

仮想サーバの使用準備

2017年4月24日

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
 - この教室にしながら、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

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 16.04）
 - 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



- サーバ情報の書かれた紙を配ります
 - これを自由に使ってWebサービスを実装してもらいます

ログイン

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

[会員IDでログイン](#)
[IPアドレスでログイン](#)

サーバのIPアドレスでログイン
 IPアドレスでログインすると、対象のサーバのみの操作が行えます。

IPアドレス

パスワード

ログインする

OSのインストール

- 「各種設定」->「OSインストール」
 -> 「カスタムOSインストール」
 -> 「Ubuntu 16.04 amd64」を選択->設定内容を確認&実行

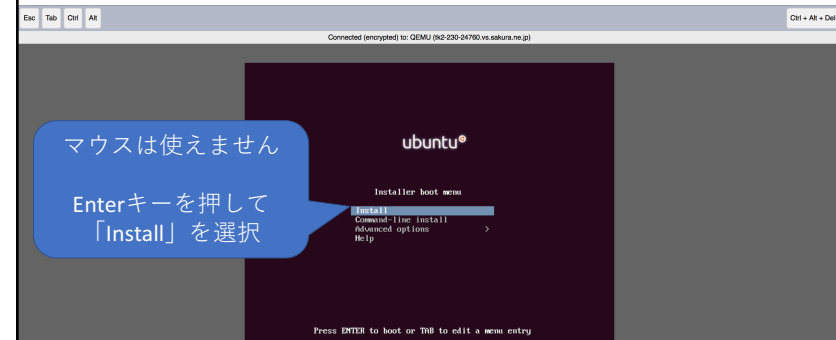


仮想コンソールの起動

- しばらくすると、画面下部にメッセージが出ます
- 「VNCコンソール」を起動してください

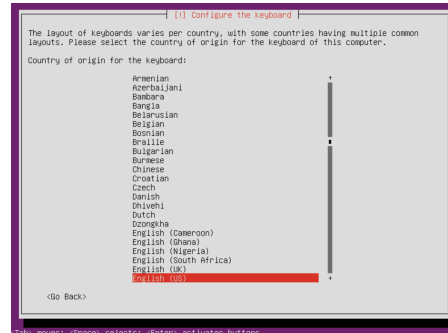


インストールの開始



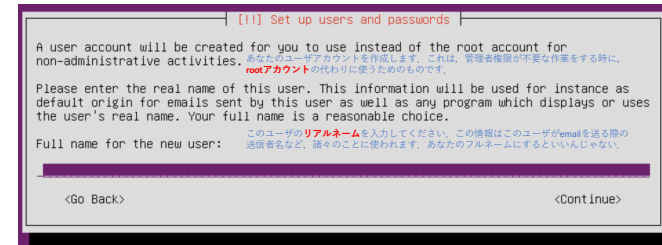
言語の選択

- 下キーでカーソルを動かし「Japanese」を選択
- 次の画面のキーボードレイアウトも「Japanese」でOK



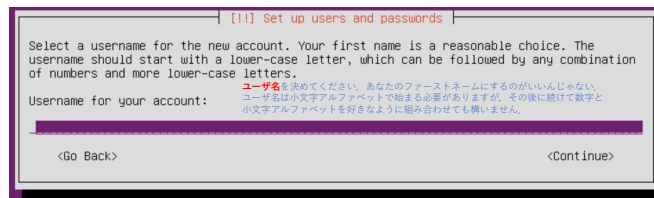
ユーザアカウントの作成

- 作業するためにはユーザアカウントが必要なので作成します
 - 誰か代表者1名を決めて、その人のアカウントとしてください
 - **Root account** : 常に全権限を振るってコマンドを実行できる特殊ユーザ（あぶない）
 - **Real name** : アカウントに紐づけられる実名のことを（UNIXシステムでは）こう呼ぶ。メールの送信者名になったりするので、「Kei Wakabayashi」などが一般的？



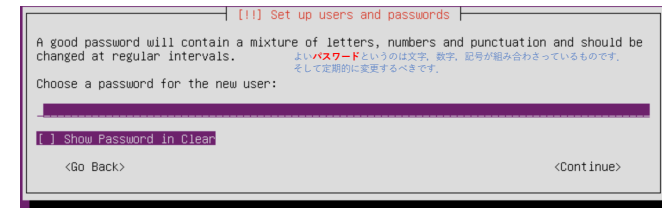
ユーザアカウントの作成

- **Username** : システム上のユーザ名。ログイン時に「ユーザ名」と「パスワード」を毎回入力するが、その時に入力するのはこれ。
- 決まったらEnter
- 戻りたいときはTabを押して「Go Back」にフォーカスを移してEnter



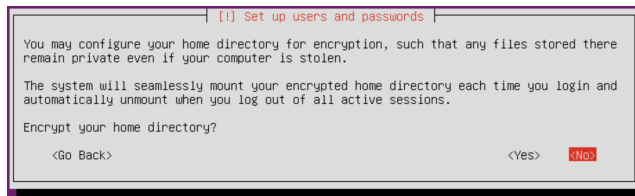
ユーザアカウントの作成

- パスワードを作ってください
 - 脆弱だと破られて入られます
 - 辞書に乗っている単語は禁止、8文字以上、英数字組み合わせで
 - 他のと使い回すのもダメ
 - 決まったらEnter, もう一度入力を求められますので入力してください



ユーザアカウントの作成

- ホームディレクトリの暗号化をするかどうか聞かれます
 - ここでは **No** を選択
- これでユーザアカウントの作成手続きは終わりです



パーティションの設定 + OSインストール

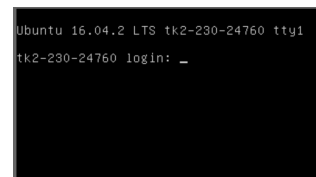
- ディスクのパーティションの設定を行い、**OS**をインストールします
 - ここはさくらのVPSのインストールガイドに従ってください
 - <https://help.sakura.ad.jp/hc/ja/articles/206207821>
 - (あるいは「さくらのVPS カスタムOSインストールガイド」などでWeb検索)
 - 上のURLから「Ubuntu 16.04」を選択して、「パーティションの設定」以降を参照



一から作り直すので
左キーを押して
Yesを選択します。
注意！

OSのインストールが完了したら...

- サーバを起動して、コンソールを開く
 - 作成したアカウント名とパスワードを入力してログインしてください
- GUIはありません**



今日やること

- ファイアウォールを設定する
 - 筑波大学以外のネットワークからアクセスはできないようにしてください
 - セキュリティや著作権等諸々の問題を緩和するためです。必須
 - 学外からアクセスしたいときは筑波大学のVPNサービスを使うのがオススメ！
- 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をお願いします
- 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ものであり、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

```
klis@tk2-230-24760:~$ pwd
/home/klis
klis@tk2-230-24760:~$ cd /
klis@tk2-230-24760:/$ ls
bin dev home lib lost+found mnt proc
boot etc initrd.img lib64 media opt root
klis@tk2-230-24760:/$ pwd
/
klis@tk2-230-24760:/$ cd
klis@tk2-230-24760:~$ pwd
/home/klis
klis@tk2-230-24760:~$ _
```

ファイアウォールの設定

- Ubuntu 16.04ではUFWというソフトウェアで設定ができる
 - 以下の4つのコマンドを実行すればOKです

```
$ sudo ufw default deny
Default incoming policy changed to 'deny'
$ sudo ufw allow from 133.51.0.0/16
Rules updated
$ sudo ufw allow from 130.158.0.0/16
Rules updated
$ sudo ufw enable
Firewall is active and enabled on system startup
```

ユーザアカウントの作成

- 他のメンバのアカウントを作成してください
 - パスワードは本人に手打ちしてもらうとよいでしょう

```
$ sudo useradd -m xxxxxxxxxx
$ sudo passwd xxxxxxxxxx
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
```

xxxxxxxxxxは
作成するアカウントのユーザ名
(一般的なのはファーストネーム)

- 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 -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

- サーバ内のほぼ全てのファイル・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 xxxxxxxxxx sudo
```

- 特に、教員用アカウントにはsudo権限を付与しておいてください

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

ここまでできたら...

- ここからはSSHによるリモートログイン環境で作業しましょう
 - WindowsであればPuTTYやTeraTermなど
 - Mac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 xxxxxxxxxx@160.16.aaa.bbb
```

xxxxxxxxxxはユーザ名
aaa.bbbは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
- 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でログアウトしてから、ブラウザを閉じます

```
$ exit
```

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

Apache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 install build-essential
```

build-essential パッケージをインストール

```
Reading package lists... Done
Building dependency tree
...
```

Y とタイプしてEnter

```
Do you want to continue? [Y/n]
```

```
$ sudo apt install apache2
```

apache2 パッケージをインストール

Webページの確認

- 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
- 自分のオリジナルのページに差し替えてみよう
 - (その前に, エディタをインストールしておいた方がよいかもしれない.
Vim派の人は \$ sudo apt install vim, Emacs派の人は \$ sudo apt install emacs)

```
$ cd /var/www/html
$ sudo mv index.html default.html
$ sudo vi index.html
```

エディタで日本語が表示できない時は...

- Ubuntuには日本語のフォント自体は入っている
- ただしシステムのロケールが **en_US** なので、エディタによってはデフォルトで日本語エンコードを認識しない
 - 2通りの解決策：
 1. エディタのエンコード認識設定を行う（エディタの設定変更だけで済む）
 - 例えば、「vi encoding 日本語」などでWeb検索
 2. システムのロケールを日本語に設定変更する（システム全体の言語設定が変わる）
 - 例えば、「Ubuntu locale 日本語」などでWeb検索

個人ページの有効化

- 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

```
$ 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、それぞれのユーザの個人ページを作ってみてください

```
$ 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

- 作成したページは <http://160.16.aaa.bbb/~xxxxxxxxxx/index.html> で閲覧できる

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/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 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
```

```
$ sudo service apache2 restart
```

CGIを作る

- まずはRubyのインストール

```
$ sudo apt install ruby ruby-dev
```

- 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/ruby -Ks
print "Content-type: text/html\n\n"
print "<html>"
print "Hello World!"
print "</html>"
```

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！

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

- <http://160.16.aaa.bbb/~xxxxxxxxxx/test.cgi> を見てみよう

SQLite3でデータベース連携

- SQLite3本体と, Rubyのsqlite3ライブラリをインストール

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

- 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');
```

```
$ vi dbtest.cgi
#!/usr/bin/ruby -Ks
print "Content-type: text/plain\n\n"
require("sqlite3")
d = SQLite3::Database.open("test.db")
d.execute("SELECT * FROM test;"){ | row |
  p(row)
}
d.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 xxxxxxxxx@160.16.aaa.bbb: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.