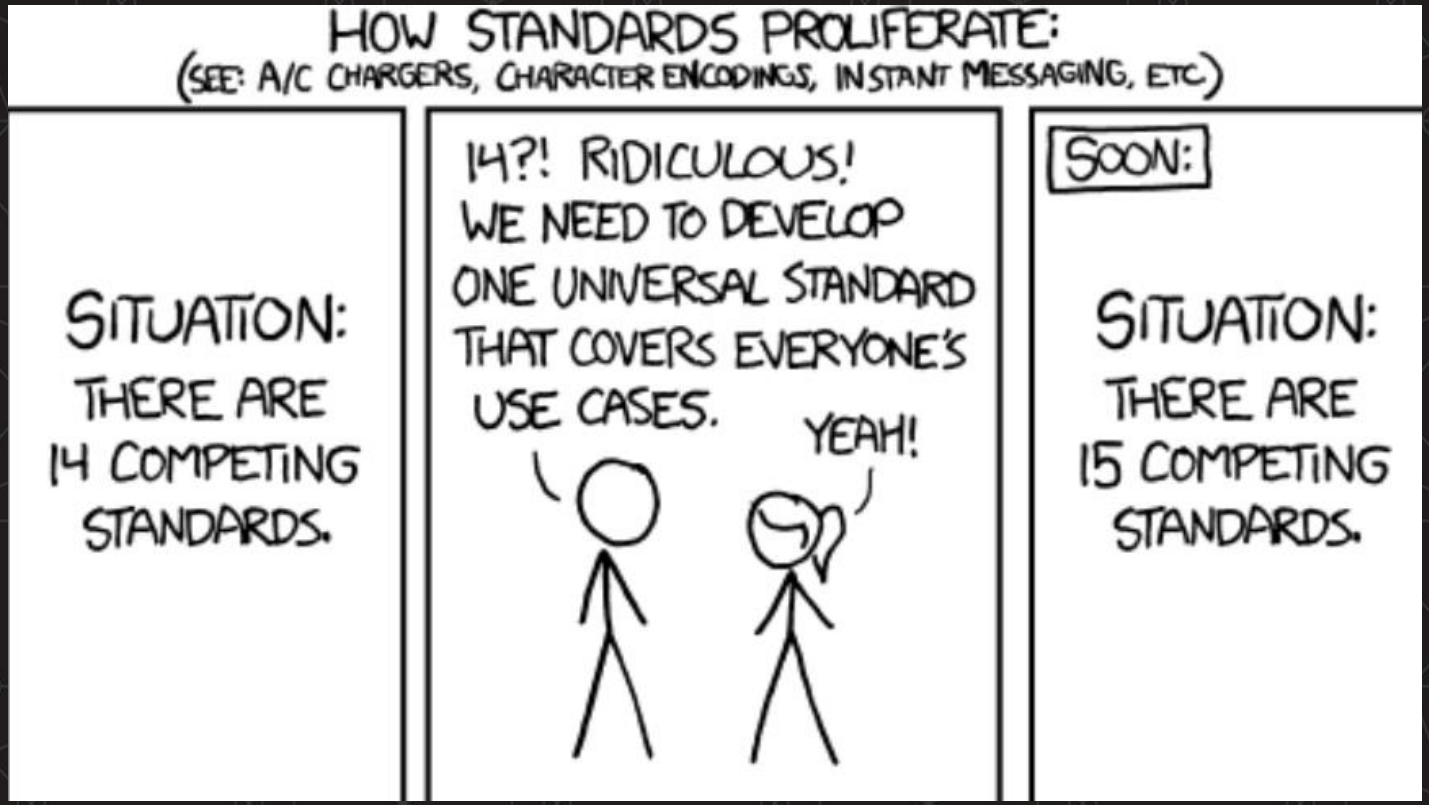
NGINX

AS A WEB SERVER

OS ENTERPRISE CONSULTING


HOW STANDARDS PROLIFERATE:
(SEE: A/C CHARGERS, CHARACTER ENCODINGS, INSTANT MESSAGING, ETC)
SITUATION:
THERE ARE
14 COMPETING
STANDARDS.
14?! RIDICULOUS!
WE NEED TO DEVELOP
ONE UNIVERSAL STANDARD
THAT COVERS EVERYONE'S
USE CASES.
YEAH!
SOON:
SITUATION:
THERE ARE
15 COMPETING
STANDARDS.

1

2

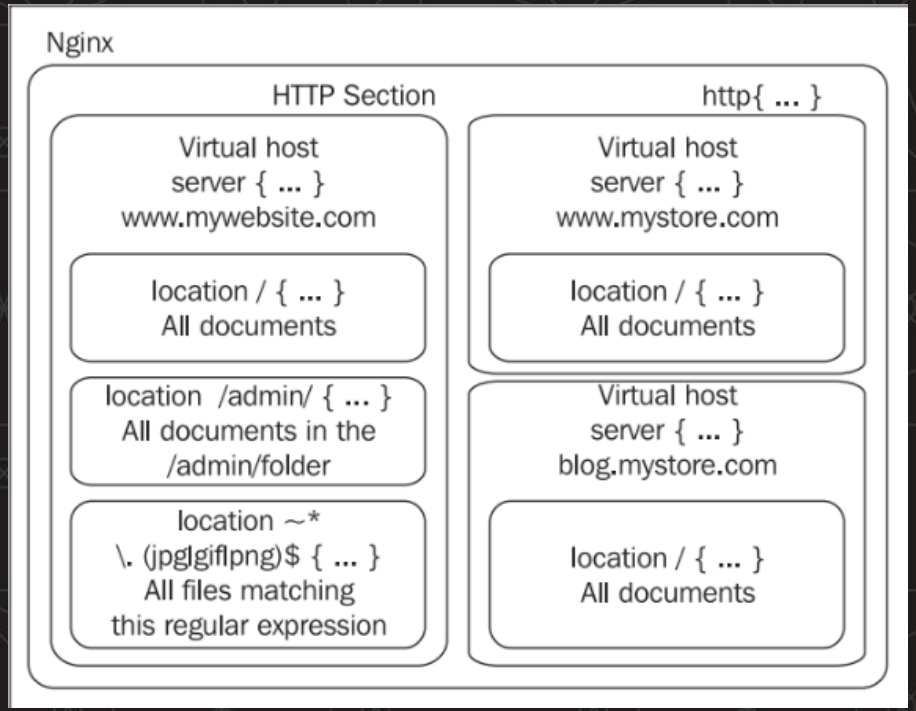
3

4

```
user  nginx;
worker_processes  1;

error_log  /var/log/nginx/error.log warn;
pid        /var/run/nginx.pid;


events {
    worker_connections  1024;
}


http {
    include       /etc/nginx/mime.types;
    default_type  application/octet-stream;

    log_format  main  '$remote_addr - $remote_user [$time_local] "$request" '
                      '$status $body_bytes_sent "$http_referer" '
                      '"$http_user_agent" "$http_x_forwarded_for"';

    access_log  /var/log/nginx/access.log  main;

    sendfile        on;
    #tcp_nopush     on;

    keepalive_timeout  65;

    #gzip  on;

    include /etc/nginx/conf.d/*.conf;
}
```

Estructura del archivo de configuración

- El módulo HTTP introduce tres bloques lógicos:
- ***http***: Este bloque se inserta en la raíz del archivo de configuración. Le permite comenzar a definir directivas y bloques.
- ***server:*** En este bloque se declara un sitio web y recibe su propia información. Se usa dentro del bloque http.
- ***location:*** Permite definir un grupo de configuraciones para aplicar a una ubicación en particular del sitio web. Este bloque puede estar dentro de un ***server*** o anidado dentro de otro ***location***.



HOST VIRTUAL

```
server {
    listen       80;
    server_name  localhost;
    location / {
        root   html;
        index  index.html index.htm;
    }
```

- Principales directivas a configurar para poder crear un virtual host.

Listen

Contexto:

listen *[address][:port] [additional options]*

Additional Options:

default_server

ssl

http2

proxy_protocol

HOST VIRTUAL

```
server {
    listen       80;
    server_name  localhost;
    location / {
        root   html;
        index  index.html index.htm;
    }
```

server_name

Contexto:

server_name *hostname1 [hostname2]* …

Ejemplos:

server_name www.ose.pe;

server_name *.ose.pe;

server_name *ose.*

server_name ose.pe “”;

server_name _ “”;

HOST VIRTUAL

```
server {
    listen       80;
    server_name  localhost;
    location / {
        root   html;
        index  index.html index.htm;
    }
```

location

Contexto:

location *optional_modifier location-match* {}

Ejemplos:

location /site {}

location = /page1 {}

location ~* \.(jpe?g|png|gif|ico) {}

location ^~ /costumes {}

```
server {
    listen       80;
    server_name  localhost;
    location / {
        root   html;
        index  index.html index.htm;
    }
```

Location con el modificador:

= : si coincide exactamente con la URI.

^~ : si la expresión comienza con la URI.

~ : si la expresión coincide con la URI.

~* : si la expresión coincide con la URI.

Sin modificador: si coincide o comienza con la URI.

HOST VIRTUAL

```
server {
    listen       80;
    server_name  localhost;
    location / {
        root   html;
        index  index.html index.htm;
    }
```

root

Contexto:

root */where/is/the/web/page;*

index

Contexto:

index *index.html index.0.html /index.html;*

ERROR PAGES

- Módulos que trabajan junto a su bloque location.

- Redirigir todos los errores o individualmente cada uno.

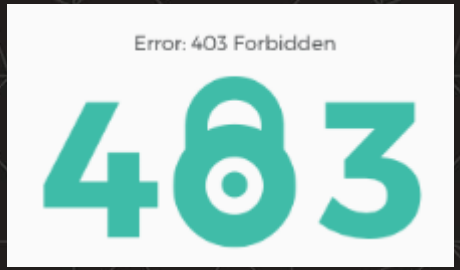
error_page

Contexto:

error_page *code … [=[response]] uri;*

Ejemplos:

```
error_page 404       /404.html;
error_page 500 502 503 504 /50x.html;
location ~* \.(jpe?g|png|gif|ico) {}
location ^~ /costumes {}
```



Módulo rewrite

- Elemento clave para la SEO.
- Nuevo procesamiento de solicitudes.
- Dar una vista mas agradable a la URL.

Redirección 301 y 302

- Al realizar el mantenimiento de la página.
- Hemos movido a una nueva dirección nuestro contenido web.
- Presentan el siguiente contexto:

```
rewrite ^/oldlocation$ http://newdomain.com/ redirect;
rewrite ^/oldlocation$ http://newweb/ permanent;
```

Questions and Next Steps

NGINX