドでも

```
$ scp 送りたいファイル名 xxxxxxxxxxxx@aaa.bbb.ccc.ddd:送り先のディレクトリ
```

- rsyncコマンドを使う方法
- gitを使う方法
- etc.

参考winSCP

- ①WinSCPをインストール
- ②WinSCPを起動
サーバのIPアドレスを指定してログイン
- ③下記のようなウィンドウが表示され、
ドラッグ&ドロップでファイルをアップロードすることができる



次回までに検討すること

- 開発言語を決めておく
 - Pythonにするか. それ以外でも可
- 開発に用いるフレームワークについて相談しておく
 - 高機能なフレームワークほど複雑で学習コストが上がる
 - 普通にCGIを書く方が敷居は低い
 - チームの全員がプログラムを把握してコミットできる方法が望ましい
- 全員がGitHubのアカウントを作成しておく
 - <https://github.com/>

Pythonフレームワーク
Django
Flask など