Lecture 17

408/508 *Computational Techniques for Linguists*

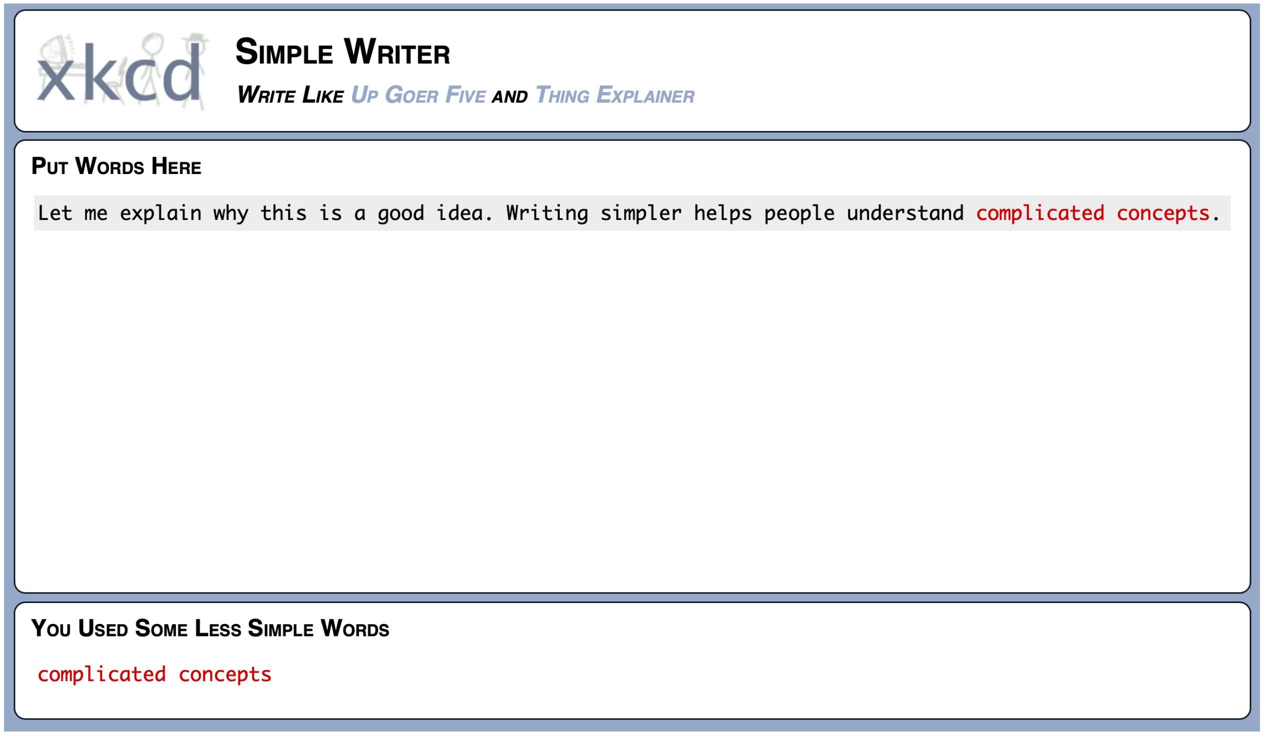
Today's Topics

- Another term project idea
- Creating a webserver (`httpd`)
    - install the Apache2 webserver
        - installed by default on macOS,
        - install using apt install on Ubuntu
    - how to start/stop service
    - DocumentRoot
    - User web files

- Homework 8 (heads-up):
    - *set up your own webserver*
    - *given out on Thursday*

xkcd

https://xkcd.com/simplewriter/



Simple Writer

Write Like Up Goer Five and Thing Explainer

Put Words Here

Let me explain why this is a good idea. Writing simpler helps people understand complicated concepts.

You Used Some Less Simple Words

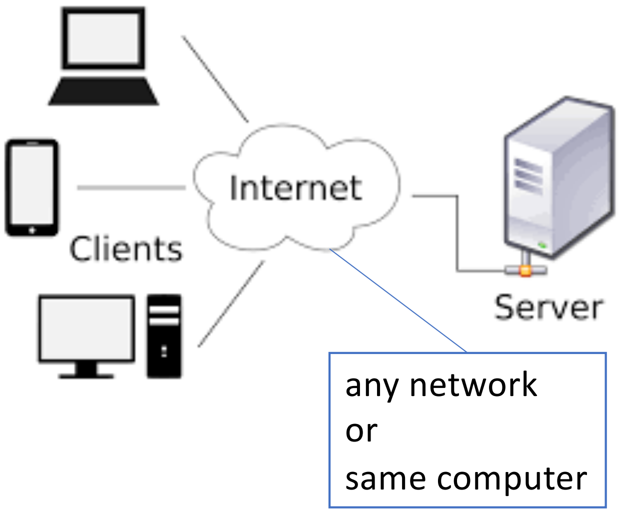
complicated concepts

Demo

- as you type
- Javascript

The server side

- So far, all the web programming has been **client-side** only
  - i.e. the Javascript code is running *inside* the browser
- Let's build a webserver
  - the client-side will send form information to the **server-side** to be processed



any network
or
same computer

Building a Webserver

- We'll use `cgi–bin` and `bash` scripts initially …
- Apache2 is the most common webserver software
  - *https://httpd.apache.org*
  - has been around more than a quarter century
  - *unfortunately, configurations are similar but different on macOS and Ubuntu*

  (we will cover both today)

Common Gateway Interface (CGI)

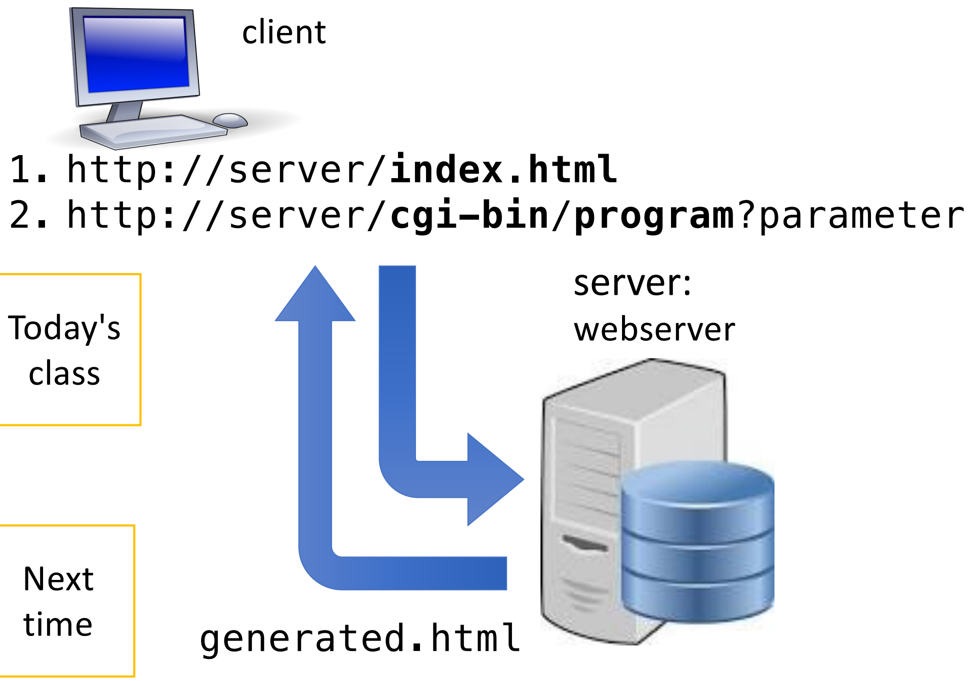
- The glue between a webserver and programs that run on the computer (= server) hosting the webserver

1. Normally, a webserver sends out **static webpages** in response to (URL) requests from a client (your web browser).

Today's class

2. Sometimes, we want the **request to run a program** (a script or binary) on the server that does some computation and generates some result to be displayed on the client (as a webpage).

Next time



```
1. http://server/index.html
2. http://server/cgi-bin/program?parameter
```

httpd

- httpd is the Apache HyperText Transfer Protocol (HTTP) server program.
    - It is designed to be run as a standalone **daemon** process. When used like this it will create a pool of child processes or threads to handle requests.
    - "We fancifully began to use the word **daemon** to describe background processes that worked tirelessly to perform system chores". (Prof. Corbató, MIT)
    - In general, httpd should not be invoked directly, but rather should be invoked via `apachctl` on Unix-based systems

Apache Webserver on macOS

Commands to be entered at a Terminal

```
~$ which httpd
/usr/sbin/httpd
```

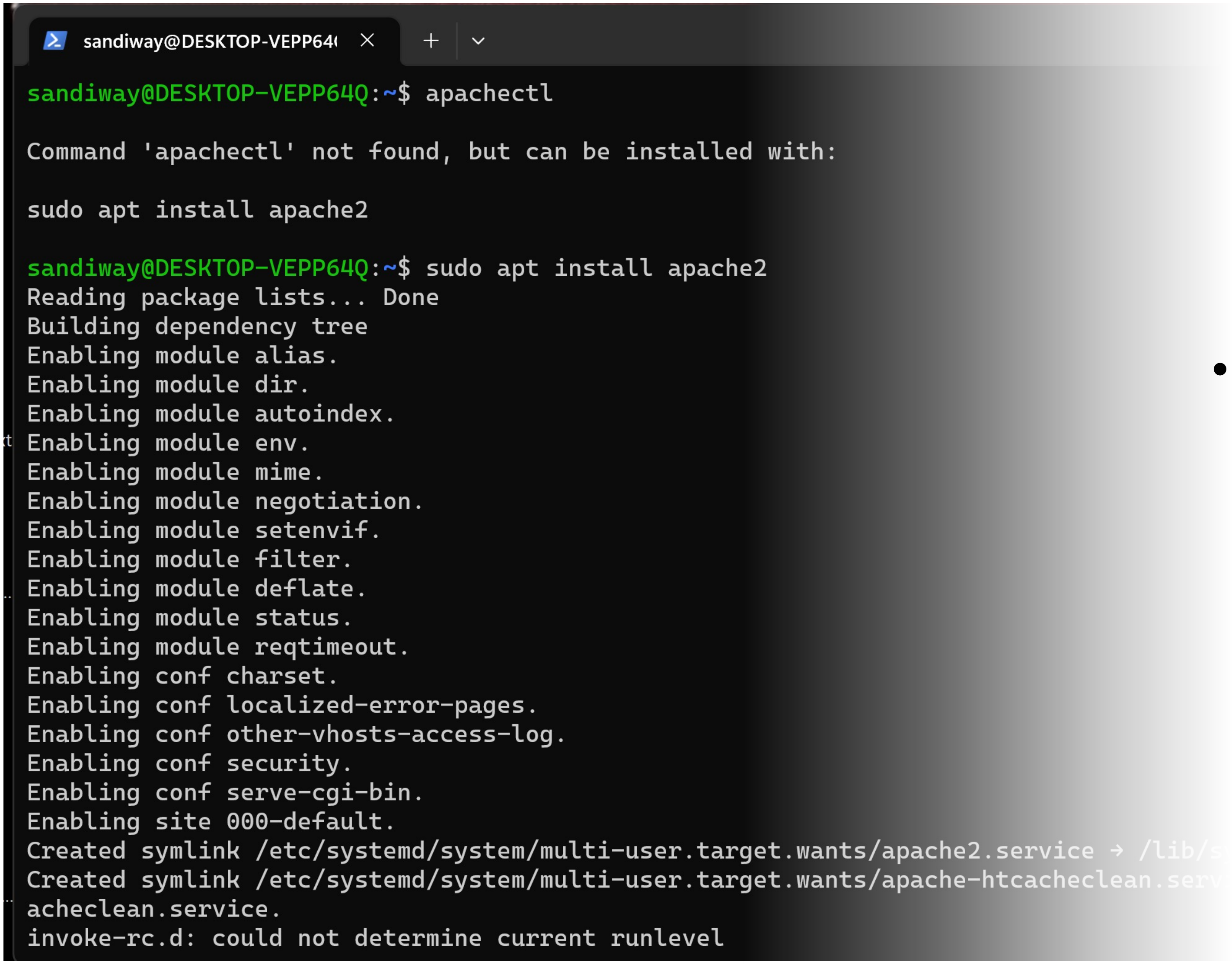
- Apache version (macOS 14.0 *Sonoma*):
  ```
  • ~$ httpd -v
  • Server version: Apache/2.4.56 (Unix)
  • Server built:   Aug  5 2023 06:30:16
  ```

Apache 2.4

- Apache version (OSX 10.13 *High Sierra*):
  ```
  • ~$ httpd -v
  • Server version: Apache/2.4.33 (Unix)
  • Server built:   Apr  3 2018 17:54:07
  ```

Apache Webserver on Ubuntu

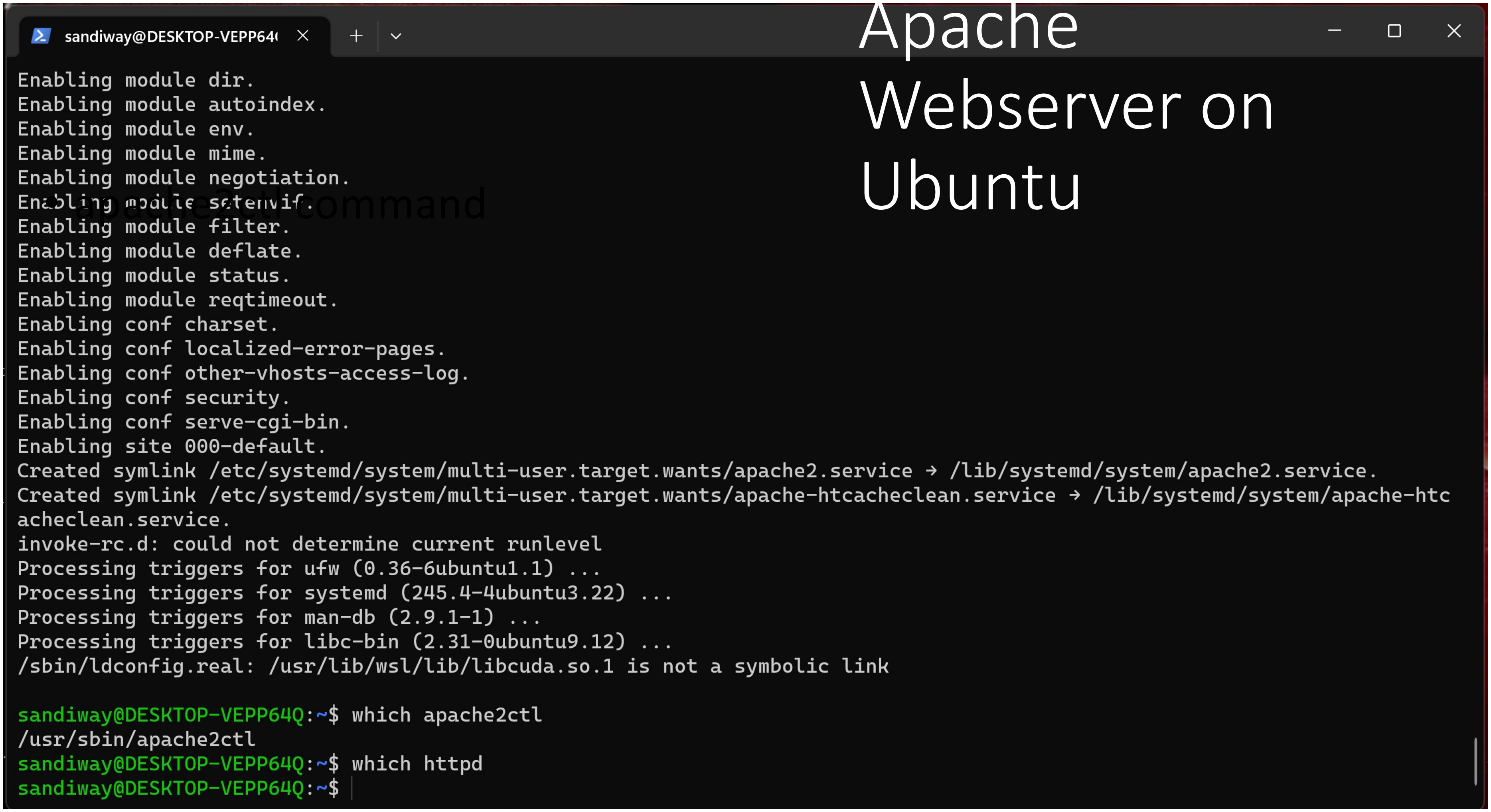
- Ubuntu:
  - `sudo apt-get update`



```
sandiway@DESKTOP-VEPP64Q:~$ apachectl

Command 'apachectl' not found, but can be installed with:

sudo apt install apache2

sandiway@DESKTOP-VEPP64Q:~$ sudo apt install apache2
Reading package lists... Done
Building dependency tree
Enabling module alias.
Enabling module dir.
Enabling module autoindex.
Enabling module env.
Enabling module mime.
Enabling module negotiation.
Enabling module setenvif.
Enabling module filter.
Enabling module deflate.
Enabling module status.
Enabling module reqtimeout.
Enabling conf charset.
Enabling conf localized-error-pages.
Enabling conf other-vhosts-access-log.
Enabling conf security.
Enabling conf serve-cgi-bin.
Enabling site 000-default.
Created symlink /etc/systemd/system/multi-user.target.wants/apache2.service → /lib/s
Created symlink /etc/systemd/system/multi-user.target.wants/apache-htcacheclean.serv
acheclean.service.
invoke-rc.d: could not determine current runlevel
```

Apache Webserver on Ubuntu

apache2ctl command



```
Enabling module dir.
Enabling module autoindex.
Enabling module env.
Enabling module mime.
Enabling module negotiation.
Enabling module setenvif.
Enabling module filter.
Enabling module deflate.
Enabling module status.
Enabling module reqtimeout.
Enabling conf charset.
Enabling conf localized-error-pages.
Enabling conf other-vhosts-access-log.
Enabling conf security.
Enabling conf serve-cgi-bin.
Enabling site 000-default.
Created symlink /etc/systemd/system/multi-user.target.wants/apache2.service → /lib/systemd/system/apache2.service.
Created symlink /etc/systemd/system/multi-user.target.wants/apache-htcacheclean.service → /lib/systemd/system/apache-htc
acheclean.service.
invoke-rc.d: could not determine current runlevel
Processing triggers for ufw (0.36-6ubuntu1.1) ...
Processing triggers for systemd (245.4-4ubuntu3.22) ...
Processing triggers for man-db (2.9.1-1) ...
Processing triggers for libc-bin (2.31-0ubuntu9.12) ...
/sbin/ldconfig.real: /usr/lib/wsl/lib/libcuda.so.1 is not a symbolic link

sandiway@DESKTOP-VEPP64Q:~$ which apache2ctl
/usr/sbin/apache2ctl
sandiway@DESKTOP-VEPP64Q:~$ which httpd
sandiway@DESKTOP-VEPP64Q:~$
```

Apache Webserver on macOS

Commands to be entered at a Terminal

- Apache webserver control:

```
• ~$ which apachectl
• /usr/sbin/apachectl
• sudo apachectl start
• sudo apachectl stop
• sudo apachectl –k restart            (after configuration change)
• apachectl configtest                 (check configuration)
• Syntax OK
• ps –ax | grep httpd                  (ps = process status)
• sudo apachectl stop
• ps –ax | grep httpd
```

ps command

```
NAME
     ps – process status

SYNOPSIS
     ps [-AaCcEefhjlMmrSTvwXx] …

DESCRIPTION
     The ps utility displays a header line, followed by lines containing
     information about all of your processes that have controlling terminals.

    -a      Display information about other users' processes as well as your
             own.  This will skip any processes which do not have a
             controlling terminal, unless the -x option is also specified.
    -x      When displaying processes matched by other options, include
             processes which do not have a controlling terminal.
```

grep command

```
NAME
     grep, egrep, fgrep, rgrep, bzgrep, bzegrep, bzfgrep, zgrep, zegrep,
     zfgrep – file pattern searcher

SYNOPSIS
     grep … [pattern] [file ...]

DESCRIPTION
     The grep utility searches any given input files, selecting lines that
     match one or more patterns.  By default, a pattern matches an input line
     if the regular expression (RE) in the pattern matches the input line
     without its trailing newline.  An empty expression matches every line.
     Each input line that matches at least one of the patterns is written to
     the standard output.
```

Apache Webserver on macOS

```
~$ ps -ax | grep httpd
  154 ??         0:19.63 /usr/sbin/httpd -D FOREGROUND
  517 ??         0:00.00 /usr/sbin/httpd -D FOREGROUND
17404 ttys001    0:00.00 grep httpd
~$ sudo apachectl stop
Password:
~$ ps -ax | grep httpd
17409 ttys001    0:00.00 grep httpd
(base) ~$ sudo apachectl start
(base) ~$ ps -ax | grep httpd
17414 ??         0:00.13 /usr/sbin/httpd -D FOREGROUND
17430 ??         0:00.00 /usr/sbin/httpd -D FOREGROUND
17432 ttys001    0:00.00 grep httpd
```

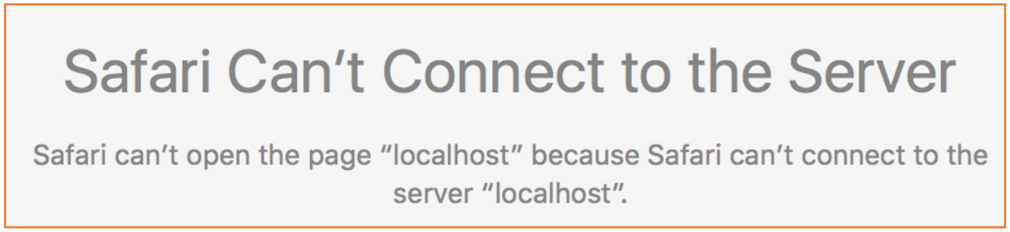
Apache Webserver on macOS

```
~$ sudo apachectl start
Password:
/System/Library/LaunchDaemons/org.apache.httpd.plist: service
already loaded
Load failed: 37: Operation already in progress

~$ apachectl configtest
AH00558: httpd: Could not reliably determine the server's fully
qualified domain name, using Sandiways-MacBook-4.local. Set the
'ServerName' directive globally to suppress this message
Syntax OK
```

Apache Webserver on macOS

- `sudo apachectl start`
- On a browser, enter: http://localhost/ this displays as default the file `/Library/WebServer/Documents/index.html`

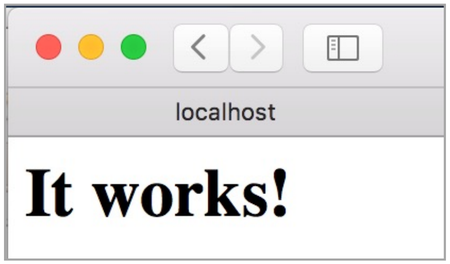
not running…



Safari Can’t Connect to the Server

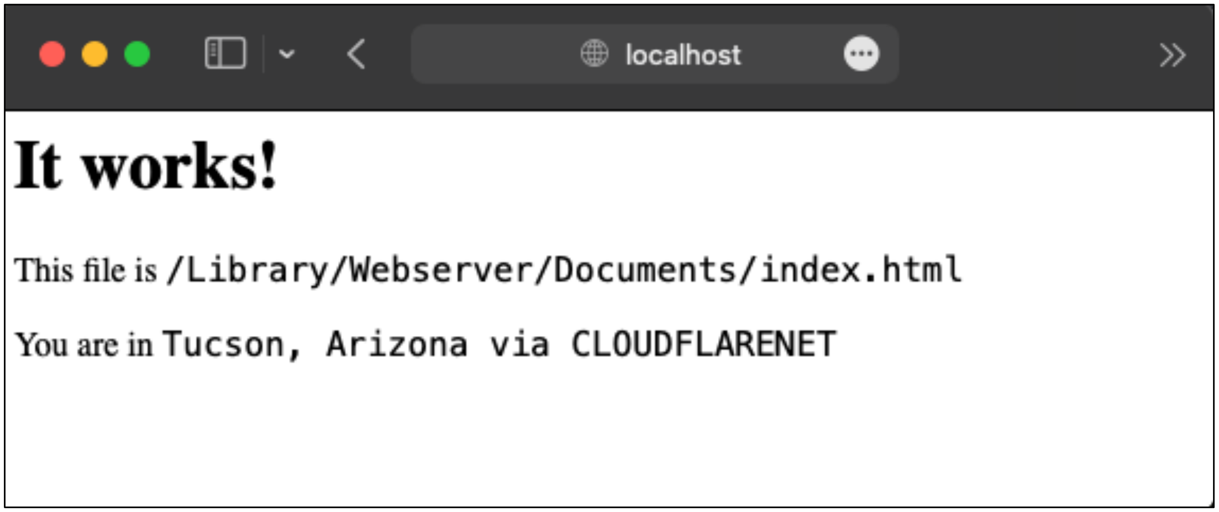
Safari can’t open the page “localhost” because Safari can’t connect to the server “localhost”.

running



It works!

Apache Webserver on macOS



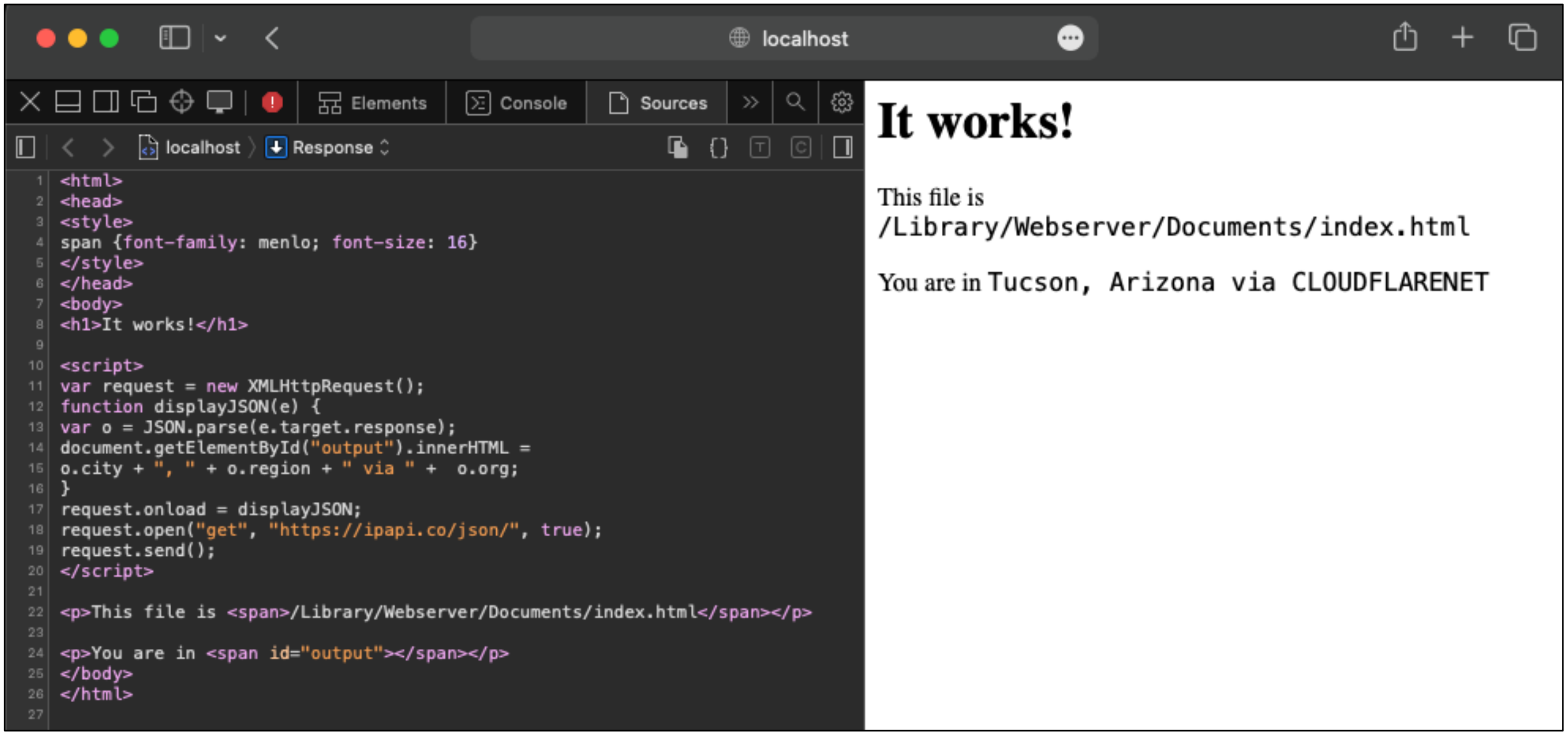
It works!

This file is `/Library/Webserver/Documents/index.html`

You are in `Tucson, Arizona via CLOUDFLARENET`

Apache Webserver on macOS

- On macOS:



```
<html>
<head>
<style>
span {font-family: menlo; font-size: 16}
</style>
</head>
<body>
<h1>It works!</h1>

<script>
var request = new XMLHttpRequest();
function displayJSON(e) {
var o = JSON.parse(e.target.response);
document.getElementById("output").innerHTML =
o.city + ", " + o.region + " via " +  o.org;
}
request.onload = displayJSON;
request.open("get", "https://ipapi.co/json/", true);
request.send();
</script>

<p>This file is <span>/Library/Webserver/Documents/index.html</span></p>

<p>You are in <span id="output"></span></p>
</body>
</html>
```

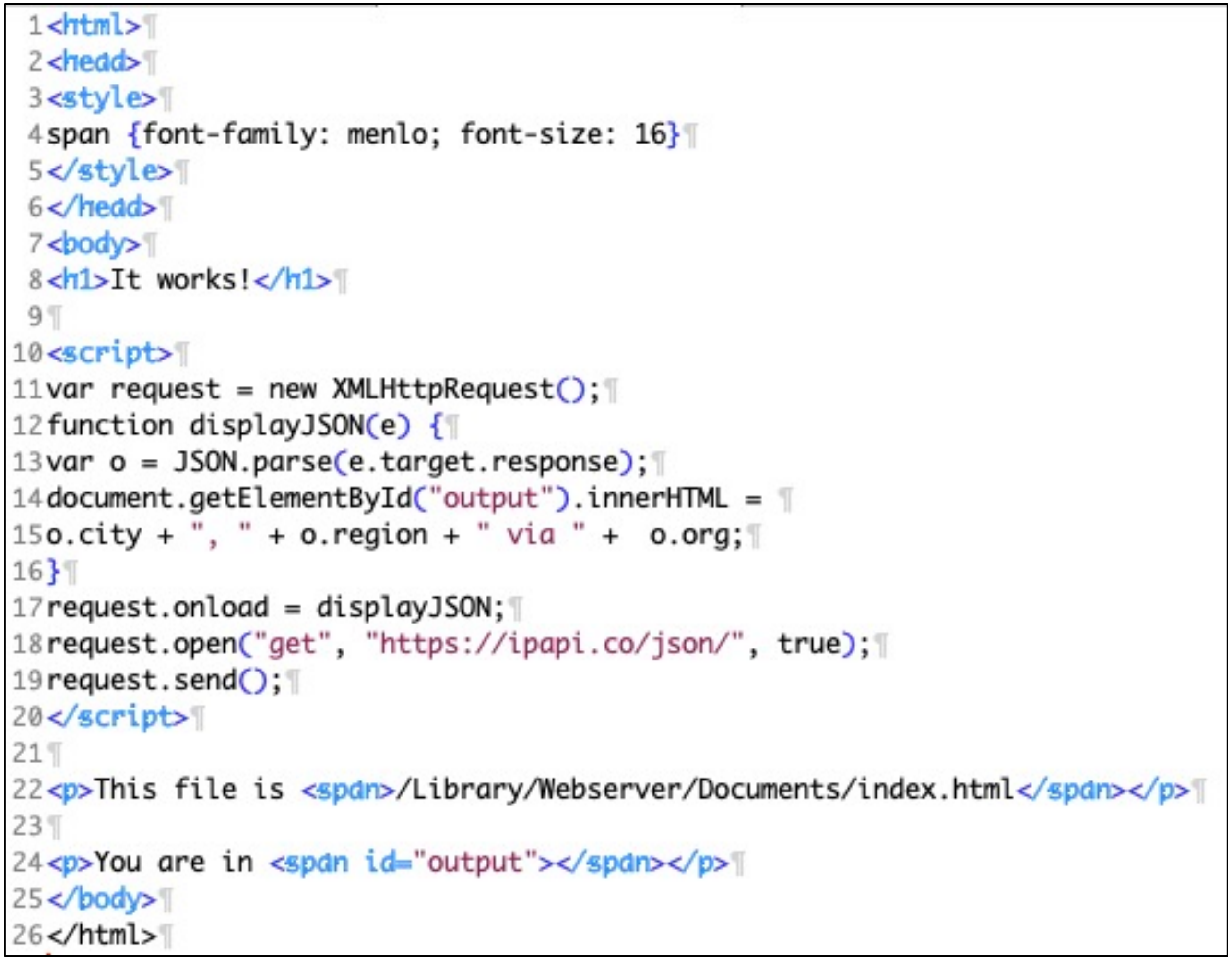
It works!

This file is `/Library/Webserver/Documents/index.html`

You are in `Tucson, Arizona via CLOUDFLARENET`

Sample Site webpage

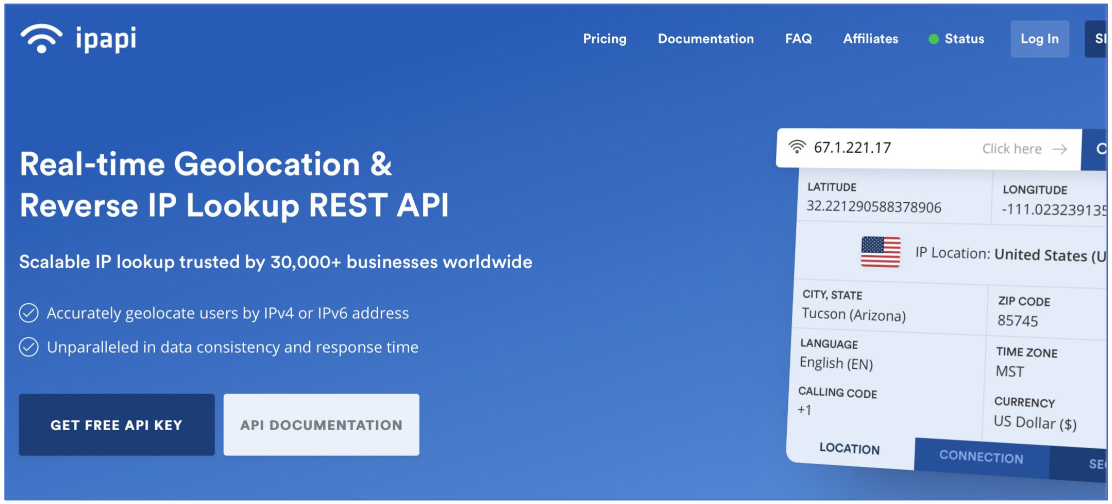
- Normally, Javascript in the browser is sandboxed for security.
  - It has no access to your machine details, e.g. IP address or filesystem
- **So how did we do this?**
- File:
  - `sample-index.html`
  - I renamed it as `index.html` on my computer



```
<html>
<head>
<style>
span {font-family: menlo; font-size: 16}
</style>
</head>
<body>
<h1>It works!</h1>

<script>
var request = new XMLHttpRequest();
function displayJSON(e) {
var o = JSON.parse(e.target.response);
document.getElementById("output").innerHTML = 
o.city + ", " + o.region + " via " +  o.org;
}
request.onload = displayJSON;
request.open("get", "https://ipapi.co/json/", true);
request.send();
</script>

<p>This file is <span>/Library/Webserver/Documents/index.html</span></p>

<p>You are in <span id="output"></span></p>
</body>
</html>
```

Lookup

```
https://ipapi.co/json/
```



Real-time Geolocation & Reverse IP Lookup REST API

Scalable IP lookup trusted by 30,000+ businesses worldwide

- Accurately geolocate users by IPv4 or IPv6 address
- Unparalleled in data consistency and response time

67.1.221.17

LATITUDE 32.221290588378906

LONGITUDE -111.023239135

IP Location: United States (U

CITY, STATE Tucson (Arizona)

ZIP CODE 85745

LANGUAGE English (EN)

TIME ZONE MST

CALLING CODE +1

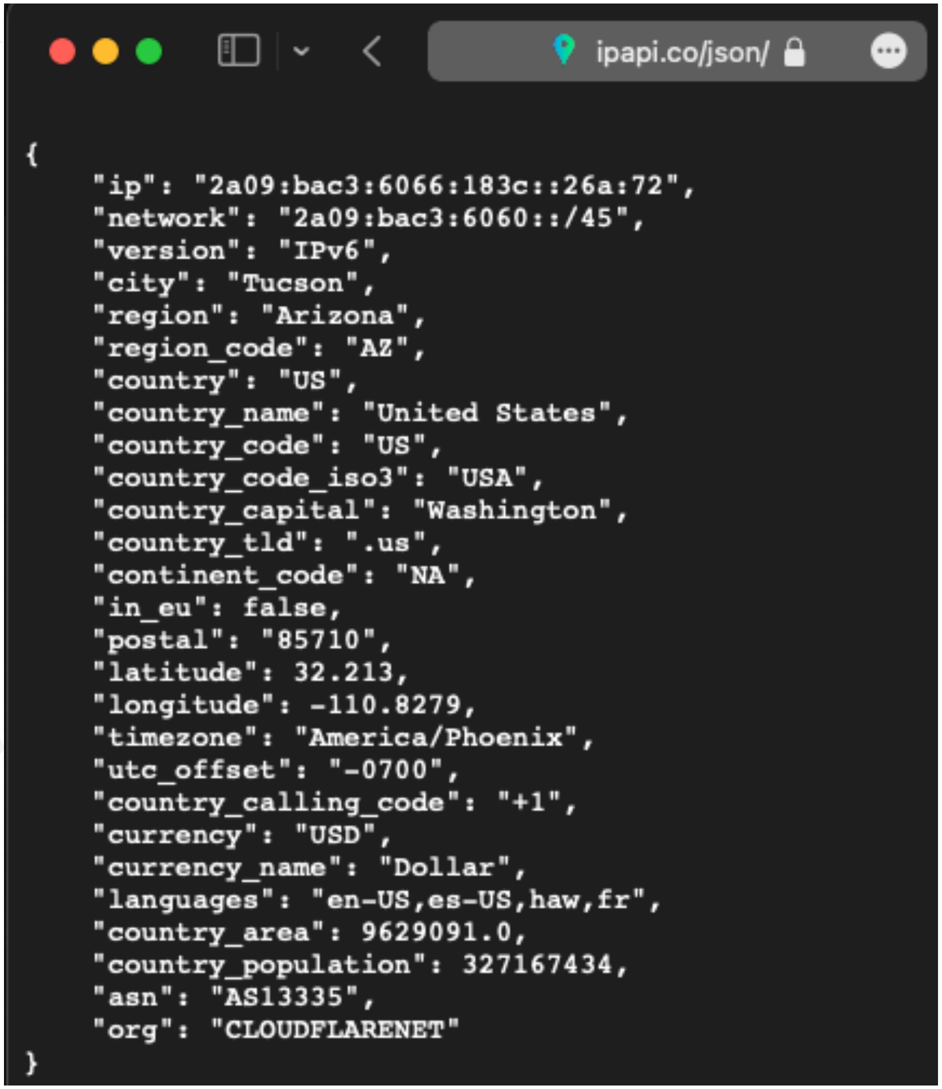
CURRENCY US Dollar ($)

- Terminal:

```
 ~$ curl ifconfig.co/
2a09:bac3:6066:296::42:3d
```

Lookup

- JSON = Javascript Object Notation
- IPv6 format
  - x:x:x:x:x:x:x:x
  - each x is 4 hex digits
  - total: 128 bits
- IPv4 format (*since 1983*)
  - x.x.x.x each x is 2 hex digits
  - total: 32 bits

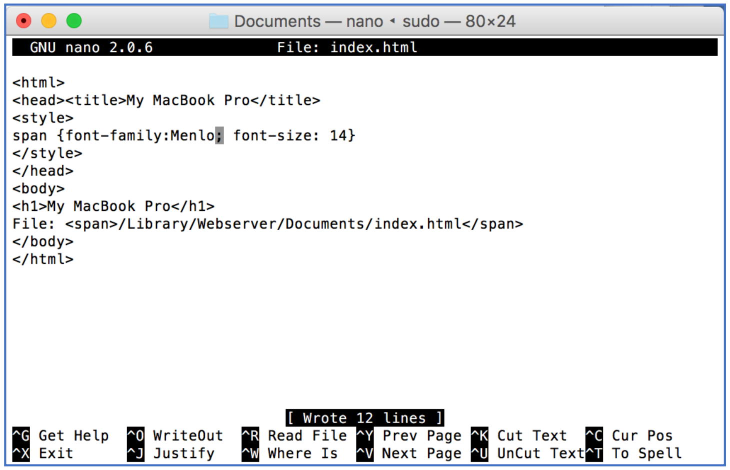


```
{
    "ip": "2a09:bac3:6066:183c::26a:72",
    "network": "2a09:bac3:6060::/45",
    "version": "IPv6",
    "city": "Tucson",
    "region": "Arizona",
    "region_code": "AZ",
    "country": "US",
    "country_name": "United States",
    "country_code": "US",
    "country_code_iso3": "USA",
    "country_capital": "Washington",
    "country_tld": ".us",
    "continent_code": "NA",
    "in_eu": false,
    "postal": "85710",
    "latitude": 32.213,
    "longitude": -110.8279,
    "timezone": "America/Phoenix",
    "utc_offset": "-0700",
    "country_calling_code": "+1",
    "currency": "USD",
    "currency_name": "Dollar",
    "languages": "en-US,es-US,haw,fr",
    "country_area": 9629091.0,
    "country_population": 327167434,
    "asn": "AS13335",
    "org": "CLOUDFLARENET"
}
```

Apache Webserver on macOS

Default static webpage storage location:

- http://localhost/
- **`/Library/WebServer/Documents/index.html.en~orig`**
- Let's create **`index.html`** ourselves!
- **`sudo nano /Library/Webserver/Documents/index.html.en~orig`**



```
  GNU nano 2.0.6                File: index.html

<html>
<head><title>My MacBook Pro</title>
<style>
span {font-family:Menlo; font-size: 14}
</style>
</head>
<body>
<h1>My MacBook Pro</h1>
File: <span>/Library/Webserver/Documents/index.html</span>
</body>
</html>

                              [ Wrote 12 lines ]
^G Get Help  ^O WriteOut  ^R Read File ^Y Prev Page ^K Cut Text  ^C Cur Pos
^X Exit      ^J Justify   ^W Where Is  ^V Next Page ^U UnCut Text^T To Spell
```

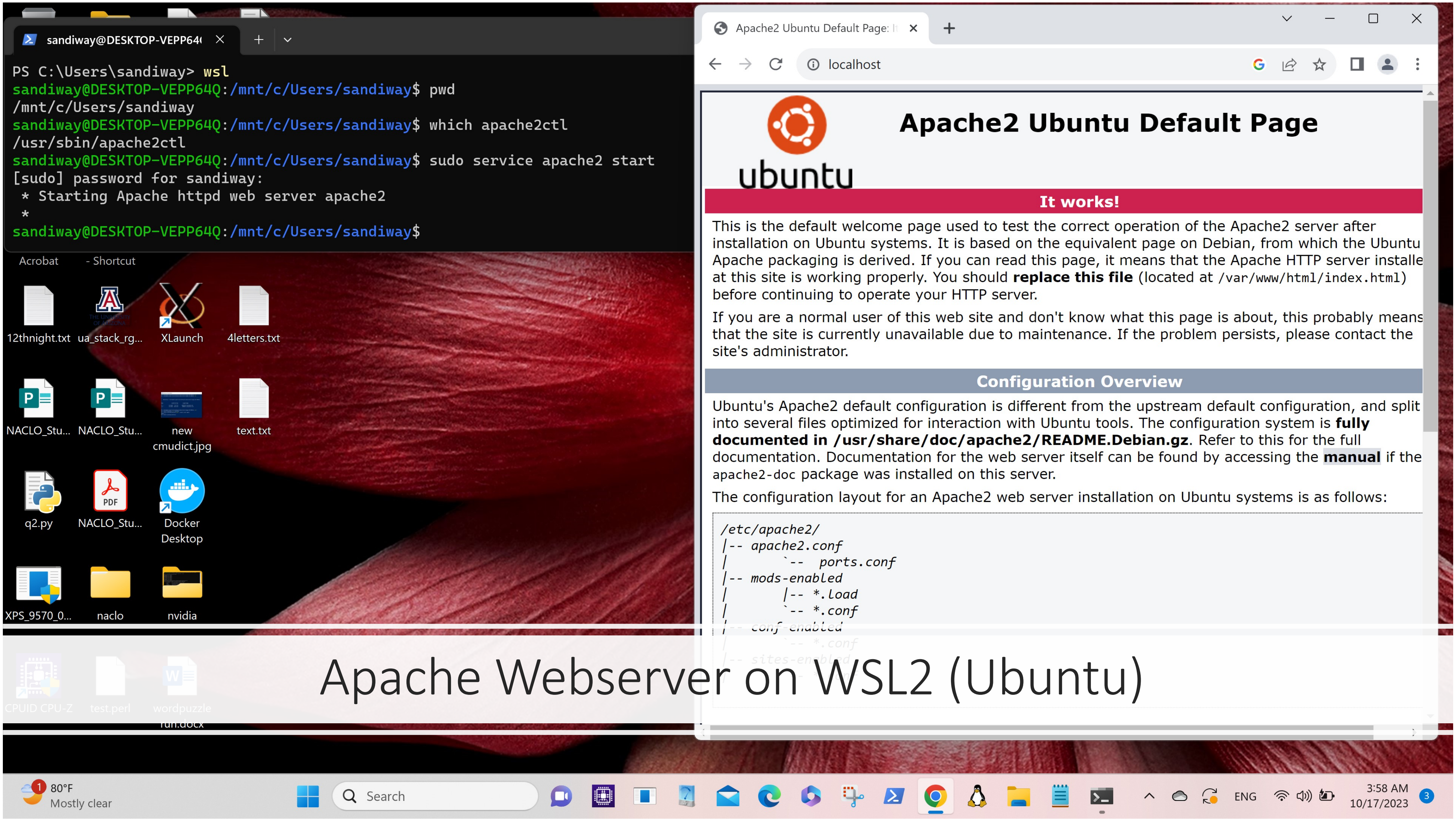
nano is a simple text editor

^ means use the Control key

save file as
`/Library/Webserver/Documents/index.html`

Apache Webserver on WSL2 (Ubuntu)

- From Windows Powershell:
  - `wsl`
- Default page (Document Root):
  - `/var/www/html/index.html`
  - owner of file is `root`; you need to use `sudo` prefix, e.g. `sudo  nano`, to edit it
- Commands:
  - `sudo service apache2 start`
  - `sudo service apache2 stop`
  - `sudo service apache2 restart`



```
PS C:\Users\sandiway> wsl
sandiway@DESKTOP-VEPP64Q:/mnt/c/Users/sandiway$ pwd
/mnt/c/Users/sandiway
sandiway@DESKTOP-VEPP64Q:/mnt/c/Users/sandiway$ which apache2ctl
/usr/sbin/apache2ctl
sandiway@DESKTOP-VEPP64Q:/mnt/c/Users/sandiway$ sudo service apache2 start
[sudo] password for sandiway:
 * Starting Apache httpd web server apache2
 *
sandiway@DESKTOP-VEPP64Q:/mnt/c/Users/sandiway$
```

Apache2 Ubuntu Default Page

It works!

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 installe 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.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

Configuration Overview

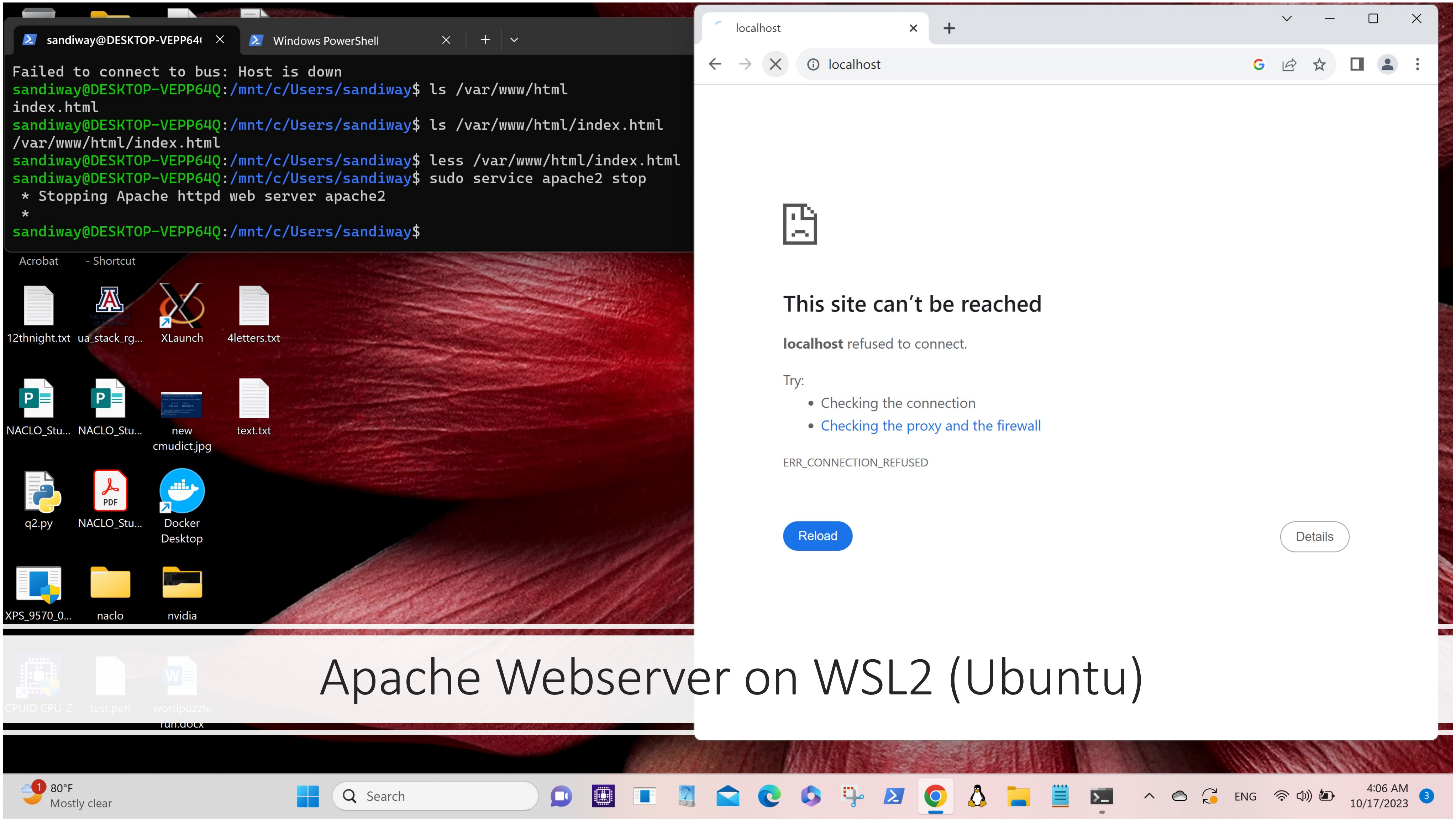
Ubuntu's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Ubuntu tools. The configuration system is **fully documented in /usr/share/doc/apache2/README.Debian.gz**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the `apache2-doc` package was installed on this server.

The configuration layout for an Apache2 web server installation on Ubuntu systems is as follows:

```
/etc/apache2/
|-- apache2.conf
|       `--  ports.conf
|-- mods-enabled
|       |-- *.load
|       `-- *.conf
|-- conf-enabled
|       `-- *.conf
|-- sites-enabled
```

Apache Webserver on WSL2 (Ubuntu)

Apache Webserver on WSL2 (Ubuntu)



```
Failed to connect to bus: Host is down
sandiway@DESKTOP-VEPP64Q:/mnt/c/Users/sandiway$ ls /var/www/html
index.html
sandiway@DESKTOP-VEPP64Q:/mnt/c/Users/sandiway$ ls /var/www/html/index.html
/var/www/html/index.html
sandiway@DESKTOP-VEPP64Q:/mnt/c/Users/sandiway$ less /var/www/html/index.html
sandiway@DESKTOP-VEPP64Q:/mnt/c/Users/sandiway$ sudo service apache2 stop
 * Stopping Apache httpd web server apache2
 *
sandiway@DESKTOP-VEPP64Q:/mnt/c/Users/sandiway$
```

This site can’t be reached

localhost refused to connect.

Try:

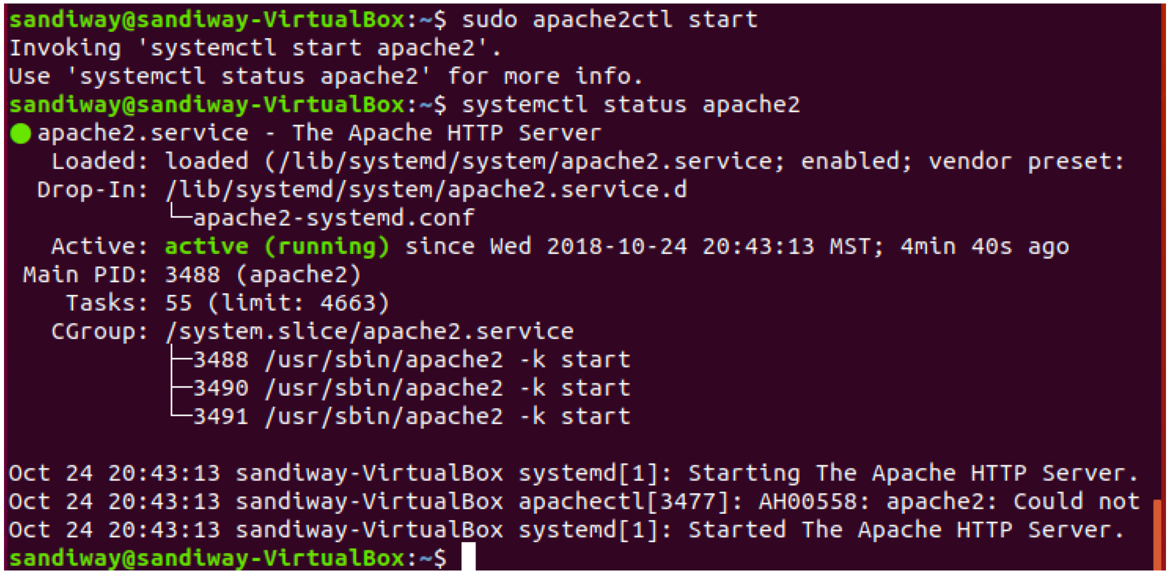
- Checking the connection
- Checking the proxy and the firewall

ERR_CONNECTION_REFUSED

Apache2 on Ubuntu

- Apache2 webserver (on VirtualBox or WSL2):
  - `sudo apache2ctl start`
  - `sudo apache2ctl stop`
  - `sudo apache2ctl restart`

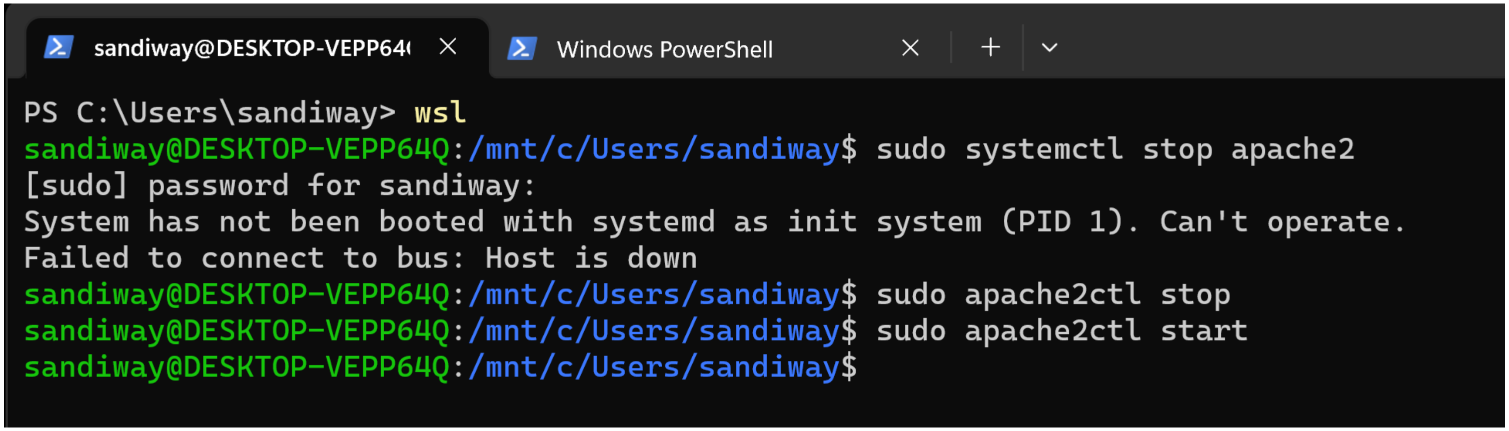
Note: macOS is different, uses `apachectl`



```
sandiway@sandiway-VirtualBox:~$ sudo apache2ctl start
Invoking 'systemctl start apache2'.
Use 'systemctl status apache2' for more info.
sandiway@sandiway-VirtualBox:~$ systemctl status apache2
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset:
  Drop-In: /lib/systemd/system/apache2.service.d
           └─apache2-systemd.conf
   Active: active (running) since Wed 2018-10-24 20:43:13 MST; 4min 40s ago
 Main PID: 3488 (apache2)
    Tasks: 55 (limit: 4663)
   CGroup: /system.slice/apache2.service
           ├─3488 /usr/sbin/apache2 -k start
           ├─3490 /usr/sbin/apache2 -k start
           └─3491 /usr/sbin/apache2 -k start

Oct 24 20:43:13 sandiway-VirtualBox systemd[1]: Starting The Apache HTTP Server.
Oct 24 20:43:13 sandiway-VirtualBox apachectl[3477]: AH00558: apache2: Could not
Oct 24 20:43:13 sandiway-VirtualBox systemd[1]: Started The Apache HTTP Server.
sandiway@sandiway-VirtualBox:~$ 
```

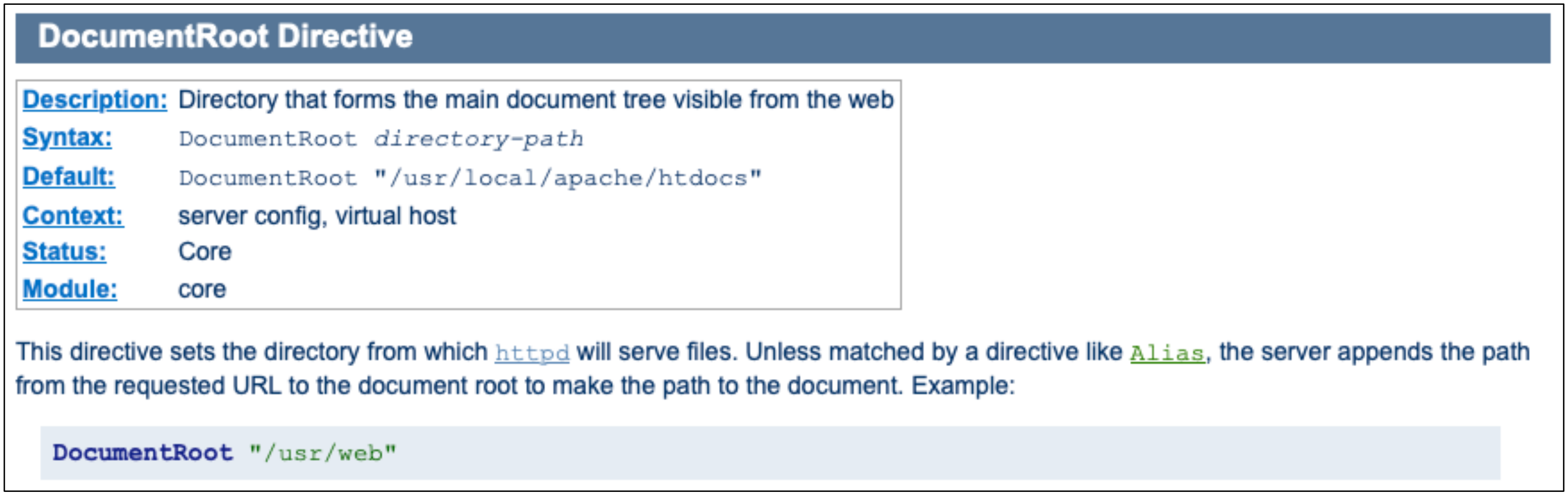
Apache2 on Ubuntu

- Apache webserver (works on VirtualBox, doesn't work in WSL2):
  - `sudo systemctl start apache2`
  - `sudo systemctl stop apache2`
  - `sudo systemctl restart apache2`



```
PS C:\Users\sandiway> wsl
sandiway@DESKTOP-VEPP64Q:/mnt/c/Users/sandiway$ sudo systemctl stop apache2
[sudo] password for sandiway:
System has not been booted with systemd as init system (PID 1). Can't operate.
Failed to connect to bus: Host is down
sandiway@DESKTOP-VEPP64Q:/mnt/c/Users/sandiway$ sudo apache2ctl stop
sandiway@DESKTOP-VEPP64Q:/mnt/c/Users/sandiway$ sudo apache2ctl start
sandiway@DESKTOP-VEPP64Q:/mnt/c/Users/sandiway$
```

DocumentRoot



DocumentRoot Directive

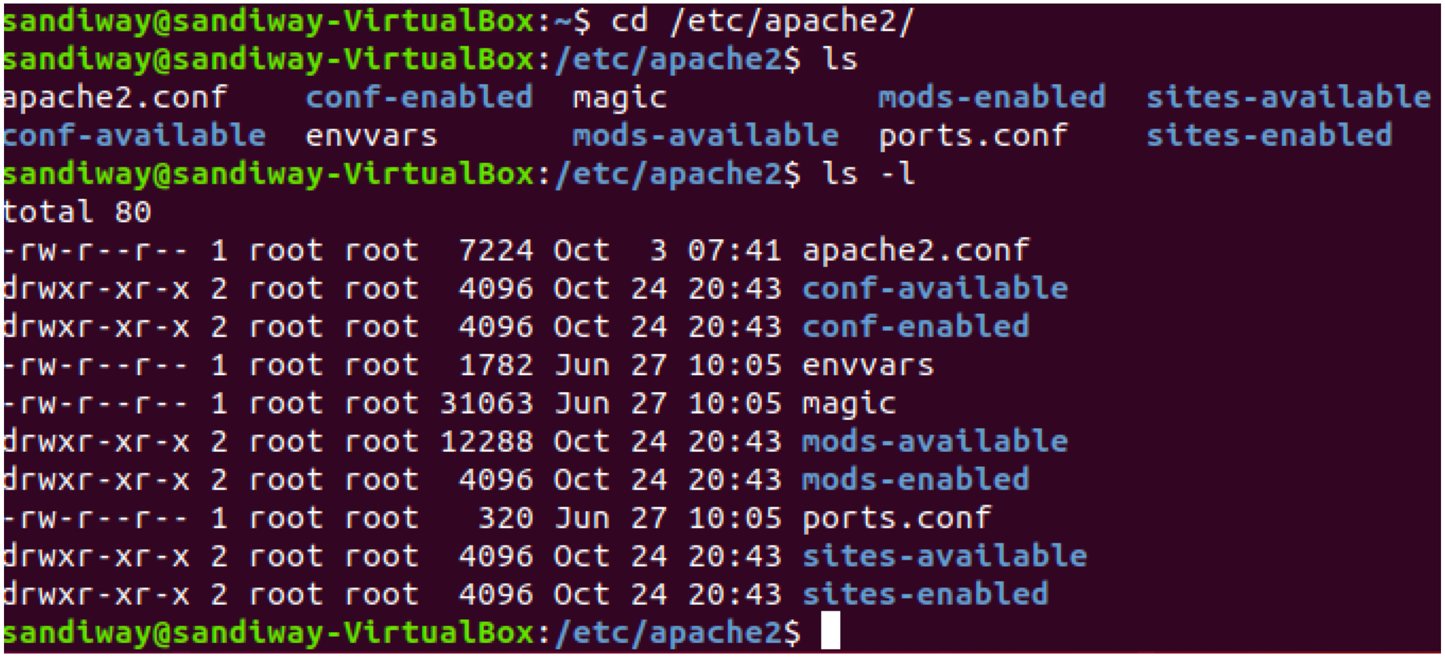
Description:	Directory that forms the main document tree visible from the web
Syntax:	`DocumentRoot` *`directory-path`*
Default:	`DocumentRoot "/usr/local/apache/htdocs"`
Context:	server config, virtual host
Status:	Core
Module:	core

This directive sets the directory from which `httpd` will serve files. Unless matched by a directive like `Alias`, the server appends the path from the requested URL to the document root to make the path to the document. Example:

```
DocumentRoot "/usr/web"
```

Apache2 on Ubuntu

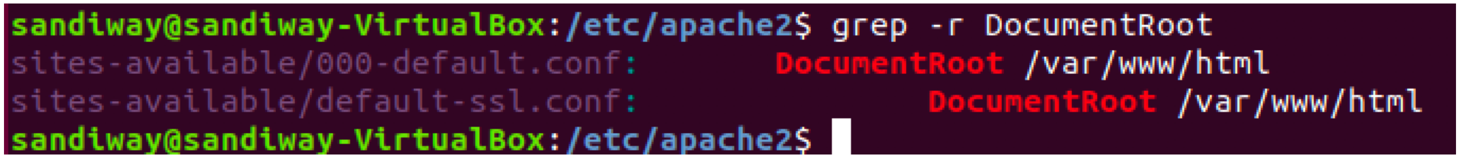
- Master configuration file:
  - **/etc/apache2/httpd.conf**



```
sandiway@sandiway-VirtualBox:~$ cd /etc/apache2/
sandiway@sandiway-VirtualBox:/etc/apache2$ ls
apache2.conf    conf-enabled  magic           mods-enabled  sites-available
conf-available  envvars       mods-available  ports.conf    sites-enabled
sandiway@sandiway-VirtualBox:/etc/apache2$ ls -l
total 80
-rw-r--r-- 1 root root  7224 Oct  3 07:41 apache2.conf
drwxr-xr-x 2 root root  4096 Oct 24 20:43 conf-available
drwxr-xr-x 2 root root  4096 Oct 24 20:43 conf-enabled
-rw-r--r-- 1 root root  1782 Jun 27 10:05 envvars
-rw-r--r-- 1 root root 31063 Jun 27 10:05 magic
drwxr-xr-x 2 root root 12288 Oct 24 20:43 mods-available
drwxr-xr-x 2 root root  4096 Oct 24 20:43 mods-enabled
-rw-r--r-- 1 root root   320 Jun 27 10:05 ports.conf
drwxr-xr-x 2 root root  4096 Oct 24 20:43 sites-available
drwxr-xr-x 2 root root  4096 Oct 24 20:43 sites-enabled
sandiway@sandiway-VirtualBox:/etc/apache2$ 
```

Apache2 on Ubuntu

- `cd /etc/apache2/`
- `grep –r DocumentRoot` (`grep –r` searches into subdirectories for string)



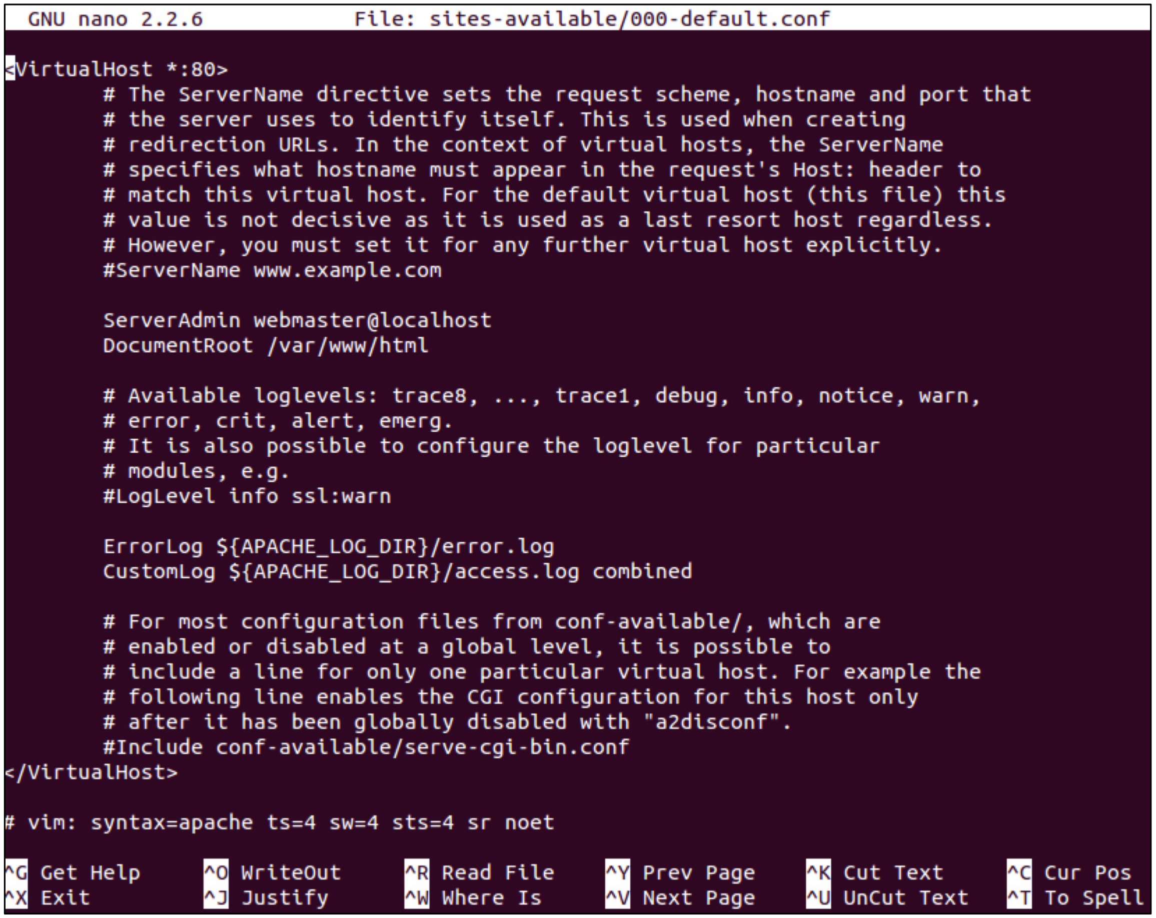
```
sandiway@sandiway-VirtualBox:/etc/apache2$ grep -r DocumentRoot
sites-available/000-default.conf:       DocumentRoot /var/www/html
sites-available/default-ssl.conf:               DocumentRoot /var/www/html
sandiway@sandiway-VirtualBox:/etc/apache2$
```

- For httpd, relevant file is:

`/etc/apache2/sites–enabled/000–default.conf`

Apache2 on Ubuntu

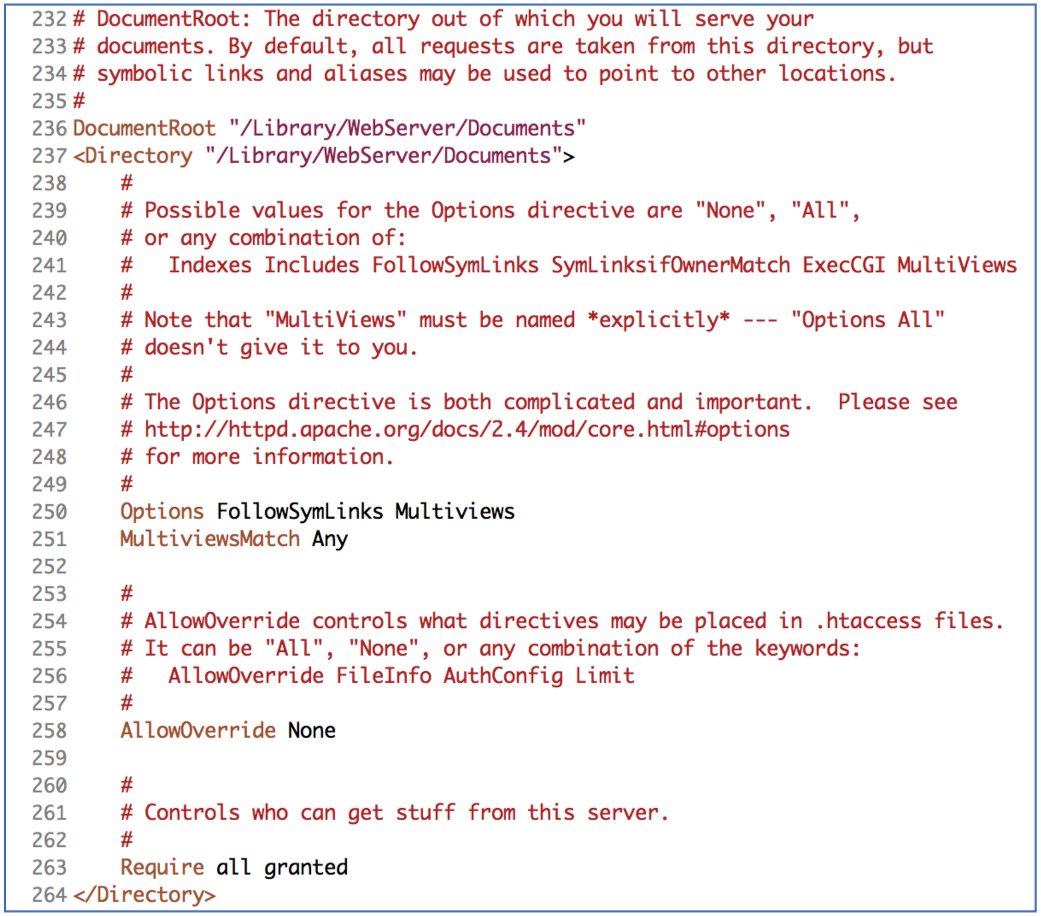
```
nano /etc/apache2/sites-enabled/000-default.conf
```



```
<VirtualHost *:80>
        # The ServerName directive sets the request scheme, hostname and port that
        # the server uses to identify itself. This is used when creating
        # redirection URLs. In the context of virtual hosts, the ServerName
        # specifies what hostname must appear in the request's Host: header to
        # match this virtual host. For the default virtual host (this file) this
        # value is not decisive as it is used as a last resort host regardless.
        # However, you must set it for any further virtual host explicitly.
        #ServerName www.example.com

        ServerAdmin webmaster@localhost
        DocumentRoot /var/www/html

        # Available loglevels: trace8, ..., trace1, debug, info, notice, warn,
        # error, crit, alert, emerg.
        # It is also possible to configure the loglevel for particular
        # modules, e.g.
        #LogLevel info ssl:warn

        ErrorLog ${APACHE_LOG_DIR}/error.log
        CustomLog ${APACHE_LOG_DIR}/access.log combined

        # For most configuration files from conf-available/, which are
        # enabled or disabled at a global level, it is possible to
        # include a line for only one particular virtual host. For example the
        # following line enables the CGI configuration for this host only
        # after it has been globally disabled with "a2disconf".
        #Include conf-available/serve-cgi-bin.conf
</VirtualHost>

# vim: syntax=apache ts=4 sw=4 sts=4 sr noet
```

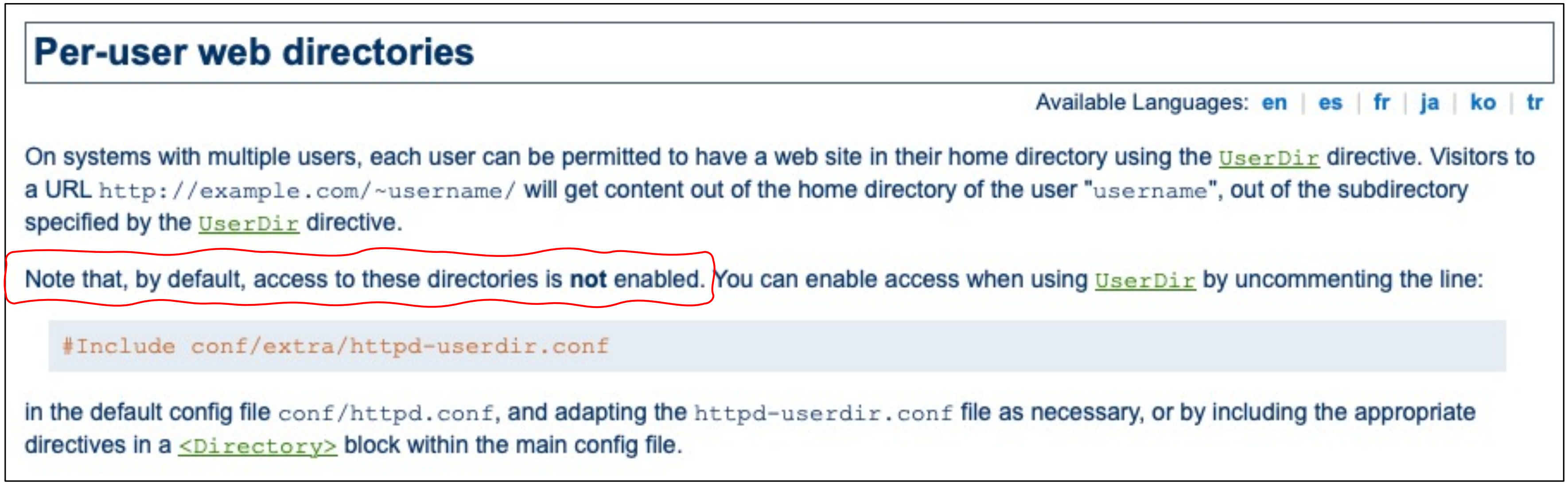
Apache Webserver on macOS

- Configuration file: `/etc/apache2/httpd.conf`
  - DocumentRoot



```
# DocumentRoot: The directory out of which you will serve your
# documents. By default, all requests are taken from this directory, but
# symbolic links and aliases may be used to point to other locations.
#
DocumentRoot "/Library/WebServer/Documents"
<Directory "/Library/WebServer/Documents">
    #
    # Possible values for the Options directive are "None", "All",
    # or any combination of:
    #   Indexes Includes FollowSymLinks SymLinksifOwnerMatch ExecCGI MultiViews
    #
    # Note that "MultiViews" must be named *explicitly* --- "Options All"
    # doesn't give it to you.
    #
    # The Options directive is both complicated and important.  Please see
    # http://httpd.apache.org/docs/2.4/mod/core.html#options
    # for more information.
    #
    Options FollowSymLinks Multiviews
    MultiviewsMatch Any

    #
    # AllowOverride controls what directives may be placed in .htaccess files.
    # It can be "All", "None", or any combination of the keywords:
    #   AllowOverride FileInfo AuthConfig Limit
    #
    AllowOverride None

    #
    # Controls who can get stuff from this server.
    #
    Require all granted
</Directory>
```

User web files



Per-user web directories

Available Languages: en | es | fr | ja | ko | tr

On systems with multiple users, each user can be permitted to have a web site in their home directory using the `UserDir` directive. Visitors to a URL `http://example.com/~username/` will get content out of the home directory of the user "`username`", out of the subdirectory specified by the `UserDir` directive.

Note that, by default, access to these directories is **not** enabled. You can enable access when using `UserDir` by uncommenting the line:

```
#Include conf/extra/httpd-userdir.conf
```

in the default config file `conf/httpd.conf`, and adapting the `httpd-userdir.conf` file as necessary, or by including the appropriate directives in a `<Directory>` block within the main config file.

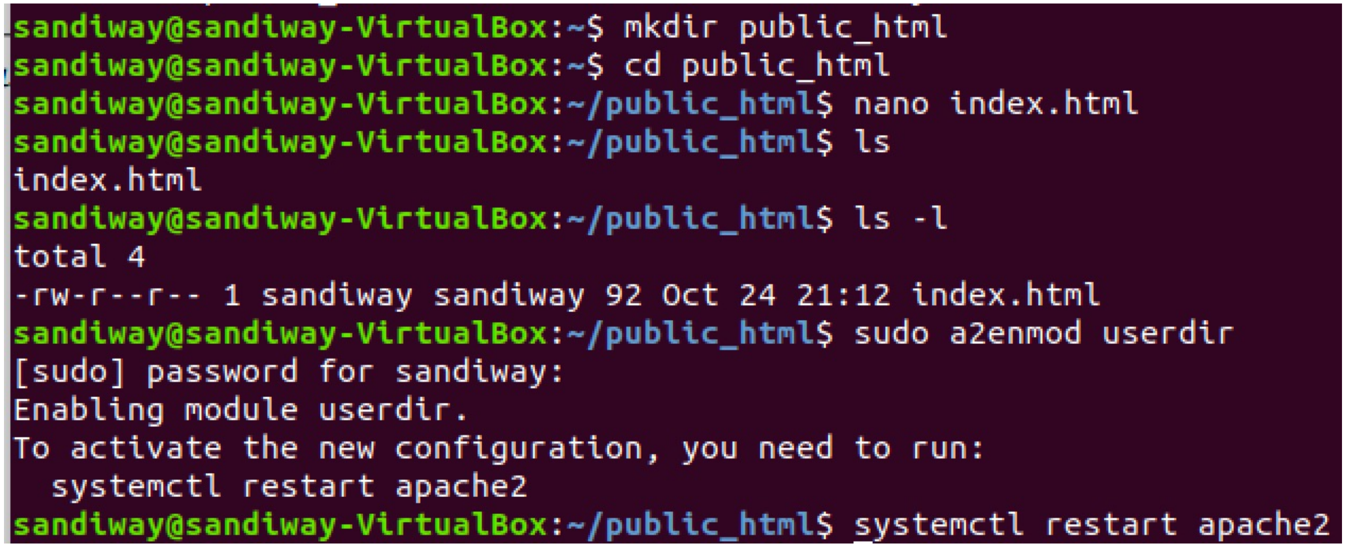
Apache2 on Ubuntu vs macOS

- Logs are in directory:
  `/var/log/apache2/`
  - `access.log`
  - `error.log`
- User web files in:
  - `~/public_html`
- In your home directory do:
  - `mkdir public_html`
  - `nano public_html/index.html`

- Logs are in directory:
  `/var/log/apache2/`
  - `access_log`
  - `error_log`
- User web files in:
  - `~/Sites`
- In your home directory do:
  - `mkdir Sites`
  - `nano Sites/index.html`

Apache2 on Ubuntu

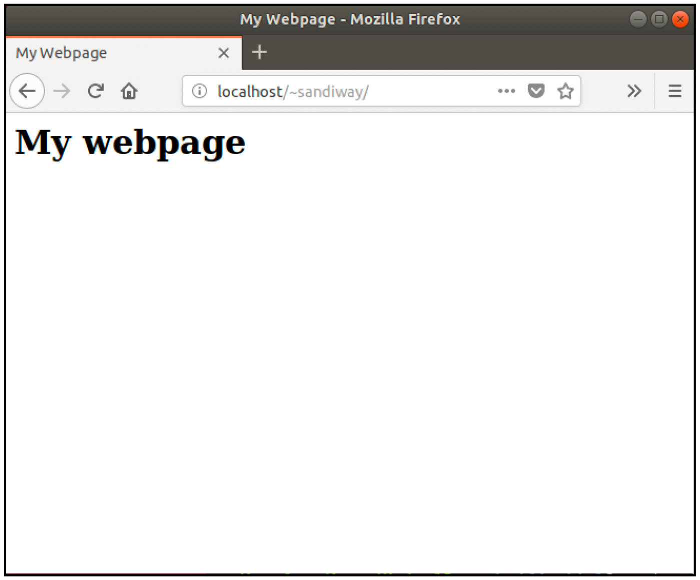
- To enable user web files in **~/public_html**
  - **sudo a2enmod userdir**
  - **sudo systemctl restart apache2**
  - http://localhost/~sandiway/



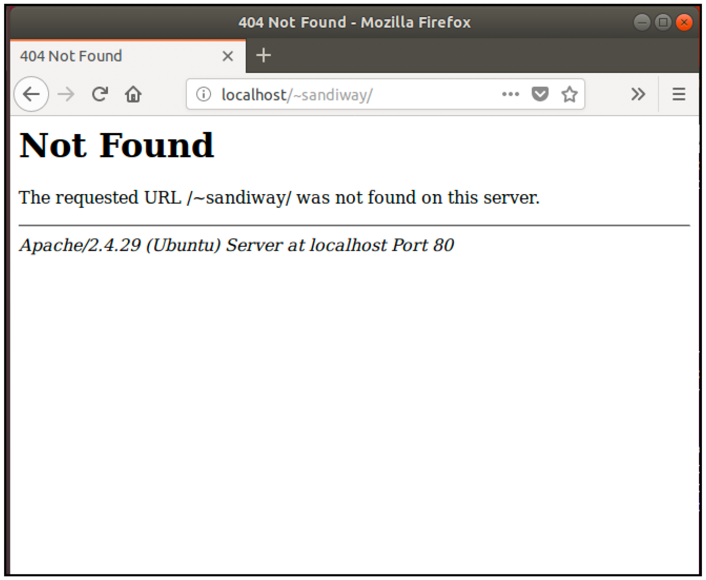
```
sandiway@sandiway-VirtualBox:~$ mkdir public_html
sandiway@sandiway-VirtualBox:~$ cd public_html
sandiway@sandiway-VirtualBox:~/public_html$ nano index.html
sandiway@sandiway-VirtualBox:~/public_html$ ls
index.html
sandiway@sandiway-VirtualBox:~/public_html$ ls -l
total 4
-rw-r--r-- 1 sandiway sandiway 92 Oct 24 21:12 index.html
sandiway@sandiway-VirtualBox:~/public_html$ sudo a2enmod userdir
[sudo] password for sandiway:
Enabling module userdir.
To activate the new configuration, you need to run:
  systemctl restart apache2
sandiway@sandiway-VirtualBox:~/public_html$ systemctl restart apache2
```

Apache2 on Ubuntu

- To enable user web files in **~/public_html**
  - `sudo a2enmod userdir` **(a2dismod)**
  - `sudo service apache2 restart`
  - http://localhost/~sandiway/



My webpage



Not Found

The requested URL /~sandiway/ was not found on this server.

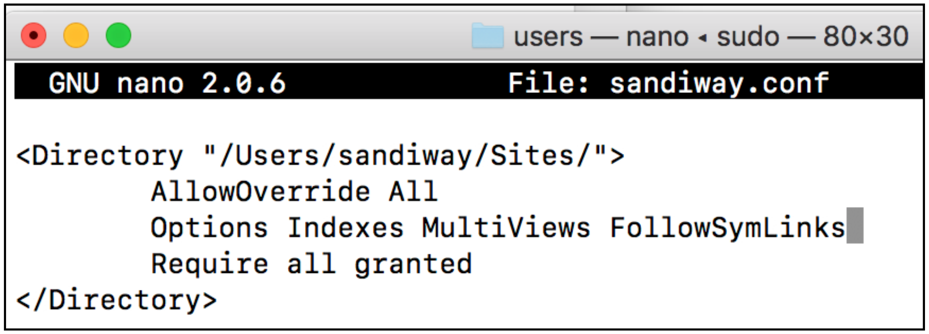
Apache/2.4.29 (Ubuntu) Server at localhost Port 80

Apache Webserver on macOS

Static webpages

- storage locations:
  - http://localhost/~sandiway/ **(no need to be superuser)**
  - **`mkdir ~sandiway/Sites`** (**`/Users/`*`username`*`/Sites`**)
  - **`~/Sites/index.html`** (create this file!)
  - **`sudo nano /etc/apache2/users/sandiway.conf`**

create this file …



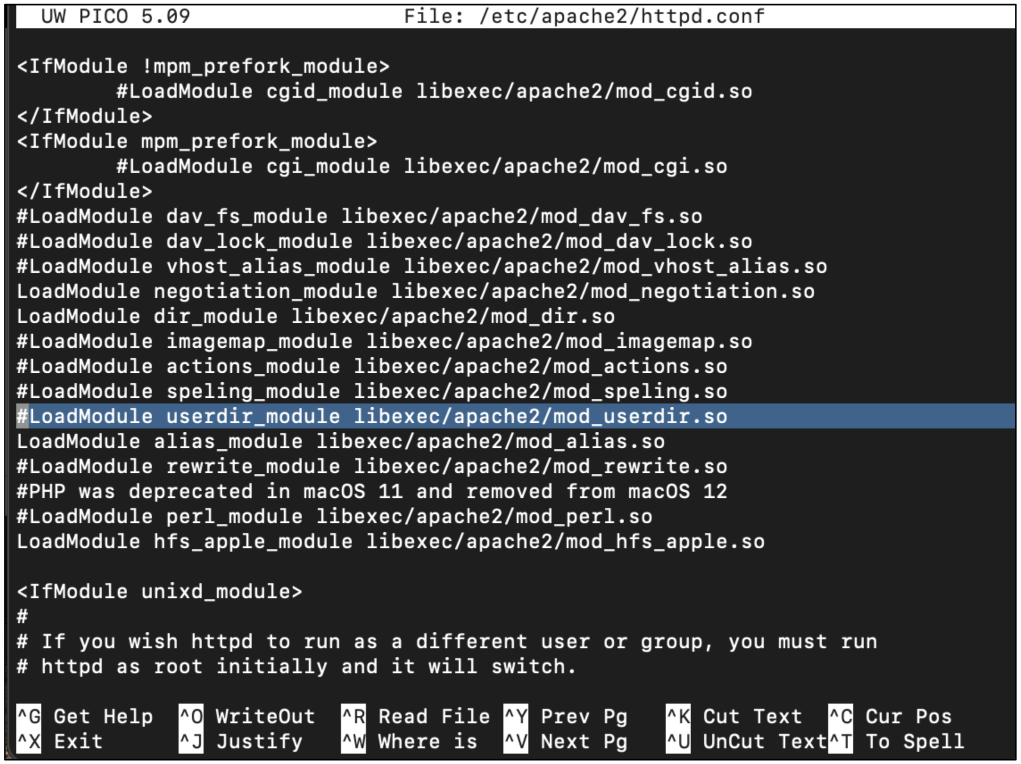
```
  GNU nano 2.0.6                File: sandiway.conf

<Directory "/Users/sandiway/Sites/">
        AllowOverride All
        Options Indexes MultiViews FollowSymLinks
        Require all granted
</Directory>
```

Apache Webserver on macOS

Static webpages

  - http://localhost/~sandiway/

- storage locations:
  - `sudo nano /etc/apache2/httpd.conf`
- uncomment `mod_userdir.so` line
  - (*remove the comment char #*)

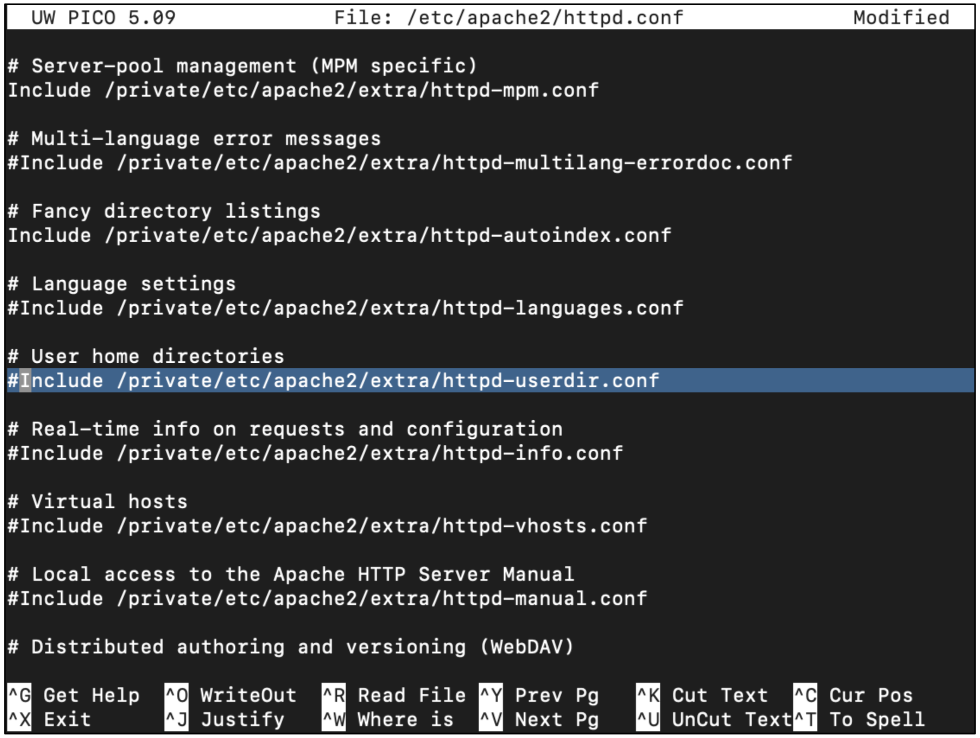


```
  UW PICO 5.09                         File: /etc/apache2/httpd.conf

<IfModule !mpm_prefork_module>
        #LoadModule cgid_module libexec/apache2/mod_cgid.so
</IfModule>
<IfModule mpm_prefork_module>
        #LoadModule cgi_module libexec/apache2/mod_cgi.so
</IfModule>
#LoadModule dav_fs_module libexec/apache2/mod_dav_fs.so
#LoadModule dav_lock_module libexec/apache2/mod_dav_lock.so
#LoadModule vhost_alias_module libexec/apache2/mod_vhost_alias.so
LoadModule negotiation_module libexec/apache2/mod_negotiation.so
LoadModule dir_module libexec/apache2/mod_dir.so
#LoadModule imagemap_module libexec/apache2/mod_imagemap.so
#LoadModule actions_module libexec/apache2/mod_actions.so
#LoadModule speling_module libexec/apache2/mod_speling.so
#LoadModule userdir_module libexec/apache2/mod_userdir.so
LoadModule alias_module libexec/apache2/mod_alias.so
#LoadModule rewrite_module libexec/apache2/mod_rewrite.so
#PHP was deprecated in macOS 11 and removed from macOS 12
#LoadModule perl_module libexec/apache2/mod_perl.so
LoadModule hfs_apple_module libexec/apache2/mod_hfs_apple.so

<IfModule unixd_module>
#
# If you wish httpd to run as a different user or group, you must run
# httpd as root initially and it will switch.

^G Get Help  ^O WriteOut  ^R Read File ^Y Prev Pg   ^K Cut Text  ^C Cur Pos
^X Exit      ^J Justify   ^W Where is  ^V Next Pg   ^U UnCut Text^T To Spell
```

Apache Webserver on macOS

Static webpages

- also need to uncomment another line in `/etc/apache2/httpd.conf`
- uncomment the `httpd_userdir.conf` line
- (*remove the #*)

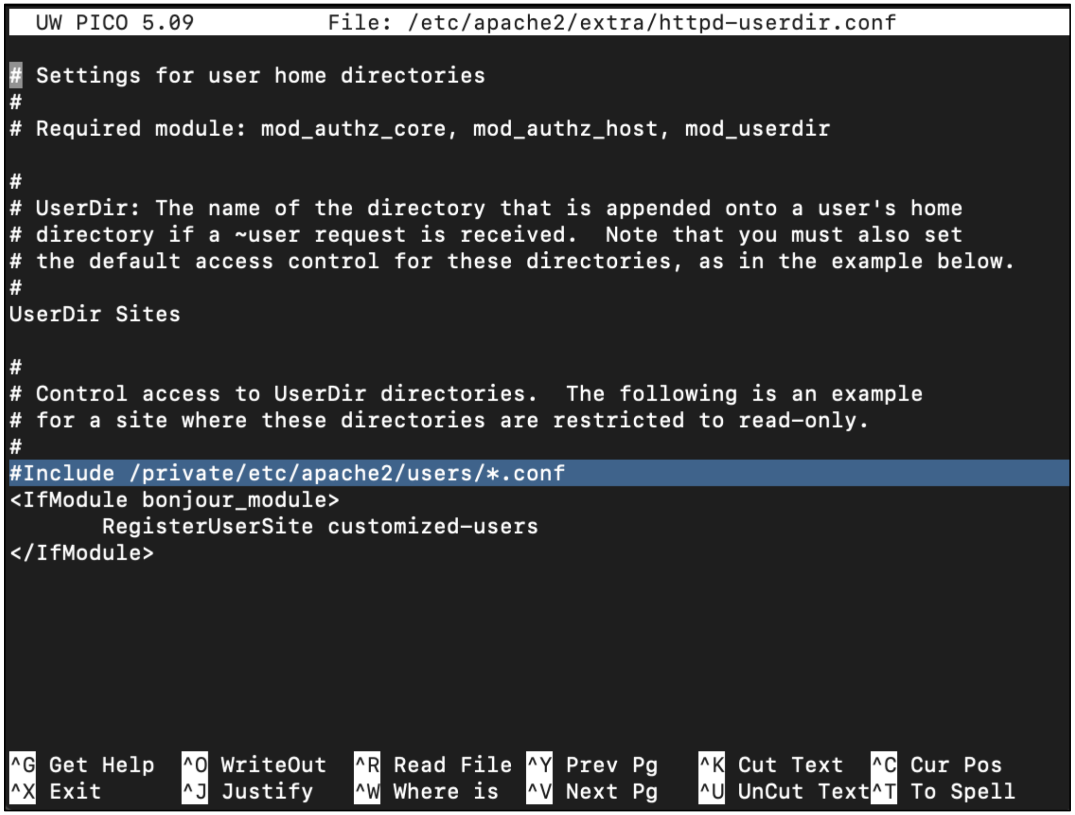


```
  UW PICO 5.09              File: /etc/apache2/httpd.conf              Modified

# Server-pool management (MPM specific)
Include /private/etc/apache2/extra/httpd-mpm.conf

# Multi-language error messages
#Include /private/etc/apache2/extra/httpd-multilang-errordoc.conf

# Fancy directory listings
Include /private/etc/apache2/extra/httpd-autoindex.conf

# Language settings
#Include /private/etc/apache2/extra/httpd-languages.conf

# User home directories
#Include /private/etc/apache2/extra/httpd-userdir.conf

# Real-time info on requests and configuration
#Include /private/etc/apache2/extra/httpd-info.conf

# Virtual hosts
#Include /private/etc/apache2/extra/httpd-vhosts.conf

# Local access to the Apache HTTP Server Manual
#Include /private/etc/apache2/extra/httpd-manual.conf

# Distributed authoring and versioning (WebDAV)

^G Get Help  ^O WriteOut  ^R Read File ^Y Prev Pg   ^K Cut Text  ^C Cur Pos
^X Exit      ^J Justify   ^W Where is  ^V Next Pg   ^U UnCut Text^T To Spell
```

Apache Webserver on macOS

Static webpages

- `sudo nano /etc/apache2/extra/httpd-userdir.conf`
- uncomment this include
- (*remove the #*)

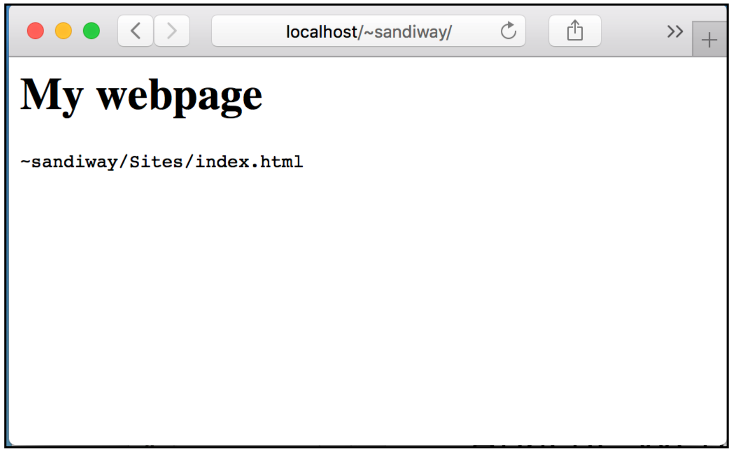


```
  UW PICO 5.09            File: /etc/apache2/extra/httpd-userdir.conf

# Settings for user home directories
#
# Required module: mod_authz_core, mod_authz_host, mod_userdir

#
# UserDir: The name of the directory that is appended onto a user's home
# directory if a ~user request is received.  Note that you must also set
# the default access control for these directories, as in the example below.
#
UserDir Sites

#
# Control access to UserDir directories.  The following is an example
# for a site where these directories are restricted to read-only.
#
#Include /private/etc/apache2/users/*.conf
<IfModule bonjour_module>
       RegisterUserSite customized-users
</IfModule>
```

Apache Webserver on macOS

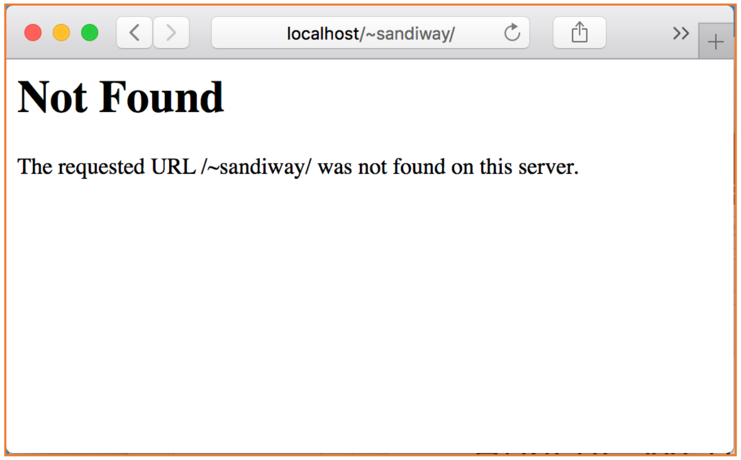
Static webpages

- storage locations:
  - `sudo apachectl –k restart`
  - create a file `~sandiway/Sites/index.html`



My webpage

`~sandiway/Sites/index.html`

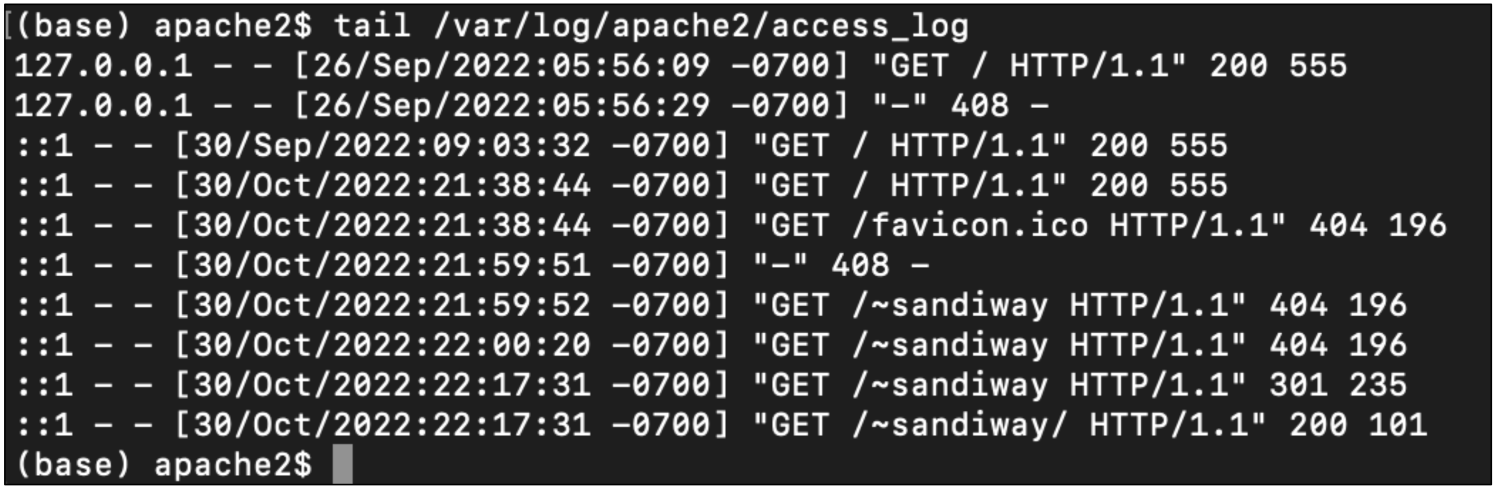


Not Found

The requested URL /~sandiway/ was not found on this server.

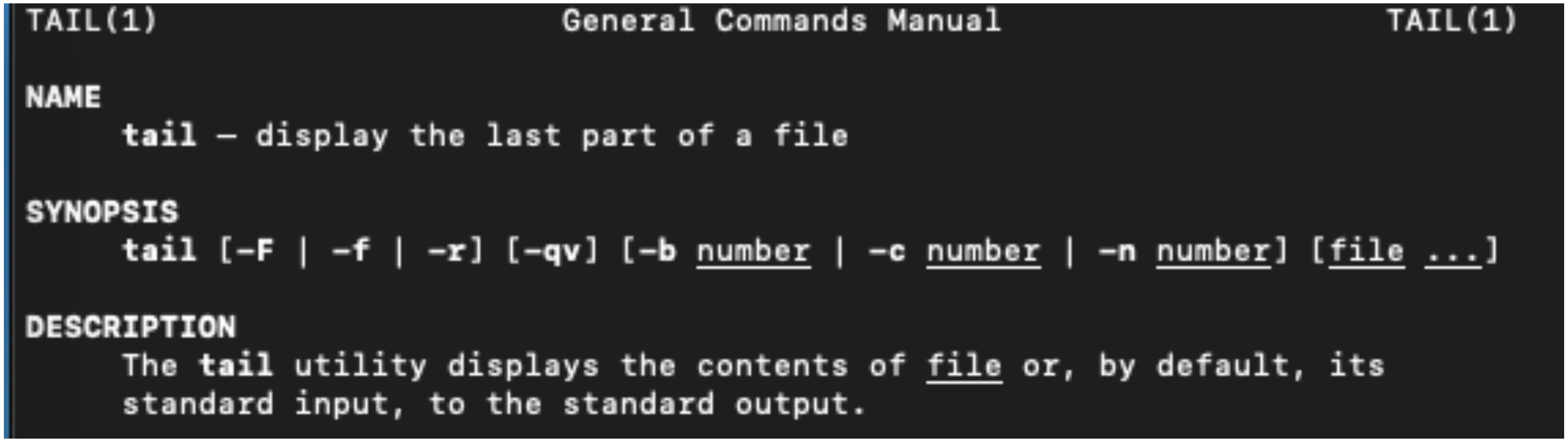
Apache Webserver on macOS

- `tail /var/log/apache2/access_log`



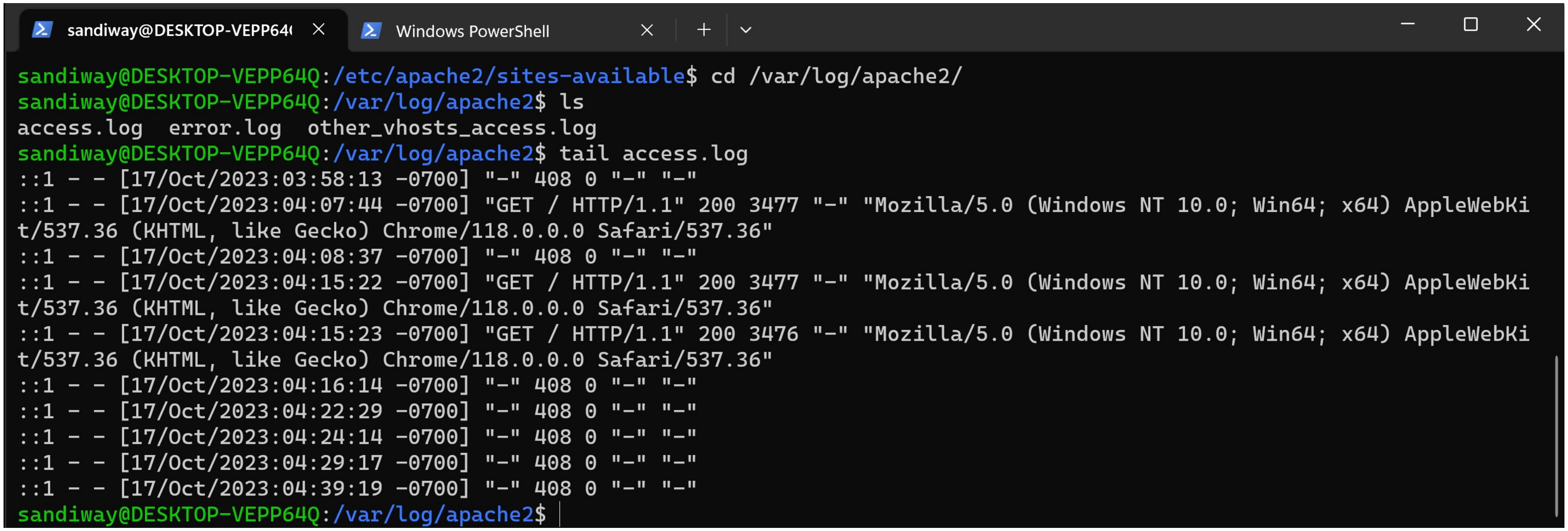
```
[(base) apache2$ tail /var/log/apache2/access_log
127.0.0.1 - - [26/Sep/2022:05:56:09 -0700] "GET / HTTP/1.1" 200 555
127.0.0.1 - - [26/Sep/2022:05:56:29 -0700] "-" 408 -
::1 - - [30/Sep/2022:09:03:32 -0700] "GET / HTTP/1.1" 200 555
::1 - - [30/Oct/2022:21:38:44 -0700] "GET / HTTP/1.1" 200 555
::1 - - [30/Oct/2022:21:38:44 -0700] "GET /favicon.ico HTTP/1.1" 404 196
::1 - - [30/Oct/2022:21:59:51 -0700] "-" 408 -
::1 - - [30/Oct/2022:21:59:52 -0700] "GET /~sandiway HTTP/1.1" 404 196
::1 - - [30/Oct/2022:22:00:20 -0700] "GET /~sandiway HTTP/1.1" 404 196
::1 - - [30/Oct/2022:22:17:31 -0700] "GET /~sandiway HTTP/1.1" 301 235
::1 - - [30/Oct/2022:22:17:31 -0700] "GET /~sandiway/ HTTP/1.1" 200 101
(base) apache2$ 
```

tail command



```
TAIL(1)                   General Commands Manual                  TAIL(1)

NAME
     tail – display the last part of a file

SYNOPSIS
     tail [-F | -f | -r] [-qv] [-b number | -c number | -n number] [file ...]

DESCRIPTION
     The tail utility displays the contents of file or, by default, its
     standard input, to the standard output.
```

Apache Webserver on Ubuntu



```
sandiway@DESKTOP-VEPP64Q:/etc/apache2/sites-available$ cd /var/log/apache2/
sandiway@DESKTOP-VEPP64Q:/var/log/apache2$ ls
access.log  error.log  other_vhosts_access.log
sandiway@DESKTOP-VEPP64Q:/var/log/apache2$ tail access.log
::1 - - [17/Oct/2023:03:58:13 -0700] "-" 408 0 "-" "-"
::1 - - [17/Oct/2023:04:07:44 -0700] "GET / HTTP/1.1" 200 3477 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKi
t/537.36 (KHTML, like Gecko) Chrome/118.0.0.0 Safari/537.36"
::1 - - [17/Oct/2023:04:08:37 -0700] "-" 408 0 "-" "-"
::1 - - [17/Oct/2023:04:15:22 -0700] "GET / HTTP/1.1" 200 3477 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKi
t/537.36 (KHTML, like Gecko) Chrome/118.0.0.0 Safari/537.36"
::1 - - [17/Oct/2023:04:15:23 -0700] "GET / HTTP/1.1" 200 3476 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKi
t/537.36 (KHTML, like Gecko) Chrome/118.0.0.0 Safari/537.36"
::1 - - [17/Oct/2023:04:16:14 -0700] "-" 408 0 "-" "-"
::1 - - [17/Oct/2023:04:22:29 -0700] "-" 408 0 "-" "-"
::1 - - [17/Oct/2023:04:24:14 -0700] "-" 408 0 "-" "-"
::1 - - [17/Oct/2023:04:29:17 -0700] "-" 408 0 "-" "-"
::1 - - [17/Oct/2023:04:39:19 -0700] "-" 408 0 "-" "-"
sandiway@DESKTOP-VEPP64Q:/var/log/apache2$
```