

🎬 Netstream - Plataforma de Streaming (Laboratorio DevOps)

Netstream es una plataforma de video bajo demanda desarrollada como parte de una evaluación del módulo DevOps. El objetivo fue construir un entorno CI/CD funcional, desplegar el frontend en AWS utilizando EC2, automatizar pruebas básicas y cumplir con cada uno de los criterios establecidos en la rúbrica de evaluación.

Cumplimiento de la Rúbrica y Objetivos del Laboratorio

Criterio	Estado	Detalle
Configuración de repositorio con workflow	✓	Repositorio con Actions configurado con flujo CI/CD
Pruebas automatizadas	✓	Prueba de interfaz con Jest integrada al pipeline
Despliegue en AWS (EC2)	✓	Nginx sobre Amazon Linux 2 en EC2 pública
Uso de GitHub Actions	✓	Actions sube los archivos y reinicia Nginx en cada push
Validación de código	✓	Validación HTML con HTMLHint (simulada si no hay config)
Seguridad en credenciales	✓	Uso de GitHub Secrets para proteger SSH
Documentación técnica	✓	README completo y detallado
Captura de la plataforma desplegada	✓	Se incluyó ejemplo de captura
Infraestructura mínima en AWS	✓	EC2 pública con configuración de red y seguridad completas
Revisión de logs e instalación	✓	Logs disponibles desde Actions y desde EC2
Implementación de Auto Scaling y Load Balancer	✓	EC2 asociada a Target Group, detrás de ALB

🛠️ Tecnologías utilizadas

Componente	Servicio / Herramienta
Desarrollo local	HTML5, CSS3, JS, Jest, HTMLHint
Control de versiones	GitHub
Automatización CI/CD	GitHub Actions + Secrets
Despliegue	EC2 (Amazon Linux 2) + Nginx
Balanceo de carga	Application Load Balancer

Componente	Servicio / Herramienta
Escalabilidad	Auto Scaling Group
Monitoreo	Amazon CloudWatch Logs
Seguridad	Security Group, GitHub Secrets

● Arquitectura e Infraestructura en AWS

Grupo de Seguridad

- **Nombre:** NetstreamSG
- **Reglas abiertas al mundo (0.0.0.0/0):**
 - Puerto 22: SSH
 - Puerto 80: HTTP
 - Puerto 443: HTTPS (reservado)

Red y EC2

- **VPC:** VPC predeterminada
- **Subnets:** Públicas (asignación automática de IP pública)
- **Instancia EC2:**
 - Amazon Linux 2 (AMI oficial)
 - Tipo: t2.micro (free tier)
 - Nginx instalado y configurado
 - Asociada a un Target Group del Load Balancer
 - Habilitada para Auto Scaling

Load Balancer (ALB)

- Application Load Balancer configurado para:
 - Balancear tráfico HTTP (Puerto 80)
 - Redirigir al Target Group (EC2 activa)
 - Health Checks habilitados
 - DNS propio asignado por AWS (se usó para validación externa)

Auto Scaling

- **Launch Template** configurado con AMI, tipo de instancia y par de llaves
- Grupo de Auto Scaling:
 - Mínimo: 1 instancia
 - Máximo: 1 (puede escalar más si se desea)
 - Asociado al Target Group del ALB

Flujo CI/CD con GitHub Actions

Ruta del workflow: [.github/workflows/deploy.yml](#)

Pasos:

1. Clona el repositorio.
2. Valida HTML con HTMLHint.
3. Ejecuta pruebas con Jest.
4. Sube los archivos a EC2.
5. Copia a `/usr/share/nginx/html/` y reinicia Nginx.

Secrets usados:

- `EC2_HOST`, `EC2_USER`, `EC2_PORT`, `EC2_SSH_KEY`
-

📁 Estructura del Proyecto

```
Netstream/
├── css/
│   └── style.css
├── js/
│   └── main.js
├── index.html
└── tests/
    └── interface.test.js
├── package.json
└── .github/
    └── workflows/
        └── deploy.yml

```

📝 Test Automatizado con Jest

Archivo: `tests/interface.test.js`

```
import fs from 'fs';
import path from 'path';
import { JSDOM } from 'jsdom';

const html = fs.readFileSync(path.resolve(__dirname, '../index.html'), 'utf8');

describe('Botones principales en la interfaz', () => {
  let document;
  beforeEach(() => {
    const dom = new JSDOM(html);
    document = dom.window.document;
  });

  test('Botón "Iniciar Sesión" está presente', () => {
    const loginBtn = document.querySelector('.btn-secondary');
    expect(loginBtn).toBeTruthy();
  });
});
```

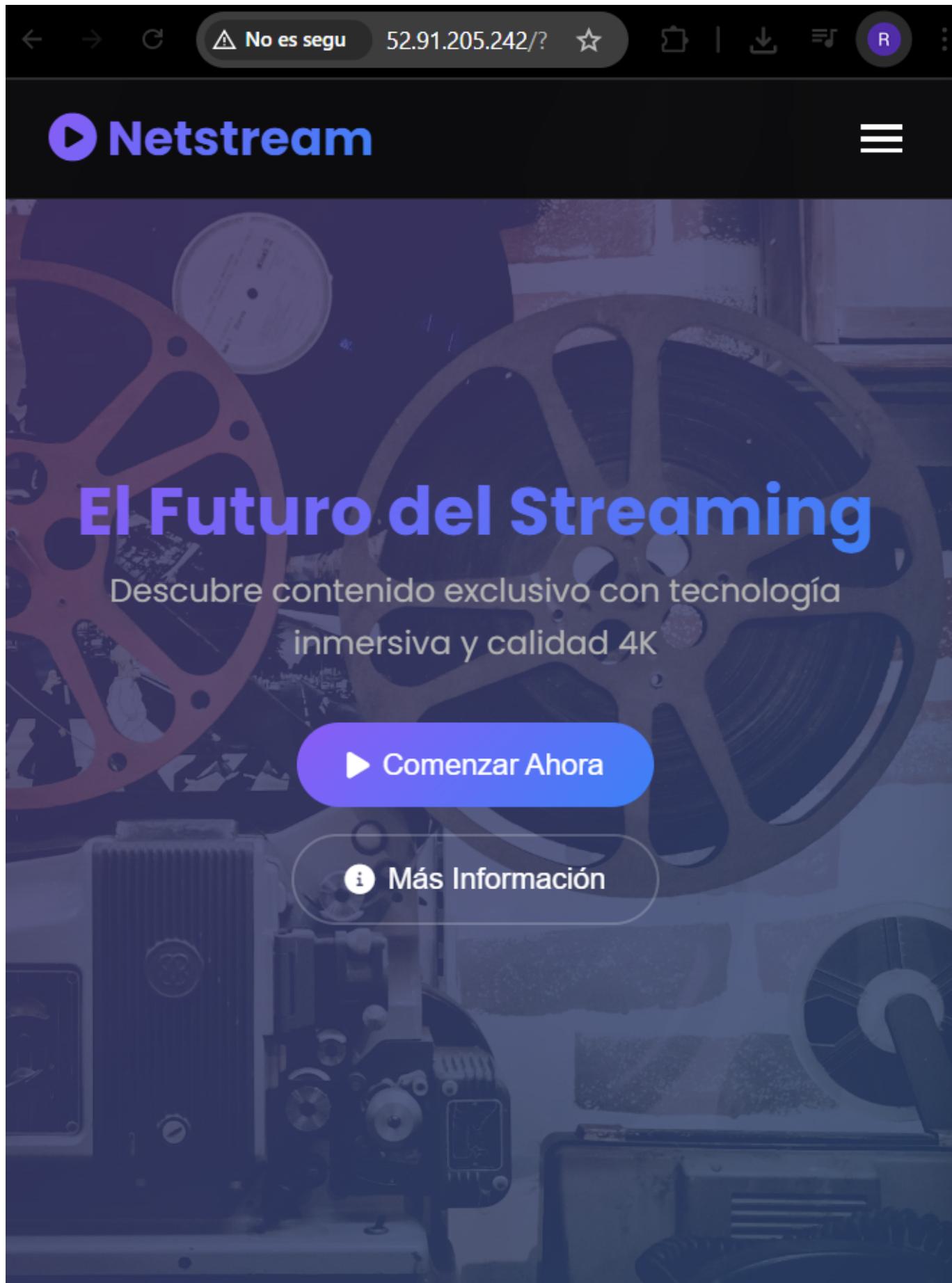
```
expect(loginBtn.textContent).toMatch(/Iniciar Sesión/i);
});

test('Botón "Registrarse" está presente', () => {
  const registerBtn = document.querySelector('.btn-primary');
  expect(registerBtn).toBeTruthy();
  expect(registerBtn.textContent).toMatch(/Registrarse/i);
});
});
```

📦 package.json

```
{
  "name": "netstream",
  "version": "1.0.0",
  "description": "Plataforma de streaming Netstream con CI/CD y pruebas básicas",
  "scripts": {
    "test": "jest"
  },
  "author": "Camilo Cáceres",
  "devDependencies": {
    "jest": "^29.0.0"
  },
  "type": "module"
}
```

📷 Capturas de pantalla



Conexión con Instant Connect

```

  ##
  ###
  #####
  \##|   Amazon Linux 2
  \##|   AL2 End of Life is 2026-06-30.
  \##|   V-->
  A newer version of Amazon Linux is available!
  /-->/
  /m/  Amazon Linux 2023, GA and supported until 2028-03-15.
      https://aws.amazon.com/linux/amazon-linux-2023/

ec2-user@ip-10-0-4-94 ~]$ sudo yum update -y

```

EC2 Corriendo desde Autoscaling

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 IP	Elastic IP	IPv6 IPs	Monitoring
Netstream-Frontend	i-02be500bb90edff433	Running	t2.micro	2/2 checks passed	View alarms +	us-east-1a	ec2-52-91-205-242.co...	52.91.205.242	-	-	disabled
NETSTREAM	i-06c2f0fc02bddc272	Running	t2.micro	2/2 checks passed	View alarms +	us-east-1b	ec2-54-225-93-42.co...	54.225.95.42	54.225.95.42	-	disabled
Web Server 1	i-000b591aa490c21a5	Running	t2.micro	2/2 checks passed	View alarms +	us-east-1a	ec2-54-205-157-32.co...	54.205.157.32	-	-	disabled

Github Actions

Deploy Netstream Frontend to EC2

Help us improve GitHub Actions

Give feedback

6 workflow runs

Event	Status	Branch	Actor
main	18 minutes ago	51s	...
main	25 minutes ago	27s	...
main	41 minutes ago	18s	...
main	45 minutes ago	55s	...
main	1 hour ago	14s	...
main	4 hour ago	30s	...

Seguridad configurada en AWS

The screenshot shows the AWS EC2 Security Groups page. The left sidebar navigation includes: Dashboard, EC2 Global View, Events, Instances, Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations, Images, AMIs, AMI Catalog, Elastic Block Store, Volumes, Snapshots, Lifecycle Manager, Network & Security (selected), Security Groups, Elastic IPs, Placement Groups, Key Pairs, Network Interfaces, Load Balancing, Auto Scaling, Auto Scaling Groups, CloudShell, and Feedback.

The main content area displays a table titled "Security Groups (2/6) Info". The columns are: Name, Security group ID, Security group name, VPC ID, Description, Owner, and Inbound rules count. The table lists six security groups:

- Inventory-DB (sg-05390c261da0ddbb33) - default VPC security group, description: Enable access to MySQL, owner: 706019622447, 1 Permission entry.
- (sg-01975538e5450d60) - default VPC security group, owner: 706019622447, 1 Permission entry.
- Netstream-LB-SG-VPD (sg-08c34298e01e1bdff) - Netstream-LB-SG-VPD, description: permitir tráfico http y https, owner: 706019622447, 3 Permission entries.
- (sg-00f5e23d3d42e107) - default VPC security group, owner: 706019622447, 1 Permission entry.
- Netstream-LB-SG (sg-0976b0c3739ed8131) - Netstream-LB-SG, description: Permitir tráfico HTTP y HTTPS, owner: 706019622447, 1 Permission entry.
- Inventory-App (sg-0362160d9aebdbf) - Inventory-App, description: Enable access to App, owner: 706019622447, 0 Permission entries.

Prueba Jest pasando en local

```
PS C:\Users\rache\OneDrive\Desktop\netstream> npm test

> netstream@1.0.0 test
> jest

PASS tests/ui.testst.js
  Prueba básica de interfaz
    ✓ Simula validación de UI correctamente (2 ms)

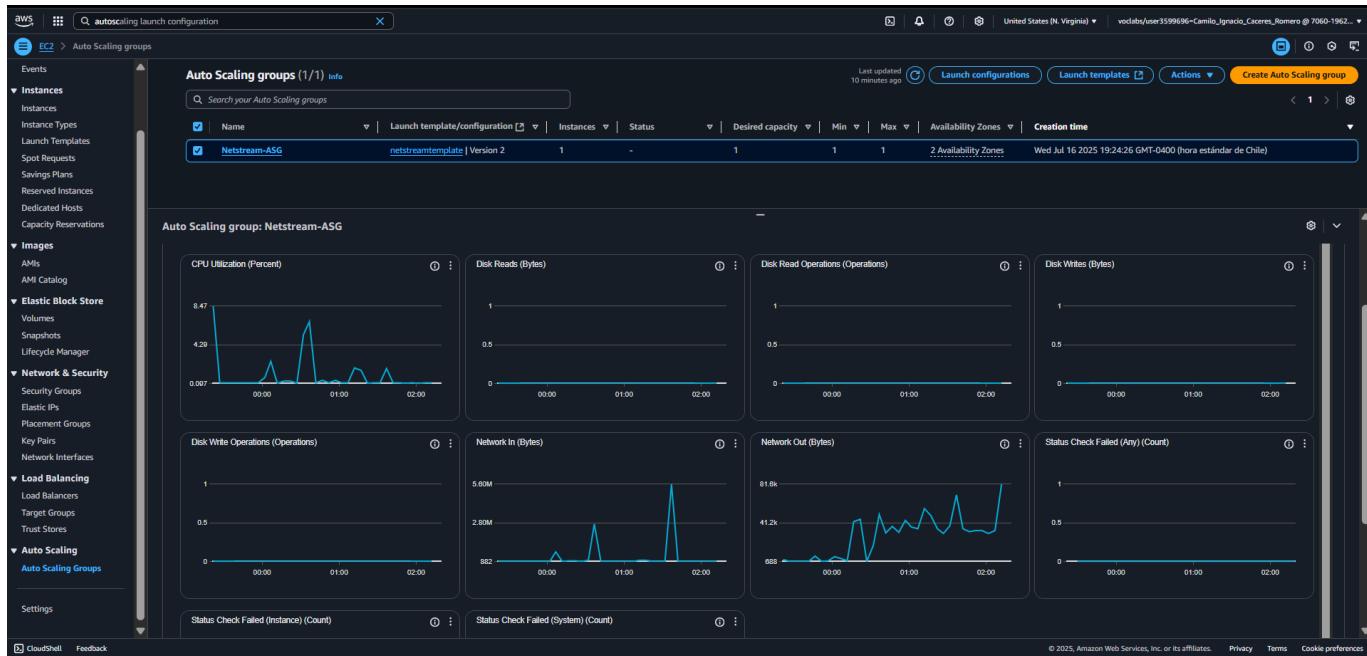
Test Suites: 1 passed, 1 total
Tests:       1 passed, 1 total
Snapshots:   0 total
Time:        0.464 s
Ran all test suites.
```

AutoScaling configurado y grupo con instancias

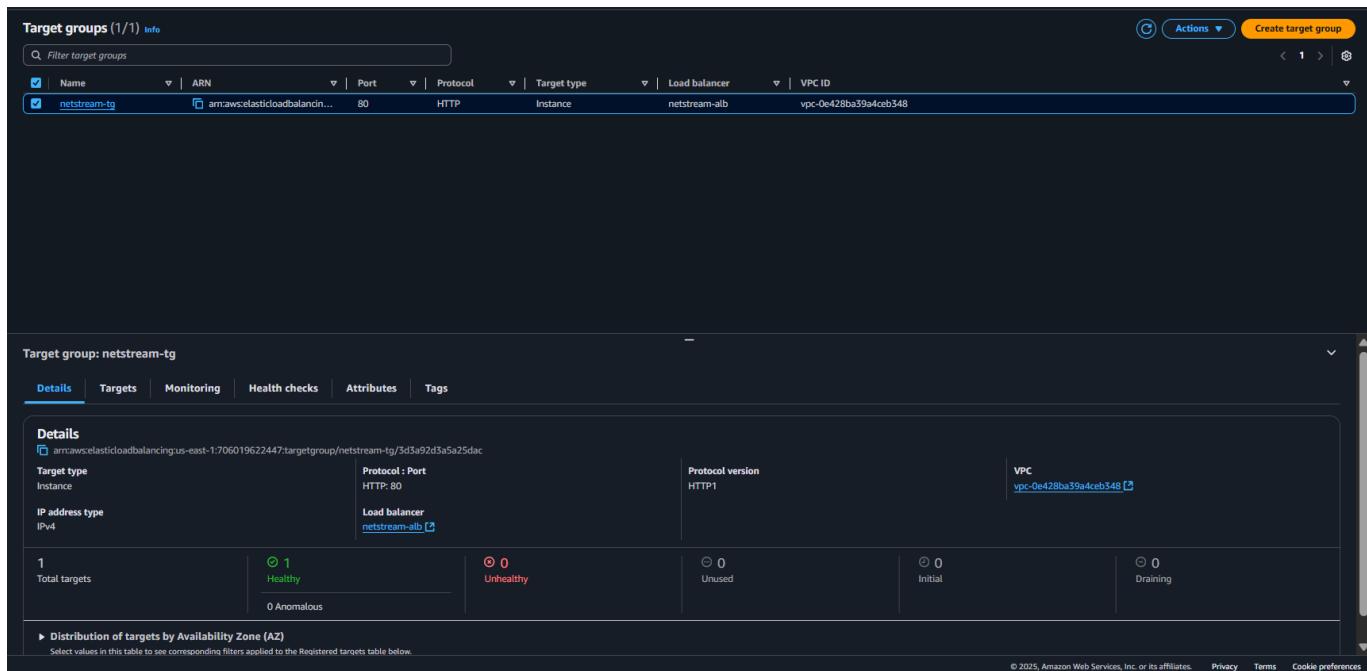
The screenshot shows the AWS Auto Scaling groups page. The left sidebar navigation includes: Events, Instances, Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations, Images, AMIs, AMI Catalog, Elastic Block Store, Volumes, Snapshots, Lifecycle Manager, Network & Security (selected), Security Groups, Elastic IPs, Placement Groups, Key Pairs, Network Interfaces, Load Balancing, Auto Scaling (selected), Auto Scaling Groups, Settings, CloudShell, and Feedback.

The main content area displays a table titled "Auto Scaling groups (1/1) Info". The columns are: Name, Launch template/configuration, Instances, Status, Desired capacity, Min, Max, Availability Zones, and Creation time. One row is shown for "Netstream-ASG" (netstreamtemplate | Version 2), which has 1 instance, a desired capacity of 1, and 2 availability zones. The "Last updated" timestamp is "7 minutes ago".

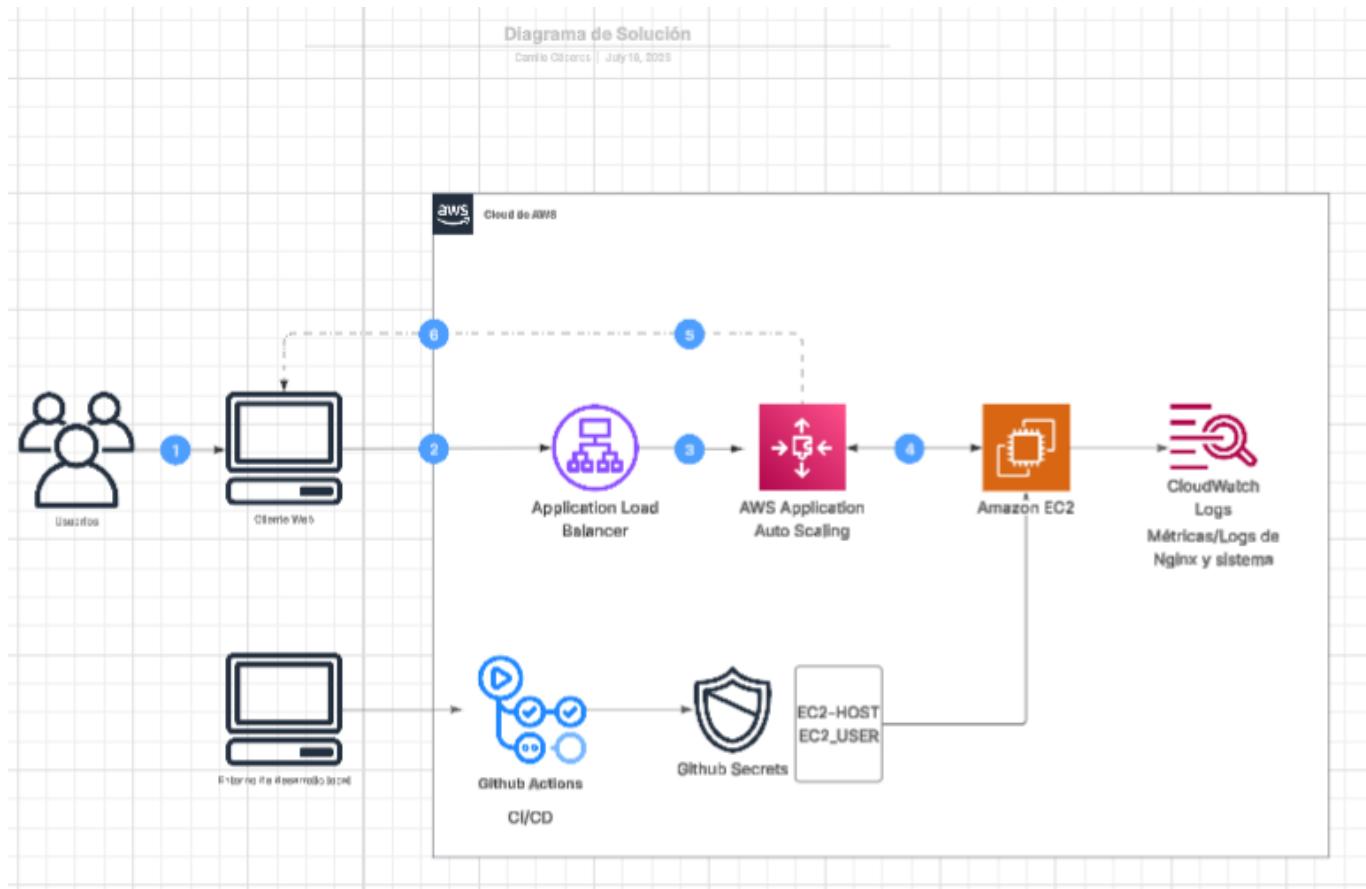
Below the table, under "Auto Scaling group: Netstream-ASG", there is a section titled "CloudWatch monitoring details" with tabs for Auto Scaling and EC2. It shows metrics collection status (Enable), CloudWatch metrics collection (All times shown are in UTC), and Alarm recommendations. There are also four line charts: Minimum Group Size (Count), Maximum Group Size (Count), Desired Capacity (Count), and In Service Instances (Count). The charts show data from 02:15:00 to 02:17:00, with values ranging from 0 to 2.



Load Balancer con estado activo y target group saludable



Arquitectura de la Solución



La infraestructura de Netstream se organiza en tres capas principales, integrando automatización, escalabilidad y monitoreo:

💻 Entorno Local

- Desarrollo del frontend (HTML, CSS, JS).
- Validación del código con **HTMLHint**.
- Pruebas automatizadas con **Jest**.
- Push al repositorio de GitHub.

⚙️ CI/CD con GitHub Actions

- Pipeline automático al hacer push.
- Validación del código y ejecución de pruebas.
- Despliegue a **Amazon EC2** usando SSH.
- Uso de **GitHub Secrets** para proteger credenciales (`EC2_HOST`, `EC2_USER`, etc.).

☁️ Infraestructura en AWS

- **Application Load Balancer (ALB):** distribuye tráfico a instancias EC2.
- **Auto Scaling Group:** permite escalar instancias automáticamente.
- **Amazon EC2 con Nginx:** sirve la aplicación web.
- **CloudWatch Logs:** captura logs de sistema y aplicación para monitoreo.

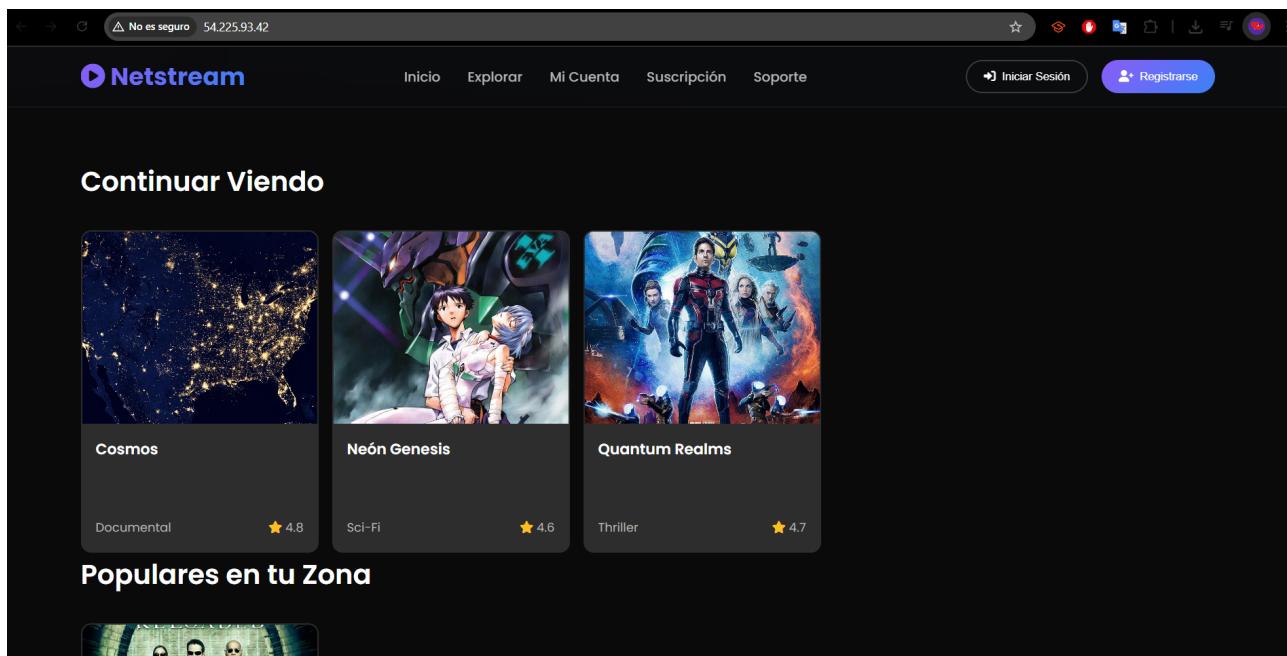
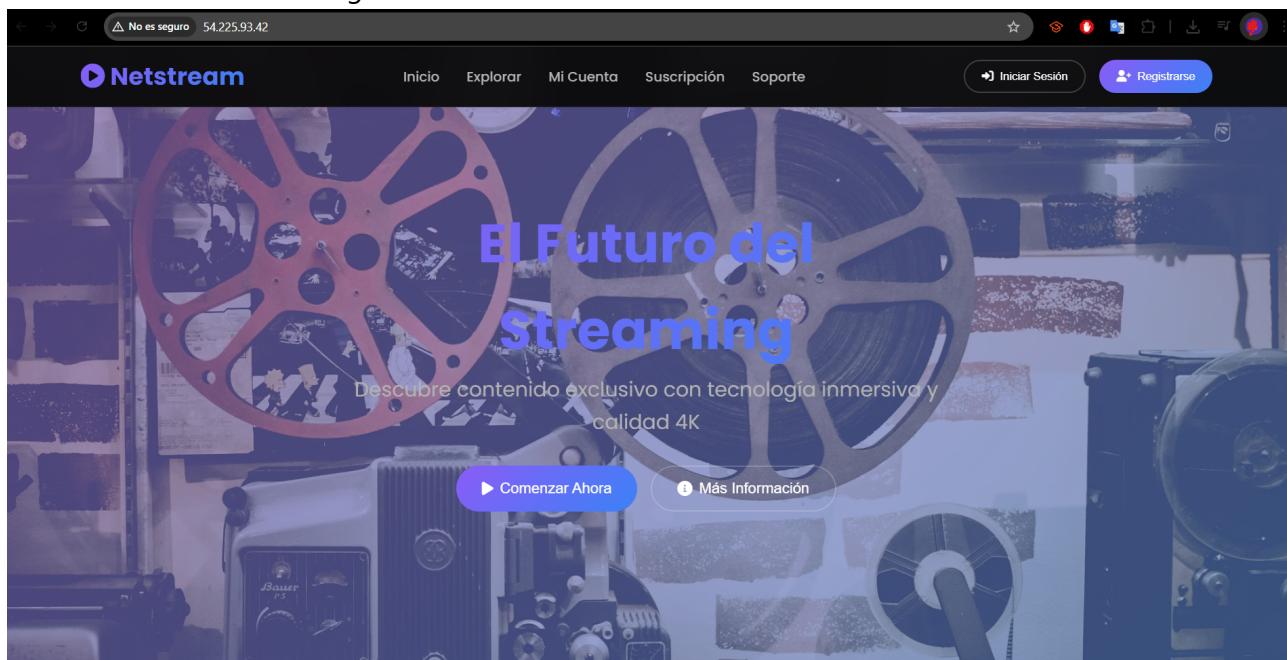
Esta arquitectura cumple con los objetivos del módulo DevOps, integrando despliegue continuo, validaciones automáticas, seguridad y escalabilidad.

🔒 Seguridad y buenas prácticas

- Uso estricto de GitHub Secrets.
- Claves PEM y configuraciones privadas excluidas del repositorio.
- `.gitignore` actualizado para prevenir exposición.

🖼 Capturas de la Plataforma Desplegada

1. Vista de Netstream en navegador desde ALB:



Netstream

Inicio Explorar Mi Cuenta Suscripción Soporte Iniciar Sesión Registrarse

Populares en tu Zona



Matrix Reloaded

Sci-Fi ★ 4.9

Series Sci-Fi



Netstream

Inicio Explorar Mi Cuenta Suscripción Soporte Iniciar Sesión Registrarse



Vida Marina

Naturaleza ★ 4.9

Netstream

La plataforma de streaming del futuro con contenido exclusivo y tecnología inmersiva.

[f](#) [t](#) [i](#) [y](#)

Navegación

Inicio Explorar Mi Cuenta Suscripción

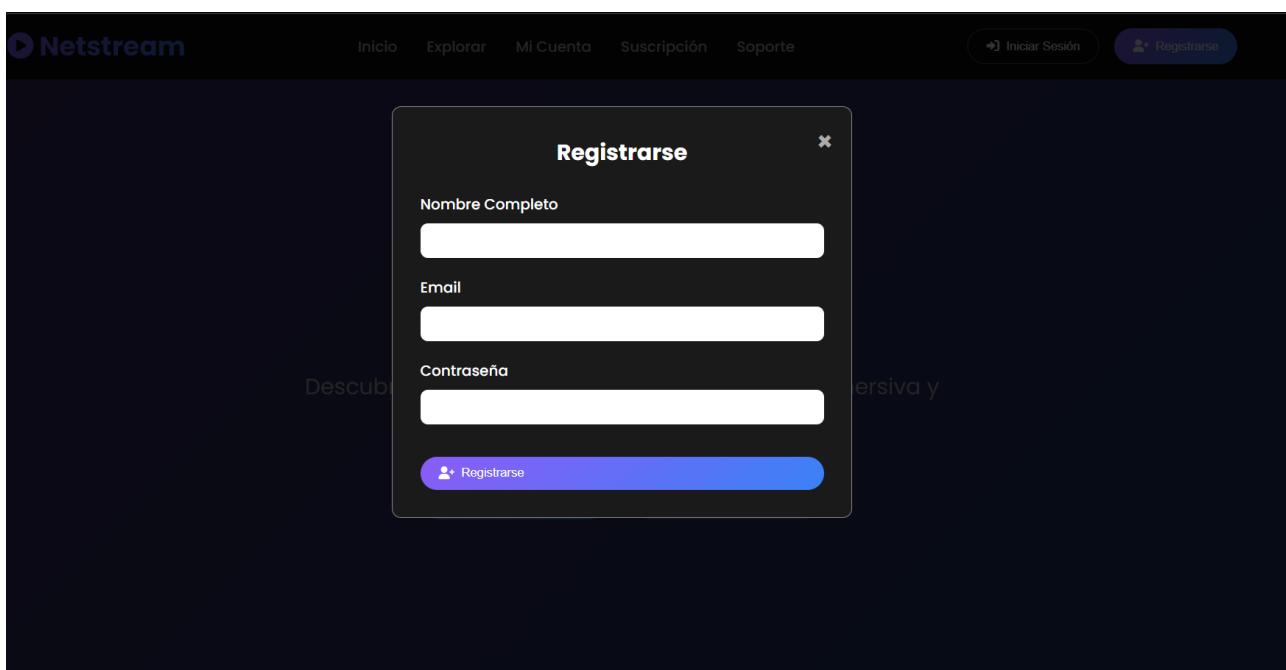
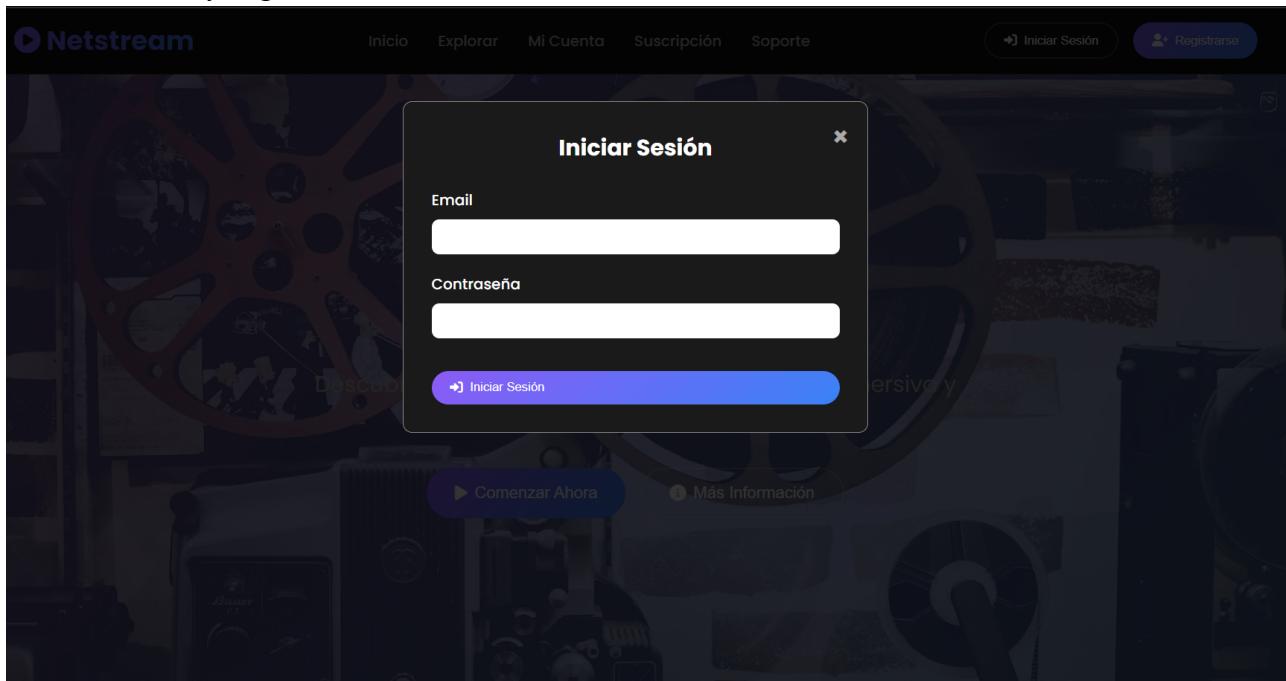
Soporte

Centro de Ayuda Contacto FAQ

Legal

Privacidad Términos Cookies [Español ▾](#)

2. Inicio de Sesión y Registro



🚀 Mejoras futuras

- Backend con autenticación y base de datos.
- Migración a arquitectura serverless (S3 + CloudFront).
- Integración de métricas con CloudWatch.
- HTTPS con ACM + redirección automática.

✍ Desarrollado como parte de un laboratorio de DevOps. Proyecto educativo con enfoque práctico y profesional.