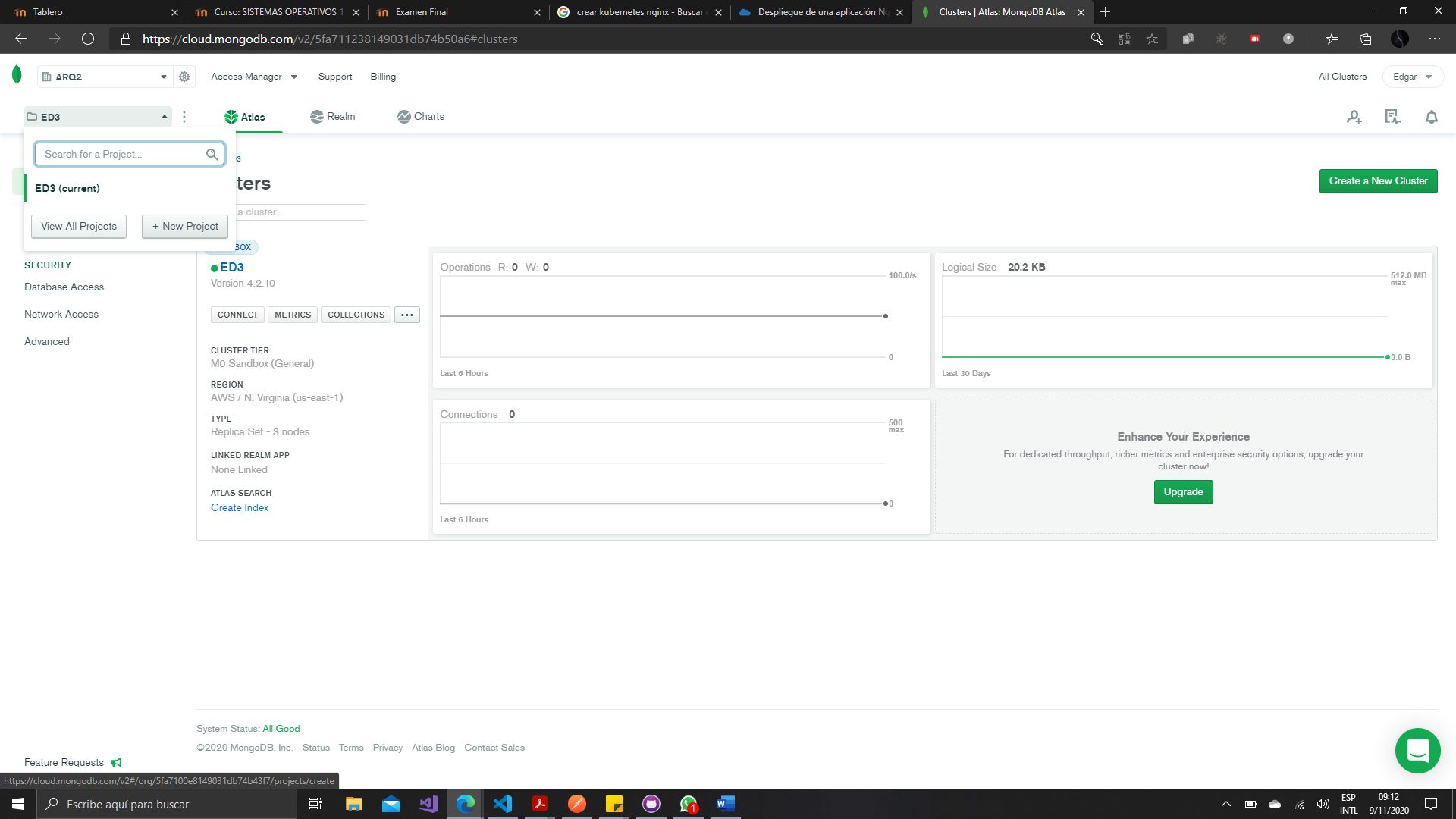
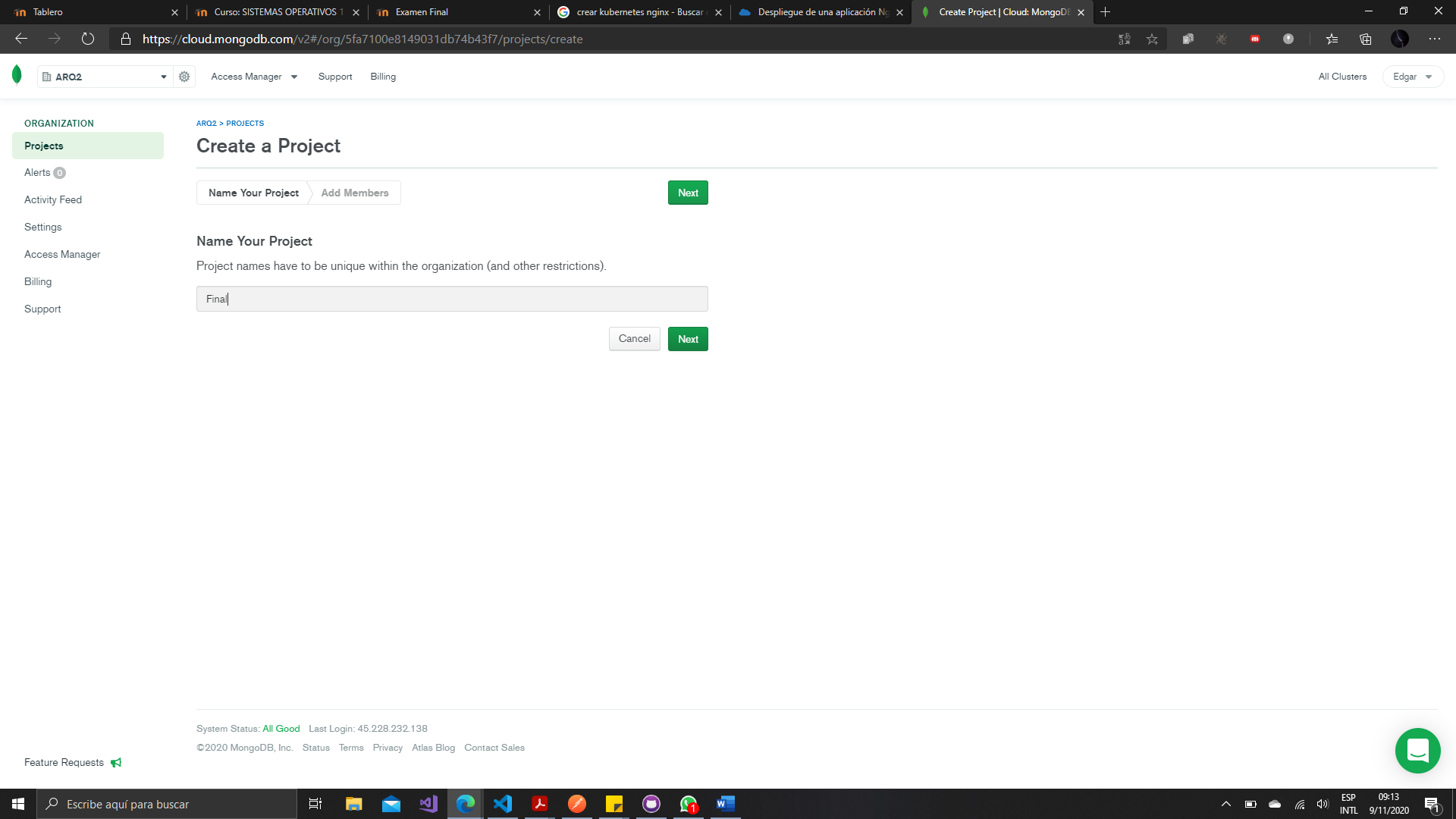
# Manual de configuraciones

# Creación y configuración de la base de datos

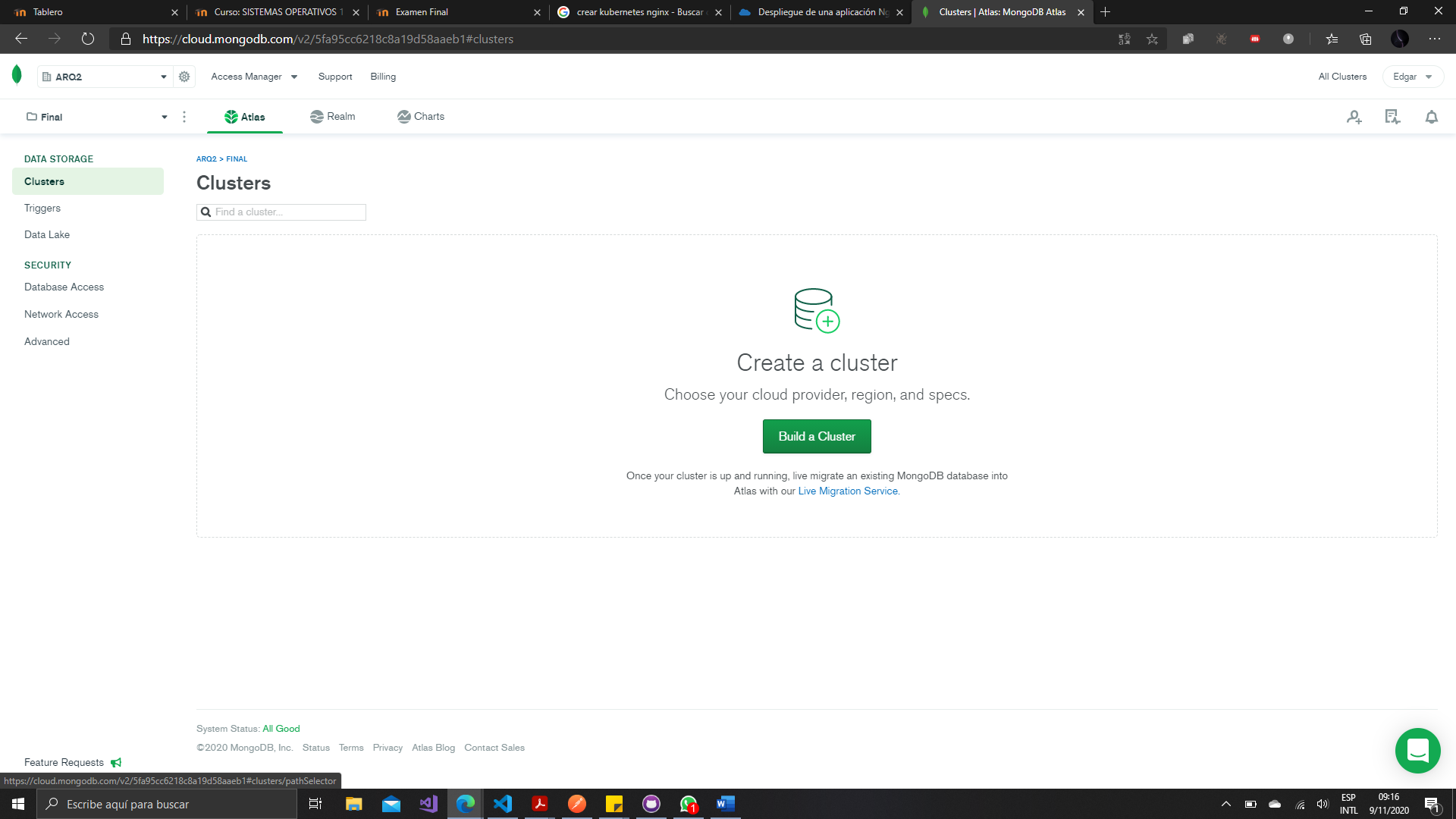
Se crea un nuevo proyecto.

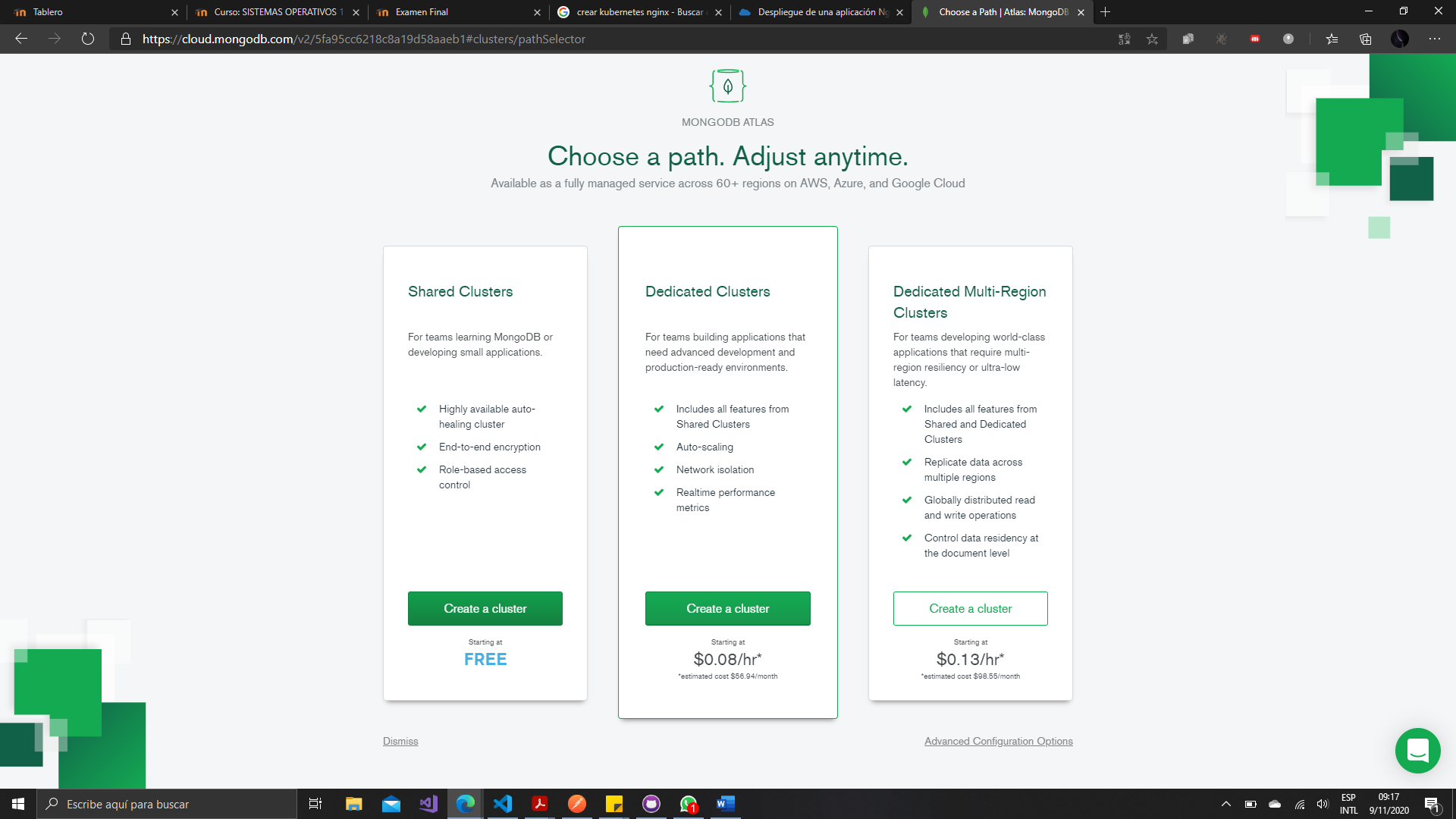


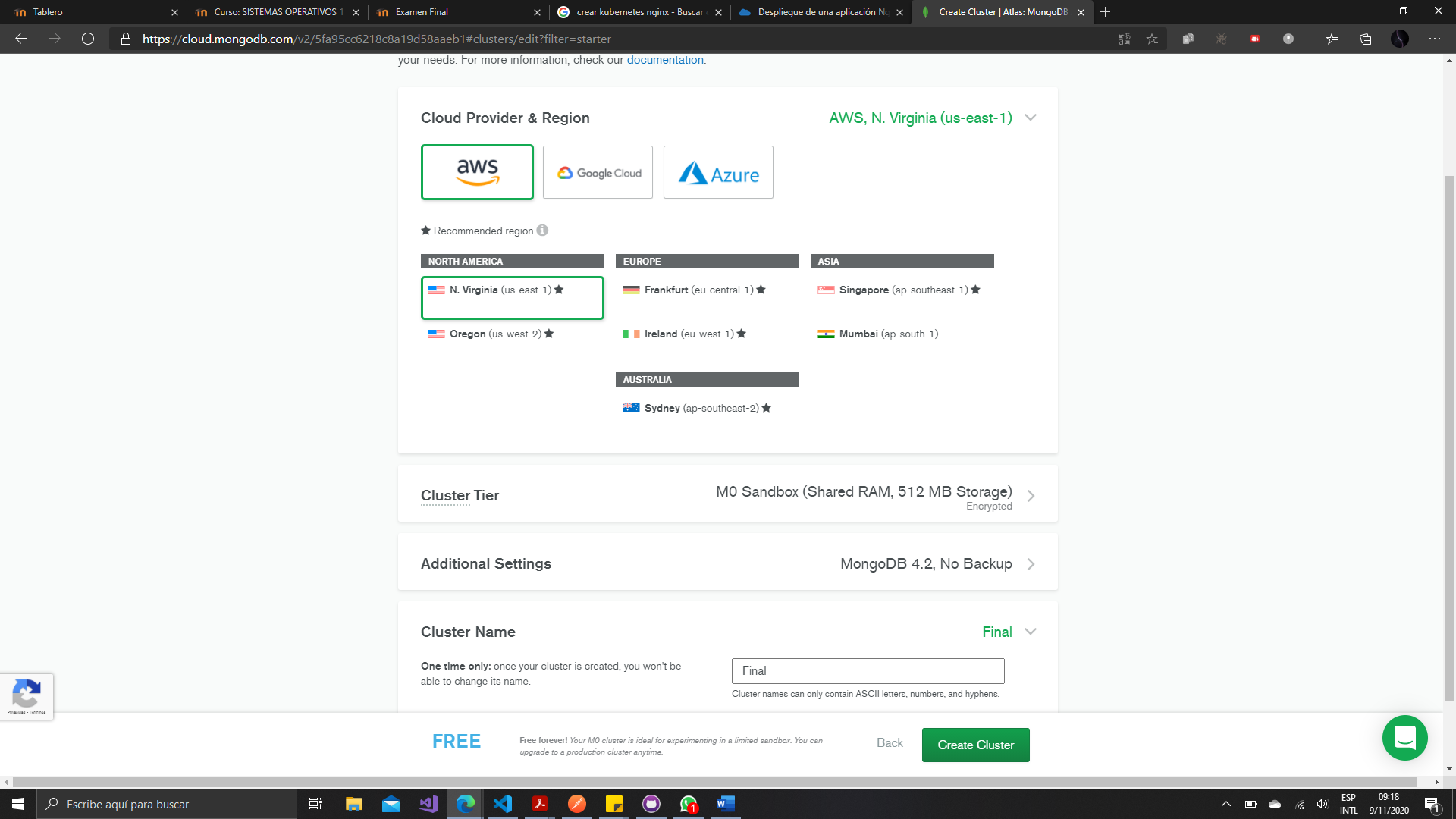
Se asigna el nombre, luego next y luego crear.

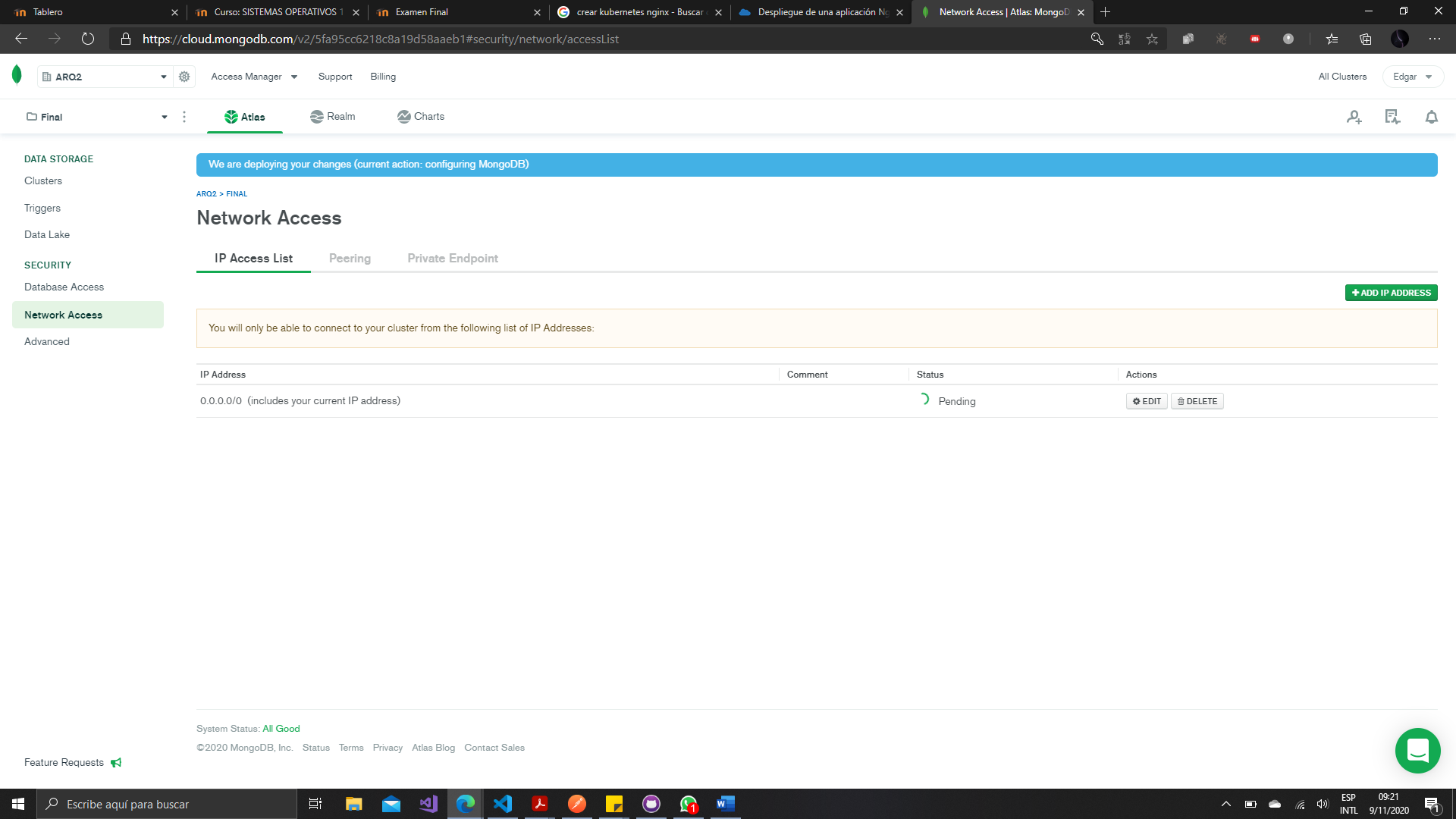


Se crea un nuevo cluster

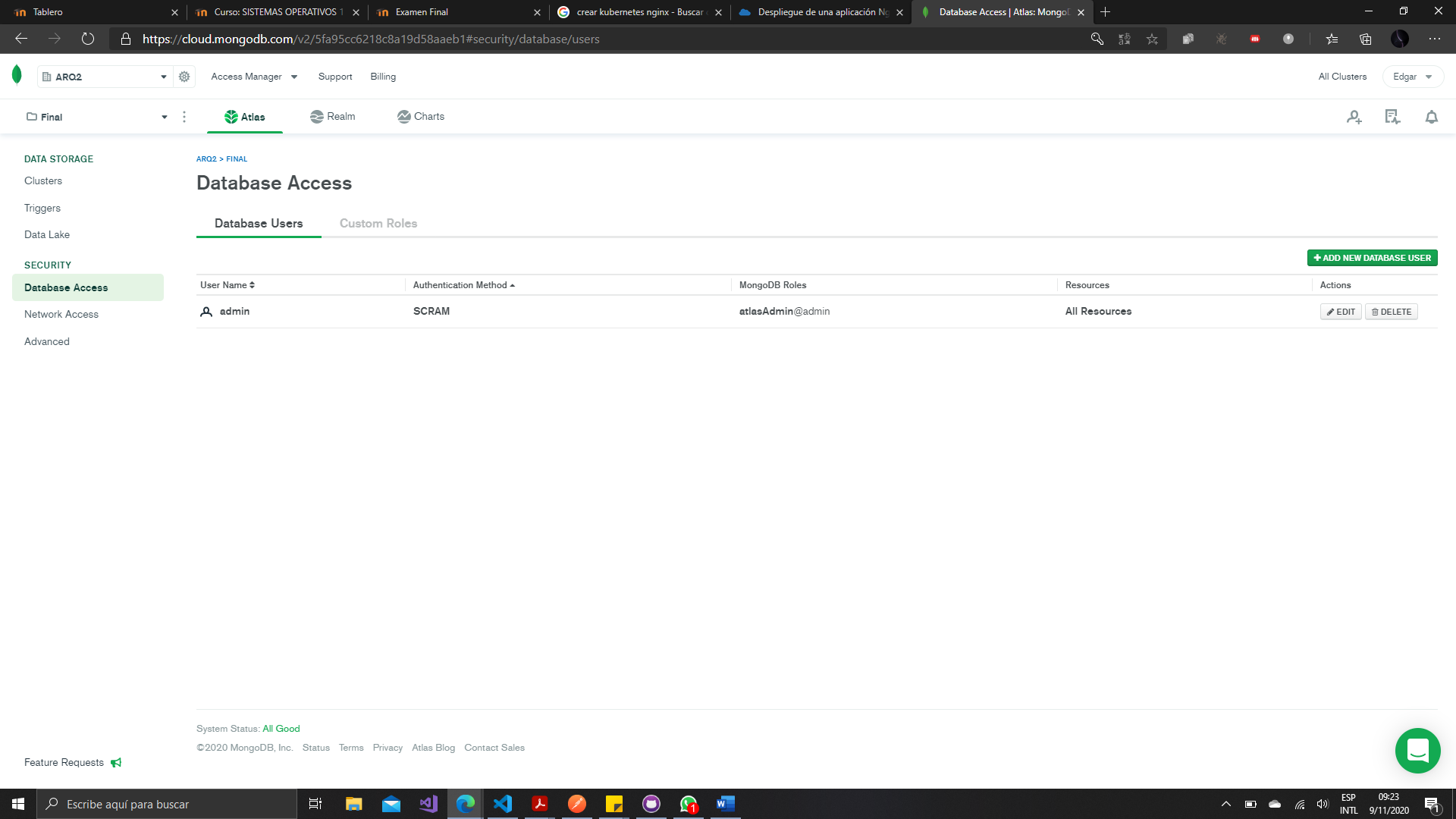


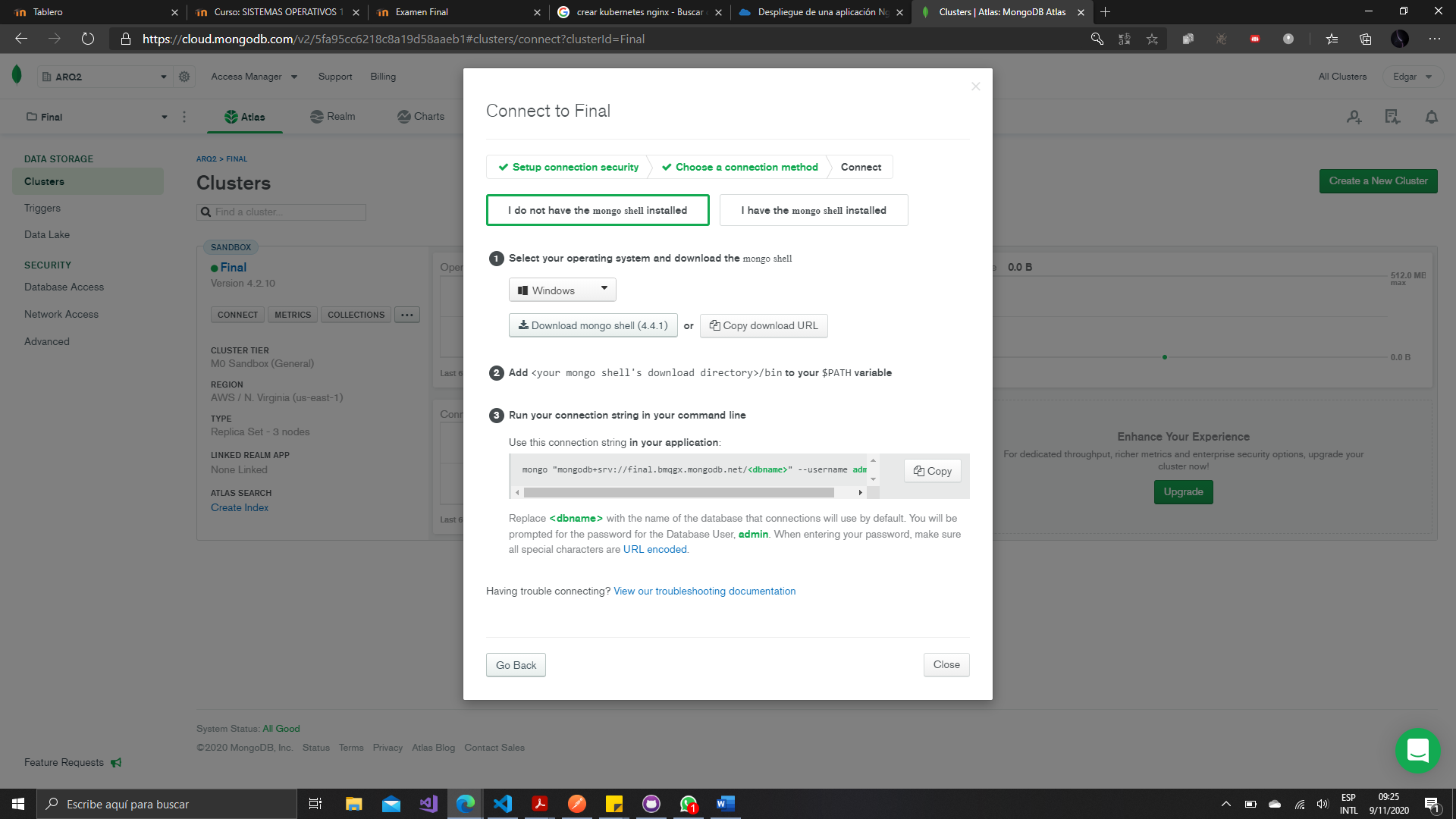
Se selecciona la opción que se desee, en este caso usaremos la opción gratuita.

Se selecciona el proveedor, se asigna el nombre y clic en crear cluster. 

Se agrega un acceso una dirección ip

Se crea un usuario con permisos de administrador.

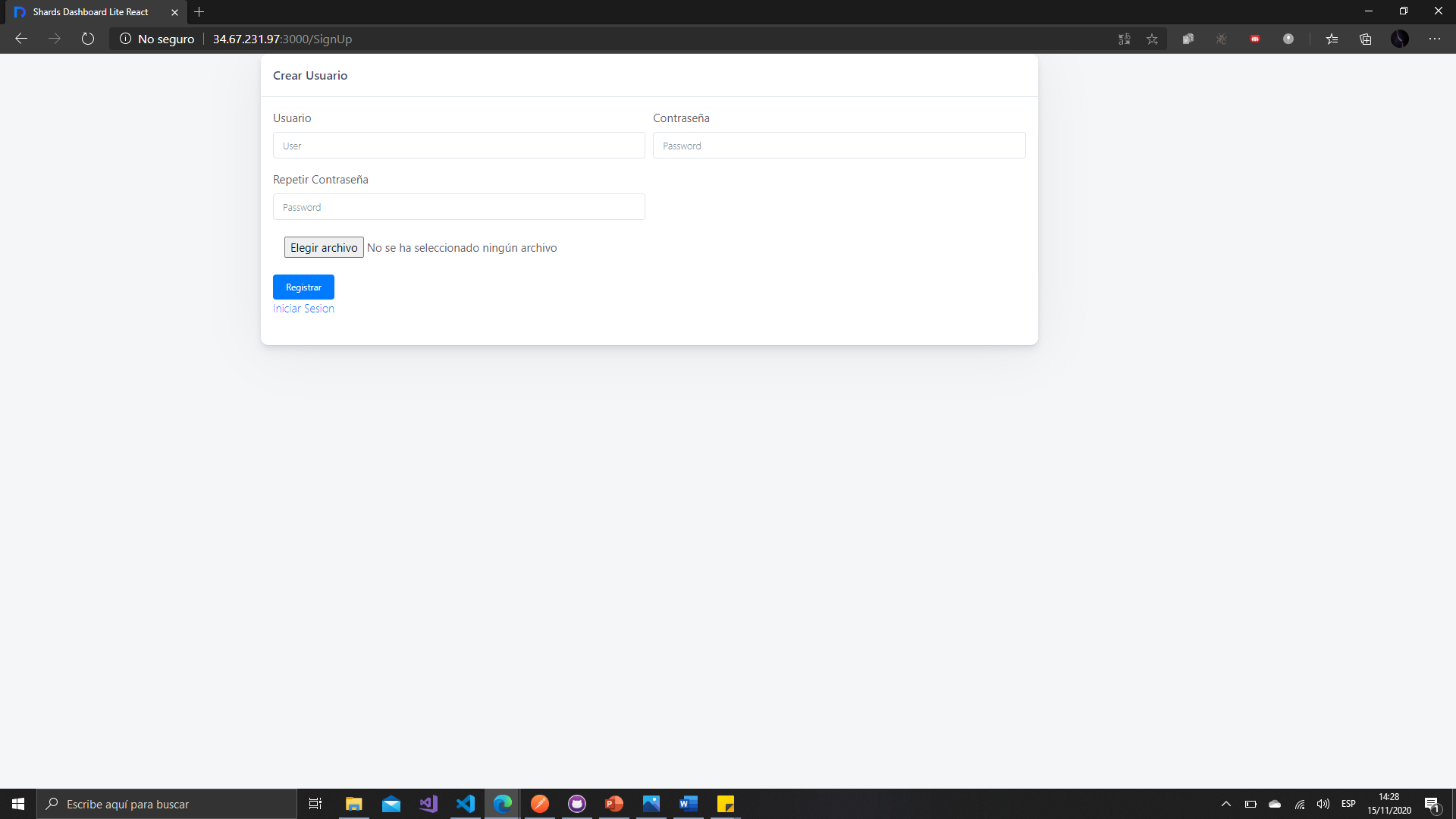


Nos dirigimos a clusters y luego a connect donde se desplegarán las opciones de coneccion, se selecciona la deseada. En este caso tenemos la siguiente url: mongodb+srv://admin:1234@final.bmqgx.mongodb.net/Final?retryWrites=true&w=majority

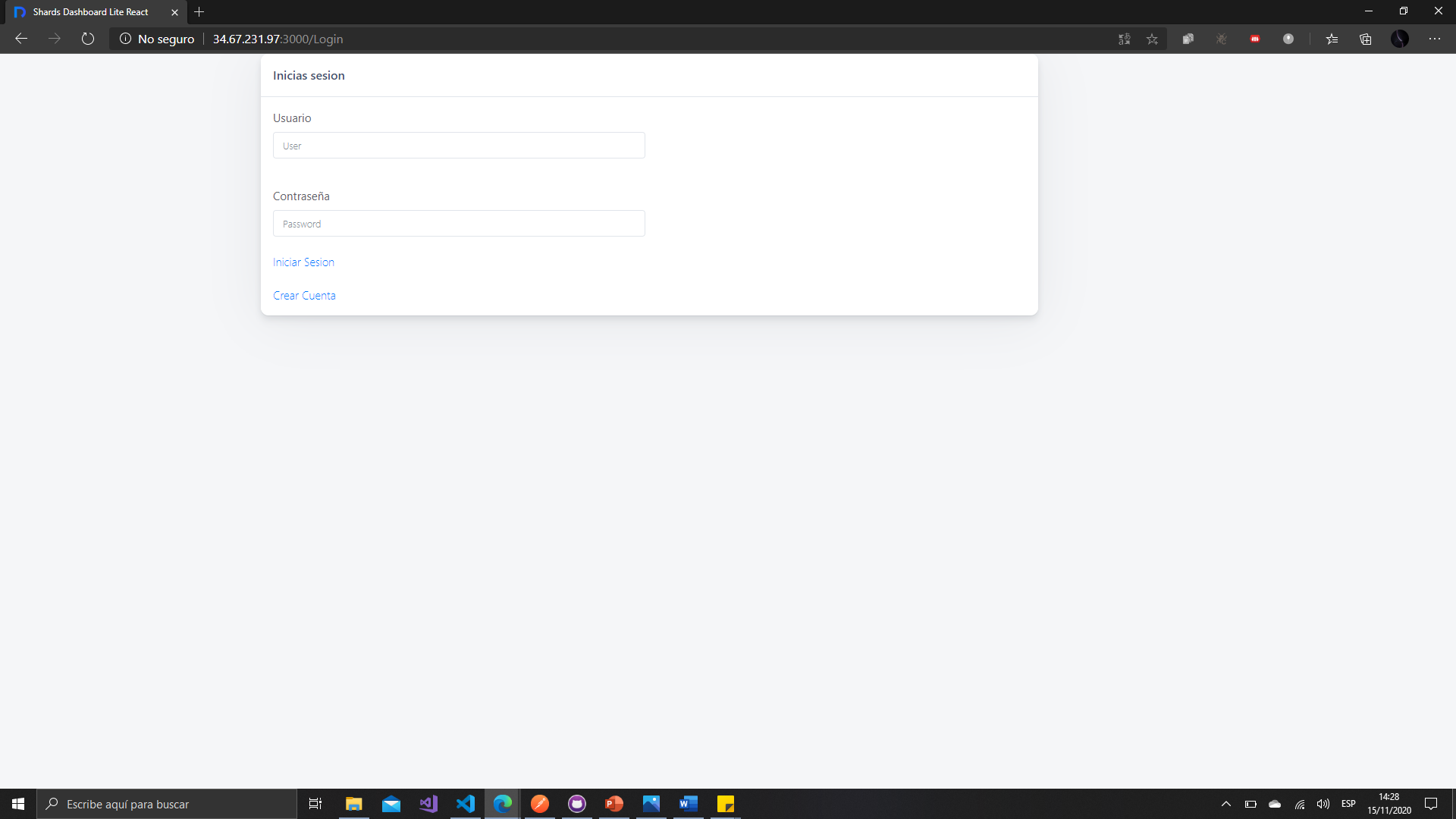
## Frontend

Se desarrollo una pagina web estática con html5, css y javascrpt.

Registro



Login



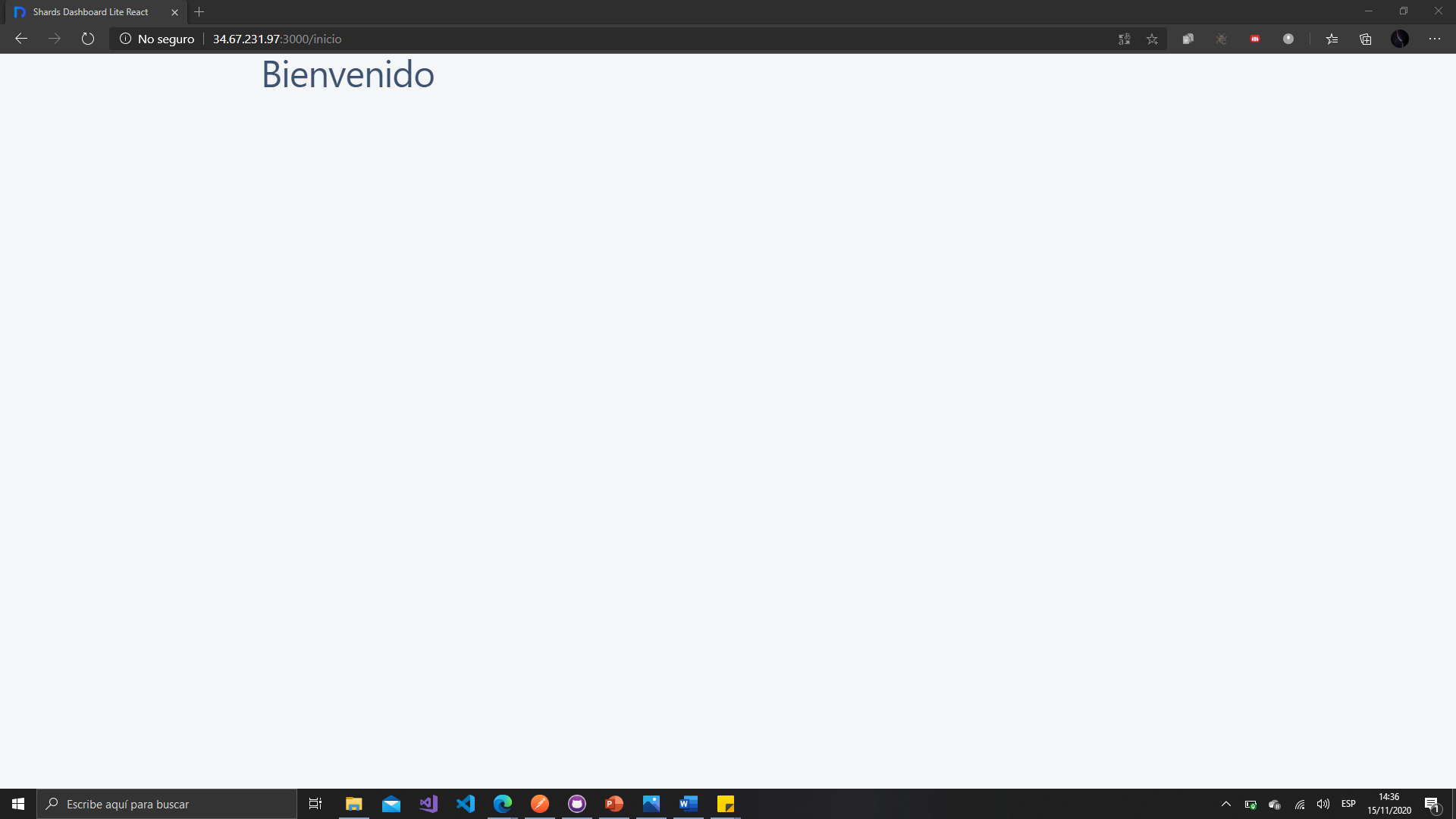
## Backend

Aplicación en nodejs con las siguientes rutas

35.188.121.170:5000/Login

35.188.121.170:5000/registro

Inicio



# Kubernetes

## ymls

Backend

apiVersion: apps/v1

kind: Deployment

metadata:

  name: back-deployment

spec:

  replicas: 3

  selector:

    matchLabels:

      app: back

  template:

    metadata:

      labels:

        app: back

    spec:

      containers:

      - name: server-back

        image: gcr.io/sopes-295304/back:1.0

        ports:

        - containerPort: 5000

Frontend

apiVersion: apps/v1

kind: Deployment

metadata:

  name: front-deployment

spec:

  replicas: 3

  selector:

    matchLabels:

      app: front

  template:

    metadata:

      labels:

        app: front

    spec:

      containers:

      - name: server-front

        image: gcr.io/sopes-295304/front2:1.0

        ports:

        - containerPort: 3000

Balanceador de carga backend

apiVersion: v1

kind: Service

metadata:

  name: back-service

spec:

  selector:

    app: back

  ports:

    - port: 5000

      targetPort: 5000

  type: LoadBalancer

Balanceador de carga frontend

apiVersion: v1

kind: Service

metadata:

  name: front-service

spec:

  selector:

    app: front

  ports:

    - port: 3000

      targetPort: 3000

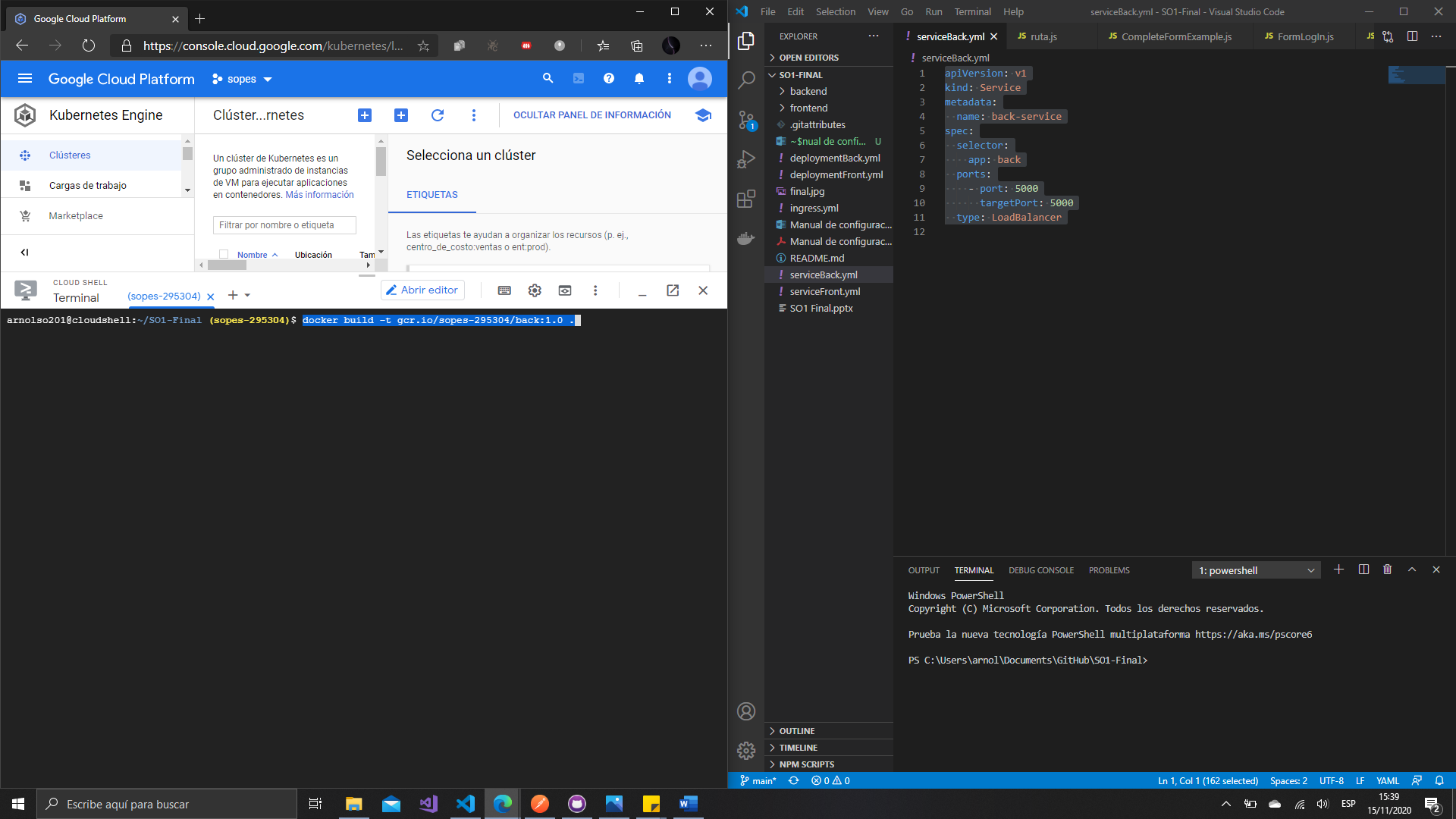
  type: LoadBalancer

## Creación Pods y balanceador de carga

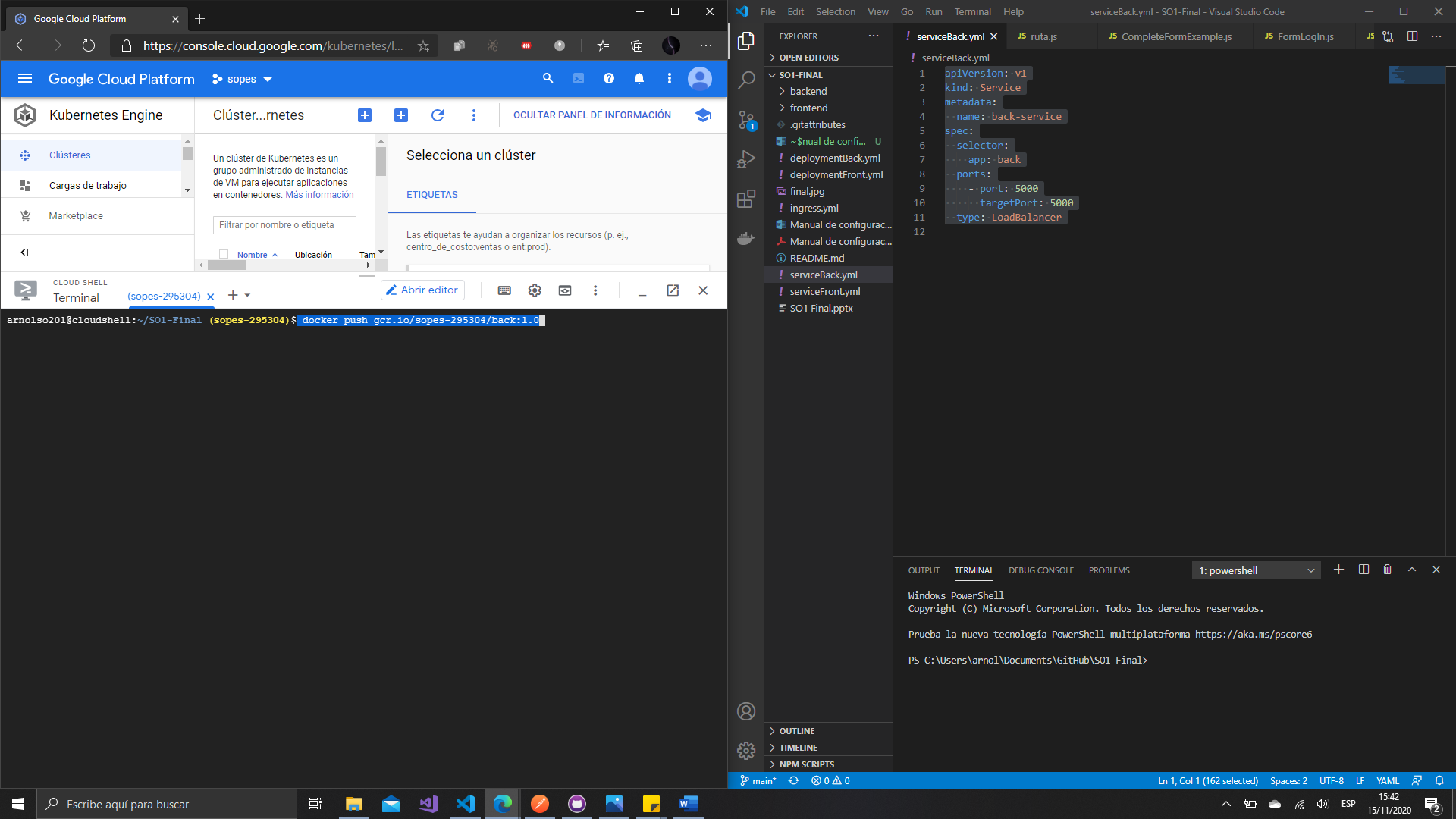
Para iniciar con la creación de los pods es necesario primero construir las imágenes que se utilizaran en los pod, para eso se utiliza el siguiente comando

Se le asigna el nombre y versión deseada.

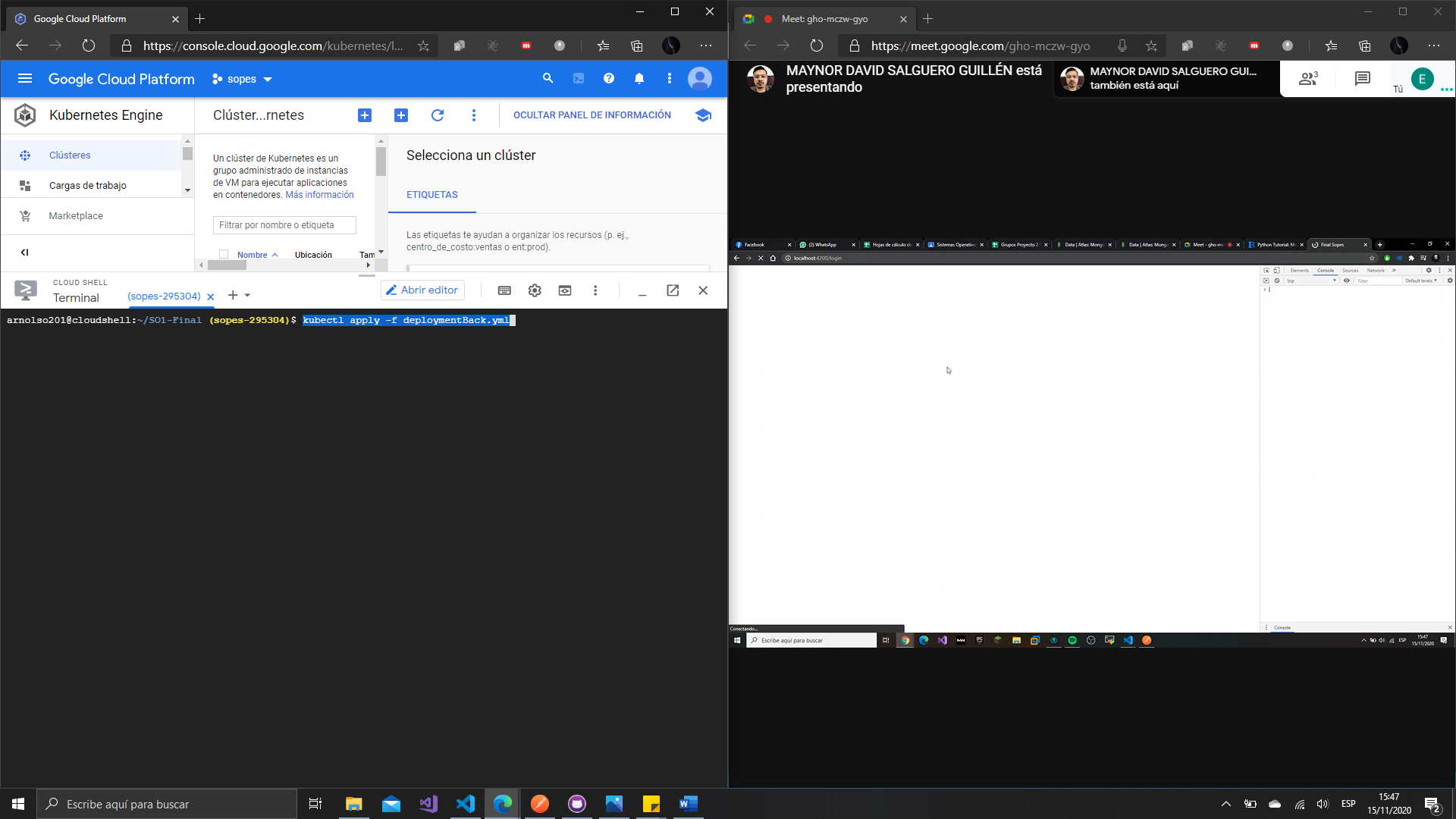
Para el frontend se aplican los mismos comandos.



Luego de construir las plantillas se les hace push para posterior mente utilizar la plantilla.



Con las plantillas listas solamente queda ejecutar los yml que se encargaran de la construcción de los pods y los balanceadores de carga. Se debe de realizar la construcción del frontend y backend antes que los balanceadores de carga. Para la construcción de los yml se utiliza el siguiente comando



Con los siguientes comandos es posible ver los pods y servicios que están corriendo.

