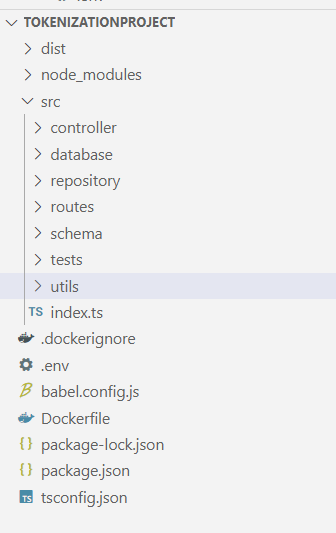
El Proyecto está conformado por un desarrollo backend nodejs (paquetes importantes de express js, redis y mysql).



El Proyecto nodejs está estructurado en

Index.js: la ejecución inicial del Proyecto

Utils: se encuentra las validaciones de la entrada de datos.

Tests: se encuentra el código de test con las validaciones importantes para la ejecución local básica.

Routes: se encuentra las rutas que conforman las urls.

Controller: contiene la lógica de la aplicación.

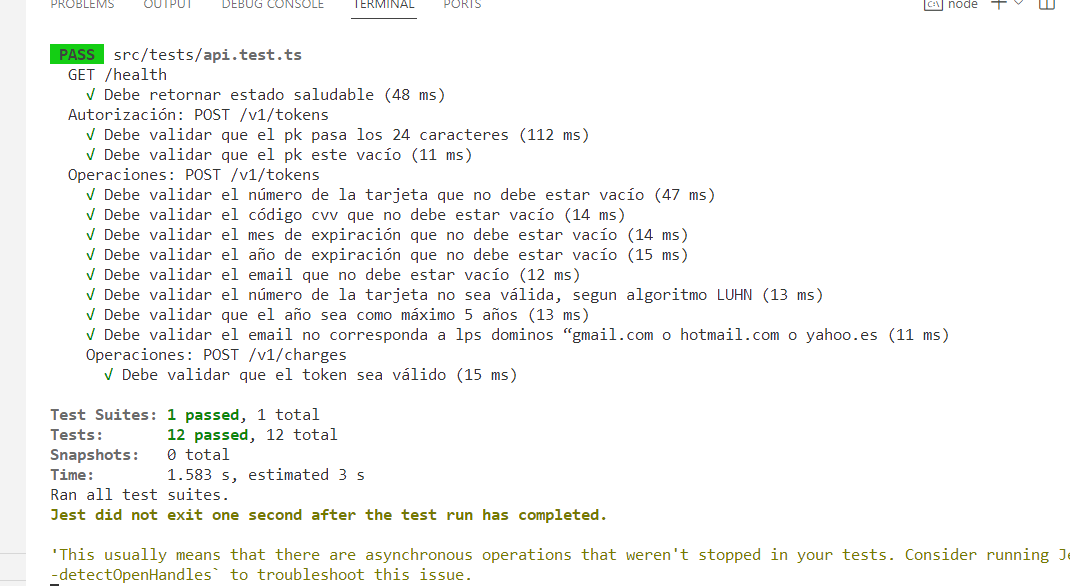
Schema: contiene la estructura del modelo para poder interactuar con las demás capas.

Repository: es el intermediario entre el controller y la capa de datos.

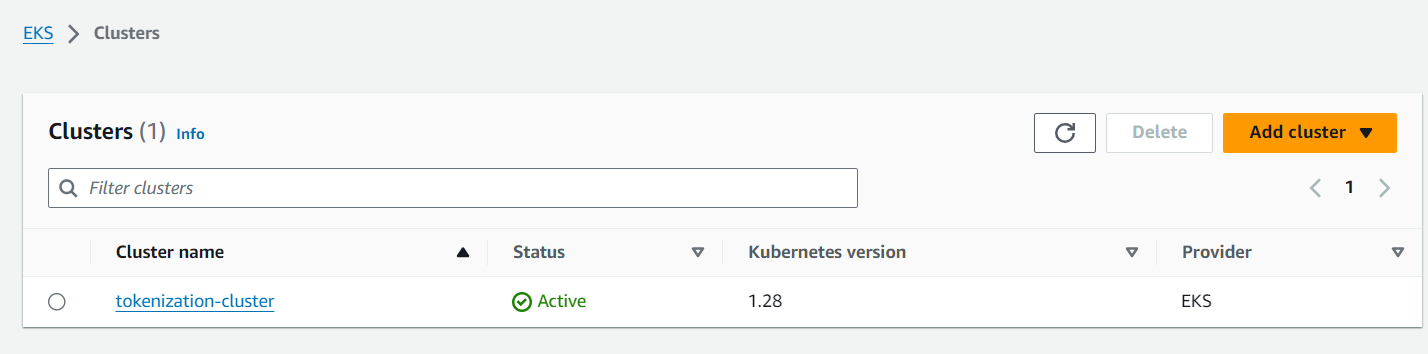
Database: contiene las conexiones y los métodos de las base de datos redis y mysql. Trabaja conjuntamente con la capa de repositorio mediante una clase de factoría para elegir con que base de datos trabajar. Así que el controlador no sabe qué base de datos se usa.

La ejecución se realiza del test se realiza con el comando npm run test; la compilación se realiza con el comando npm run build; la ejecucion local se realiza con el comando npm run start. Sin embargo para que funcione todo el flujo debe estar en un ambiente EKS de Amazon.

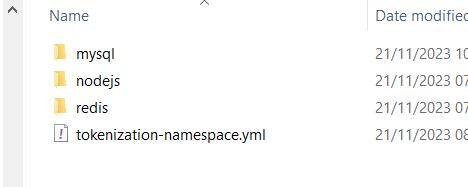
Por ejemplo, en la siguiente imagen se encuentra el test realizado.



Para el despliegue a un clúster EKS, se requirió el programa kubectl para crear el cluster y los despliegues en AWS.

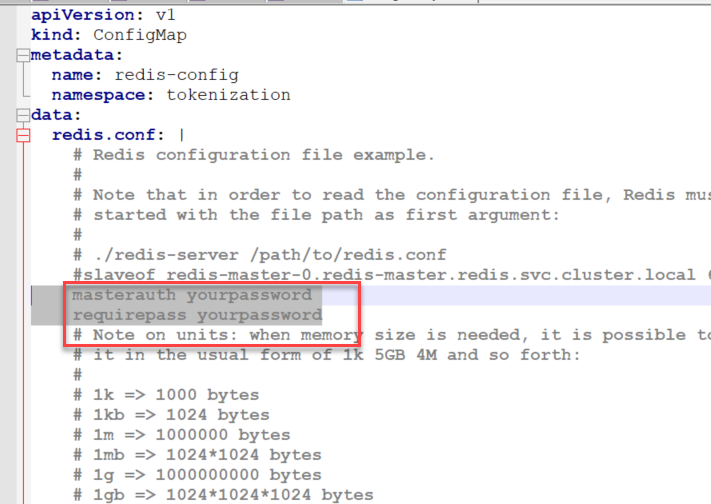


En la carpeta kubernetes se encuentran los archivos para la ejecución de los despliegues.



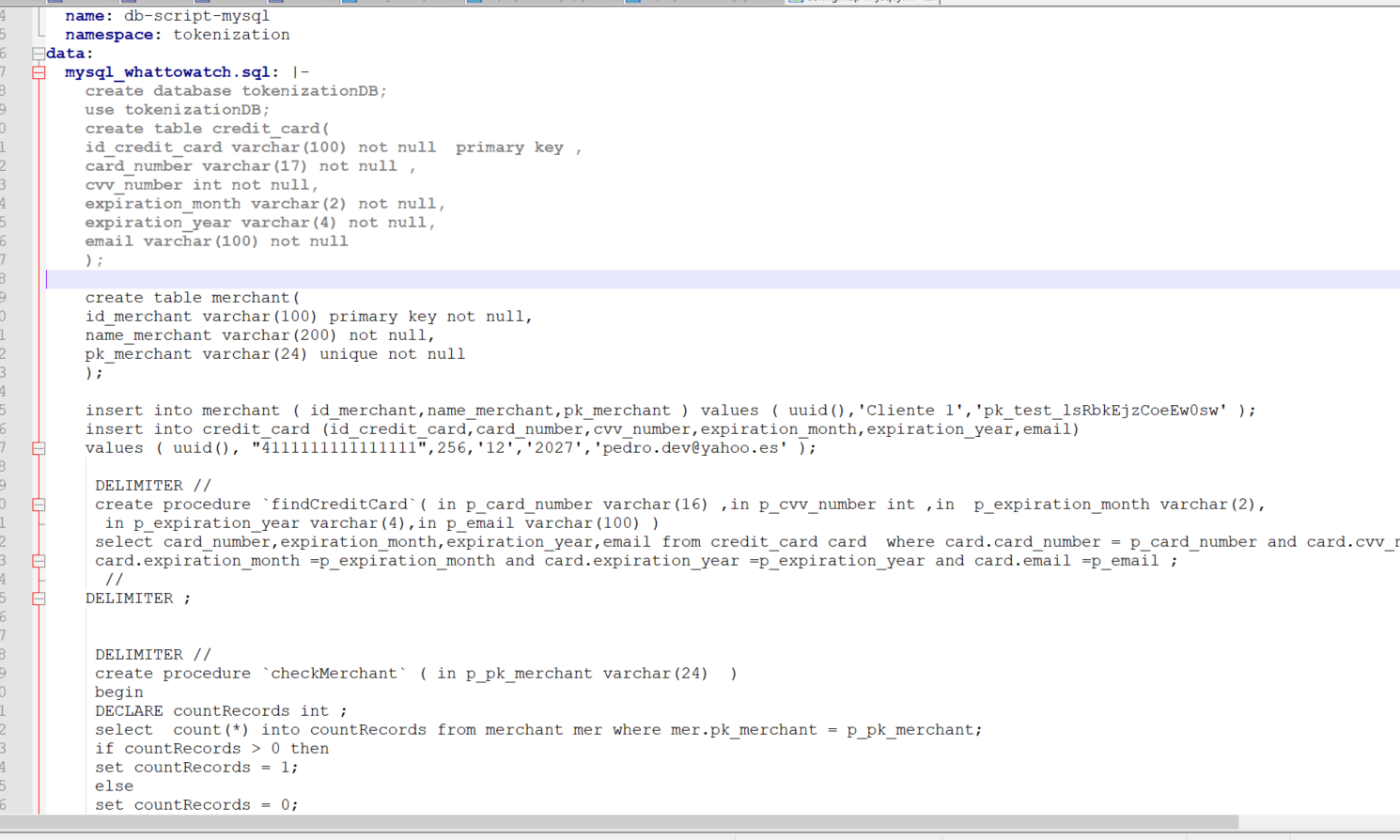
Se debe ubicar en cada carpeta desde la consola cmd, powershell, etc. primero se ejecuta el archivo tokenization-namespace.yml que crea el namespace o grupo lógico para mantener un orden en los despliegues. Para ello se ejecuta el comando kubectl apply -f [nombre del archivo]. 

Luego se ejecuta el comando kubectl create secret generic credentials --from-literal=password='yourpassword' --namespace=tokenization para crear una contraseña oculta en el cluster. Asimismo, en el archivo kubernetes/config-redis.yml se debe colocar la contraseña creada

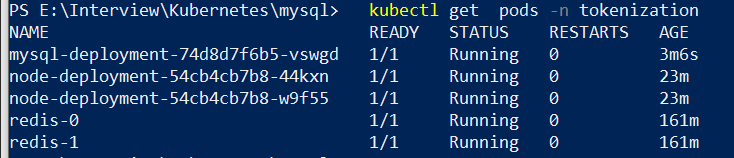


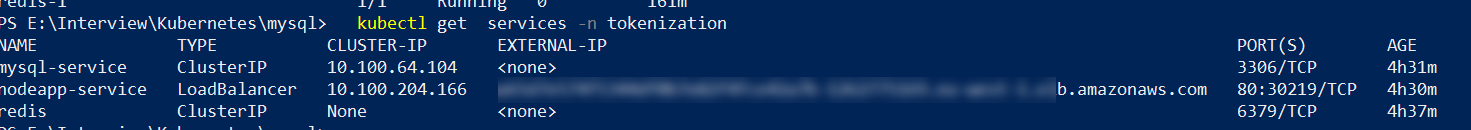
Por ultimo ejecutar todos los archivos yml con el comando kubectl apply -f [nombre del archivo].

Tambien se debe tener en cuenta en que la carpeta mysql/ configmap-mysql.yml se encuentra el script que se ejecutará en la db mysql durante el despliegue



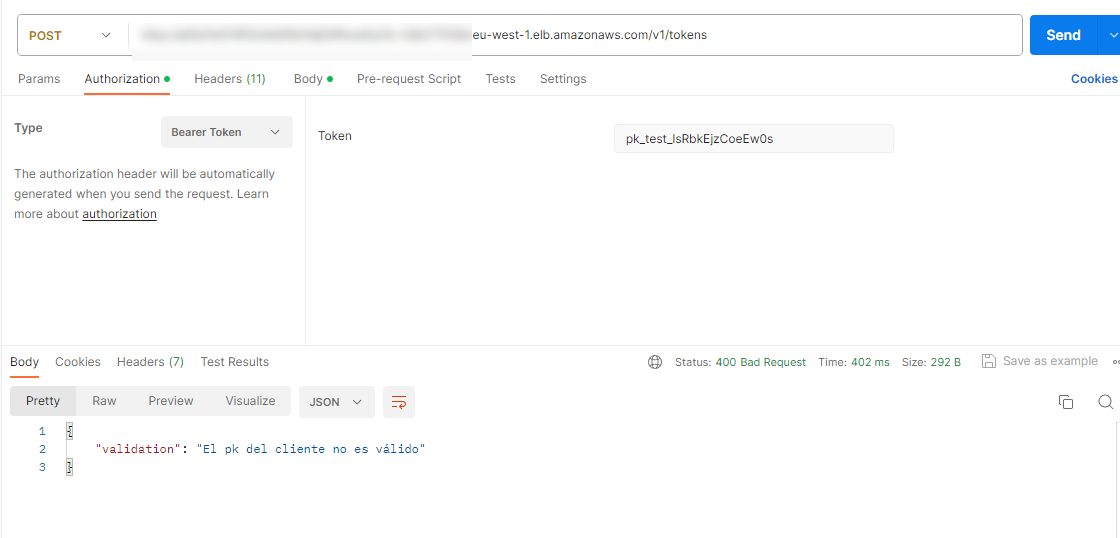
Luego del despliegue se tiene creado los pods, servicios, storage, configs y contenedores mysql, redis y nodejs. En las imágenes se muestra el despliegue de los servicios y pods.

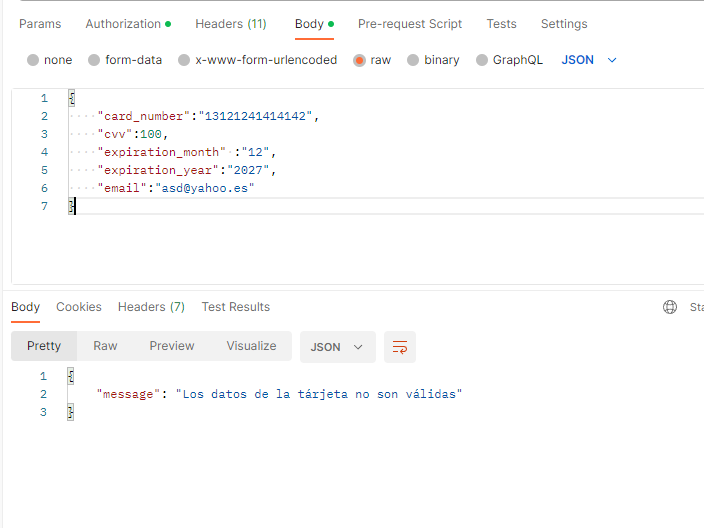


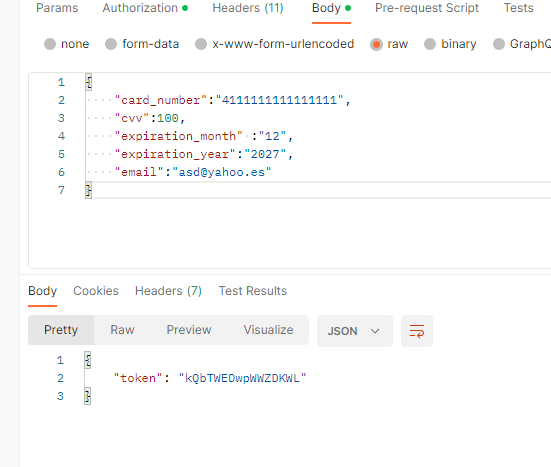


En el servicio nodeapp se encuentra la url externa. Con ello se ejecutara los apis.

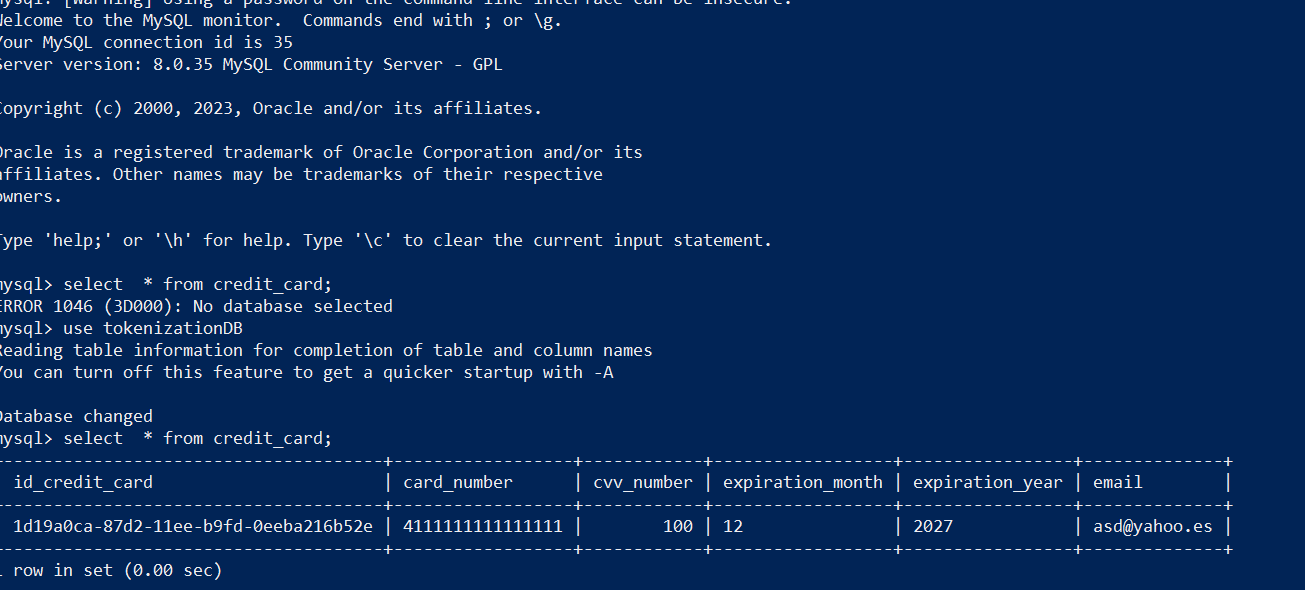
Con ello se puede verificar el funcionamiento.



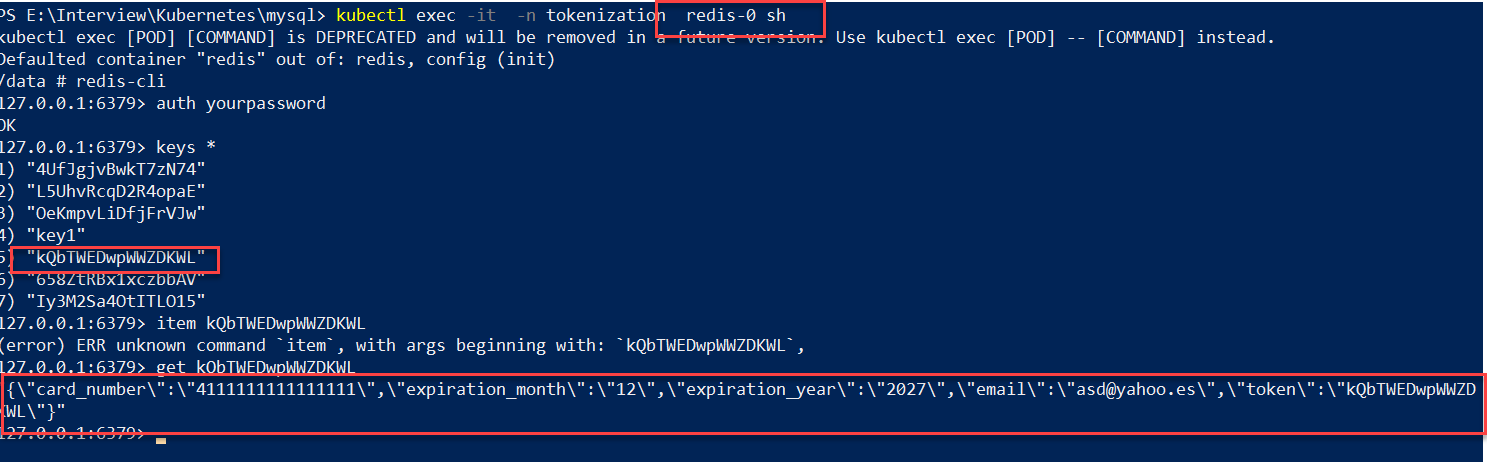


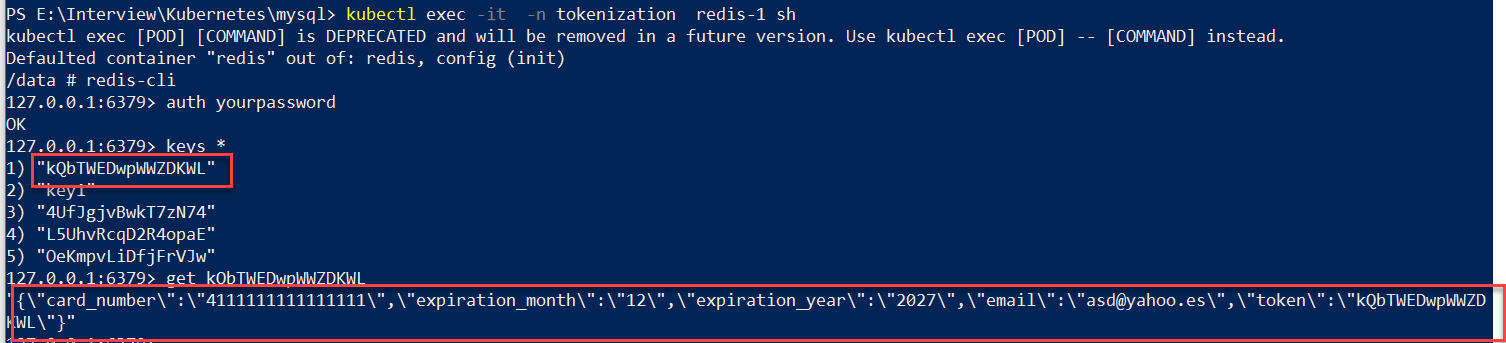


Para validar los datos de la tarjeta de crédito se puede visualizar en el pod de mysql.

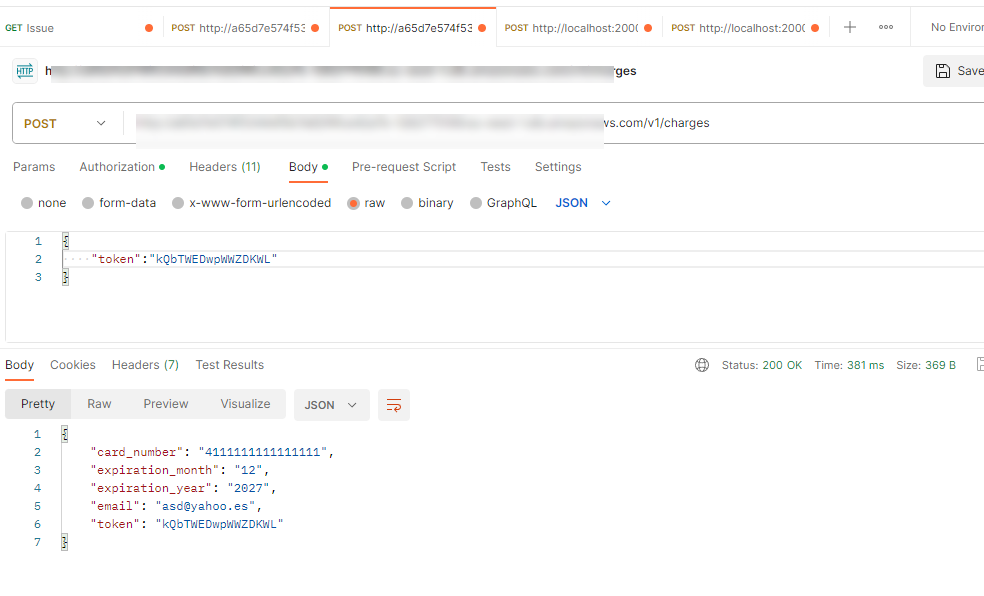


Por otro lado, en la siguiente imagen se puede visualizar el token kQbTWEDwpWWZDKWL almacenado en redis master y slave (replica).





Para la api de consulta de token, en la siguiente imagen se muestra el resultado



Para validar el token expirado, se puede forzar en en el la db redis, ya que tiene la opcion de expirar sus items.

