

Exercisi 1:

Creamos la carpeta:

```
debian12abhiexam@debian12abhiexam:/var/www$ sudo mkdir -p /var/www/examen
debian12abhiexam@debian12abhiexam:/var/www$ ls
examen  html
debian12abhiexam@debian12abhiexam:/var/www$ |
```

Asignamos el www-data como propietario :

```
debian12abhiexam@debian12abhiexam:/var/www$ sudo chown -R www-data:www-data /var/www/examen/
debian12abhiexam@debian12abhiexam:/var/www$ ls
examen  html
debian12abhiexam@debian12abhiexam:/var/www$ ls -la examen/
total 8
drwxr-xr-x 2 www-data www-data 4096 dic 14 15:47 .
drwxr-xr-x 4 root      root    4096 dic 14 15:47 ..
debian12abhiexam@debian12abhiexam:/var/www$ |
```

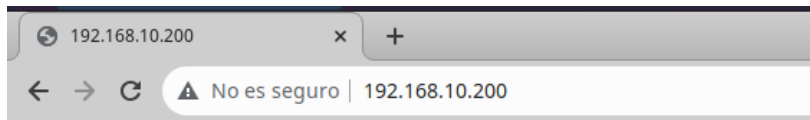
Asignamos los permisos:

```
debian12abhiexam@debian12abhiexam:~$ sudo chmod -R 755 /var/www/examen/
debian12abhiexam@debian12abhiexam:~$ ls -la /var/www/examen/
total 8
drwxr-xr-x 2 www-data www-data 4096 dic 14 15:47 .
drwxr-xr-x 4 root      root    4096 dic 14 15:47 ..
debian12abhiexam@debian12abhiexam:~$ |
```

Creamos el **index.html** dentro del **/var/www/examen**

editamos el archivo default en /etc/nginx/sites-available :

1. Accedemos desde el cliente (xubuntu en mi caso):



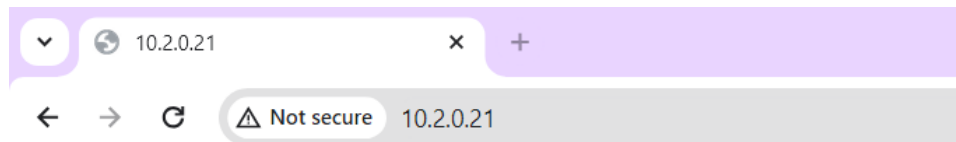
Examen

Pagina examen 1ra evaluacion

Abhijeet Singh

2. Podemos acceder desde la maquina anfitriona porque nuestro debian tiene 2 redes, una interna y otra adaptador puente con lo cual se puede acceder con el ip publica.

3. acceso desde anfitrion:



Examen

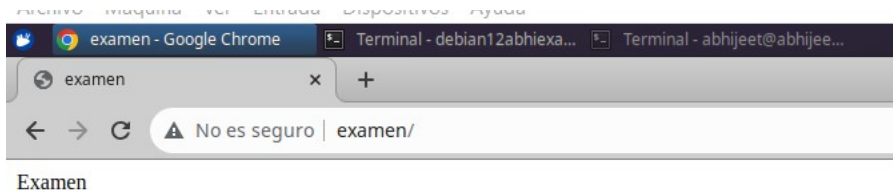
Pagina examen 1ra evaluacion

Abhijeet Singh

4.
editamos el fichero `/etc/hosts` para poder acceder con el nombre del sitio web:

```
Terminal - abhijeet@abhijeet-VirtualBox: ~
Archivo  Editar  Ver  Terminal  Pestañas  Ayuda
GNU nano 6.2 /etc/hosts
127.0.0.1 localhost
127.0.1.1 abhijeet-VirtualBox
#10.2.0.143 abhi
#10.2.0.143 perfect-learn
#10.2.0.143 webserver1
#10.2.0.45 webserver2
#10.2.0.227 balanceig
192.168.10.200 examen
```

Podemos acceder correctamente:



Pagina examen 1ra evaluacion

Abhijeet Singh

Exercisi 2:

Creamos el usuario y la contraseña:

```
Archivo  Editar  Ver  Terminal  Pestañas  Ayuda
debian12abhiexam@debian12abhiexam:/etc/nginx$ sudo sh -c "echo -n 'abhijeet:' >> /etc/nginx/.htpasswd"
[sudo] contraseña para debian12abhiexam:
debian12abhiexam@debian12abhiexam:/etc/nginx$ sudo sh -c "openssl passwd -apr1 >> /etc/nginx/.htpasswd"
Password:
Verifying - Password:
debian12abhiexam@debian12abhiexam:/etc/nginx$ |
```

comprobamos que se ha generado el usuario y la contraseña correctamente:

```
debian12abhiexam@debian12abhiexam:/etc/nginx$ cat /etc/nginx/.htpasswd
abhijeet:$apr1$wUl6JQk2$MwLzy2UGB88jthWFWr4hu0
debian12abhiexam@debian12abhiexam:/etc/nginx$ |
```

creamos la carpeta:

```
debian12abhiexam@debian12abhiexam:/var/www/examen$ sudo mkdir privat
debian12abhiexam@debian12abhiexam:/var/www/examen$ ls
index.html  privat
```

creamos el archivo index.html dentro del directorio privat:

```
Terminal - debian12abhiexam@debian12abhiexam: /var/www/examen/privat
Archivo  Editar  Ver  Terminal  Pestañas  Ayuda
GNU nano 7.2 index.html *
<html>
    <body>
        <h1>Area privada, Guillermo eres un genio</h1>
    </body>
</html>
```

modificamos el server block para que solo puedan usuarios registrados :

```
root /var/www/examen/;

# Add index.php to the list if you are using PHP
index index.html index.htm index.nginx-debian.html;

server_name examen;

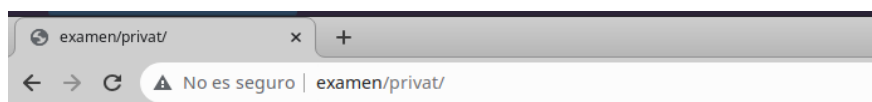
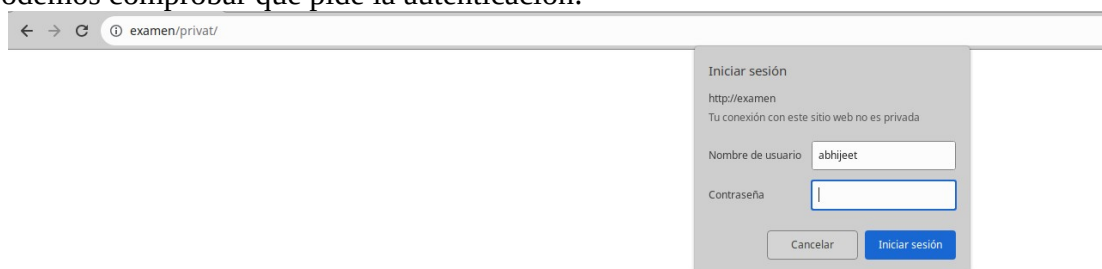
location /privat {
    # First attempt to serve request as file, then
    # as directory, then fall back to displaying a 404.
    auth_basic "Area privada";
    auth_basic_user_file /etc/nginx/.htpasswd;
    try_files $uri $uri/ =404;
}

# pass PHP scripts to FastCGI server
```

reiniciamos el servidor:

```
debian12abhiexam@debian12abhiexam:/etc/nginx/sites-available$ sudo systemctl restart nginx.service
debian12abhiexam@debian12abhiexam:/etc/nginx/sites-available$ sudo nano default
```

Podemos comprobar que pide la autenticacion:



Area privada, Guillermo eres un genio

Configuramos de forma que se pueda acceder cualquier desde maquina anfitrión y solo los usuarios registrados desde cliente.

```
GNU nano 7.2 default *
index index.html index.htm index.nginx-debian.html;

server_name examen;

location /privat {
    # First attempt to serve request as file, then
    # as directory, then fall back to displaying a 404.
    satisfy any;
    allow 10.2.0.21/24;
    auth_basic "Area privada";
    auth_basic_user_file /etc/nginx/.htpasswd;

    try_files $uri $uri/ =404;
    deny all;
}

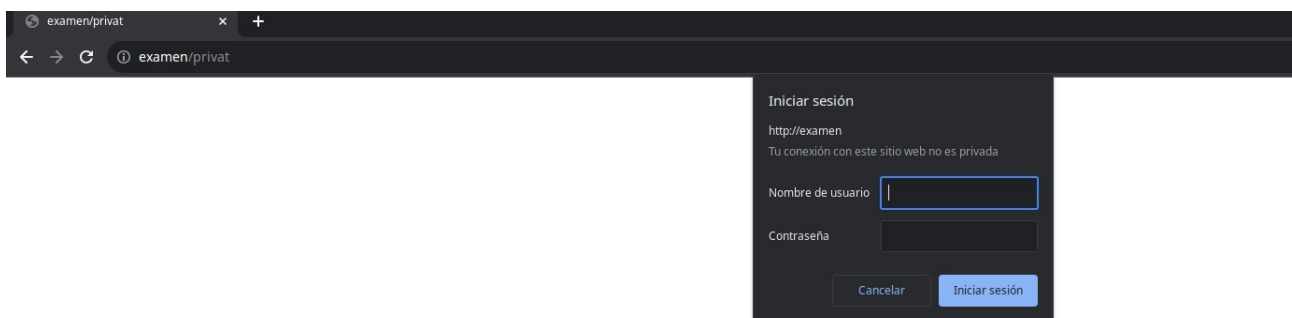
# pass PHP scripts to FastCGI server
#
```

reinicamos el servidor:

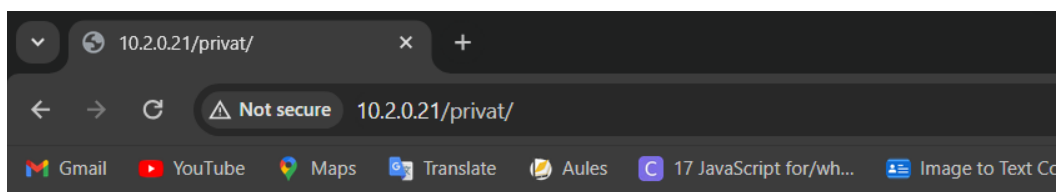
```
debian12abhiexam@debian12abhiexam:/etc/nginx/sites-available$ sudo systemctl restart nginx.service
debian12abhiexam@debian12abhiexam:/etc/nginx/sites-available$ sudo nano default
```

comprobamos de nuevo.

Pide la autenticación desde cliente:



accedemos directamente desde maquina anfitrión:



Area privada, Guillermo eres un genio

Exercisi 3:

Cambiamos el ip de servidor a 192.168.10.201:

```
link/ether 08:00:27:ee:61:02 brd ff:ff:ff:ff:ff:ff
inet 192.168.10.201/24 brd 192.168.10.255 scope global eth0
    valid_lft forever preferred_lft forever
inet6 fe80::a00:27ff:feee:6102/64 scope link
    valid_lft forever preferred_lft forever
```

Cambiamos el ip de proxy a 192.168.10.202

```
link/ether 08:00:27:a6:47:01 brd ff:ff:ff:ff:ff:ff
inet 192.168.10.202/24 brd 192.168.10.255 scope global eth0
    valid_lft forever preferred_lft forever
inet6 fe80::a00:27ff:fea6:4701/64 scope link
    valid_lft forever preferred_lft forever
bian12abhiexam@debian12abhiexam:~$
```

cambiamos el servidor que escuche por 8080

```
GNU nano 7.2
server {
    #listen 80 default_server;
    #listen [::]:80 default_server;
    listen 8080,
    listen [::]:8080;
    # este es el servidor 1
    # ssl_certificate
```

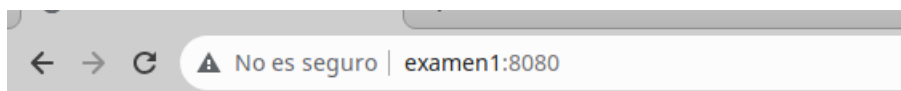
cambiamos el index.html:

```
GNU nano 7.2
<html>
  <head>Examen</head>
  <body>
    <h1>Pagina examen 1ra evaluacion</h1>
    <h2>Abhijeet Singh</h2>
    <h3> Server1</h3>
  </body>
</html>
```

cambiamos en /etc/hosts:

```
#10.2.0.45 webserver2
#10.2.0.227 balanceig
192.168.10.200 examen1
```

accedemos:



Examen

Pagina examen 1ra evaluacion

Abhijeet Singh

Server1

cambiamos el servidor2 que escuche por 8080

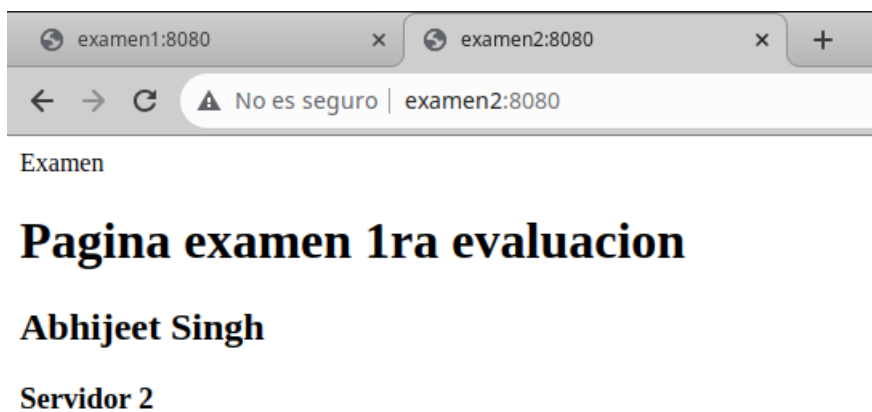
```
#
server {
    #listen 80 default_server;
    #listen [::]:80 default_server;

    listen 8080;
    listen [::]:8080;
    # SSL configuration
```

```
debian12abhiexam@debian12abhiexam:/var/www/examen$ cat index.html
<html>
  <head>Examen</head>
  <body>
    <h1>Pagina examen 1ra evaluacion</h1>
    <h2>Abhijeet Singh</h2>
    <h3>Servidor 2</h3>
  </body>
</html>
debian12abhiexam@debian12abhiexam:/var/www/examen$
```

```
#10.2.0.22/ balanceig
192.168.10.200 examen1
192.168.10.201 examen2
```

accedemos:



configuramos le balancerador:

```
upstream backend_host{
    random;
    server 192.168.10.200:8080;
    server 192.168.10.201:8080;
}
```

```

nano /./                default ^
listen [::]:80 default_server;

# SSL configuration
#
# listen 443 ssl default_server;
# listen [::]:443 ssl default_server;
#
# Note: You should disable gzip for SSL traffic.
# See: https://bugs.debian.org/773332
#
# Read up on ssl_ciphers to ensure a secure configuration.
# See: https://bugs.debian.org/765782
#
# Self signed certs generated by the ssl-cert package
# Don't use them in a production server!
#
# include snippets/snakeoil.conf;

# Add index.php to the list if you are using PHP

server_name balancejador;

location / {
    # First attempt to serve request as file, then
    # as directory, then fall back to displaying a 404.

    proxy_pass      http://backend_hosts;
}

```

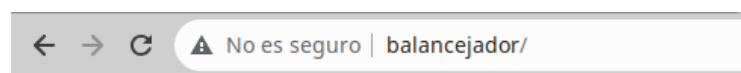
```

#10.2.0.45    webserver2
#10.2.0.227   balanceig
192.168.10.200 examen1
192.168.10.201 examen2
192.168.10.202 balancejador

# The following lines are desirable for IPv6 capable servers
:::1 ip6-localhost ip6-loopback

```

accedemos :

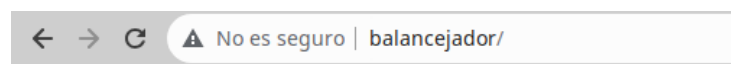


Examen

Pagina examen 1ra evaluacion

Abhijeet Singh

Server1



Examen

Pagina examen 1ra evaluacion

Abhijeet Singh

Servidor 2

4. Vercel

clonamos el repositorio

```
abhijeet@abhijeet:~/examen$ sudo git clone https://github.com/stackabuse/color-shades-generator
Clonando en 'color-shades-generator'...
remote: Enumerating objects: 61, done.
remote: Counting objects: 100% (61/61), done.
remote: Compressing objects: 100% (49/49), done.
remote: Total 61 (delta 21), reused 49 (delta 12), pack-reused 0
Recibiendo objetos: 100% (61/61), 287.91 KiB | 1.81 MiB/s, listo.
Resolviendo deltas: 100% (21/21), listo.
abhijeet@abhijeet:~/examen$ ls
color-shades-generator
abhijeet@abhijeet:~/examen$ cd color-shades-generator/
abhijeet@abhijeet:~/examen/color-shades-generator$ ls
package.json  package-lock.json  public  README.md  src
abhijeet@abhijeet:~/examen/color-shades-generator$
```

Instalamos las dependencias:

```
abhijeet@abhijeet:~/examen/color-shades-generator$ sudo npm install
npm WARN deprecated source-map-url@0.4.1: See https://github.com/lydell/source-map-url#deprecated
npm WARN deprecated source-map-resolve@0.6.0: See https://github.com/lydell/source-map-resolve#deprecated
npm WARN deprecated svgo@1.3.2: This SVGO version is no longer supported. Upgrade to v2.x.x.

added 1396 packages, and audited 1397 packages in 27s

167 packages are looking for funding
  run `npm fund` for details

31 vulnerabilities (10 moderate, 16 high, 5 critical)

To address issues that do not require attention, run:
  npm audit fix

To address all issues (including breaking changes), run:
  npm audit fix --force

Run `npm audit` for details.
abhijeet@abhijeet:~/examen/color-shades-generator$
```

generamps el build:

```
abhiyeet@abhiyeet:~/examen/color-shades-generator$ sudo npm run build

> color-shades-generator@0.1.0 build
> react-scripts build

Creating an optimized production build...
Browserslist: caniuse-lite is outdated. Please run:
  npx browserslist@latest --update-db
  Why you should do it regularly: https://github.com/browserslist/browserslist#browsers-data-updating
Browserslist: caniuse-lite is outdated. Please run:
  npx browserslist@latest --update-db
  Why you should do it regularly: https://github.com/browserslist/browserslist#browsers-data-updating
Compiled successfully.

File sizes after gzip:

  48.93 kB  build/static/js/main.56aeaed6.js
  965 B     build/static/css/main.el06b4ce.css

The project was built assuming it is hosted at /.
You can control this with the homepage field in your package.json.

The build folder is ready to be deployed.
You may serve it with a static server:

  npm install -g serve
  serve -s build

Find out more about deployment here:

  https://cra.link/deployment

abhiyeet@abhiyeet:~/examen/color-shades-generator$ ls
build  node_modules  package.json  package-lock.json  public  README.md  src
abhiyeet@abhiyeet:~/examen/color-shades-generator$
```

Instalamos vercel:

```
abhijeet@abhijeet:~/examen/color-shades-generator$ sudo npm i -g vercel
npm WARN deprecated debug@4.1.1: Debug versions >=3.2.0 <3.2.7 || >=4 <4.3.1 have a low-severity ReDos regression when used in a Node.js environment. It is recommended you upgrade to 3.2.7 or 4.3.1. (https://github.com/visionmedia/debug/issues/797)
npm WARN deprecated uuid@3.3.2: Please upgrade to version 7 or higher. Older versions may use Math.random() in certain circumstances, which is known to be problematic. See https://v8.dev/blog/math-random for details.

changed 246 packages, and audited 247 packages in 17s

17 packages are looking for funding
  run `npm fund` for details

3 moderate severity vulnerabilities

To address all issues, run:
  npm audit fix --force

Run `npm audit` for details.
abhijeet@abhijeet:~/examen/color-shades-generator$ vercel --version
Vercel CLI 32.7.2
32.7.2
abhijeet@abhijeet:~/examen/color-shades-generator$
```

generamos el token:

Create Token

Enter a unique name for your token to differentiate it from other tokens.
Then select the scope for the token.

TOKEN NAME	SCOPE	EXPIRATION
<input type="text" value="token_examen"/>	<div>Abhijeet Singh's projects</div>	<div>7 Days</div>

This token will expire on 12/21/23

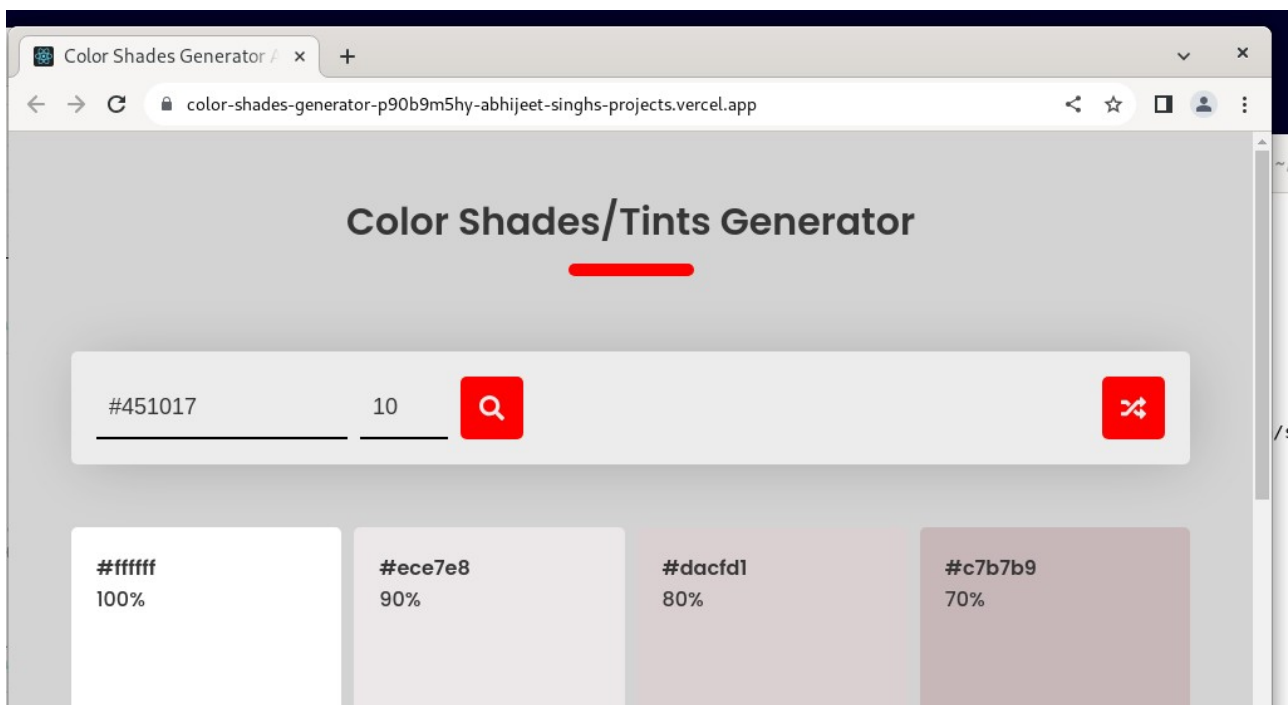
Learn more about [Access Tokens](#)

Create

hacemos el deploy contoken:

```
abhijeet@abhijeet:~/examen/color-shades-generator$ vercel -t l0moPMhBkPYeGkn9qjBCZR7c
Vercel CLI 32.7.2
? Set up and deploy "~/examen/color-shades-generator"? [Y/n] y
? Which scope do you want to deploy to? Abhijeet Singh's projects
? Link to existing project? [y/N] n
? What's your project's name? color-shades-generator
? In which directory is your code located? ./
Local settings detected in vercel.json:
Auto-detected Project Settings (Create React App):
- Build Command: react-scripts build
- Development Command: react-scripts start
- Install Command: `yarn install`, `pnpm install`, `npm install`, or `bun install`
- Output Directory: build
? Want to modify these settings? [y/N] n
Error: EACCES: permission denied, mkdir '/home/abhijeet/examen/color-shades-generator/.vercel'
abhijeet@abhijeet:~/examen/color-shades-generator$ sudo vercel -t l0moPMhBkPYeGkn9qjBCZR7c
Vercel CLI 32.7.2
? Set up and deploy "/home/abhijeet/examen/color-shades-generator"? [Y/n] y
? Which scope do you want to deploy to? Abhijeet Singh's projects
? Found project "abhijeet-singhs-projects/color-shades-generator". Link to it? [Y/n] y
🔗 Linked to abhijeet-singhs-projects/color-shades-generator (created .vercel and added it to .gitignore)
🔍 Inspect: https://vercel.com/abhijeet-singhs-projects/color-shades-generator/7GTiA71mS5zof8ACgDKpGAAqEcCE [1s]
✅ Preview: https://color-shades-generator-p90b9m5hy-abhijeet-singhs-projects.vercel.app [1s]
🚀 To deploy to production (color-shades-generator-nine.vercel.app), run `vercel --prod`
abhijeet@abhijeet:~/examen/color-shades-generator$
```

accedemos:



deploy en prdocucuin:

📌 To deploy to production (color-shades-generator-nine.vercel.app), run `vercel --prod`

```
abhiyeet@abhiyeet:~/examen/color-shades-generator$ vercel --prod
vercel --prod
Vercel CLI 32.7.2
🔍 Inspect: https://vercel.com/abhiyeet-singhs-projects/color-shades-generator/36skNXguZu9zXBRj3i2fg2w2C546 [1s]
✅ Production: https://color-shades-generator-k4vmqzugg-abhiyeet-singhs-projects.vercel.app [1s]
abhiyeet@abhiyeet:~/examen/color-shades-generator$
```