

EJERCICIO 2: Crea un nuevo documento para esta práctica, llámalo P2_AWS_Route53_tunombre . Muestra en él una captura de pantalla de la instancia creada. Puede ser una captura de la ventana de la administración de instancias. Muestra también captura de pantalla de la pestaña de Detalles de la instancia creada.

<input type="checkbox"/>	UbuntuServer	i-04c30be3cab7fb2b6	En ejecución	t3.micro	3/3 comprobador	Ver alarmas +	us-east-1c	ec2-3-90-201-45.comp...	3.90.201.45	-
<input type="checkbox"/>	LinuxJuice	i-03f94057411bd154b	En ejecución	t3.micro	Inicializando	Ver alarmas +	us-east-1c	ec2-98-84-106-8.comp...	98.84.106.8	-
<input type="checkbox"/>	UbuntuCliente	i-0f447c62b9a5ae7b0	En ejecución	t3.micro	3/3 comprobador	Ver alarmas +	us-east-1c	ec2-98-84-173-143.co...	98.84.173.143	-

i-03f94057411bd154b (LinuxJuice)

Detalles

Estado y alarmas

Monitoreo

Seguridad

Redes

Almacenamiento

Etiquetas

▼ Resumen de instancia Información

ID de la instancia

i-03f94057411bd154b

Dirección IPv6

-

Tipo de nombre de anfitrión

Nombre de IP: ip-172-31-17-157.ec2.internal

Responder al nombre DNS de recurso privado

IPv4 (A)

Dirección IP asignada automáticamente

98.84.106.8 [IP pública]

Dirección IPv4 pública

98.84.106.8 | dirección abierta

Estado de la instancia

En ejecución

Nombre DNS de IP privada (solo IPv4)

ip-172-31-17-157.ec2.internal

Tipo de instancia

t3.micro

ID de VPC

vpc-085c6a2e9da2bdfb0

Direcciones IPv4 privadas

172.31.17.157

DNS público

ec2-98-84-106-8.compute-1.amazonaws.com | dirección abierta

Direcciones IP elásticas

-

Hallazgo de AWS Compute Optimizer

Suscribirse a AWS Compute Optimizer para recibir recomendaciones. | Más información

EJERCICIO 3: El Código que has pegado antes, deberías ya entender qué es lo que hace. Échale un vistazo y escribe en tu documento que crees que hace cada una de las líneas.

#!/bin/bash

Dice que el script debe ejecutarse con el intérprete de Bash

yum update -y

Actualiza todos los paquetes del sistema yum (gestor de paquetes de Amazon Linux) y la -y responde si a todas las confirmaciones

yum install -y docker

Instala docker en el sistema

service docker start

Inicia el sistema de docker

systemctl enable docker.service

Configura para que el docker se abra nada más se arranque el sistema

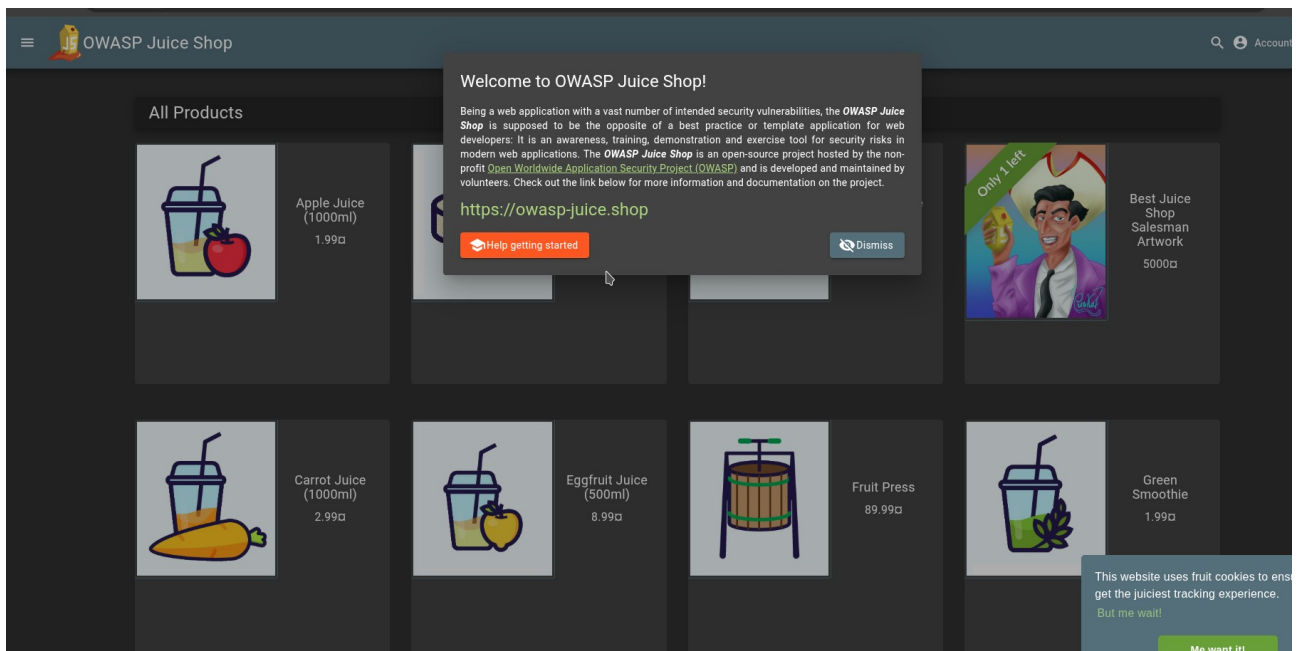
docker pull bkimminich/juice-shop

Descarga esa imagen desde docker hub

docker run -d -p 80:3000 bkimminich/juice-shop

Ejecuta la imagen descargada en un contenedor docker

EJERCICIO 4: Muestra captura de pantalla del contenido de la página desplegada (es decir, de lo que te ha mostrado el navegador).

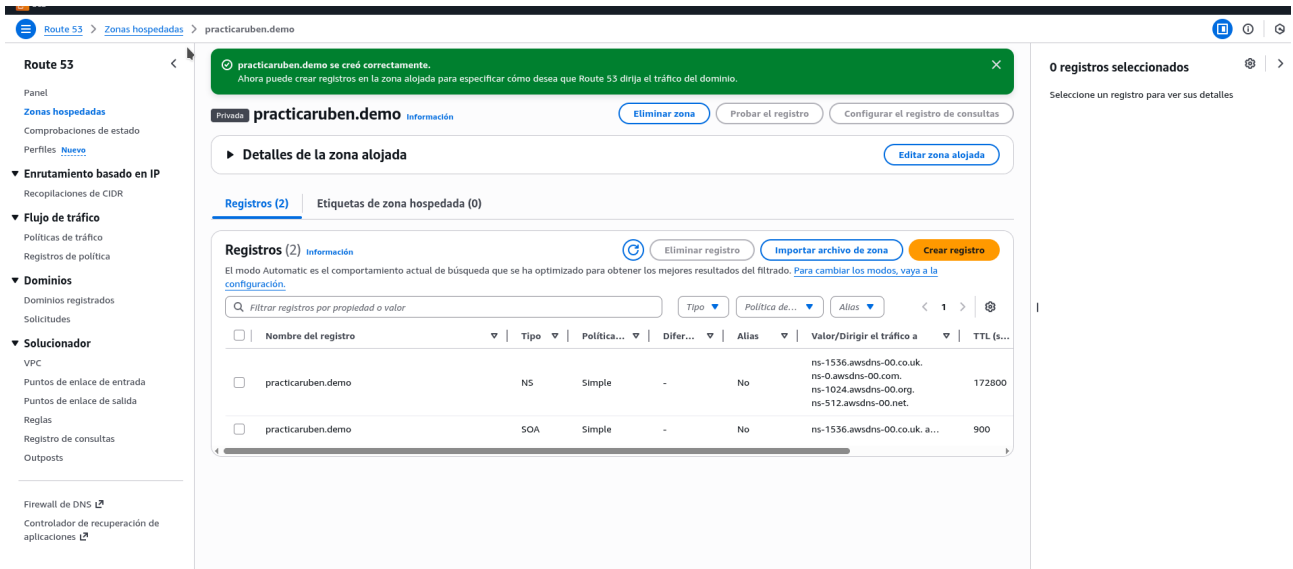


EJERCICIO 5: Muestra captura de pantalla de cómo te ha quedado la zona creada. (Imagen equivalente a la de arriba pero con tus datos) :

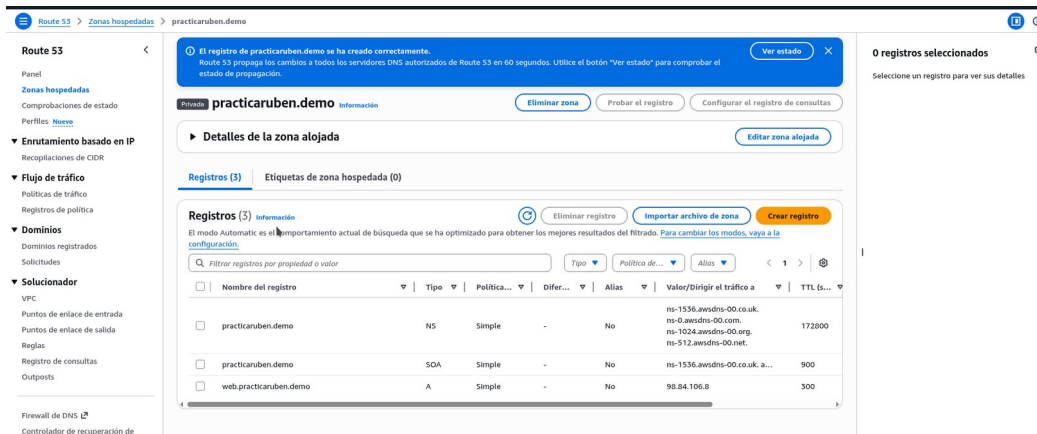
```
Last login: Thu Nov 6 08:11:00 2025 from 18.206.107.28
ubuntu@ip-172-31-18-56:~$ sudo curl 98.84.106.8
<!--
- Copyright (c) 2014-2025 Bjoern Kimminich & the OWASP Juice Shop contributors.
- SPDX-License-Identifier: MIT
-->

<!doctype html>
<html lang="en" data-beasties-container>
<head>
  <meta charset="utf-8">
  <title>OWASP Juice Shop</title>
  <meta name="description" content="Probably the most modern and sophisticated insecure web application">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link id="favicon" rel="icon" type="image/x-icon" href="assets/public/favicon_js.ico">
  <link rel="stylesheet" type="text/css" href="//cdnjs.cloudflare.com/ajax/libs/cookieconsent2/3.1.0/cookieconsent.min.css">
  <script src="//cdnjs.cloudflare.com/ajax/libs/cookieconsent2/3.1.0/cookieconsent.min.js"></script>
  <script src="//cdnjs.cloudflare.com/ajax/libs/jquery/2.2.4/jquery.min.js"></script>
  <script>
    window.addEventListener("load", function(){
      window.cookieconsent.initialise({
        "palette": {
          "popup": { "background": "var(--theme-primary)", "text": "var(--theme-text)" },
          "button": { "background": "var(--theme-accent)", "text": "var(--theme-text)" }
        },
        "theme": "classic",
        "position": "bottom-right",
        "content": { "message": "This website uses fruit cookies to ensure you get the juiciest tracking experience.", "dismiss":
9PnbKL3wuH4" }
      });
    });
  </script>
```

El curl simula la página web pero no se ve porque no tengo interfaz gráfica



EJERCICIO 6: Muestra captura de pantalla de cómo te ha quedado el registro creado. (Imagen equivalente a la de arriba, pero con tus datos)



EJERCICIO 7: Muestra captura de pantalla del resultado de tu comando curl. (Imagen equivalente al contenido del cuadro de arriba)

```
ubuntu@ip-172-31-18-56:~$ curl web.practicaruben.demo
<!--
  ~ Copyright (c) 2014-2025 Bjoern Kimminich & the OWASP Juice Shop contributors.
  ~ SPDX-License-Identifier: MIT
-->

<!doctype html>
<html lang="en" data-beasties-container>
<head>
  <meta charset="utf-8">
  <title>OWASP Juice Shop</title>
  <meta name="description" content="Probably the most modern and sophisticated insecure web application">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link id="favicon" rel="icon" type="image/x-icon" href="/assets/public/favicon_js.ico">
  <link rel="stylesheet" type="text/css" href="/cdnjs.cloudflare.com/ajax/libs/cookieconsent2/3.1.0/cookieconsent.min.css">
  <script src="/cdnjs.cloudflare.com/ajax/libs/cookieconsent2/3.1.0/cookieconsent.min.js"></script>
  <script src="/cdnjs.cloudflare.com/ajax/libs/jquery/2.2.4/jquery.min.js"></script>
  <script>
    window.addEventListener("load", function(){
      window.cookieconsent.initialise({
        "palette": {
          "popup": { "background": "var(--theme-primary)", "text": "var(--theme-text)" },
          "button": { "background": "var(--theme-accent)", "text": "var(--theme-text)" }
        },
        "theme": "classic",
        "position": "bottom-right",
        "content": { "message": "This website uses fruit cookies to ensure you get the juiciest tracking experience.", "dismiss": "Me want it!", "link": "But me wait!", "href": "https://www.youtube.co
@PnbKL3wH4" }
      });
    });
  </script>
```

EJERCICIO 8: Haz un resumen de lo que has hecho en esta práctica, a modo de conclusión y para demostrar que has comprendido el objetivo de la práctica.

Primero hemos creado una instancia con ip pública con un docker que nos ofrece AWS, por tanto lo podemos ver en google aunque avisa que no es seguro, luego con el comando curl que accede a la página pero al no tener acceso gráfico, lo muestra el html.

Después con route 53 se crea una zona privada, donde crearemos una zona de dominio y un registro que apunta a la ip pública, por último utilizamos curl pero poniendo el dominio creado