

## Inicio y Despliegue Servidor en AWS

### Asignar IP Elastica

En el Panel de AWS -> Red y Seguridad -> Direcciones IP elástica.

Ir al botón Asignar la dirección IP elástica -> Asignar

Seleccionar la dirección -> Acciones -> Asociar la dirección IP elástica

En el apartado de Instancia seleccionar la instancia y asociar.

### Conexión al terminal AWS desde consola local.

(importante tener localizado el archivo .pem con las claves de acceso)

*ssh -i archivo.pem ubuntu@IP\_Publica*

\$ ssh -i ~/Desktop/testtareas.pem [ubuntu@52.53.98.210](https://52.53.98.210)

### Actualizar Servidor.

- Descargar Actualizaciones  
Update  
\$ sudo apt update
- Instalar Actualizaciones  
Upgrade  
\$ sudo apt upgrade

### Nginx.

- Instalar nginx  
\$ sudo apt install nginx
- Conocer estado  
\$ sudo systemctl status nginx
- Hacer un test.  
\$ sudo nginx -t

- Reiniciar nginx.  
\$ sudo systemctl reload nginx

Atencion!! Recordar tener el puerto http abierto en AWS.

### **Aplicación Node.**

#### Usuario para apps.

- Crear usuario apps  
\$ sudo adduser apps
- Entramos como apps  
\$ sudo -u apps -i
- Bloquear el login de apps (desde usuario ubuntu)  
\$ sudo passwd -l apps
- Dar Permisos a la capeta apps  
\$ sudo chmod o+rx /home/apps/

#### Instalar nvm para versiones de node.

- Entramos como apps  
\$ sudo -u apps -i
- Instalar nvm. Revisar documentación -> <https://github.com/nvm-sh/nvm>  
wget -qO- [https://raw.githubusercontent.com/nvm-sh/nvm-v0.39.5/install.sh](https://raw.githubusercontent.com/nvm-sh/nvm/v0.39.5/install.sh) | bash

- Dejar de ser apps y volver a entrar, para cargar la nueva configuración
- Instalar versión de node  
\$ nvm install node

#### Descargamos el proyecto node.

- git clone https://ruta del proyecto
- Entramos en el proyecto (cd nombre\_de\_la\_carpeta)
- Instalamos dependencias (npm install)
- Creamos archivo con variables de entorno
- Podemos comprobar que arranca (npm start)

#### Instalamos MongoDB

- Como usuario Ubuntu
- Instalar mongo ->  
<https://www.mongodb.com/docs/manual/tutorial/install-mongodb-on-ubuntu/>  
sudo apt-get install gnupg curl

```
curl -fsSL https://pgp.mongodb.com/server-7.0.asc | \
sudo gpg -o /usr/share/keyrings/mongodb-server-7.0.gpg \
--dearmor
```

```
echo "deb [ arch=amd64,arm64 ] https://repo.mongodb.org/apt/ubuntu
jammy/mongodb-org/7.0 multiverse" | sudo tee
/etc/apt/sources.list.d/mongodb-org-7.0.list
```

```
sudo apt-get update
sudo apt-get install -y mongodb-org
sudo systemctl start mongod
sudo systemctl status mongod
sudo systemctl enable mongod (Para que mongo inicie con el sistema)
```

### Poner seguridad en mongodb, añadir contraseña.

- Conectar a mongo (mongosh)  
\$ mongosh
- Cambiar a la BD de admin

use admin

- Creamos el usuario para la BD

```
db.createUser(  
{  
  user: "admin",  
  pwd: "supersegura",  
  roles: [  
    { role: "userAdminAnyDatabase", db: "admin" },  
    { role: "readWriteAnyDatabase", db: "admin" }  
  ]  
}  
)
```

- con el usuario Ubuntu cambiar el archivo de configuración de mongo  
/etc/mongod.conf
- Des comentamos la línea Security:  
authorization: enabled

- Reiniciar mongobd (sudo systemctl restart mongod)
- Comprobar estado de mongo (sudo systemctl status mongod)
- Entrar en mongo (mongosh)
- Conectar usuario administrador  
mongosh --authenticationDatabase admin -u admin -p

- Crear un usuario para la BD, (después de conectar como administrador,  
ejecutar)=

```
db.createUser(  
{  
  user: "parseuser",  
  pwd: "parsepass",  
  roles: [ { role: "readWrite", db: "parsedb" } ]  
})
```

```
}  
)
```

- Para crear la variable de entorno con el usuario y la contraseña.

```
export
```

```
DATABASE_URI=mongodb://parseuser:parsepass@localhost:27017/parsedb
```

### Instalar y configurar PM2.

- Instalar pm2

```
$ npm install -g pm2
```

- Creamos ecosystem para gestionar aplicaciones

```
$ pm2 ecosystem
```

- Modificamos el ecosystem

```
$ nano ecosystem.config.js
```

- Sustituir el contenido por:

```
module.exports = {  
  apps : [{  
    "name": "testtareas",  
    "script": "npm",  
    "args": "start",  
    "cwd": "/home/apps/BackEnd"  
  }]  
};
```

- Arrancar ecosystem

```
$ pm2 start ecosystem.config.js
```

- Iniciar ecosystem

```
$ pm2 startup
```

- Saldrá un comando en consola, copia y ejecuta el comando

- Guardar la imagen del sistema para que inicie pm2  
\$ pm2 save
- Para ver el listado de las apps  
\$ pm2 list
- Para ver el listado de logs  
\$ pm2 logs
- Si existe un problema y tenemos que parar pm2  
\$ pm2 stop all  
\$ pm2 delete all

### Configurar Nginx.

- Como usuario Ubuntu en la carpeta /etc/nginx/sites-availables/
- Crear archivo de configuración

\$ sudo nano node-testtareas

- Escribir

```
server {
    listen 80;
    server_name ec2-52-53-98-210.us-west-1.compute.amazonaws.com;
```

```
    location ~ ^/(images|stylesheets)/ {
        root /home/apps/Backend/public;
        access_log off;
        expires max;
        add_header X-Owner testtareas;
    }
```

```
    location / {
        proxy_pass http://127.0.0.1:3001;
        proxy_redirect off;
        proxy_http_version 1.1;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection "Upgrade";
```

```
    proxy_set_header Host $host;
}
}
```

- Crear el link en sites-enabled  
\$ sudo ln -s /etc/nginx/sites-available/node-testtareas /etc/nginx/sites-enabled/node-testtareas
- Hacer un test.  
\$ sudo nginx -t
- Reiniciar nginx.  
\$ sudo systemctl reload nginx

### **Desplegar Node.**

- Asegurar que la rama main dispone de la última versión de develop (si procede)
- Entramos como apps  
\$ sudo -u apps -i
- Borrar BackEnd\_old  
\$ rm -rf BackEnd\_old/
- Renombramos o borramos la carpeta  
\$ mv BackEnd/ BackEnd\_old/
- Paramos pm2  
\$ pm2 stop all  
\$ pm2 delete all
- git clone https://ruta del proyecto (El repositorio tiene que ser público, o logearte)
- Entramos en el proyecto (cd nombre\_de\_la\_carpeta)
- Instalamos dependencias (npm install)
- Creamos o copiamos el archivo con variables de entorno

- ```
$ cd BackEnd
```
- ```
$ cp ../BackEnd_old/.env .
```
- Iniciamos la BD
- ```
$ npm run initDB
```
- Arrancar ecosystem
- ```
$ cd ..
```
- ```
$ pm2 start ecosystem.config.js
```

### Pruebas Despliegue.

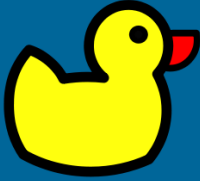
= <http://ec2-52-53-98-210.us-west-1.compute.amazonaws.com/>

- Apagamos servidor AWS desde la pagina web

### Aplicación Dominio DuckDns

<https://www.duckdns.org/domains>

Duck DNS
spec
about
why
install
faqs
logout
logged in with spacejumpkeepcoding@gmail.com |||



## Duck DNS

account spacejumpkeepcoding@gmail.com  
type free  
token 2d52c2d5-f57c-4acb-a33a-05e92a12cc2e  
token generated 4 minutes ago  
created date 16 Sep 2023, 11:46:47

success: ip address for spacejump.duckdns.org updated to 52.53.98.210

domains 1/5
http:// sub domain duckdns.org add domain

| domain    | current ip                              | ipv6                                                                                 | changed                                                    |
|-----------|-----------------------------------------|--------------------------------------------------------------------------------------|------------------------------------------------------------|
| spacejump | 52.53.98.210 <button>update ip</button> | <input type="text" value="ipv6 address"/> <input type="button" value="update ipv6"/> | 0 seconds ago <input type="button" value="delete domain"/> |

This site is protected by reCAPTCHA and the Google [Privacy Policy](#) and [Terms of Service](#) apply.



## Cambiamos archivos nginx

vamos a fichero de redirección de nginx

```
$ cd /etc/nginx/sites-available/
```

miramos archivos a modificar

```
$ ls
```

default node-testtareas

cambiamos el back end “node-testtareas” y añadimos el dominio.

```
$ sudo nano node-testtareas
```

```
server {  
    listen 80;  
    server_name      ec2-52-53-98-210.us-west-1.compute.amazonaws.com  
back.testtareas.duckdns.org;
```

```
    location ~ ^/(images|stylesheets)/ {  
        root /home/apps/Backend/public;  
        access_log off;  
        expires max;  
        add_header X-Owner testtareas;  
    }
```

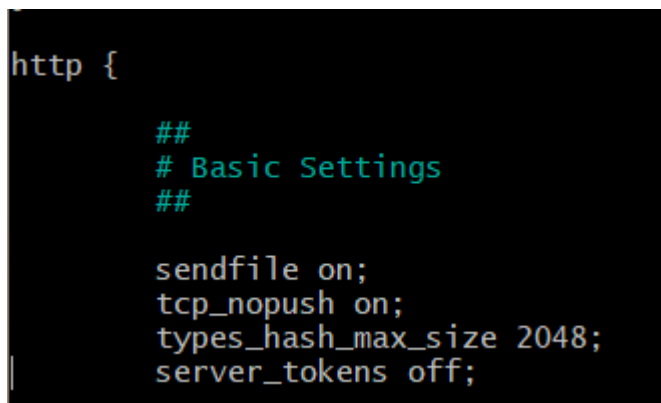
```
    location / {  
        proxy_pass http://127.0.0.1:3001;  
        proxy_redirect off;  
        proxy_http_version 1.1;  
        proxy_set_header Upgrade $http_upgrade;  
        proxy_set_header Connection "Upgrade";  
        proxy_set_header Host $host;  
    }  
}
```

## **Reiniciar nginx**

```
$ sudo nginx -t
nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
nginx: configuration file /etc/nginx/nginx.conf test is successful
$ sudo systemctl reload nginx
```

## **Evitar que nginx muestre el numero de versión**

```
$ sudo nano /etc/nginx/nginx.conf
Descomentar esta instrucción      # server_tokens off;
```



```
http {
    ##
    # Basic Settings
    ##

    sendfile on;
    tcp_nopush on;
    types_hash_max_size 2048;
    server_tokens off;
```

## **Aplicar certificado de seguridad https:**

<https://letsencrypt.org/es/getting-started/>

Aplicar con certbot

<https://certbot.eff.org/>

## My HTTP website is running ⌵ Nginx on ⌵ Ubuntu 20

Siguiendo las instrucciones de la web instalar con snap, pero instalaremos sin snap

Descargar certbot y python

```
$ sudo apt install certbot python3-certbot-nginx
```

Aplicar certificados a nginx

```
$ sudo certbot --nginx
```

Introducir el email por si no se renueva solo.

Enter email address (used for urgent renewal and security notices)

(Enter 'c' to cancel): testtareas@gmail.com

-----  
Please read the Terms of Service at

<https://letsencrypt.org/documents/LE-SA-v1.3-September-21-2022.pdf>. You must

agree in order to register with the ACME server. Do you agree?

-----  
(Y)es/(N)o: Y

-----  
Would you be willing, once your first certificate is successfully issued, to share your email address with the Electronic Frontier Foundation, a founding partner of the Let's Encrypt project and the non-profit organization that develops Certbot? We'd like to send you email about our work encrypting the web,

EFF news, campaigns, and ways to support digital freedom.  
-----

(Y)es/(N)o: N

Account registered.

Which names would you like to activate HTTPS for?

-----  
1: ec2-52-53-98-210.us-west-1.compute.amazonaws.com  
2: back.spacejump.duckdns.org  
-----

Select the appropriate numbers separated by commas and/or spaces, or leave input blank to select all options shown (Enter 'c' to cancel):

**si dejas el input en blanco selecciona todas las opciones.**

blank to select all options shown (Enter 'c' to cancel): 2 3

Requesting a certificate for testareas.duckdns.org and back.spacejump.duckdns.org

Successfully received certificate.

Certificate is saved at:  
/etc/letsencrypt/live/testareas.duckdns.org/fullchain.pem

Key is saved at: /etc/letsencrypt/live/spacejump.duckdns.org/privkey.pem

This certificate expires on 2023-12-15.

These files will be updated when the certificate renews.

Certbot has set up a scheduled task to automatically renew this certificate in the background.

Deploying certificate

Successfully deployed certificate for back.spacejump.duckdns.org to /etc/nginx/sites-enabled/node-testareas

Congratulations! You have successfully enabled HTTPS on <https://back.testareas.duckdns.org>

-----  
If you like Certbot, please consider supporting our work by:

\* Donating to ISRG / Let's Encrypt: <https://letsencrypt.org/donate>

\* Donating to EFF: <https://eff.org/donate-le>

## Activar puerto HTTPS en AWS

EC2 > Grupos de seguridad > sg-0215aabl7e3c5f4b4 - launch-wizard-1 > Editar reglas de entrada

### Editar reglas de entrada Información

Las reglas de entrada controlan el tráfico entrante que puede llegar a la instancia.

| ID de la regla del grupo de seguridad | Tipo <small>Información</small> | Protocolo <small>Información</small> | Intervalo de puertos <small>Información</small> | Origen <small>Información</small> | Descripción: opcional <small>Información</small> |          |
|---------------------------------------|---------------------------------|--------------------------------------|-------------------------------------------------|-----------------------------------|--------------------------------------------------|----------|
| sg-0a241e37dfeca52ec                  | SSH                             | TCP                                  | 22                                              | Personaliza...                    | Q<br>0.0.0.0/0 X                                 | Eliminar |
| sg-0baa9a6e35fba0e9                   | HTTP                            | TCP                                  | 80                                              | Personaliza...                    | Q<br>0.0.0.0/0 X                                 | Eliminar |
| -                                     | HTTPS                           | TCP                                  | 443                                             | Anywhere...                       | Q<br>0.0.0.0/0 X                                 | Eliminar |

Agregar regla

Cancelar Previsualizar los cambios Guardar reglas

## Nueva Rota

<https://back.testtareas.duckdns.org/>

## Proteger el sistema de boots con fail2ban

Instalar fail2ban

```
$ sudo apt install fail2ban
```

Activar fail2ban

```
$ sudo systemctl start fail2ban
```