

Caddy

Ultimate Server with Automatic HTTPS



Why you need a web server

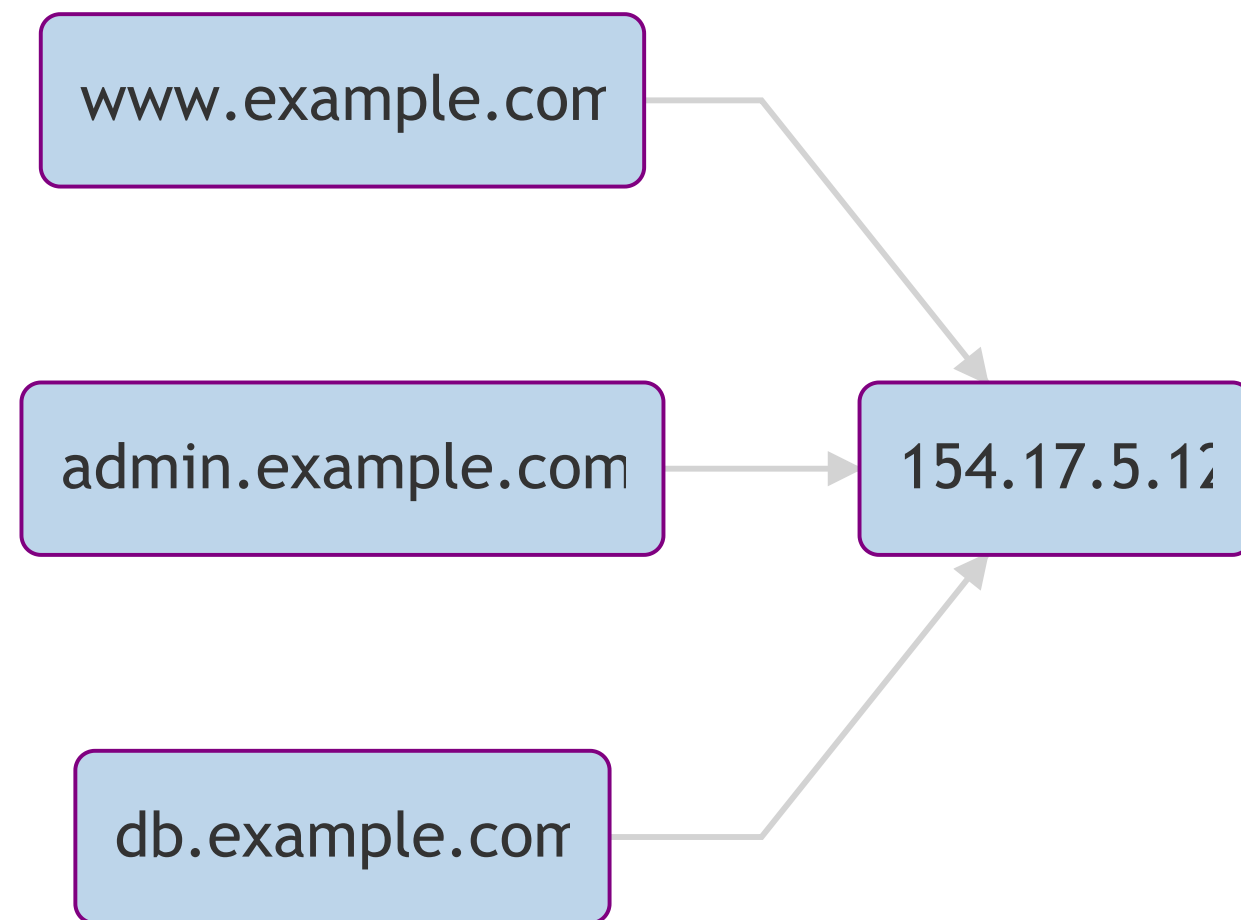
- Serving static files
 - Traditional HTML Pages
 - SPA Apps built with Vue / React
 - ``npm run serve`` is not suitable for production use!
- Reverse Proxy
 - Load-balance requests onto different instances
 - Split API / static content requests
 - HTTPS support
- Virtual Host
 - Multiple services, multiple domains
 - Same host machine

Examples

SERVING STATIC FILES WITH HTTPD ON UBUNTU

```
$ ls /var/www/html # default folder for static files
css img index.html js
```

VIRTUAL HOST



REVERSE PROXY

```
upstream api_backend { server api_backend.example.com:80;
upstream filebrowser { server filebrowser.example.com:80;
server {
    listen 80;
    location / {
        proxy_pass http://api_backend;
        proxy_set_header X-Forwarded-For $proxy_add_x_forw
        proxy_set_header Host $host;
        proxy_redirect off;
    }
    location /upload_media_files/ {
        proxy_pass http://filebrowser;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header Host $http_host;
        proxy_set_header X-Forwarded-For $proxy_add_x_forw
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection "upgrade";
    }
}
```

Common Web Servers



First 2 of them are widely used, but too hard for beginners.

EXAMPLE: USING NGINX

```
$ sudo apt install nginx
$ ls /etc/nginx/
conf.d                koi-utf              scgi_params.default
default.d             koi-win              sites-available
fastcgi.conf          mime.types            sites-enabled
fastcgi.conf.default  mime.types.default   uwsgi_params
fastcgi_params         nginx.conf            uwsgi_params.default
fastcgi_params.default scgi_params           win-utf
$ ls /etc/nginx/sites-available/ # after configuration
mapping.conf  overleaf-create-user.conf  root.conf  wp.conf
overleaf.conf recruit.conf             static.conf
```

HTTPS Support

- Crucial to privacy and safety
- In 10 years ago, you need to configure HTTPS certificates manually.
- Usually one has to pay ~\$10/yr for a certificate.

Free certificates with Let's Encrypt

- Once you've owned a domain, you can use this service to get free certificates
- However, rate limits apply, and you need to **renew** them on a regular basis
 - Otherwise, Certificate Errors! Example of Expired Cert
- Certbot+NGINX or Caddy both renew your certificates automatically
- Enable HTTPS with a few lines of code

```
$ sudo snap install --classic certbot
$ sudo ln -s /snap/bin/certbot /usr/bin/certbot
$ sudo certbot --nginx
```

Get your hands dirty!

CONFIGURE DNS RECORDS

`www.example.com` is resolved to `93.184.216.34`. Website root is `/var/www/html/`.

INSTALL CADDY ON YOUR SERVER

```
$ sudo apt install -y debian-keyring debian-archive-keyring apt-transport-https
$ curl -1sLf 'https://dl.cloudsmith.io/public/caddy/stable/gpg.key' | sudo apt-key add -
$ curl -1sLf 'https://dl.cloudsmith.io/public/caddy/stable/debian.deb.txt' | sudo tee /etc/apt/sources.list.d/caddy-stable.list
$ sudo apt update
$ sudo apt install caddy
```

CONFIGURATION

copy-paste the following lines into `/etc/caddy/Caddyfile`:

```
www.example.com
root /var/www/html/
file_server
```

RESTART CADDY

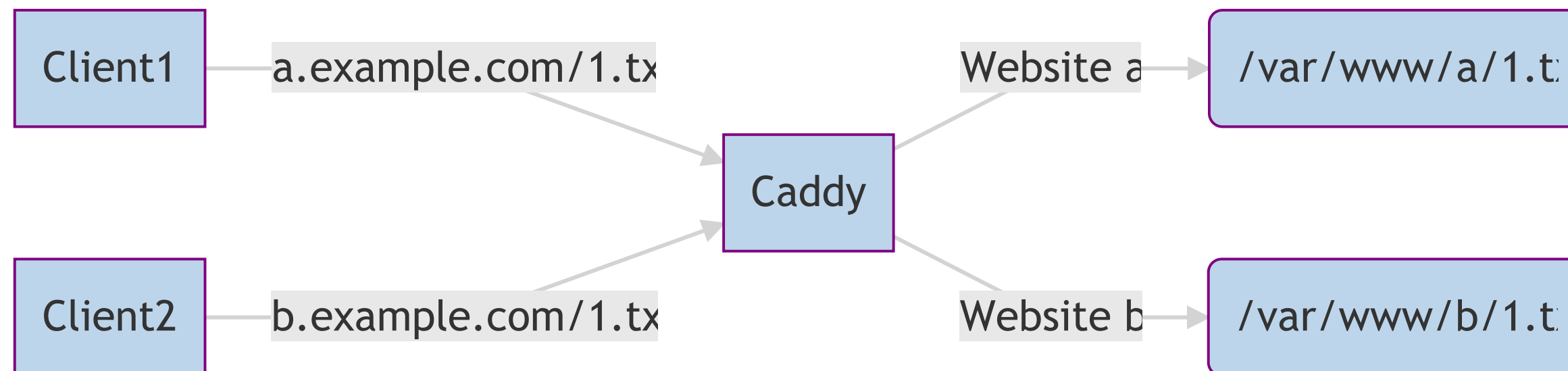
```
$ sudo systemctl restart caddy # Now you've got a fully-featured HTTPS server!
```

Virtual Hosts

- Multiple domains on a single server
- easy to deal with in `Caddyfile`

```
www.example.com {  
    # configurations specific to `www`  
}  
admin.example.com {  
    # configurations specific to `admin`  
}
```

Files under different VHosts are completely separate



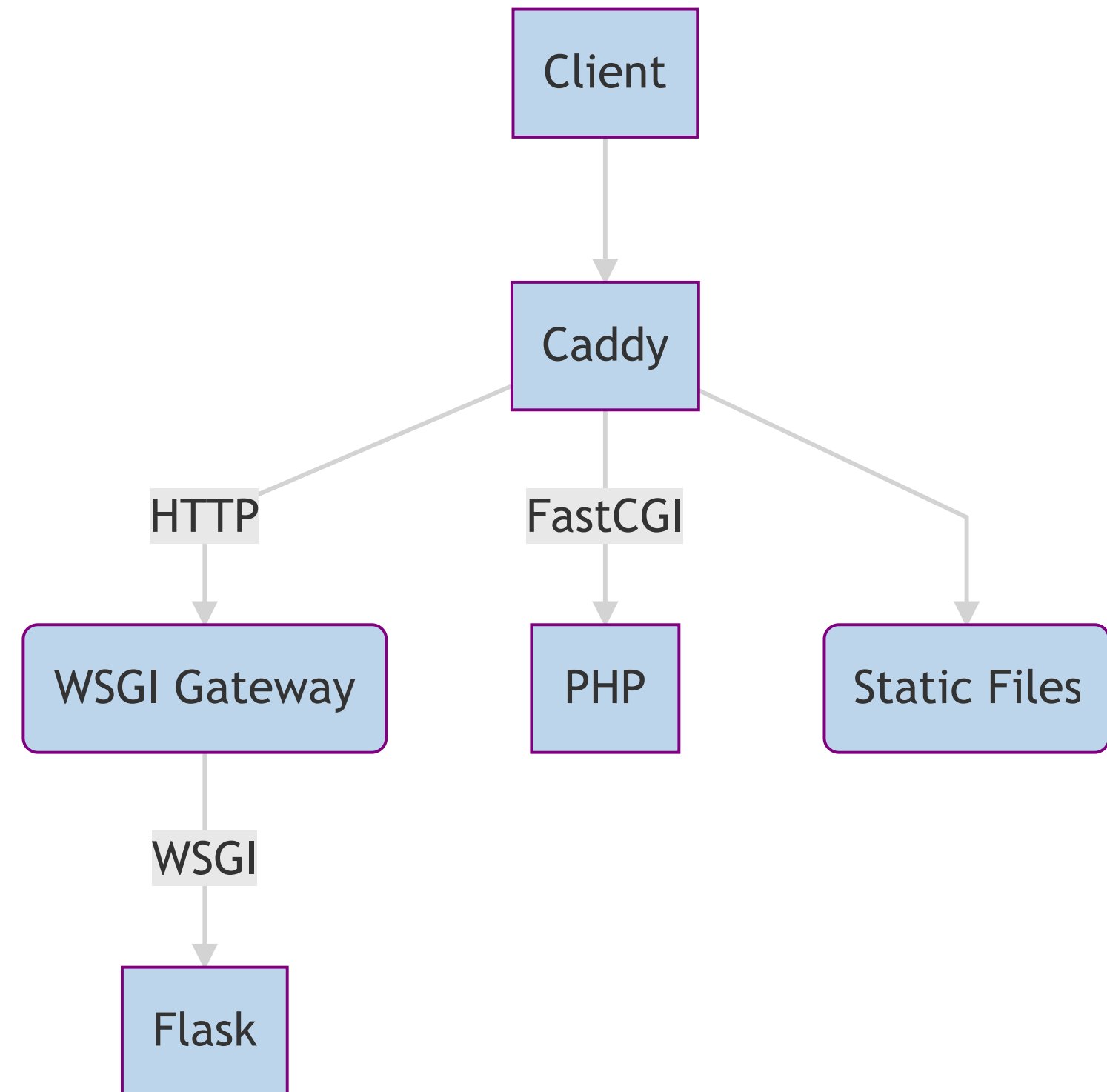
Reverse Proxy

- Typical use case: Python / PHP Servers
- Forward API requests to backend







TYPICAL CONFIG

Suppose that Gunicorn runs at `localhost:8001` & `localhost:8002`.

```
www.example.com {  
  reverse_proxy /api/* localhost:8001 localhost:8002  
  root /var/www/html/  
  file_server  
}
```



Caddy is powerful!

-  Forward Proxy
-  Logging
-  Template Rendering
-  Compression
-  Dynamic Config Update
-  Plugins

Resources

- [Official Caddyfile Tutorial](#) - Recommended
- [Let's Encrypt](#), which makes automatic HTTPS possible