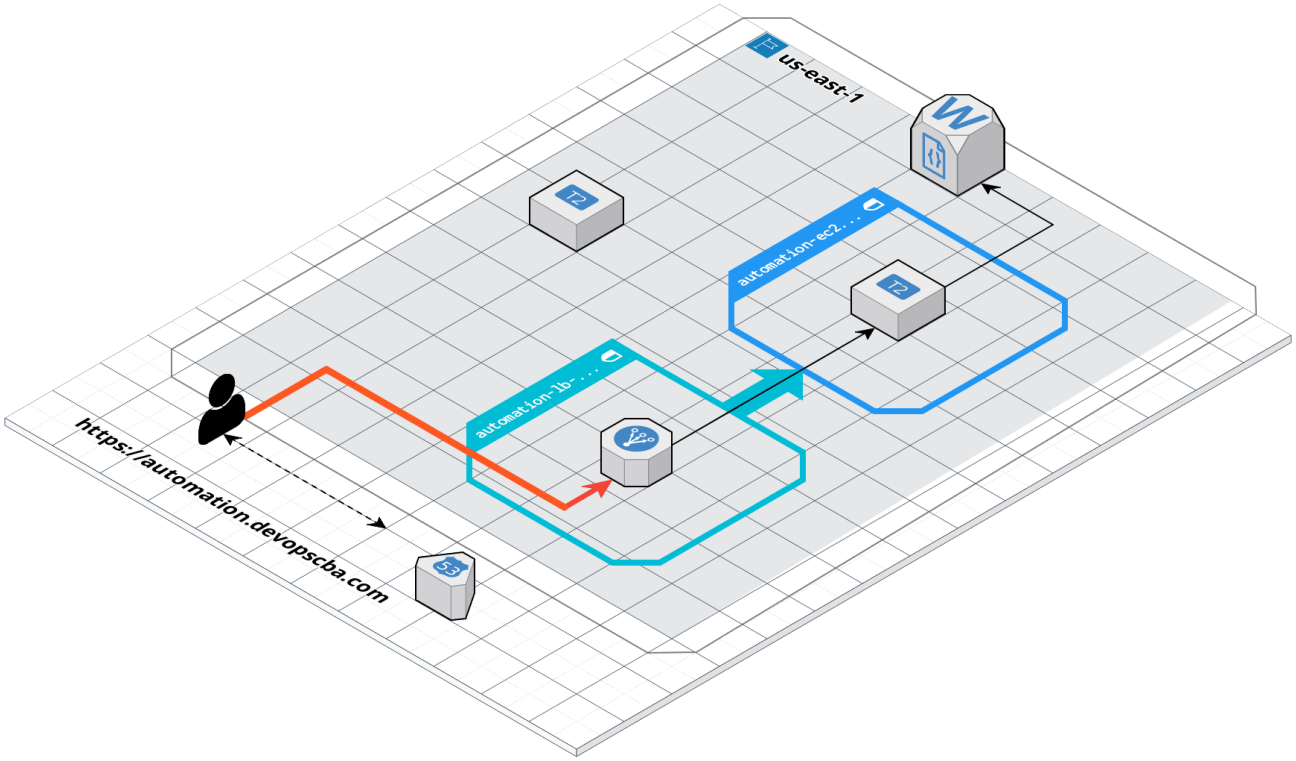
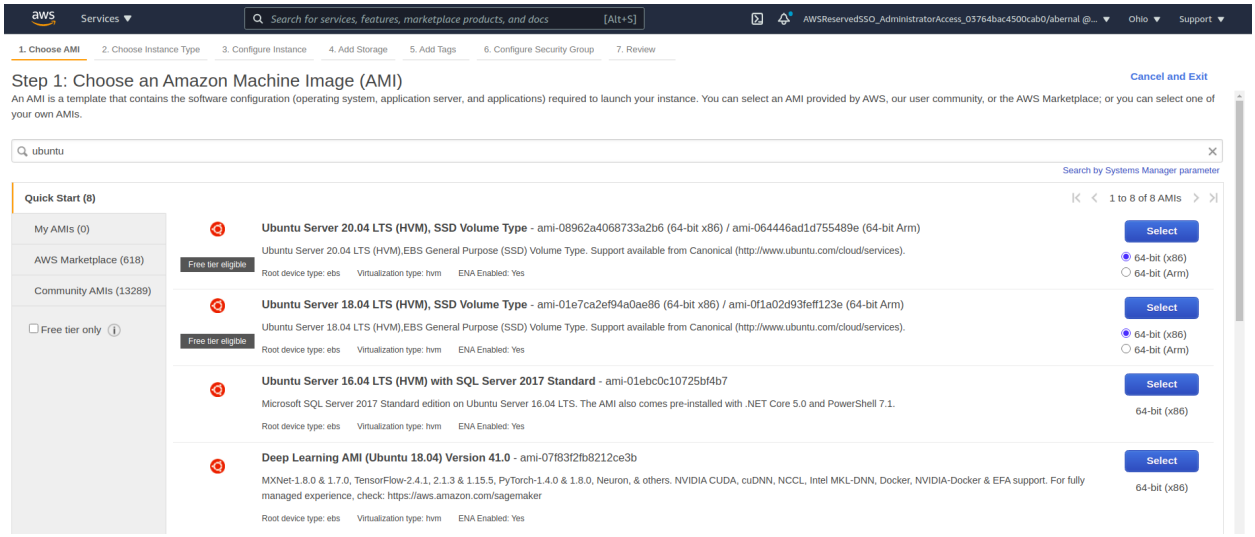
Repaso de Las Pruebas de Automatización

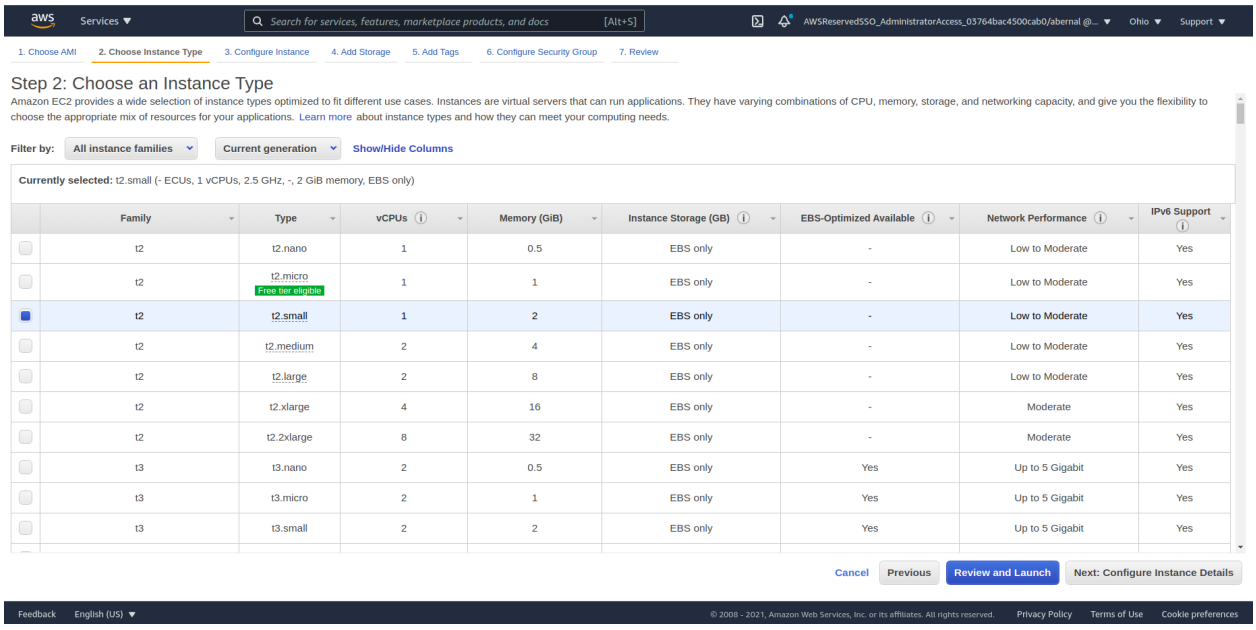
Diagrama de Componentes



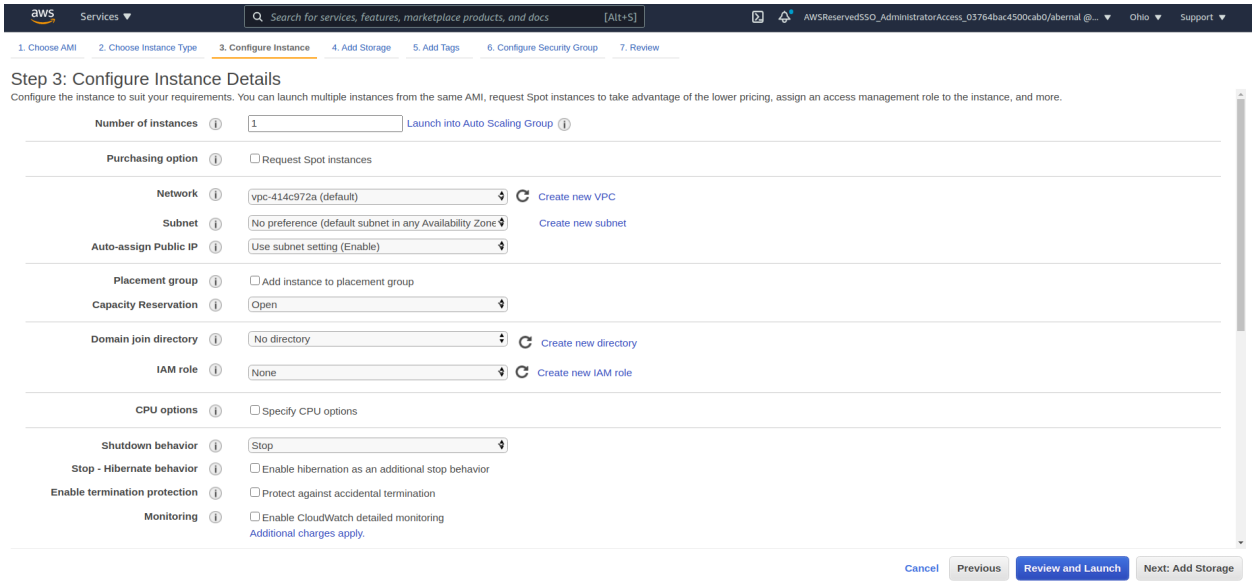
1. Crear una máquina de EC2 en la cuenta de AWS



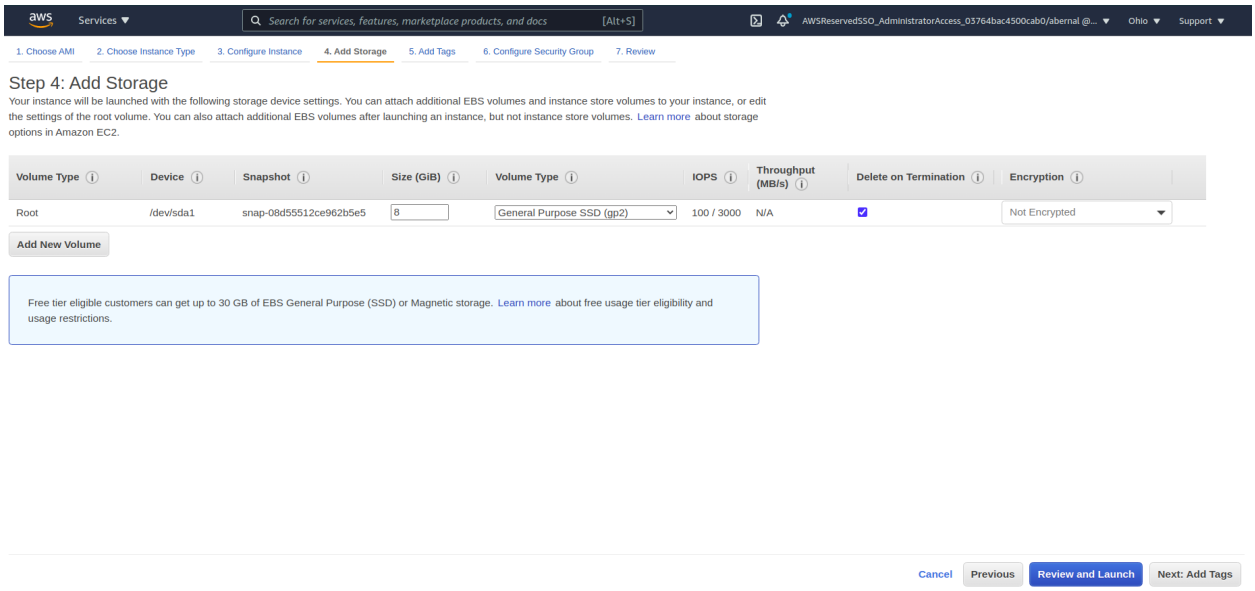
Del tipo t2 small



Con las configuraciones de red por defecto

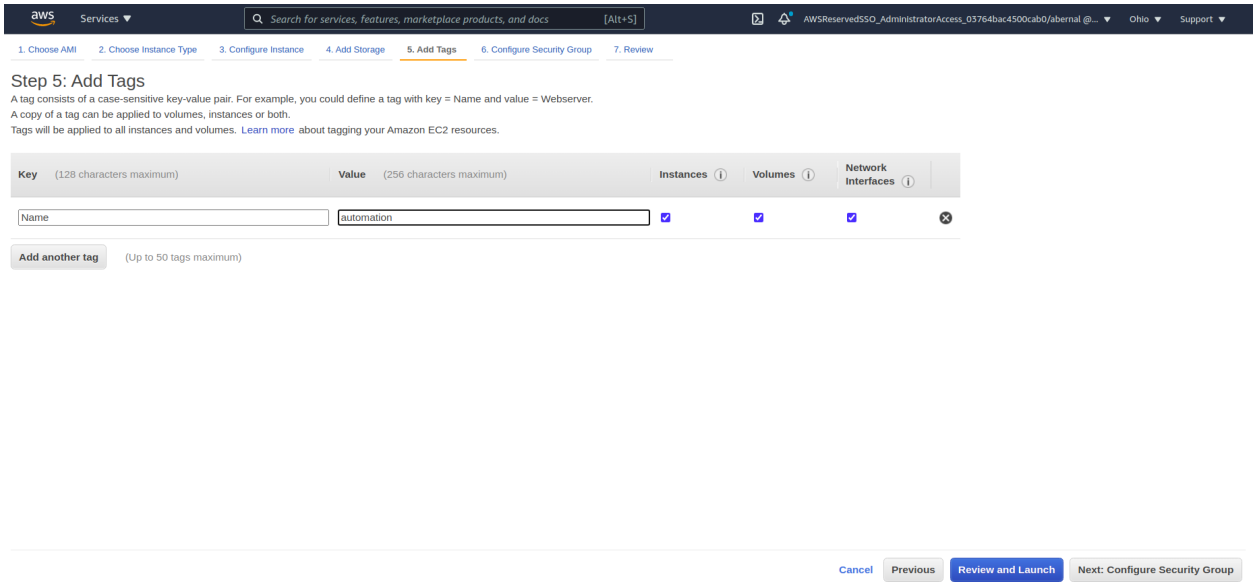


Con un disco de 8 GB

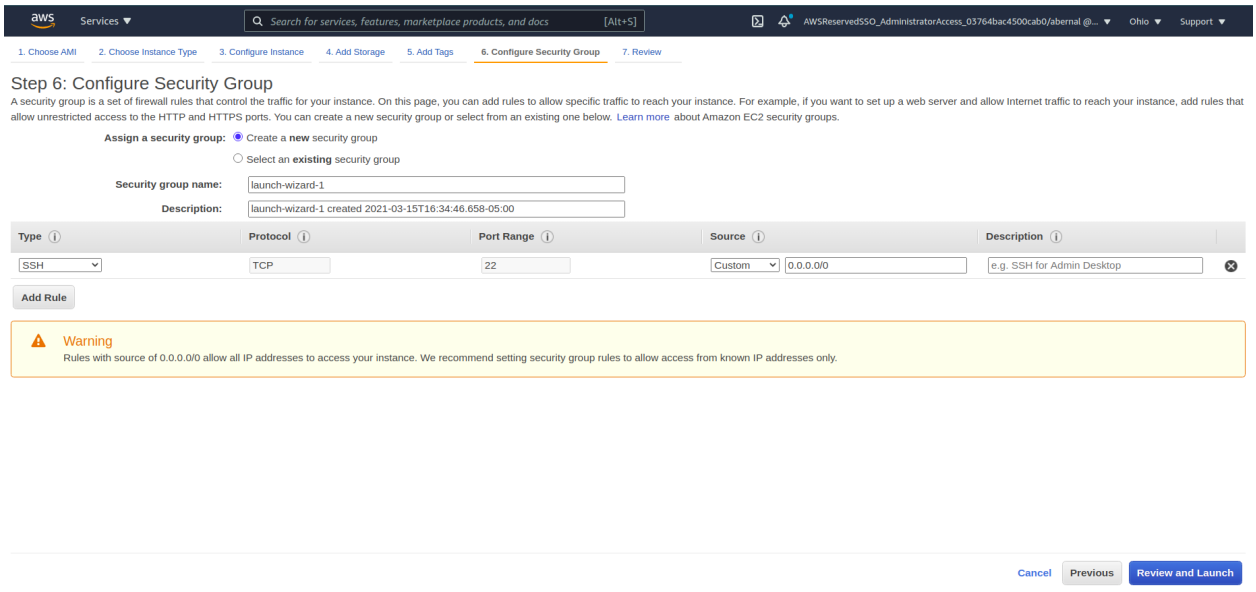


Con el tag

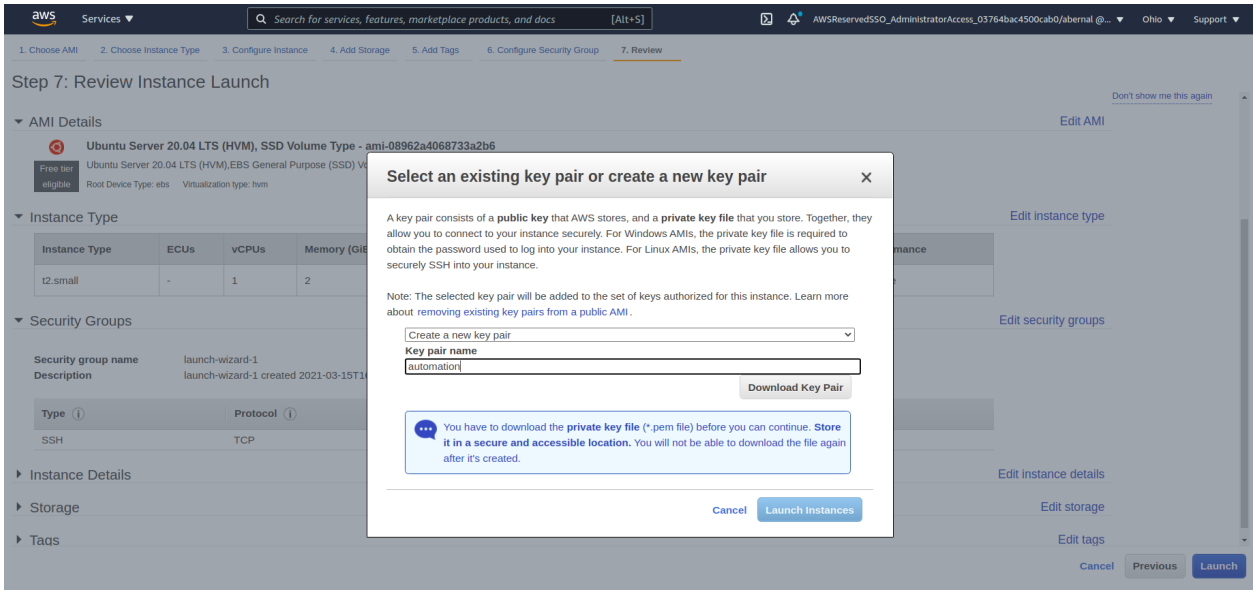
Name : Automation



Con un nuevo grupo de seguridad que habilite el acceso por el puerto 22 desde cualquier red



Cuando creemos la máquina asegurémonos de crear una llave de ssh con el nombre de automation



Descargar la llave de ssh en la máquina local

2. Ingresar a la máquina de EC2

```
ssh -i "automation.pem"
ubuntu@ec2-3-142-153-102.us-east-2.compute.amazonaws.com
```

3. Clonar el repositorio

Clonar el siguiente repositorio

- https://github.com/borjas-prodolliet/automation

Ejecutando el siguiente comando

```
git clone https://github.com/borjas-prodolliet/automation
```

4. Entrar en el repositorio

Ejecutar el siguiente comando

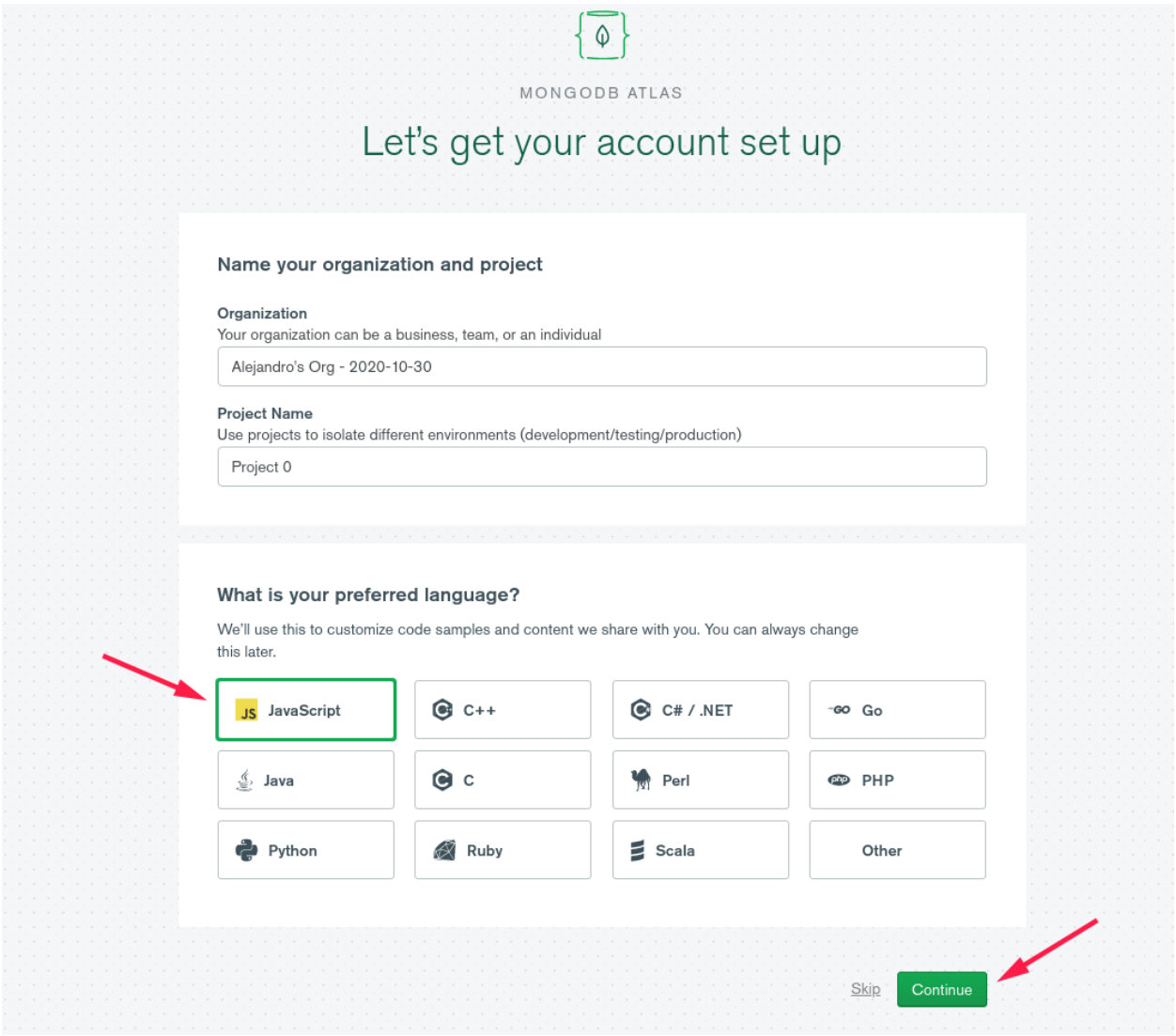
```
cd automation
```

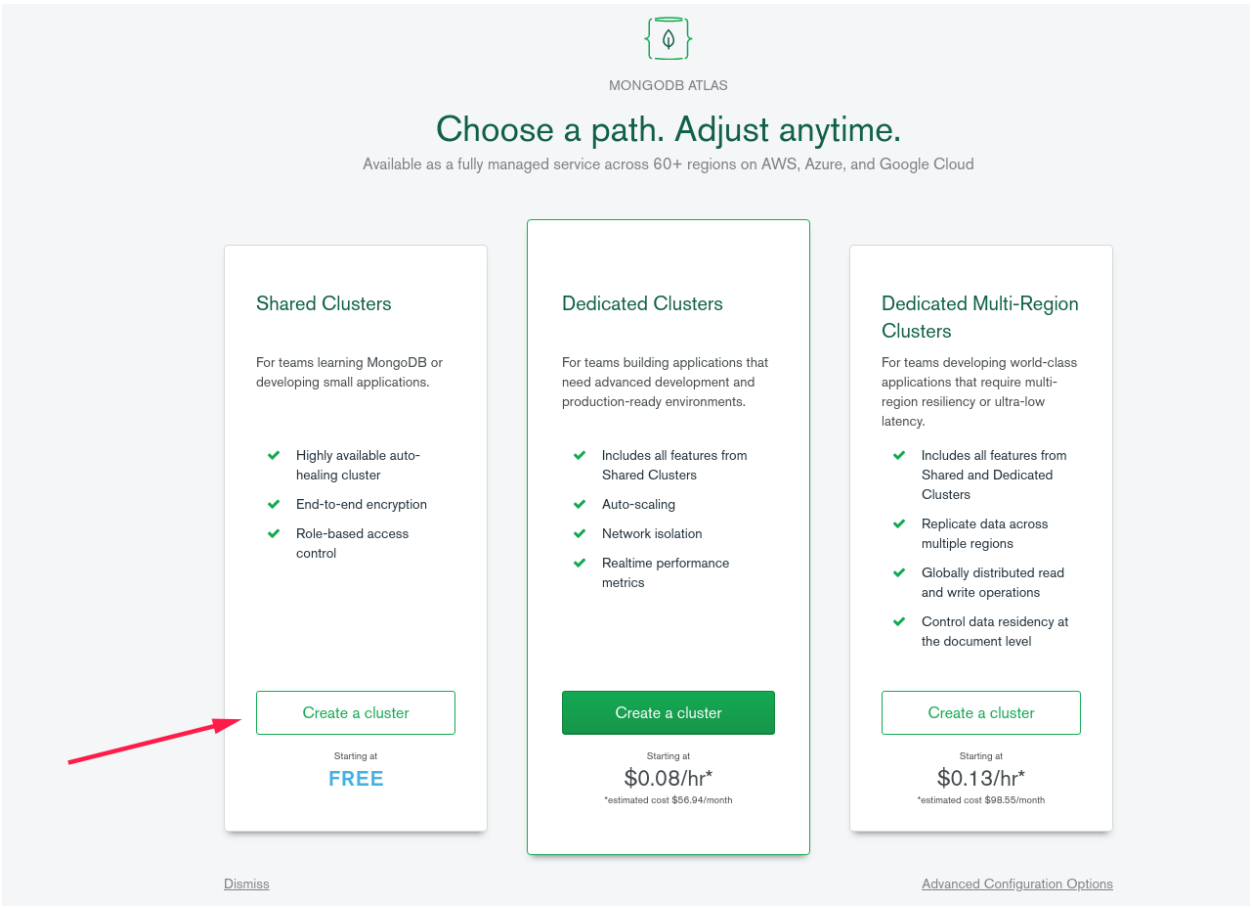
5. Configurar los datos de conexión a la base de datos

Creando nuestra base de datos en Atlas

Ir a https://account.mongodb.com/account/register

→ Registrarse


MONGODB ATLAS
Let's get your account set up
Name your organization and project
Organization
Your organization can be a business, team, or an individual
Alejandro's Org - 2020-10-30
Project Name
Use projects to isolate different environments (development/testing/production)
Project 0
What is your preferred language?
We'll use this to customize code samples and content we share with you. You can always change this later.
JavaScript
C++
C# / .NET
Go
Java
C
Perl
PHP
Python
Ruby
Scala
Other
Skip
Continue


MONGODB ATLAS
Choose a path. Adjust anytime.
Available as a fully managed service across 60+ regions on AWS, Azure, and Google Cloud
Shared Clusters
For teams learning MongoDB or developing small applications.
Highly available auto-healing cluster
End-to-end encryption
Role-based access control
Create a cluster
Starting at
FREE
Dedicated Clusters
For teams building applications that need advanced development and production-ready environments.
Includes all features from Shared Clusters
Auto-scaling
Network isolation
Realtime performance metrics
Create a cluster
Starting at
$0.08/hr*
*estimated cost $56.94/month
Dedicated Multi-Region Clusters
For teams developing world-class applications that require multi-region resiliency or ultra-low latency.
Includes all features from Shared and Dedicated Clusters
Replicate data across multiple regions
Globally distributed read and write operations
Control data residency at the document level
Create a cluster
Starting at
$0.13/hr*
*estimated cost $98.55/month
Dismiss
Advanced Configuration Options

CLUSTERS > CREATE A STARTER CLUSTER

Create a Starter Cluster

Welcome to MongoDB Atlas! We've recommended some of our most popular options, but feel free to customize your cluster to your needs. For more information, check our documentation.

Cloud Provider & Region

AWS, N. Virginia (us-east-1)

aws | Google Cloud | Azure

★ Recommended region

NORTH AMERICA	ASIA	EUROPE
N. Virginia (us-east-1) ★	Singapore (ap-southeast-1) ★	Frankfurt (eu-central-1) ★
Oregon (us-west-2) ★	Mumbai (ap-south-1)	Ireland (eu-west-1) ★
		AUSTRALIA
		Sydney (ap-southeast-2) ★

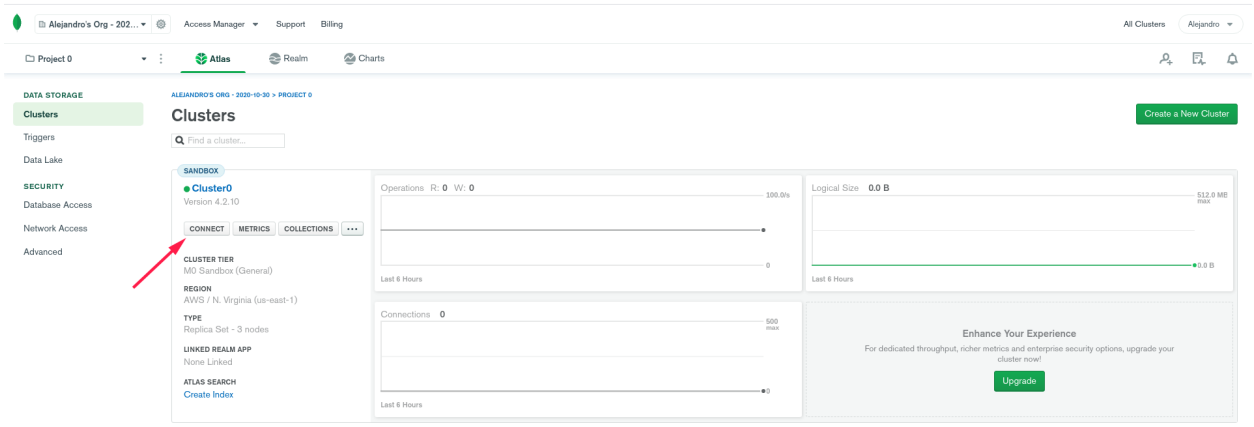
Cluster Tier

M0 Sandbox (Shared RAM, 512 MB Storage)
Encrypted

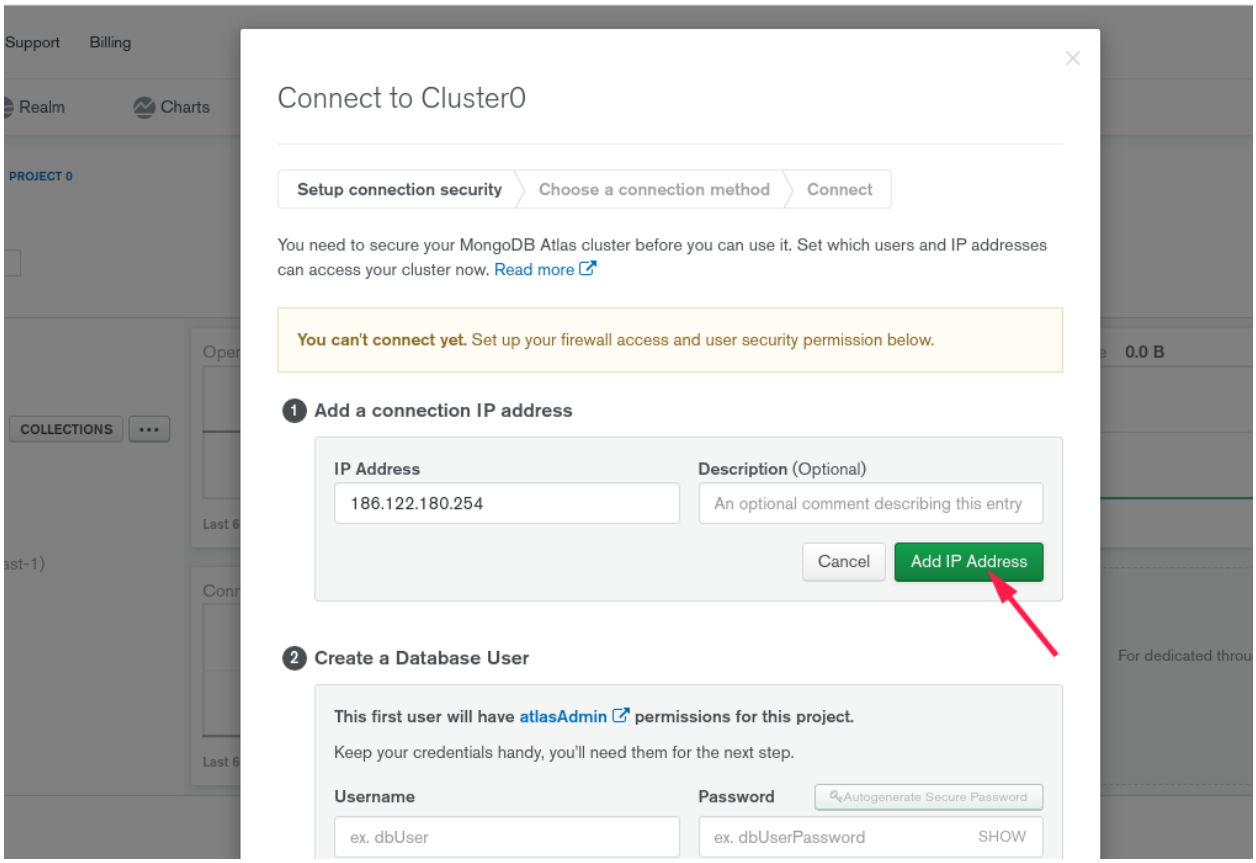
FREE

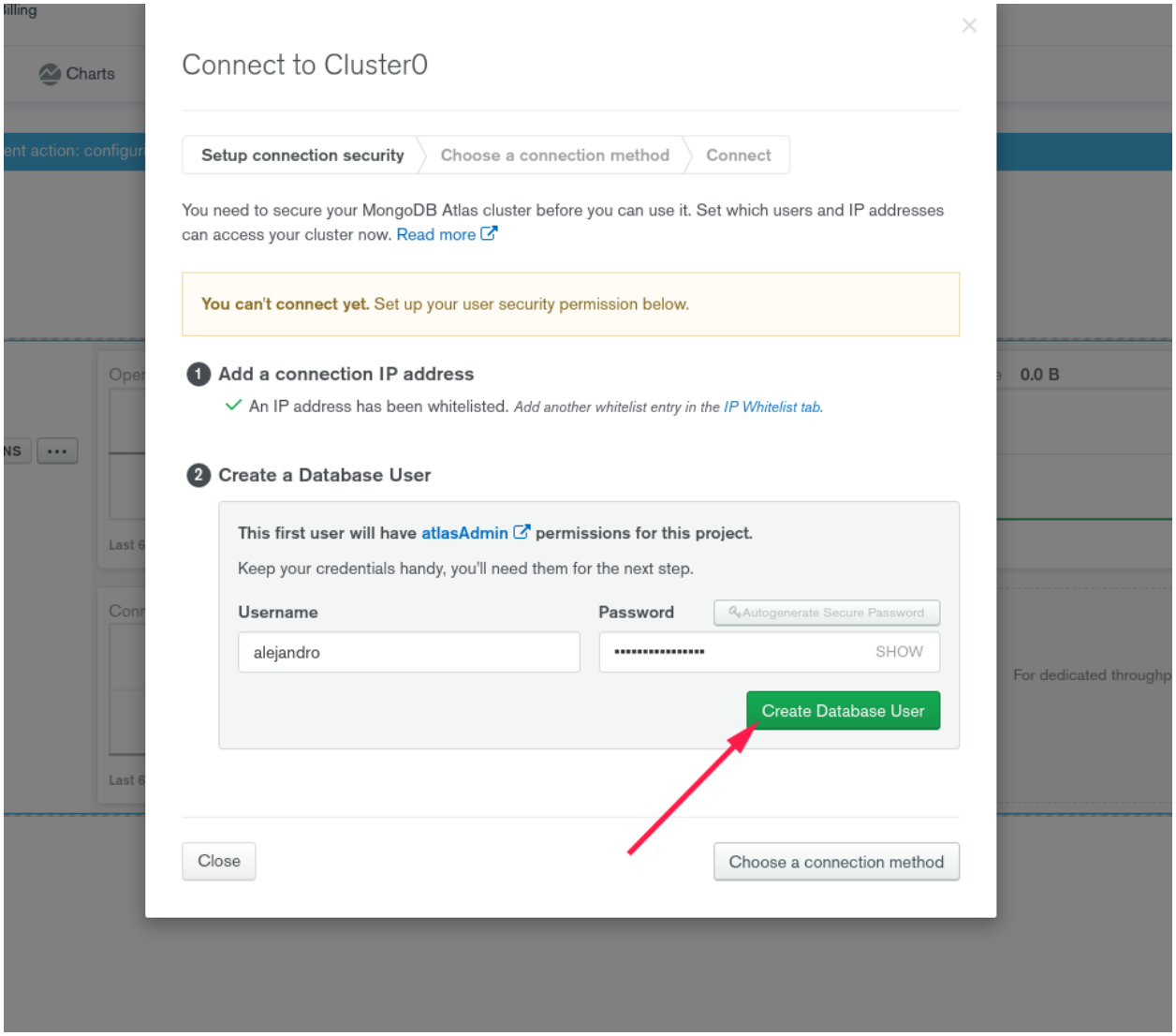
Free forever! *Your M0 cluster is ideal for experimenting in a limited sandbox. You can upgrade to a production cluster anytime.*

Back | Create Cluster



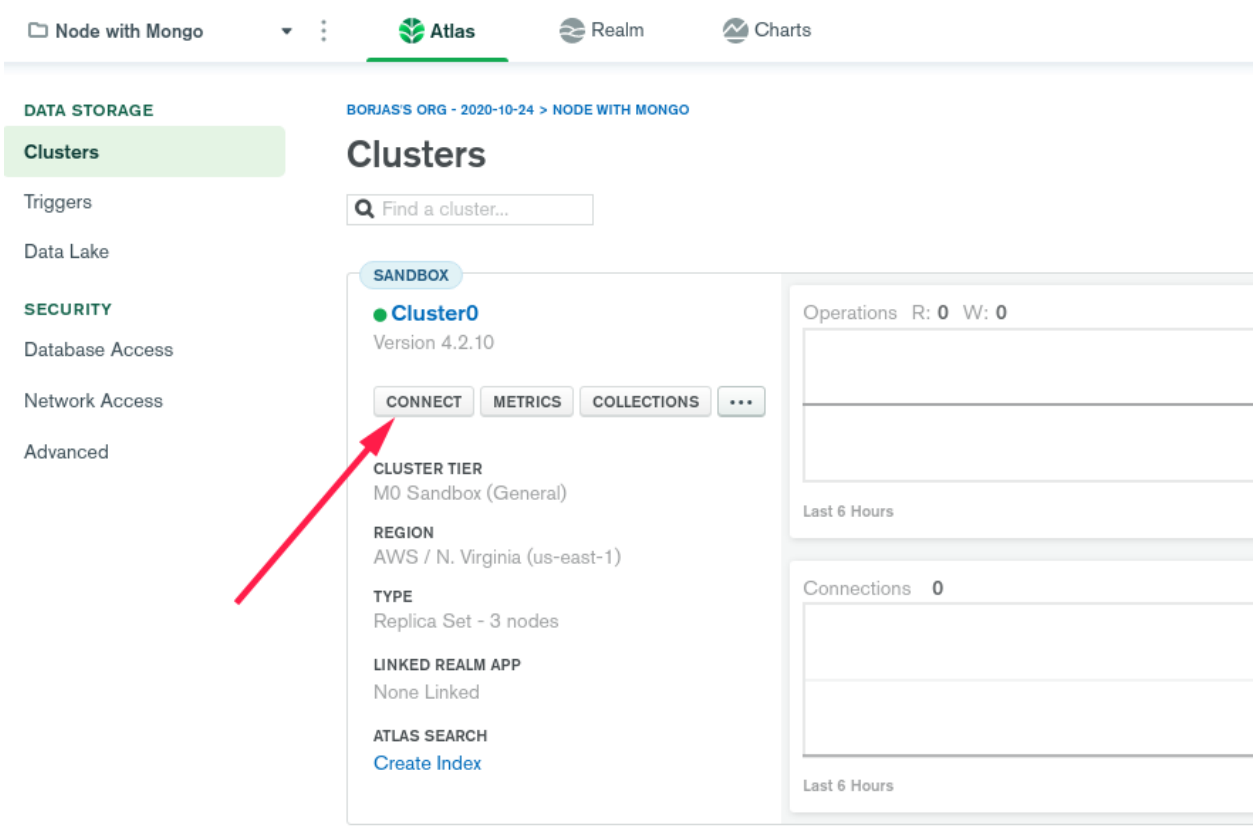
Conectándonos a nuestra base de datos

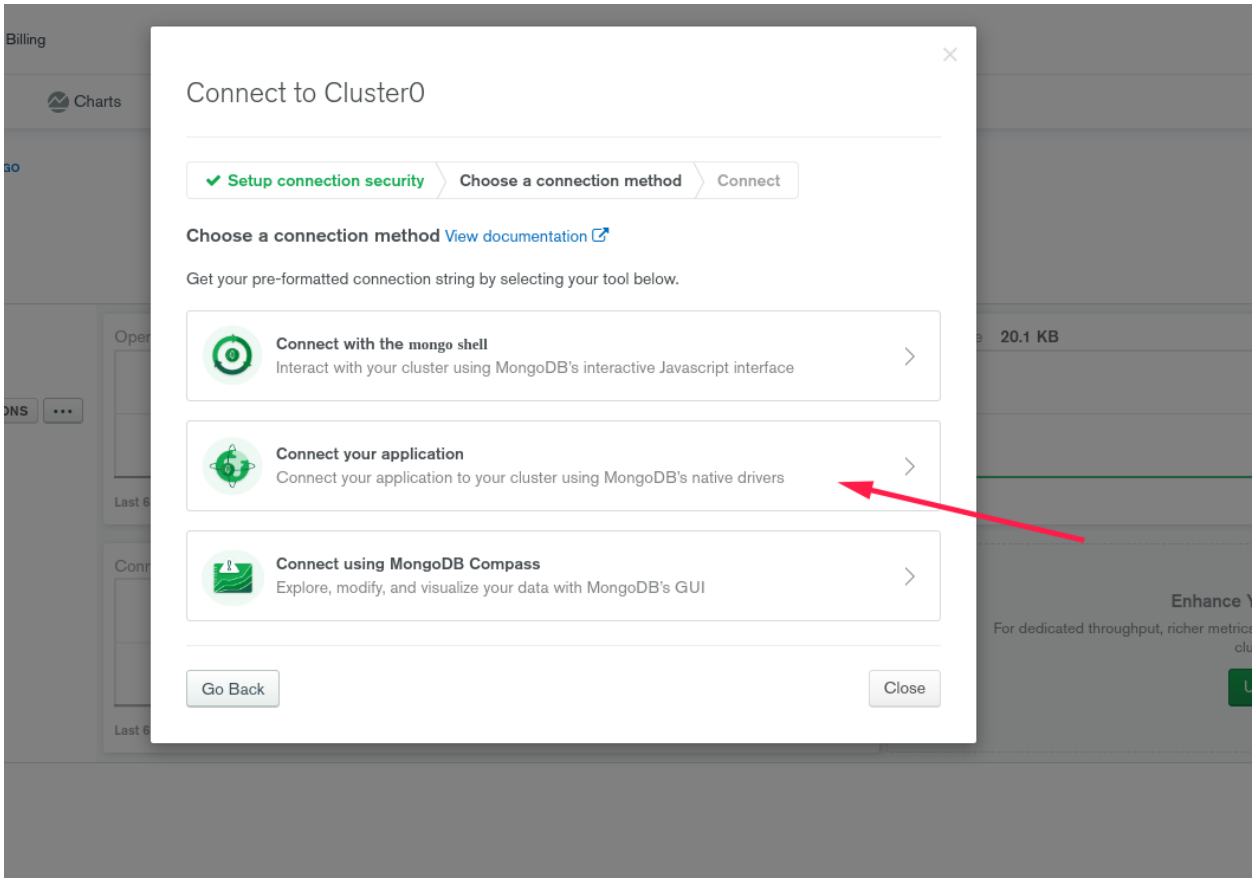


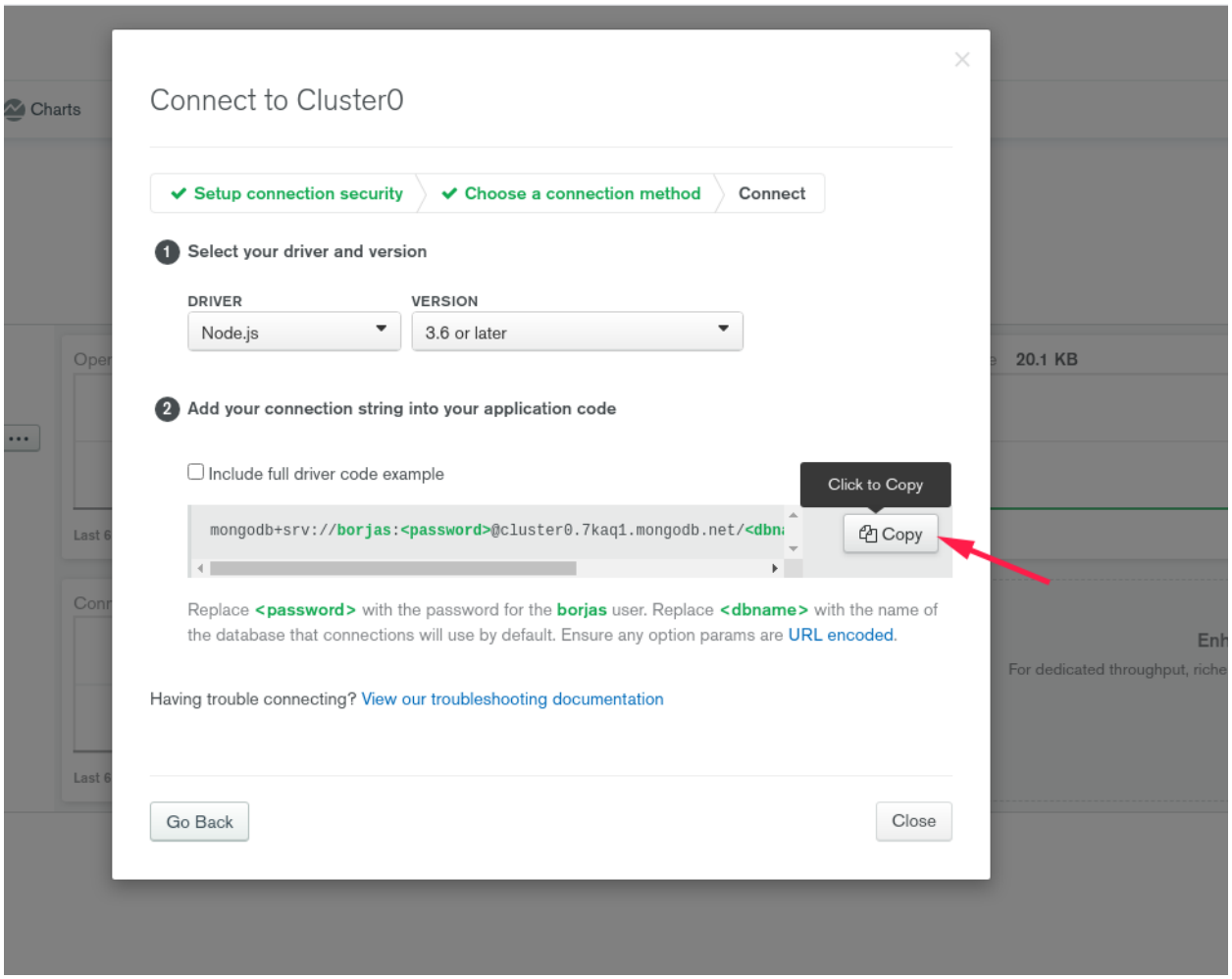


Este usuario y contraseña usaremos luego para conectarnos a la base.

Obtener el string de conexión


Node with Mongo
Atlas
Realm
Charts
DATA STORAGE
Clusters
Triggers
Data Lake
SECURITY
Database Access
Network Access
Advanced
BORJAS'S ORG - 2020-10-24 > NODE WITH MONGO
Clusters
Find a cluster...
SANDBOX
Cluster0
Version 4.2.10
CONNECT
METRICS
COLLECTIONS
CLUSTER TIER
M0 Sandbox (General)
REGION
AWS / N. Virginia (us-east-1)
TYPE
Replica Set - 3 nodes
LINKED REALM APP
None Linked
ATLAS SEARCH
Create Index
Operations R: 0 W: 0
Last 6 Hours
Connections 0
Last 6 Hours


Billing
Charts
Connect to Cluster0
Setup connection security
Choose a connection method
Connect
Choose a connection method View documentation
Get your pre-formatted connection string by selecting your tool below.
Connect with the mongo shell
Interact with your cluster using MongoDB's interactive Javascript interface
Connect your application
Connect your application to your cluster using MongoDB's native drivers
Connect using MongoDB Compass
Explore, modify, and visualize your data with MongoDB's GUI
Go Back
Close
20.1 KB



Reemplazar en el archivo **config.env**, el string de conexión y el password de tu base de mongoDB Atlas.

config.env

```
NODE_ENV=development
PORT=3000

DATABASE=string de conexion aqui
DATABASE_PASSWORD=password de base de datos aqui

JWT_SECRET=my-very-ultra-secure-long-secret
JWT_EXPIRES_IN=90d
JWT_COOKIE_EXPIRES_IN=90

EMAIL_USERNAME=4b68cdbade7bc6
EMAIL_PASSWORD=820e534040c79d
EMAIL_HOST=smtp.mailtrap.io
EMAIL_PORT=25
```

6. Popular la base de datos

Ejecutar los siguientes comandos

Instalar las librerias necesarias

```
sudo apt update
sudo apt install npm -y
npm install
```

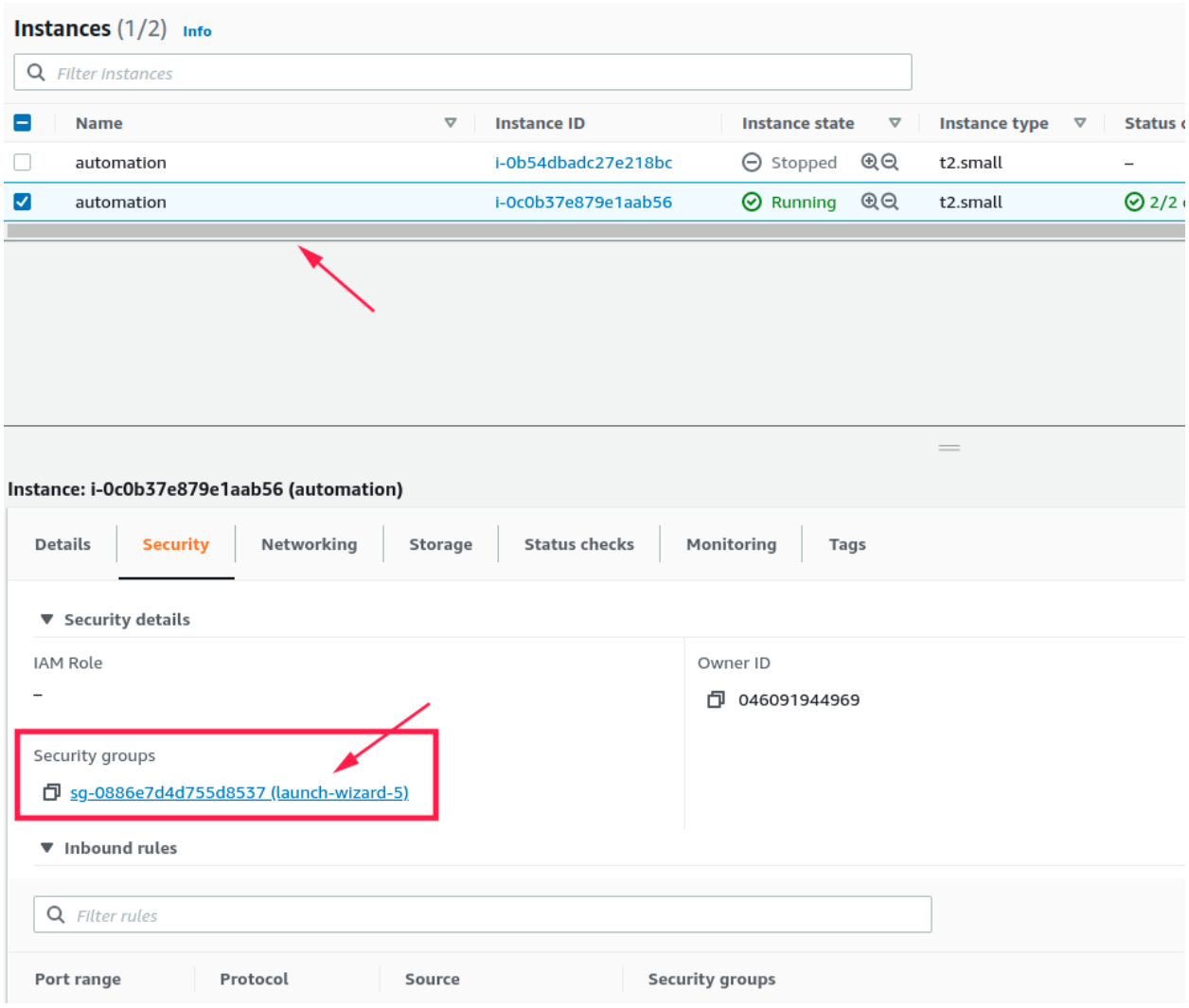
Popular la base de datos con el siguiente comando

```
node ./dev-data/data/import-dev-data.js --import
```

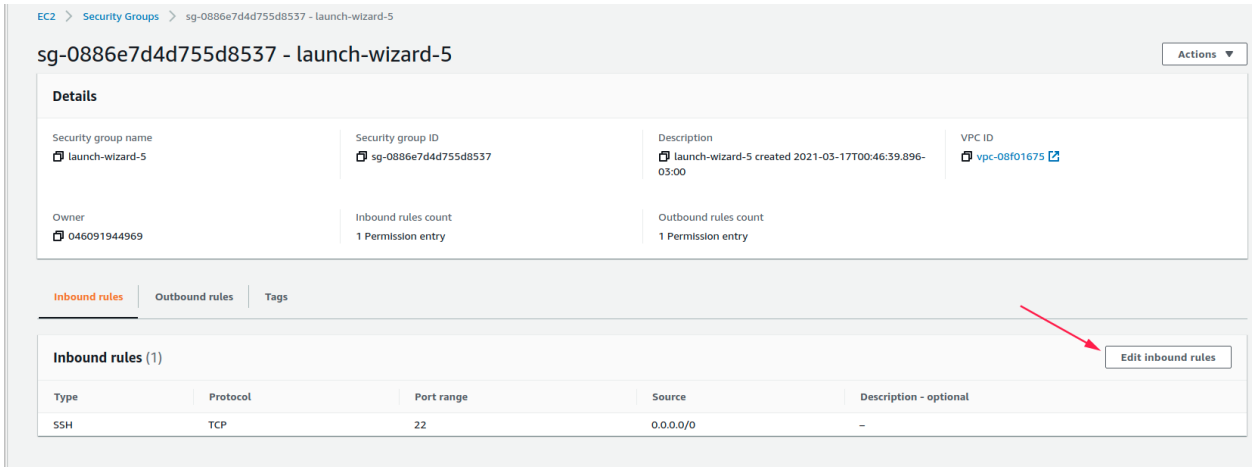
(Opcional) Para limpiar la base de datos ejecutar el siguiente comando

```
node ./dev-data/data/import-dev-data.js --delete
```

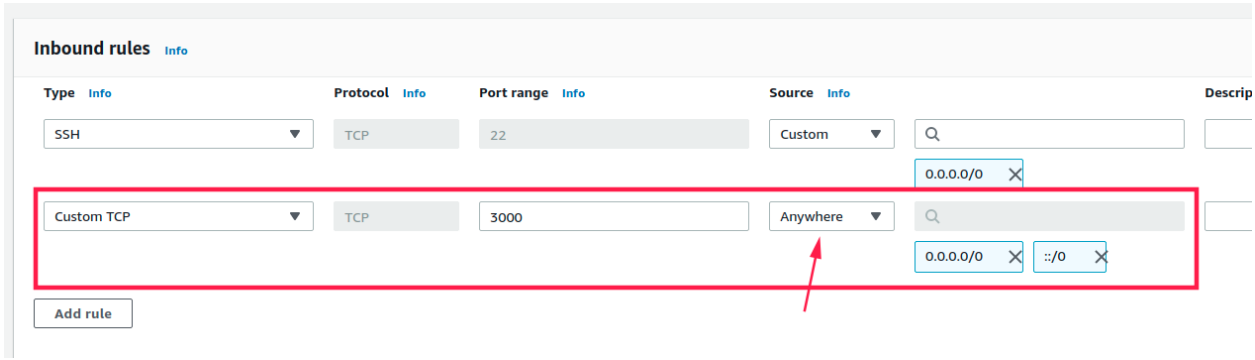
7. Modificar security group para permitir acceso a la máquina



Entramos en el security group



Luego editamos las inbound rules\



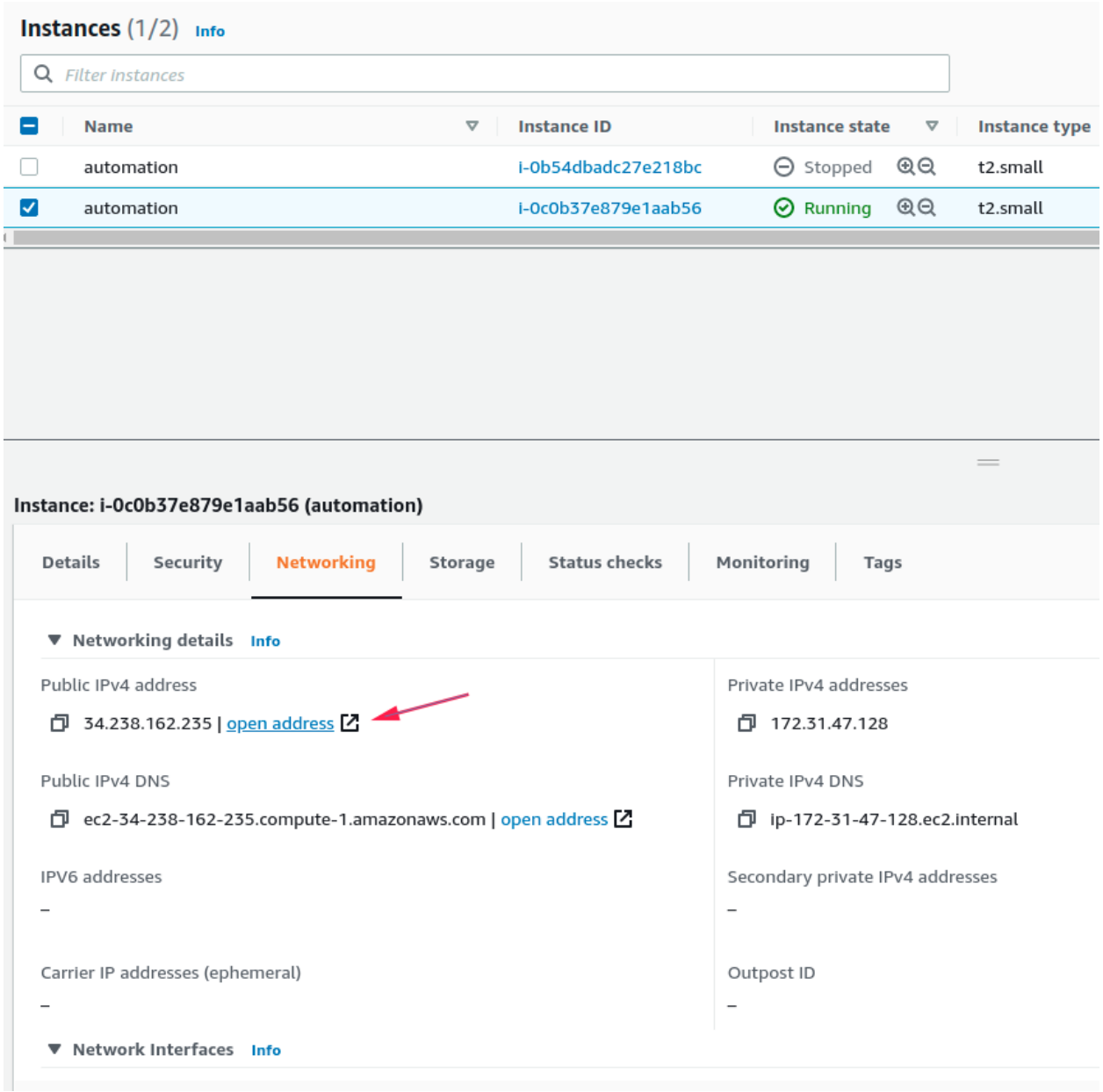
Agregamos una regla para acceder al puerto 3000 desde cualquier lado

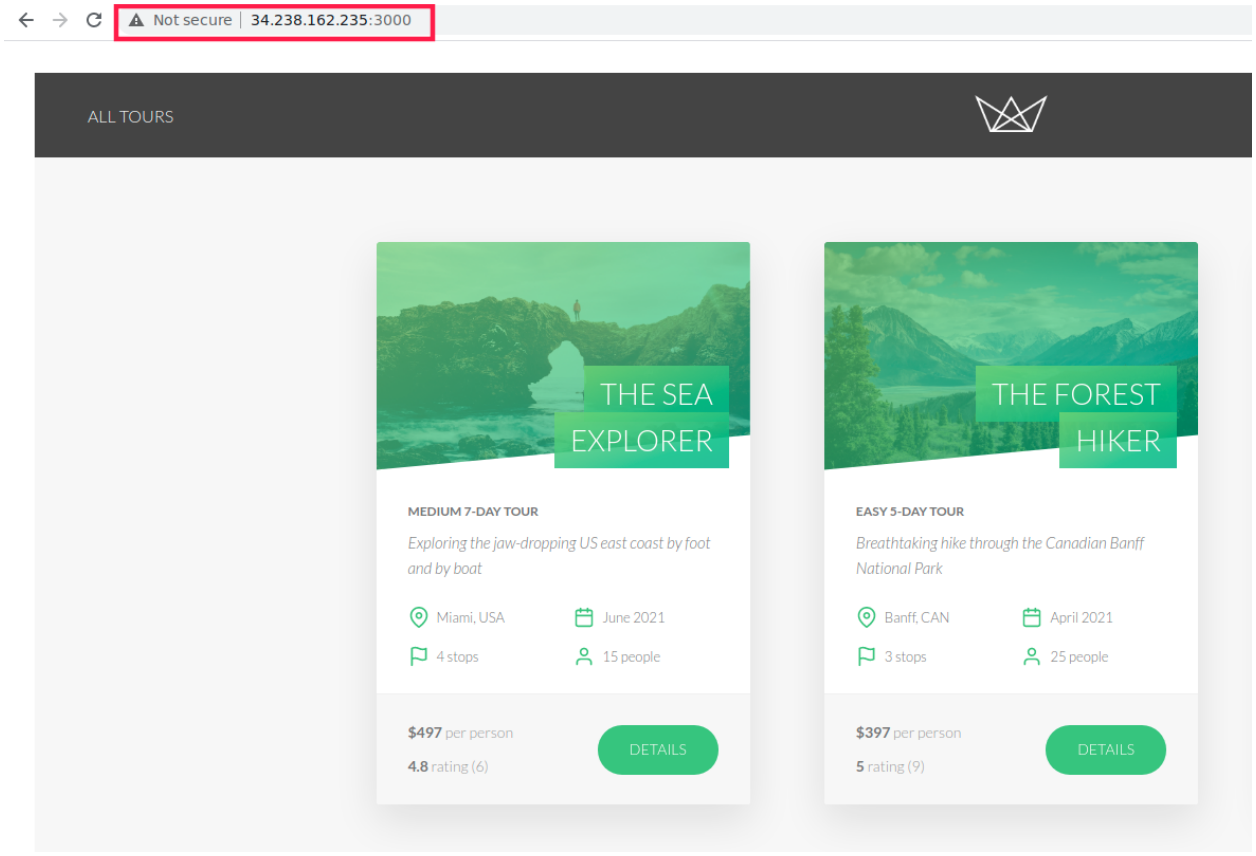
8. Correr y probar la aplicación

Ejecutar el siguiente comando

```
npm run start
```

Luego obtenemos la ip de la máquina y probamos que la aplicación esté funcionando:

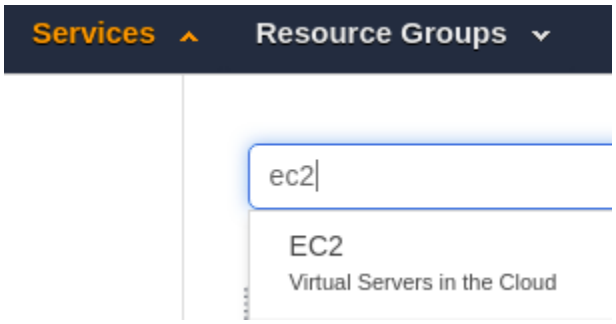

Instances (1/2) Info
Filter instances
Name
Instance ID
Instance state
Instance type
automation
i-0b54dbadc27e218bc
Stopped
t2.small
automation
i-0c0b37e879e1aab56
Running
t2.small
Instance: i-0c0b37e879e1aab56 (automation)
Details
Security
Networking
Storage
Status checks
Monitoring
Tags
Networking details Info
Public IPv4 address
34.238.162.235 | open address
Private IPv4 addresses
172.31.47.128
Public IPv4 DNS
ec2-34-238-162-235.compute-1.amazonaws.com | open address
Private IPv4 DNS
ip-172-31-47-128.ec2.internal
IPV6 addresses
–
Secondary private IPv4 addresses
–
Carrier IP addresses (ephemeral)
–
Outpost ID
–
Network Interfaces Info



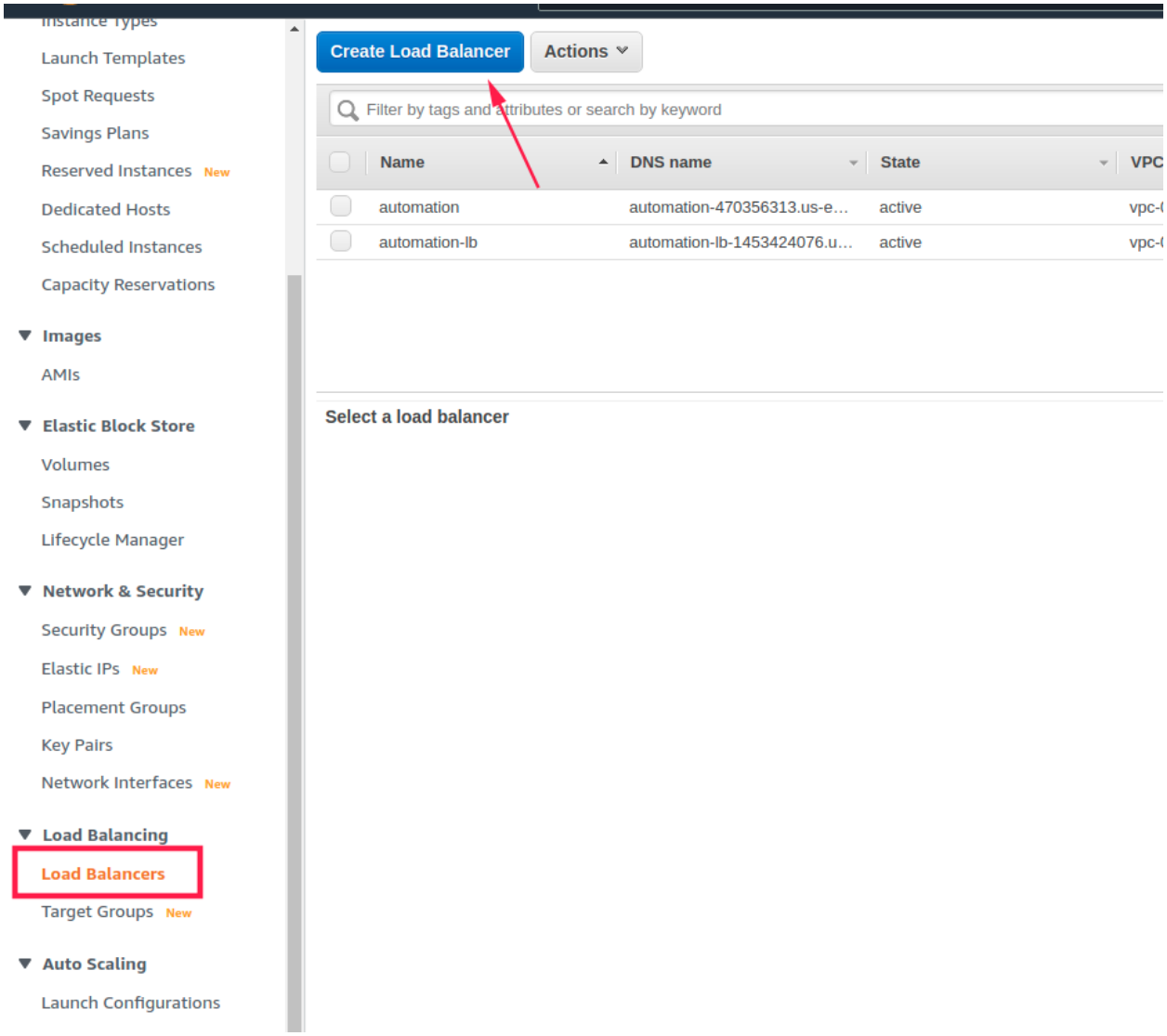
¡La aplicación está funcionando!

9. Crear Balanceador de Carga

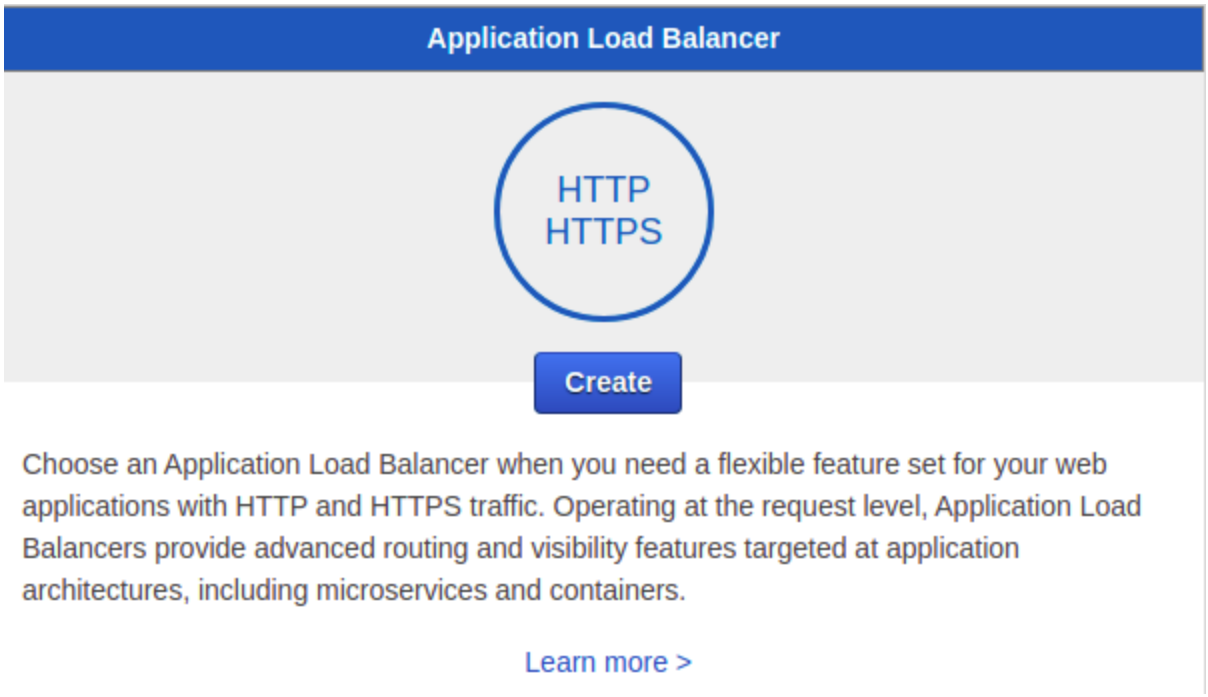
Buscar e ingresar al servicio de EC2 (Elastic Cloud Compute)



Loadbalancers “Create load balancer”



Seleccionar “Create Load Balancer”
Seleccionar “Application Load Balancer”



Step 1: Configure Load Balancer

1. Basic Configuration

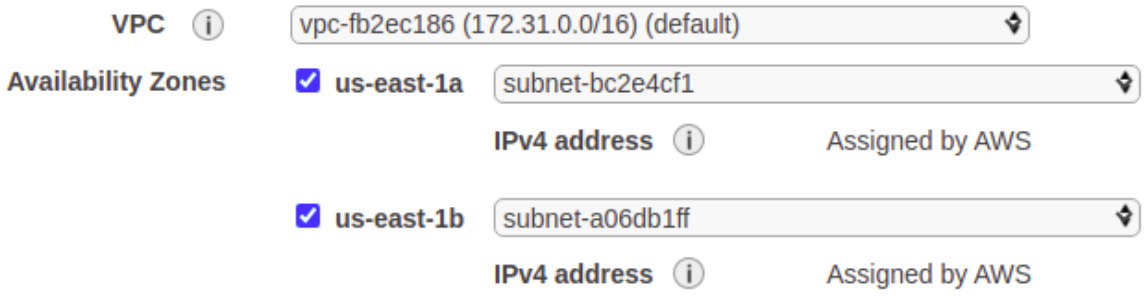
Name	automation-lb
Scheme	internet-facing
IP address type	ipv4

2. Listeners

Configurar 2 listeners HTTP y HTTPS

3. Availability Zones

Especificar al menos las zonas us-east-1a y us-east-1b

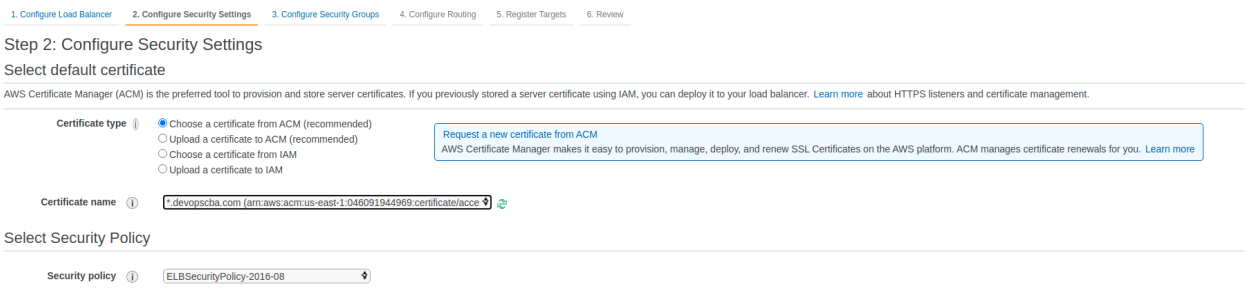


Next : Configure Security Settings

Step 2 Configure Security Settings

Seleccionar “Choose a certificate from ACM (recommended)”

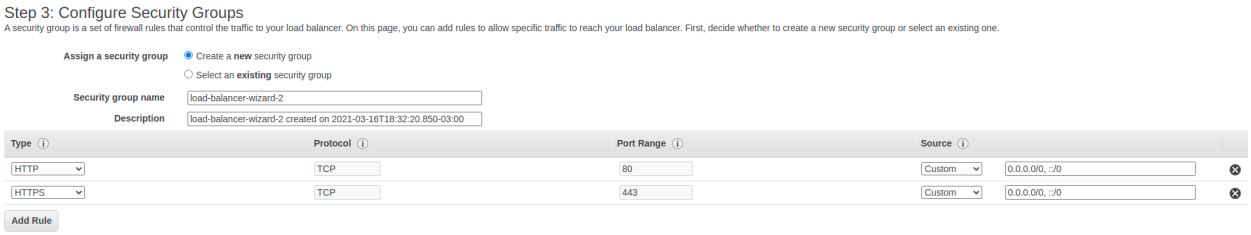
Luego seleccionar certificado *.devopscba.com



dar click en Next: Configure Security Groups

Step 3 : Configure Security Groups

“Create a new security group” y dejar valores por defecto



Dar click en Next: Configure Routing

Step 4 : Configure Routing

1. Target group

Target group	New target group
Name	automation-target-group
Target type	Instance
Protocol	HTTP
Port	3000

2. Health Checks

Protocol	HTTP
Path	/

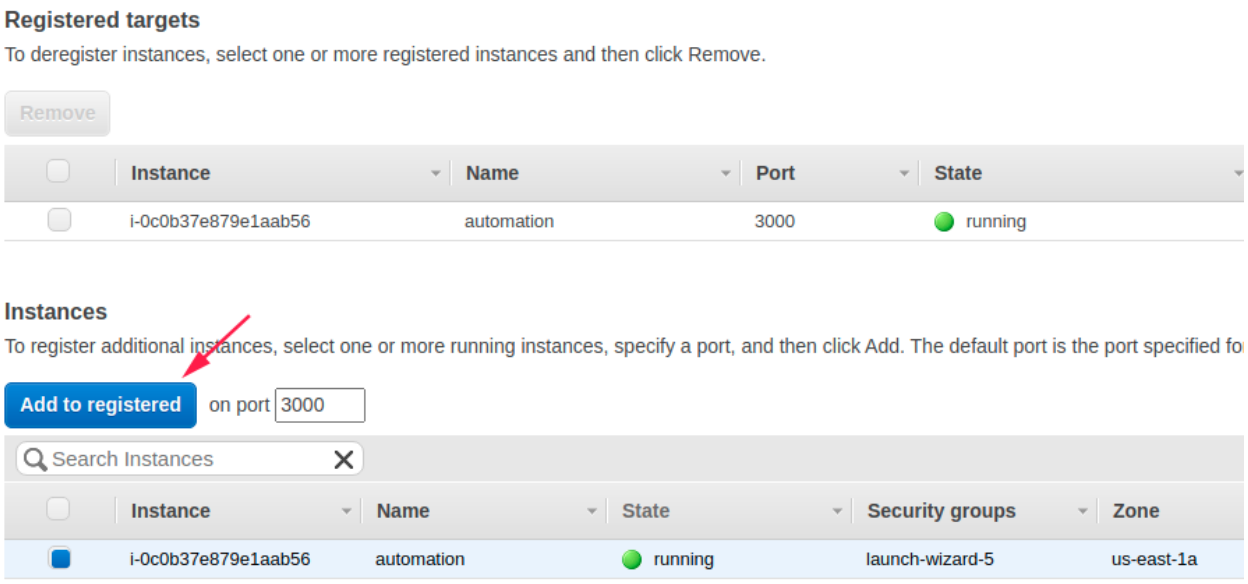
3. Advance health check settings

Dejar valores por defecto

Next : Register Targets

Step 5 : Register Targets

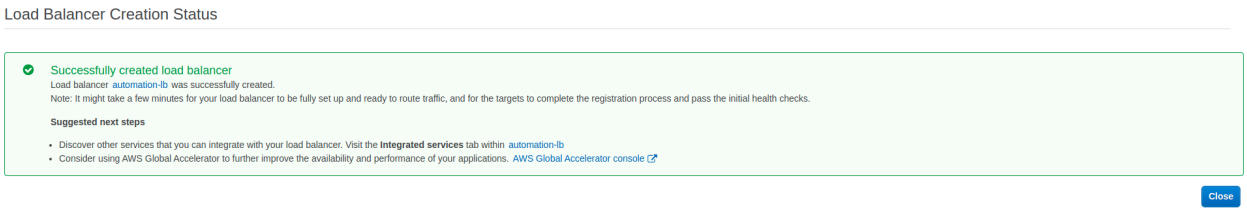
Seleccionar a la instancia de EC2 y dar click en Add to registered



Next : Review

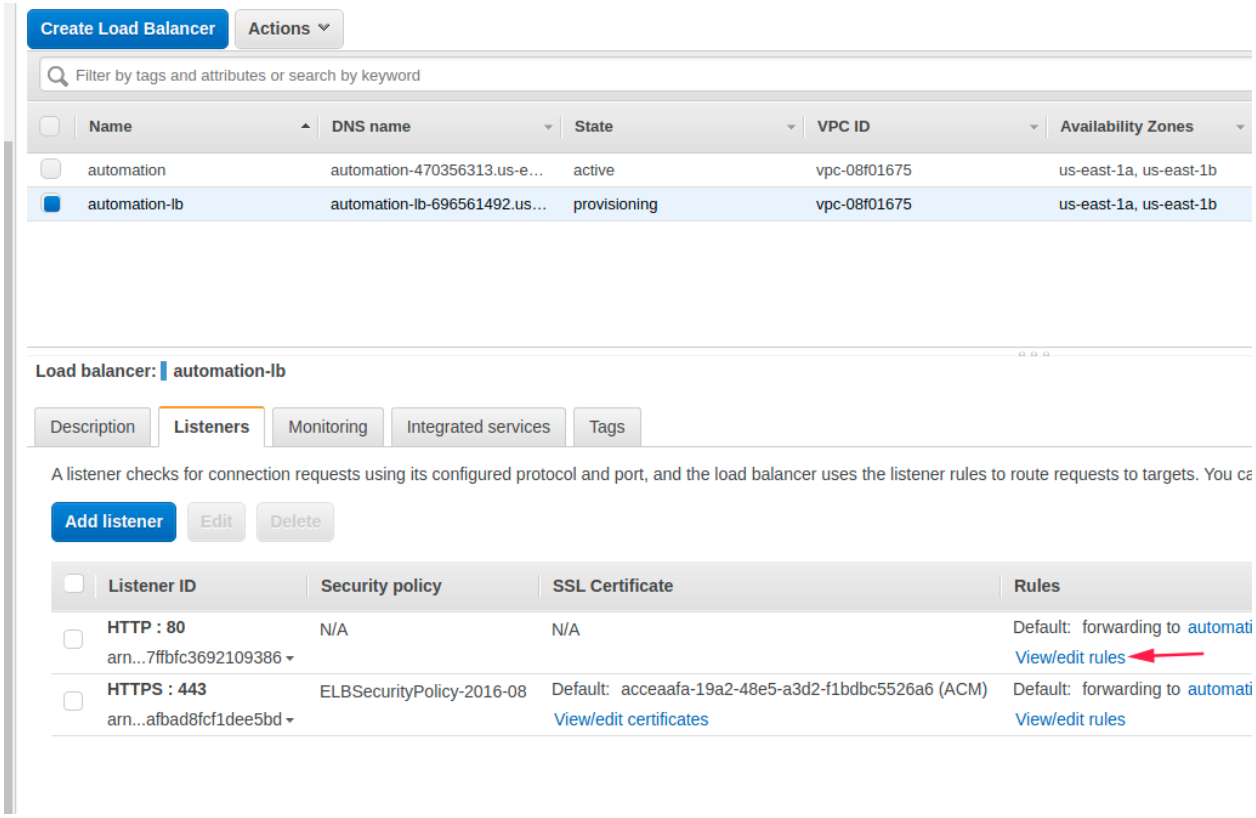
Step 6 : Review

Crear y luego dar click en Close



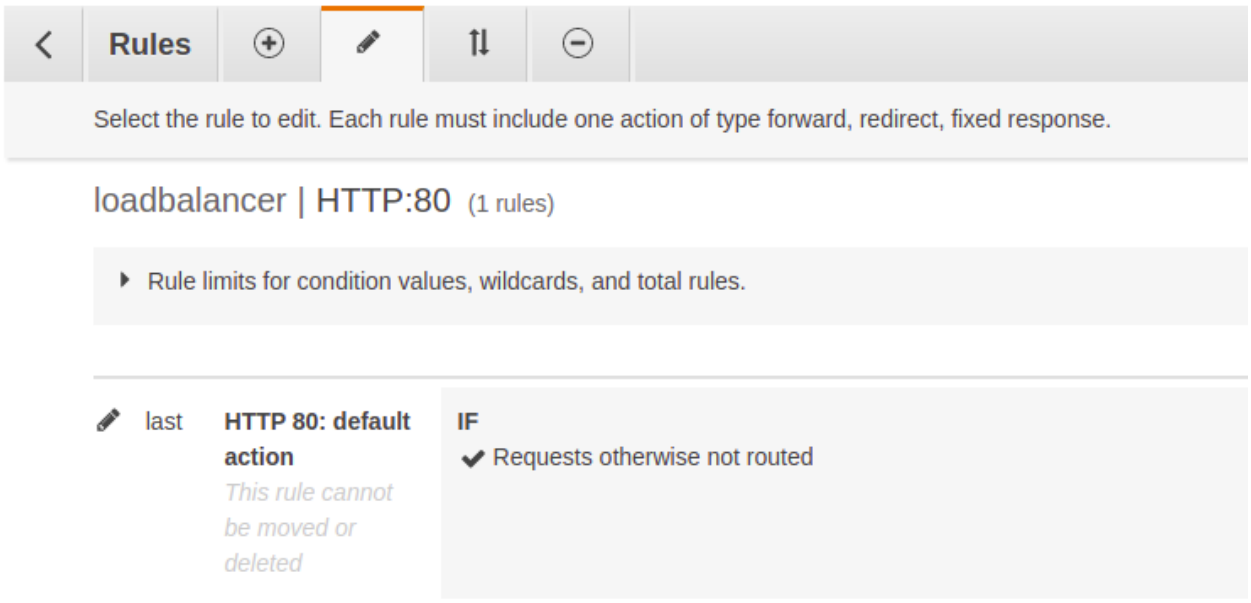
Redireccionar llamadas que vienen por HTTP a HTTPS

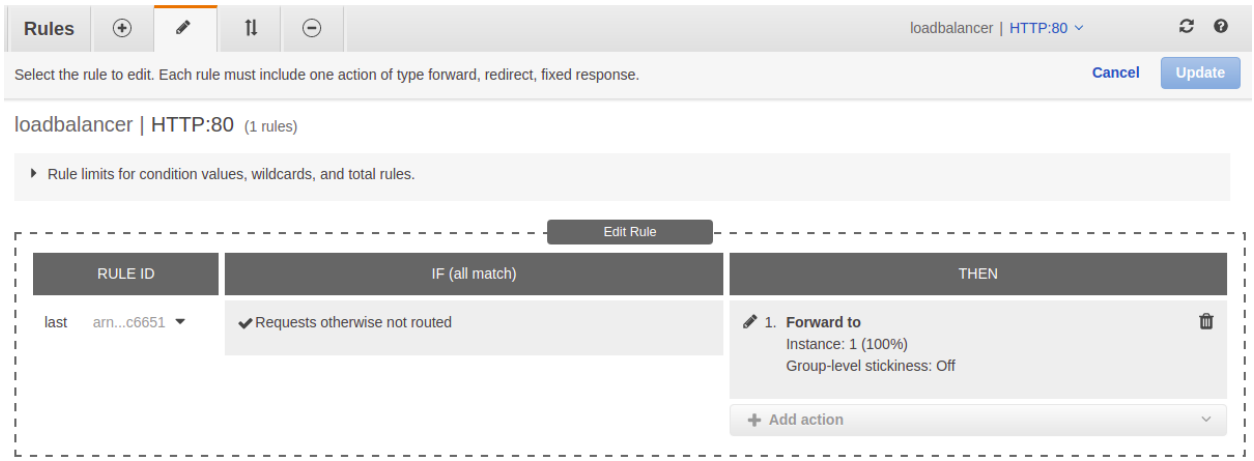
Dentro del dashboard de los balanceadores de carga seleccionar la tab “Listeners”



Dentro del Listener del puerto 80, seleccionar “View/edit rules”

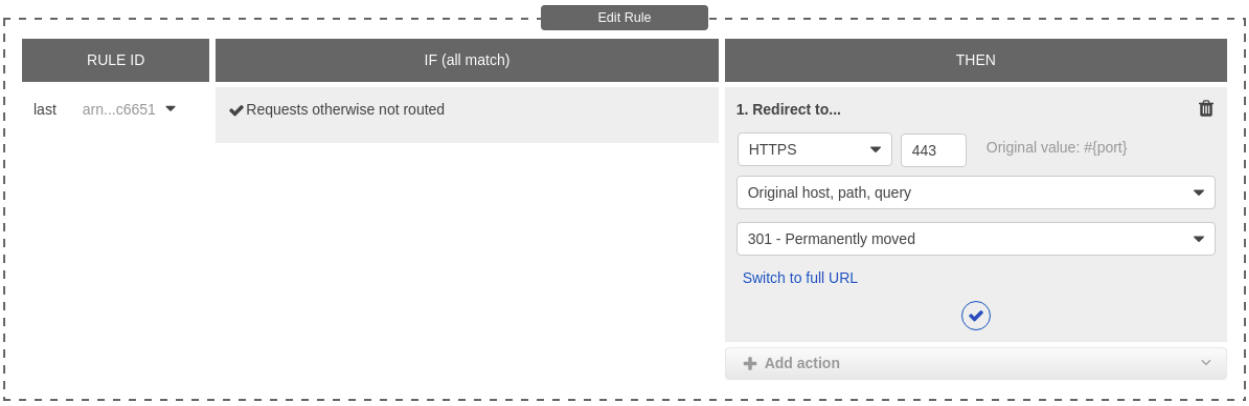
Editar la regla





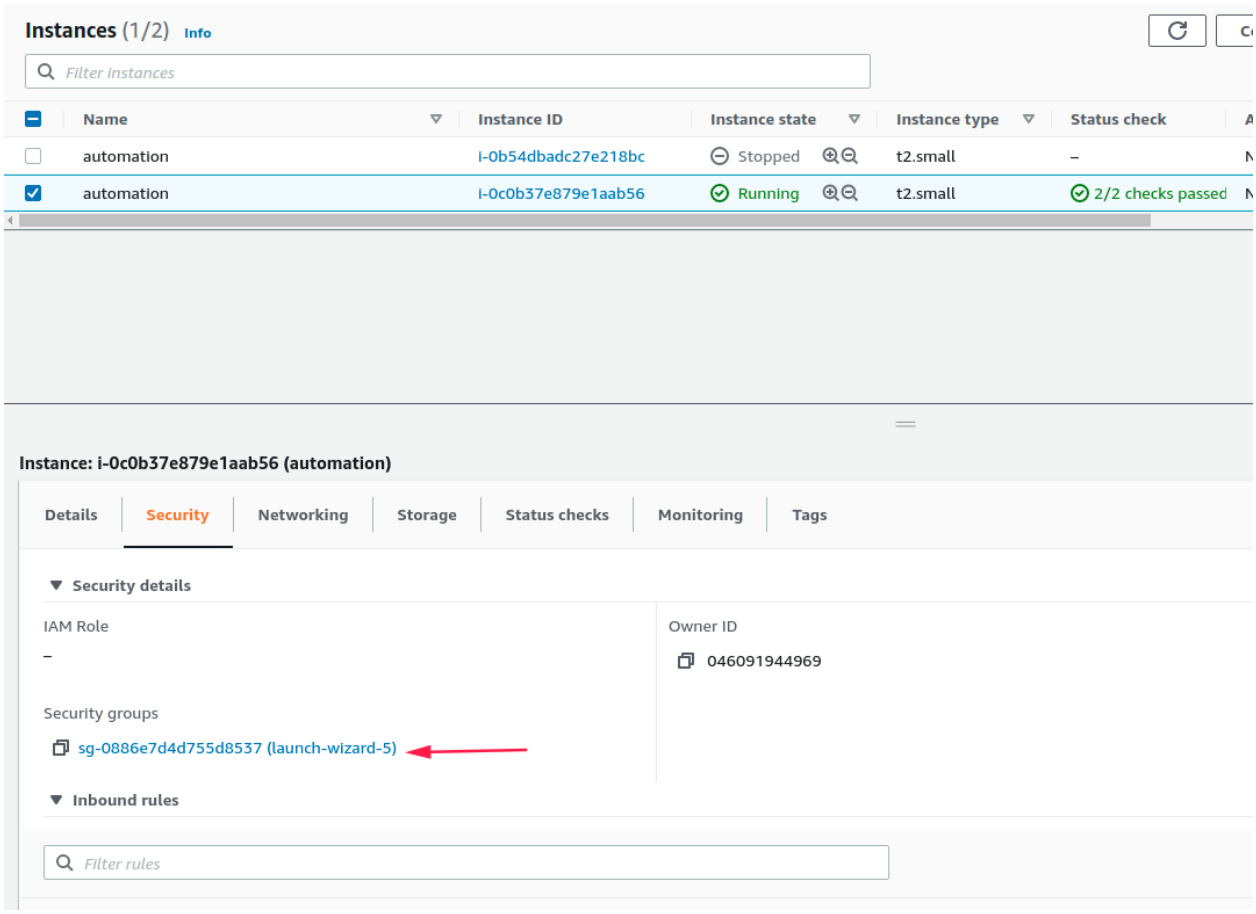
1. Borrar “Forward to”

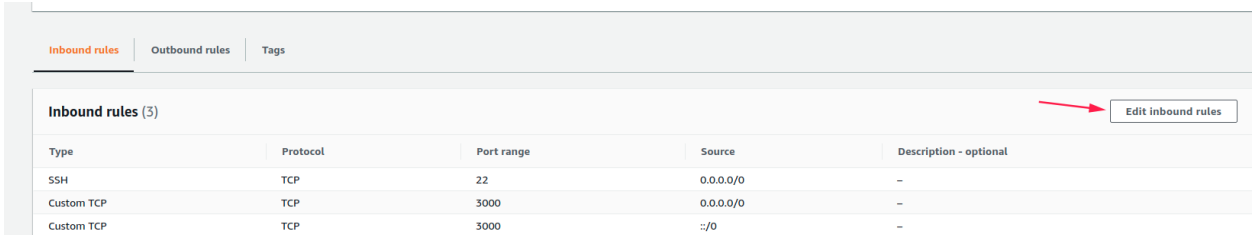
2. Add Action “Redirect to” como muestra la figura y dar click en el “check mark”



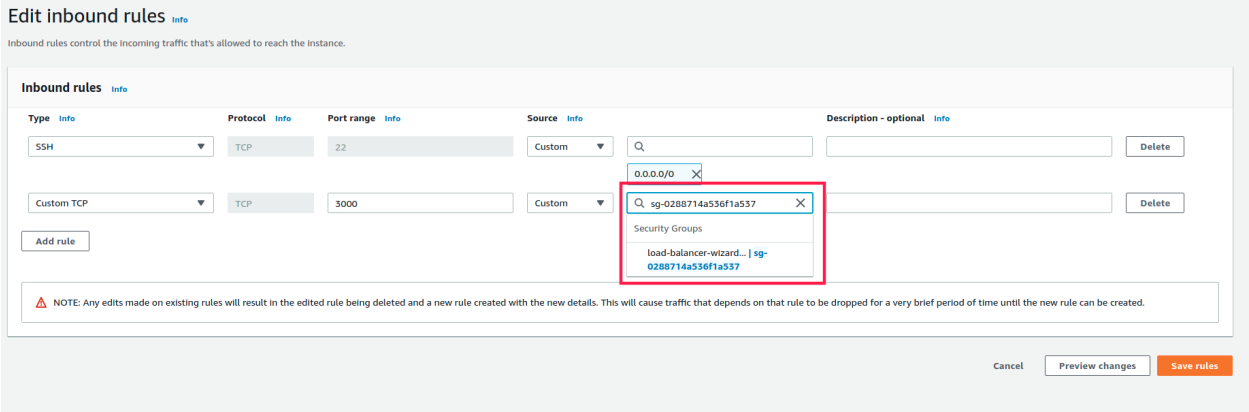
3. Dar click en Update

10. Reconfigurar el grupo de seguridad de la instancia





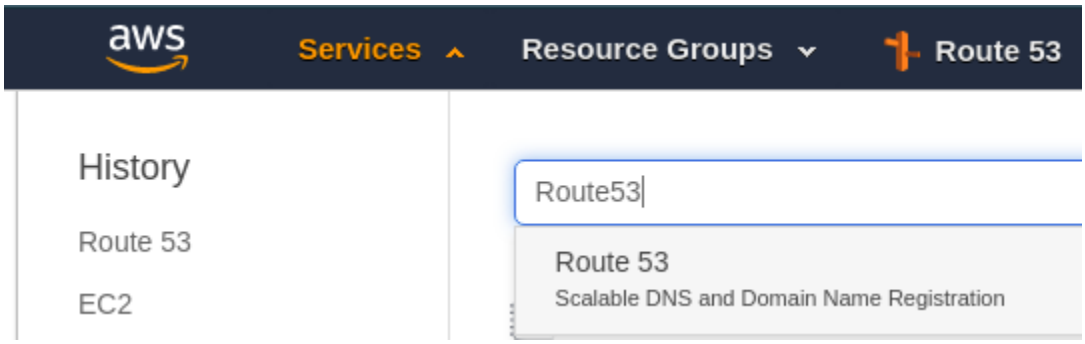
Editamos las reglas para que solo esté disponible el acceso a la instancia solamente desde el grupo de seguridad del loadbalancer



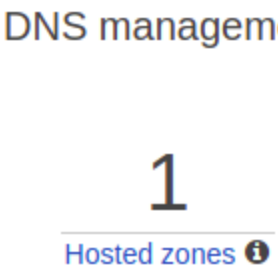
Save rules

11. Crear registro en Route53

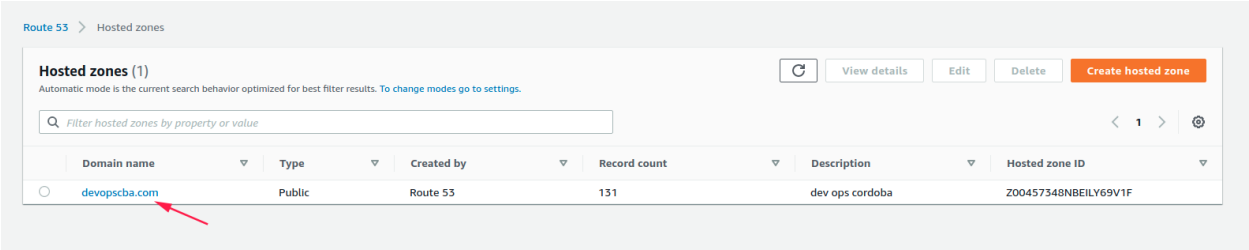
Buscar el servicio de route53



Dentro de Route53 Dashboard seleccionar “Hosted zones”



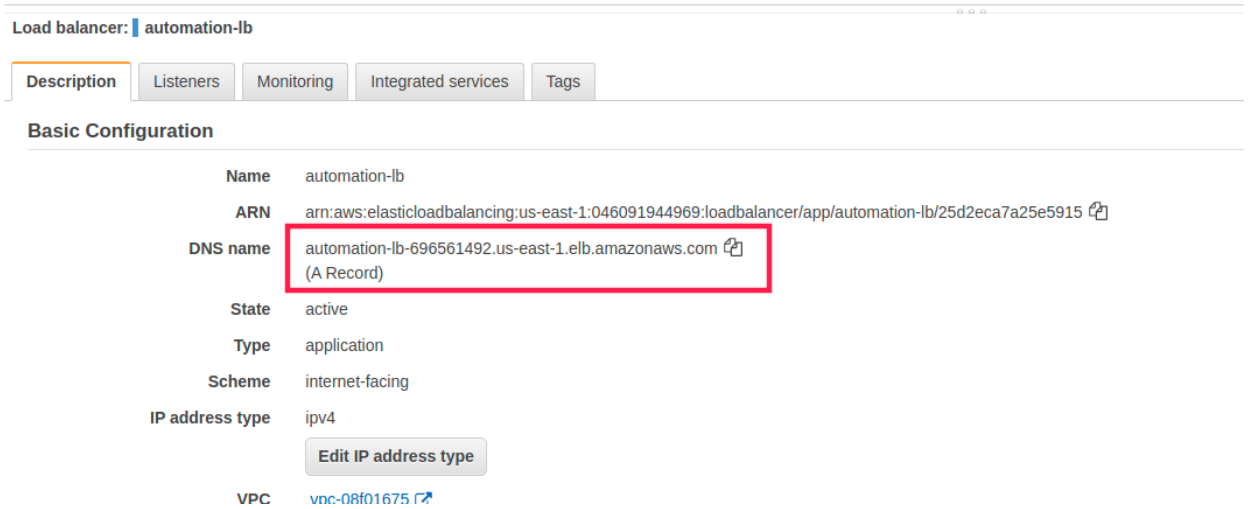
Seleccionar la “hosted zone” configurada para tu usuario

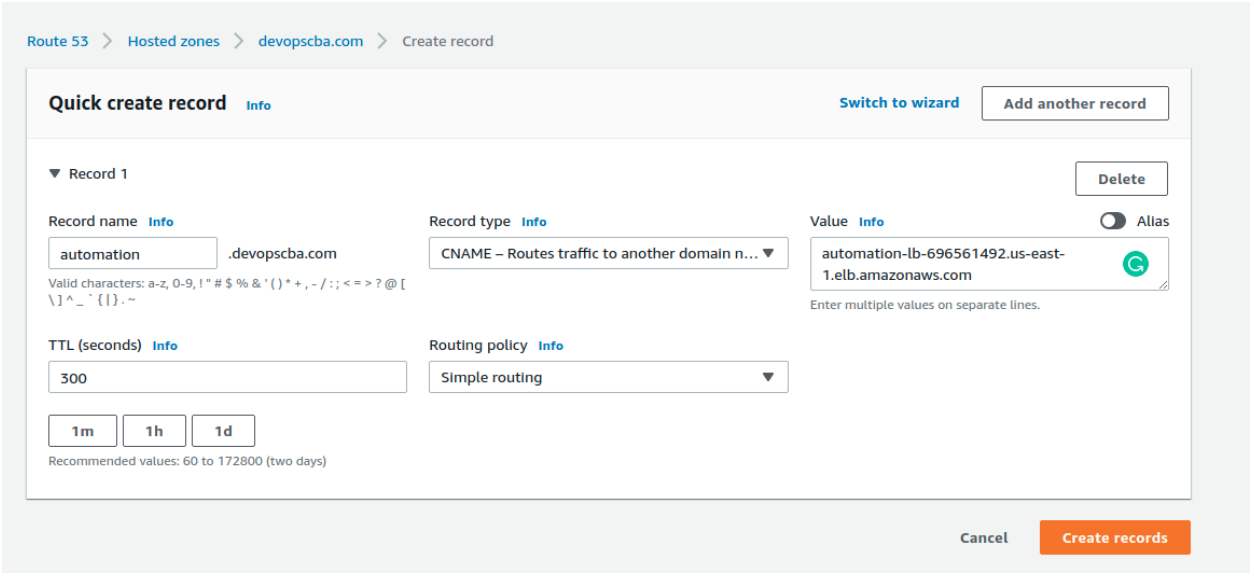


Seleccionar “Create Record Set” y completar los datos a la izquierda

Name	automation.<tuusuario>.devopscba.com
Type	CNAME
Value	DNS de loadbalancer


Create Load Balancer
Actions
Filter by tags and attributes or search by keyword
Name
DNS name
State
VPC ID
Availability Zones
Type
automation
automation-470356313.us-e…
active
vpc-08f01675
us-east-1a, us-east-1b
automation-lb
automation-lb-696561492.us…
active
vpc-08f01675
us-east-1a, us-east-1b


Load balancer: automation-lb
Description
Listeners
Monitoring
Integrated services
Tags
Basic Configuration
Name
automation-lb
ARN
arn:aws:elasticloadbalancing:us-east-1:046091944969:loadbalancer/app/automation-lb/25d2eca7a25e5915
DNS name
automation-lb-696561492.us-east-1.elb.amazonaws.com
(A Record)
State
active
Type
application
Scheme
internet-facing
IP address type
ipv4
Edit IP address type
VPC
vpc-08f01675

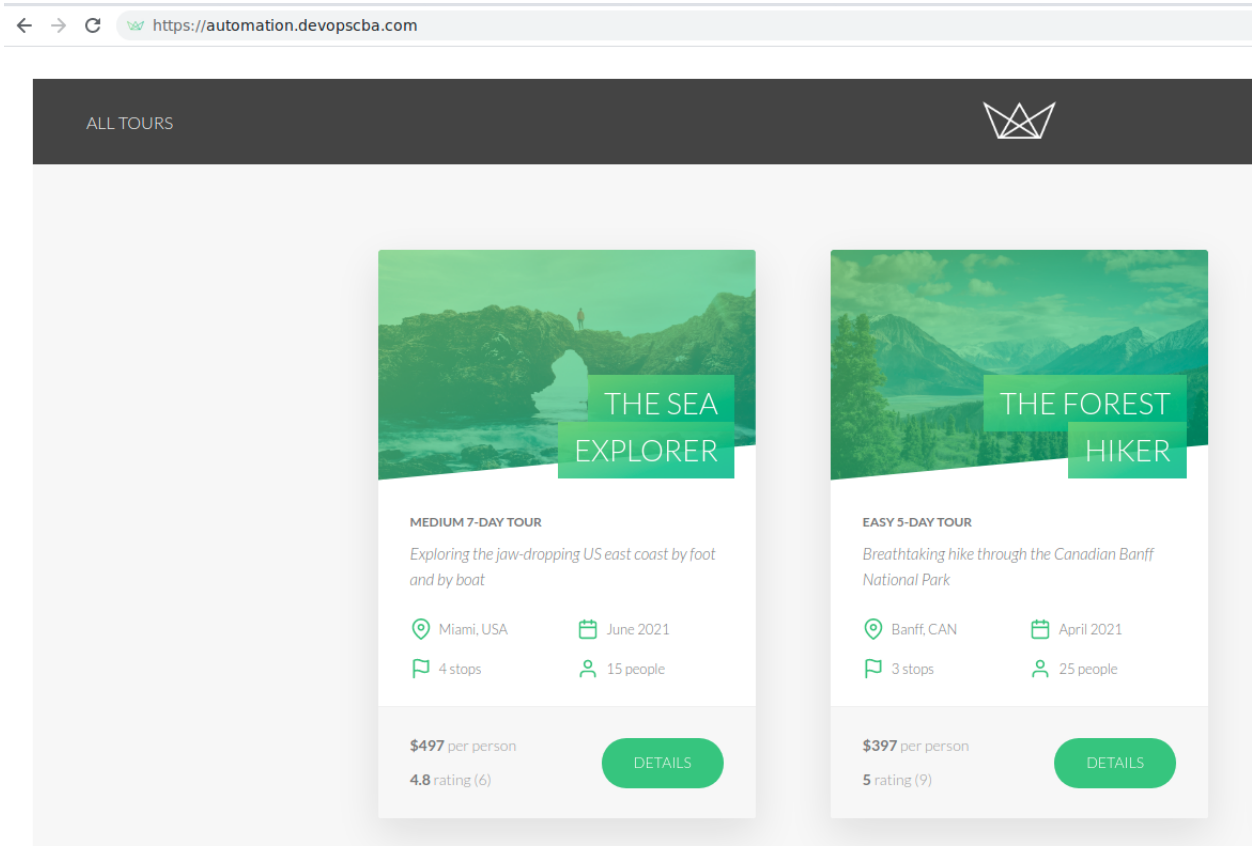

Route 53
Hosted zones
devopscba.com
Create record
Quick create record
Info
Switch to wizard
Add another record
Record 1
Delete
Record name
Info
automation
.devopscba.com
Valid characters: a-z, 0-9, ! " # $ % & ' ( ) * + , - / : ; < = > ? @ [\ ] ^ _ ` { | } . ~
Record type
Info
CNAME – Routes traffic to another domain n…
Value
Info
Alias
automation-lb-696561492.us-east-1.elb.amazonaws.com
Enter multiple values on separate lines.
TTL (seconds)
Info
300
Routing policy
Info
Simple routing
1m
1h
1d
Recommended values: 60 to 172800 (two days)
Cancel
Create records

Create Record

12. Probar acceder a la aplicacion a traves del loadbalancer

Colocar en el browser

- En el caso de este ejemplo
    - https://automation.devopscba.com



La app funciona correctamente!

Ejecutar los siguiente pasos en una máquina local

Preferiblemente con el sistema operativo Ubuntu

13. Instalar python 3.9

Ejecutar los siguientes comandos

```
sudo apt update -y
sudo apt install software-properties-common -y
sudo add-apt-repository ppa:deadsnakes/ppa
sudo apt install python3.9 python3.9-venv python3.9-dev -y
python3.9 --version
```

14. Crear un ambiente virtual de python

Ejecutar el siguiente comando

```
python3.9 -m venv ~/automation
```

15. Entrar en el folder automation

Ejecutar el siguiente comando

```
cd ~/automation
```

16. Habilitar el ambiente virtual

Ejecutar el siguiente comando

```
source /bin/activate
```

17. Instalar Selenium

17.1 Ejecutar el siguiente comando

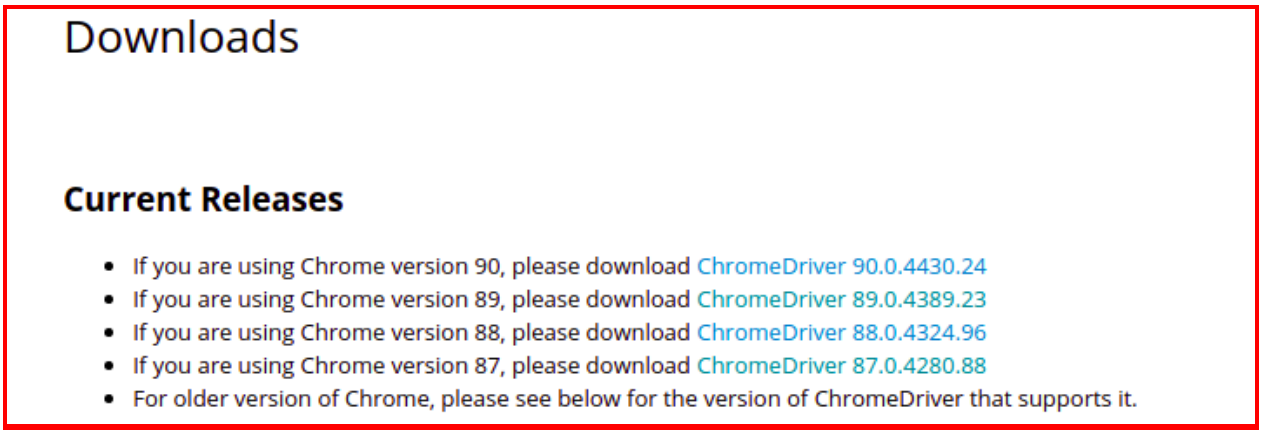
```
pip install selenium
```

17 .2 Descargar el driver de Chrome

En el siguiente link

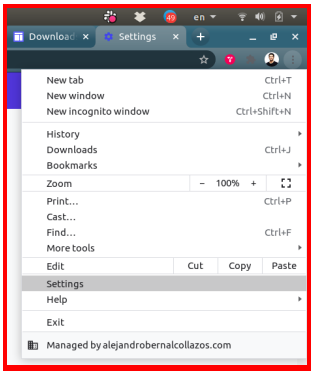
- https://chromedriver.chromium.org/downloads

Seleccionar el driver de acuerdo a la versión de chrome que tengas instalada en tu máquina local

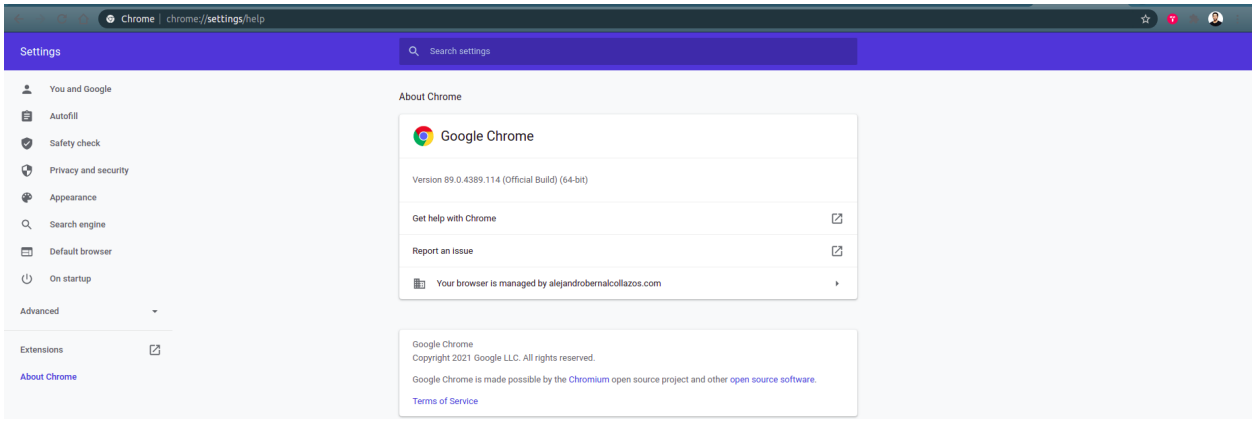


Para ver qué versión de Chrome tenemos instalado llevar a cabo los siguientes pasos

17.2.1 Ir a la parte de Settings de Chrome



17.2.2 Seleccionar la parte de “About Chrome” y verificar la versión de Chrome



17.3 Unzip del archivo que contiene el driver

Ejecutar el comando

```
unzip chromedriver_linux64.zip
```

17.4 En la máquina local, luego de descargar el driver ubicarlo dentro de una carpeta que esté en la variable PATH

Por ejemplo en el sistema operativo Linux ejecutar el siguiente comando

```
mv chromedriver /usr/local/bin
```

18. Crear un archivo que va a contener nuestras pruebas de automatización

Crear un archivo con el siguiente nombre

- automation.py

19. Agregar código en el archivo

Agregar el siguiente código

```
from selenium import webdriver
import time

browser = webdriver.Chrome()

browser.get("http://automation.devopscba.com")

# Login locator

login_xpath_locator = "//nav[@class='nav nav--user']/a[@class='nav__el']"

# Login element

login_button = browser.find_element_by_xpath(login_xpath_locator)

# Click on the login button

login_button.click()

# Wait for 3 seconds

time.sleep(3)

# Email locator
```

```
email_css_locator = "input[id='email']"

# Email element

email_input = browser.find_element_by_css_selector(email_css_locator)

# Set data into the email field

email_input.send_keys("alejandro@example.com")
```

20. Correr el programa

Ejecutar el siguiente comando

```
python automation.py
```