Desafío No. 11

Bootcamp DevOps 63703

Presentado por:
Marco Vanegas
2023

Actividades:

Haciendo uso de lo visto en clase deberás realizar la publicación de la siguiente API Pública:

https://gameofthronesquotes.xyz/

1. Desplegar una instalación de KONG en Minikube.

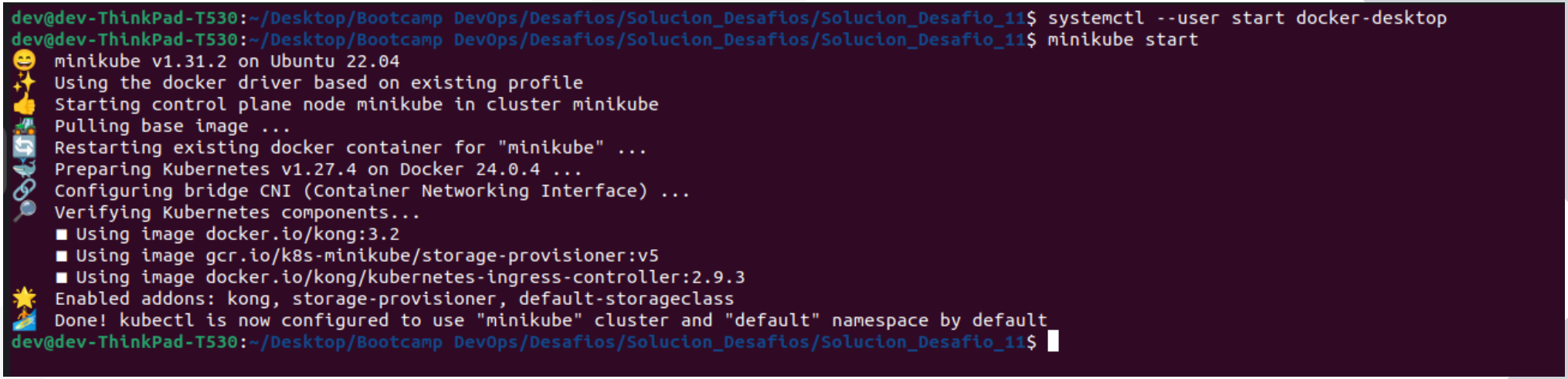
2. Publicar una API haciendo la definición del servicio y router mediante el Admin API de Kong.

3. Mediante algún cliente REST/CURL validar el funcionamiento del API Gateway llamando al método https://api.gameofthronesquotes.xyz/v1/author/tyrion/2. Acordate de actualizar la URL para que implemente tu API Gateway.

4. Agregar algún mecanismo de autenticación del lado del API Gateway.

Illustrations by Pixeltrue on

1. Se inicia Docker Desktop `systemctl --user start docker-desktop` y se inicia minikube `minikube start`.



2. Se crea el namespace `kong` con el comando `kubectl create namespace kong`.

```
dev@dev-ThinkPad-T530:~$ kubectl create namespace kong
Error from server (AlreadyExists): namespaces "kong" already exists
dev@dev-ThinkPad-T530:~$
```

3. Se crean las variables de credenciales y configuración.

```
dev@dev-ThinkPad-T530:~$ kubectl create secret generic kong-config-secret -n kong    --from-literal=portal_session_conf='{"storage":"kong","se
cret":"super_secret_salt_string","cookie_name":"portal_session","cookie_same_site":"Lax","cookie_secure":false}'    --from-literal=admin_gui_s
ession_conf='{"storage":"kong","secret":"super_secret_salt_string","cookie_name":"admin_session","cookie_same_site":"Lax","cookie_secure":fals
e}'    --from-literal=pg_host="enterprise-postgresql.kong.svc.cluster.local"    --from-literal=kong_admin_password=kong    --from-literal=pass
word=kong
error: failed to create secret secrets "kong-config-secret" already exists
dev@dev-ThinkPad-T530:~$
```

4. Se crea un secreto de licencia para Kong Gateway Free Mode:

kubectl create secret generic kong-enterprise-license --from-literal=license='"{}"' -n kong --dry-run=client -o yaml | kubectl apply -f -

```
dev@dev-ThinkPad-T530:~$ kubectl create secret generic kong-enterprise-license --from-literal=license="'{}'" -n kong --dry-run=client -o yaml
| kubectl apply -f -
secret/kong-enterprise-license created
dev@dev-ThinkPad-T530:~$
```

5. Se agrega el repositorio de Helm de Jetstack Cert Manager `helm repo add jetstack https://charts.jetstack.io ; helm repo update`.

```
dev@dev-ThinkPad-T530:~$ helm repo add jetstack https://charts.jetstack.io ; helm repo update
"jetstack" has been added to your repositories
Hang tight while we grab the latest from your chart repositories...
...Successfully got an update from the "kong" chart repository
...Successfully got an update from the "jetstack" chart repository
...Successfully got an update from the "jenkins" chart repository
Update Complete. ⎈Happy Helming!⎈
dev@dev-ThinkPad-T530:~$
```

6. Se instala Cert Manager `helm upgrade --install cert-manager jetstack/cert-manager \ --set installCRDs=true --namespace cert-manager --create-namespace`.

```
dev@dev-ThinkPad-T530:~$ helm upgrade --install cert-manager jetstack/cert-manager \
   --set installCRDs=true --namespace cert-manager --create-namespace
Release "cert-manager" does not exist. Installing it now.
NAME: cert-manager
LAST DEPLOYED: Sat Dec 16 19:52:06 2023
NAMESPACE: cert-manager
STATUS: deployed
REVISION: 1
TEST SUITE: None
NOTES:
cert-manager v1.13.3 has been deployed successfully!

In order to begin issuing certificates, you will need to set up a ClusterIssuer
or Issuer resource (for example, by creating a 'letsencrypt-staging' issuer).

More information on the different types of issuers and how to configure them
can be found in our documentation:

https://cert-manager.io/docs/configuration/

For information on how to configure cert-manager to automatically provision
Certificates for Ingress resources, take a look at the `ingress-shim`
documentation:

https://cert-manager.io/docs/usage/ingress/
dev@dev-ThinkPad-T530:~$
```

7. Se crea un emisor de certificado autofirmado.

```
dev@dev-ThinkPad-T530:~$ bash -c "cat <<EOF | kubectl apply -n kong -f -
apiVersion: cert-manager.io/v1
kind: Issuer
metadata:
 name: quickstart-kong-selfsigned-issuer-root
spec:
 selfSigned: {}
---
apiVersion: cert-manager.io/v1
kind: Certificate
metadata:
 name: quickstart-kong-selfsigned-issuer-ca
spec:
 commonName: quickstart-kong-selfsigned-issuer-ca
 duration: 2160h0m0s
    true
    Ref:
    p: cert-manager.io
   kind: Issuer
   name: quickstart-kong-selfsigned-issuer-root
 privateKey:
   algorithm: ECDSA
   size: 256
 renewBefore: 360h0m0s
 secretName: quickstart-kong-selfsigned-issuer-ca
---
apiVersion: cert-manager.io/v1
kind: Issuer
metadata:
 name: quickstart-kong-selfsigned-issuer
spec:
 ca:
   secretName: quickstart-kong-selfsigned-issuer-ca
EOF"
issuer.cert-manager.io/quickstart-kong-selfsigned-issuer-root created
certificate.cert-manager.io/quickstart-kong-selfsigned-issuer-ca created
issuer.cert-manager.io/quickstart-kong-selfsigned-issuer created
dev@dev-ThinkPad-T530:~$
```

8. Se agrega el repositorio de Helm de Kong `helm repo add kong https://charts.konghq.com ; helm repo update`.

```
dev@dev-ThinkPad-T530:~$ helm repo add kong https://charts.konghq.com ; helm repo update
"kong" has been added to your repositories
Hang tight while we grab the latest from your chart repositories...
...Successfully got an update from the "kong" chart repository
...Successfully got an update from the "jenkins" chart repository
Update Complete. ⎈Happy Helming!⎈
dev@dev-ThinkPad-T530:~$
```

9. Se instala Kong `helm install quickstart kong/kong --namespace kong --values https://bit.ly/KongGatewayHelmValuesAIO`.

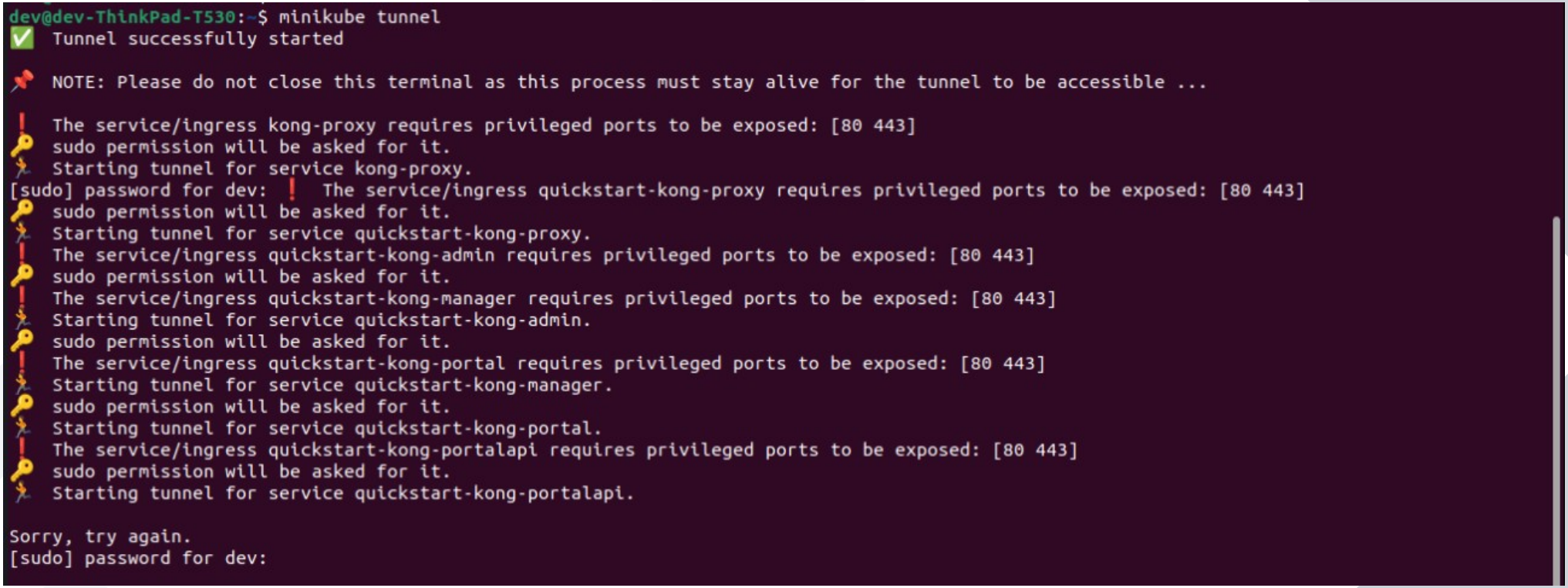
```
dev@dev-ThinkPad-T530:~$ helm install quickstart kong/kong --namespace kong --values https://bit.ly/KongGatewayHelmValuesAIO
NAME: quickstart
LAST DEPLOYED: Sat Dec 16 19:37:57 2023
NAMESPACE: kong
STATUS: deployed
REVISION: 1
TEST SUITE: None
NOTES:
To connect to Kong, please execute the following commands:

HOST=$(kubectl get svc --namespace kong quickstart-kong-proxy -o jsonpath='{.status.loadBalancer.ingress[0].ip}')
PORT=$(kubectl get svc --namespace kong quickstart-kong-proxy -o jsonpath='{.spec.ports[0].port}')
export PROXY_IP=${HOST}:${PORT}
curl $PROXY_IP

Once installed, please follow along the getting started guide to start using
Kong: https://docs.konghq.com/kubernetes-ingress-controller/latest/guides/getting-started/
dev@dev-ThinkPad-T530:~$
```

10. Se espera a que los pods estén completados y corriendo. Para ello se ejecuta `kubectl get po --namespace kong -w`.

```
dev@dev-ThinkPad-T530:~$ kubectl get po --namespace kong -w
NAME                                 READY   STATUS              RESTARTS      AGE
quickstart-kong-74486b857c-zppcw     0/2     Init:0/2            0             92s
quickstart-kong-init-migrations-lz46v   0/1     Init:0/1            0             89s
quickstart-postgresql-0              0/1     ContainerCreating   0             87s
quickstart-kong-init-migrations-lz46v   0/1     Init:0/1            0             104s
quickstart-kong-74486b857c-zppcw     0/2     Init:1/2            0             114s
quickstart-kong-74486b857c-zppcw     0/2     Init:1/2            0             2m1s
quickstart-postgresql-0              0/1     Running             0             2m24s
quickstart-postgresql-0              1/1     Running             0             2m40s
quickstart-kong-init-migrations-lz46v   0/1     PodInitializing     0             2m46s
quickstart-kong-init-migrations-lz46v   1/1     Running             0             2m51s
quickstart-kong-init-migrations-lz46v   0/1     Completed           0             4m1s
quickstart-kong-init-migrations-lz46v   0/1     Completed           0             4m4s
quickstart-kong-init-migrations-lz46v   0/1     Completed           0             4m5s
quickstart-kong-init-migrations-lz46v   0/1     Completed           0             4m6s
quickstart-kong-74486b857c-zppcw     0/2     PodInitializing     0             4m13s
quickstart-kong-74486b857c-zppcw     0/2     Error               0             4m37s
quickstart-kong-74486b857c-zppcw     0/2     Error               1 (9s ago)    4m41s
quickstart-kong-74486b857c-zppcw     0/2     CrashLoopBackOff    1 (4s ago)    4m42s
quickstart-kong-74486b857c-zppcw     1/2     CrashLoopBackOff    1 (4s ago)    4m42s
quickstart-kong-74486b857c-zppcw     1/2     Running             2 (23s ago)   5m1s
quickstart-kong-74486b857c-zppcw     2/2     Running             2 (35s ago)   5m13s
```

11. Se habilita Kong Ingress Controller mediante el comando `minikube addons enable kong`.

12. Se ejecuta `minikube tunnel` para habilitar el acceso a la API desde el exterior.



13. Se valida el funcionamiento del Kong Ingress Controller (KIC) mediante la ejecución del comando `curl -v localhost`.

```
dev@dev-ThinkPad-T530:~$ curl -v localhost
*   Trying 127.0.0.1:80...
* Connected to localhost (127.0.0.1) port 80 (#0)
> GET / HTTP/1.1
> Host: localhost
> User-Agent: curl/7.81.0
> Accept: */*
>
* Mark bundle as not supporting multiuse
< HTTP/1.1 404 Not Found
< Date: Sun, 17 Dec 2023 03:53:35 GMT
< Content-Type: application/json; charset=utf-8
< Connection: keep-alive
< Content-Length: 48
< X-Kong-Response-Latency: 0
< Server: kong/3.2.2
<
* Connection #0 to host localhost left intact
{"message":"no Route matched with those values"}dev@dev-ThinkPad-T530:~$
```

14. Se crea un servicio mediante el Admin API de Kong:

```
curl -i -s -X POST http://localhost:8001/services \
--data name=service_game_of_thrones \
--data url='https://api.gameofthronesquotes.xyz/v1/author/tyrion/2'
```

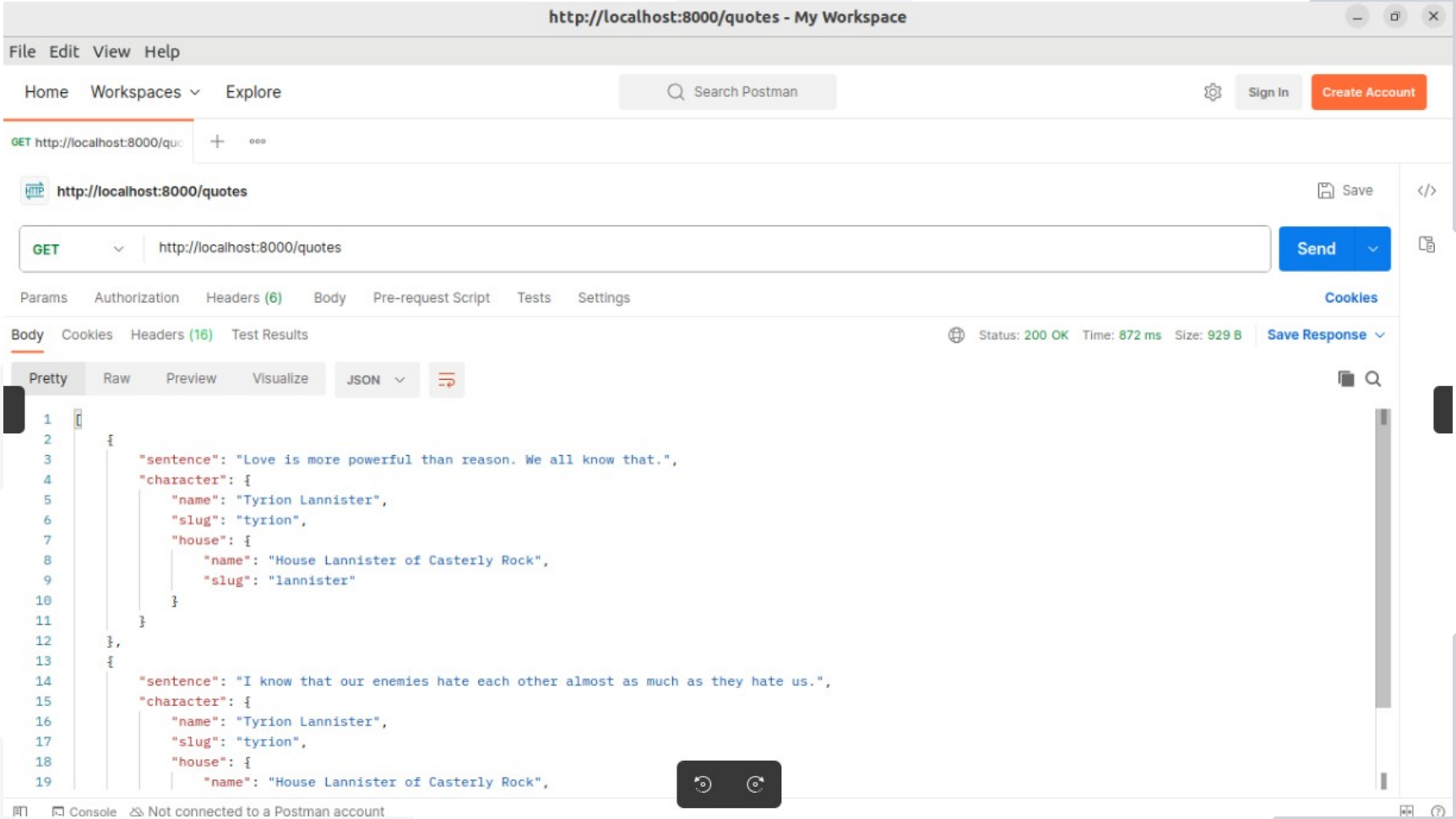
```
dev@dev-ThinkPad-T530:~/Desktop/Bootcamp DevOps/Desafios/Solucion_Desafios/Solucion_Desafio_11$ curl -i -s -X POST http://localhost:8001/servi
ces \
  --data name=game_of_thrones \
  --data url='https://gameofthronesquotes.xyz/'
HTTP/1.1 201 Created
Date: Sun, 17 Dec 2023 04:45:35 GMT
Content-Type: application/json; charset=utf-8
Connection: keep-alive
Access-Control-Allow-Origin: *
Server: kong/2.0.4
Content-Length: 310
X-Kong-Admin-Latency: 110

{"host":"gameofthronesquotes.xyz","created_at":1702788335,"connect_timeout":60000,"id":"fef4a448-1c90-4b6f-881f-39f2003274b3","protocol":"http
s","name":"game_of_thrones","read_timeout":60000,"port":443,"path":"\/","updated_at":1702788335,"retries":5,"write_timeout":60000,"tags":null,
"client_certificate":null}dev@dev-ThinkPad-T530:~/Desktop/Bootcamp DevOps/Desafios/Solucion_Desafios/Solucion_Desafio_11$
```

15. Se crea una ruta mediante el Admin API de Kong:

```
curl -i -X POST http://localhost:8001/services/service_game_of_thrones/routes \
--data 'paths[]=/quotes' \
--data 'methods[]=GET' \
--data name=route_game_of_thrones
```

```
dev@dev-ThinkPad-T530:~$ curl -i -X POST http://localhost:8001/services/game_of_thrones/routes \
    --data 'paths[]=/v1/author/tyrion/2' \
    --data name=game_of_thrones
HTTP/1.1 201 Created
Date: Sun, 17 Dec 2023 05:17:46 GMT
Content-Type: application/json; charset=utf-8
Connection: keep-alive
Access-Control-Allow-Origin: *
Server: kong/2.0.4
Content-Length: 454
X-Kong-Admin-Latency: 61

{"id":"cfc8a951-a9dd-49ff-a398-01d521028f3c","path_handling":"v0","paths":["\/v1\/author\/tyrion\/2"],"destinations":null,"headers":null,"prot
ocols":["http","https"],"methods":null,"snis":null,"service":{"id":"fef4a448-1c90-4b6f-881f-39f2003274b3"},"name":"game_of_thrones","strip_pat
h":true,"preserve_host":false,"regex_priority":0,"updated_at":1702790266,"sources":null,"hosts":null,"https_redirect_status_code":426,"tags":n
ull,"created_at":1702790266}dev@dev-ThinkPad-T530:~$
```

16. Se valida el funcionamiento mediante Postman o cURL: curl --location 'http://localhost:8000/quotes'



17. Se configura un consumidor con su respectiva clave:

curl -i -X POST http://localhost:8001/consumers/ \
--data username=marco

curl -i -X POST http://localhost:8001/consumers/marco/key-auth

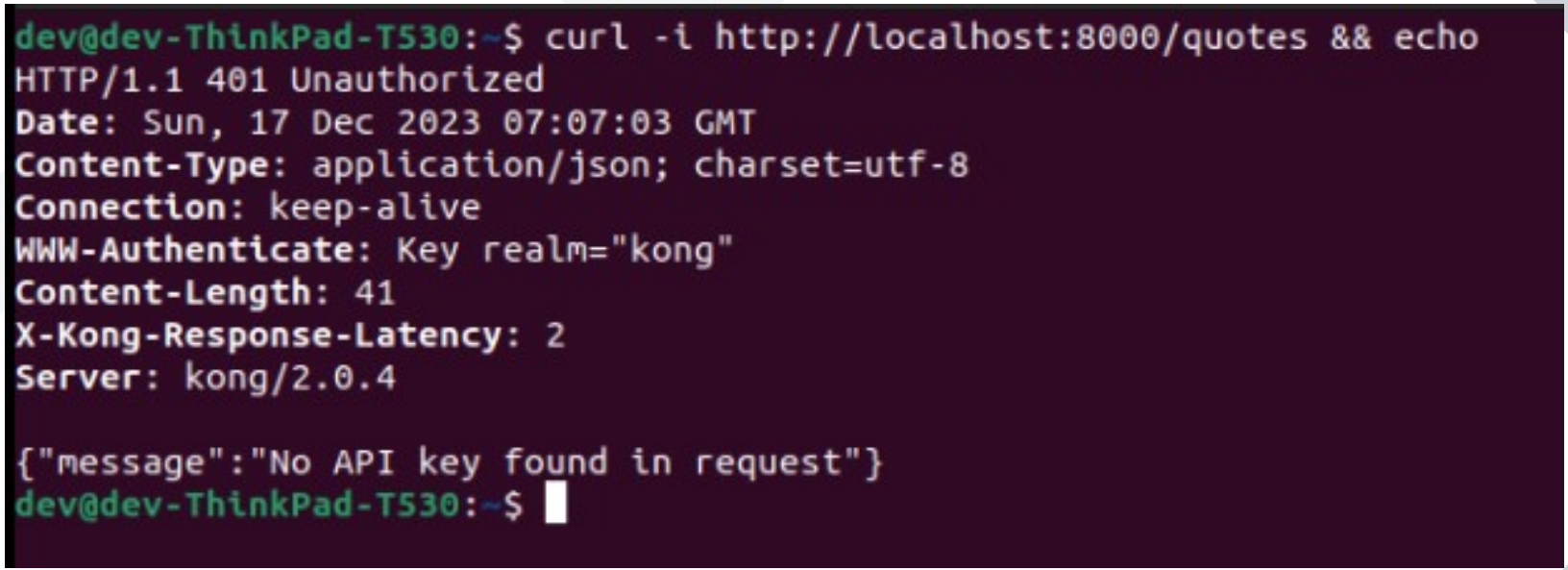
```
dev@dev-ThinkPad-T530:~$ curl -i -X POST http://localhost:8001/consumers/ \
    --data username=marco
HTTP/1.1 201 Created
Date: Sun, 17 Dec 2023 07:26:17 GMT
Content-Type: application/json; charset=utf-8
Connection: keep-alive
Access-Control-Allow-Origin: *
Server: kong/2.0.4
Content-Length: 117
X-Kong-Admin-Latency: 75

{"custom_id":null,"created_at":1702797977,"id":"47fdf1af-7574-45f2-ad41-23741eded1b0","tags":null,"username":"marco"}dev@dev-ThinkPad-T530:~$
```

18. Se habilita la autenticación de clave global mediante el Admin API de Kong:

```
curl -X POST http://localhost:8001/plugins/ \
--data "name=key-auth"  \
--data "config.key_names=apikey"
```

```
dev@dev-ThinkPad-T530:~$ curl -i -X POST http://localhost:8001/plugins \
    --data name=key-auth \
    --data config.key_names=apikey \
    --data config.hide_credentials=true
HTTP/1.1 201 Created
Date: Sun, 17 Dec 2023 06:59:29 GMT
Content-Type: application/json; charset=utf-8
Connection: keep-alive
Access-Control-Allow-Origin: *
Server: kong/2.0.4
Content-Length: 321
X-Kong-Admin-Latency: 54

{"created_at":1702796369,"config":{"key_names":["apikey"],"run_on_preflight":true,"anonymous":null,"hide_credentials":true,"key_in_body":false
},"id":"8d31d53e-0f36-4947-b4c6-c30964b41e12","service":null,"enabled":true,"protocols":["grpc","grpcs","http","https"],"name":"key-auth","con
sumer":null,"route":null,"tags":null}dev@dev-ThinkPad-T530:~$
```

19. Se valida la restricción (código de estado de respuesta HTTP 401) mediante la ejecución del comando `curl -i http://localhost:8000/quotes`.

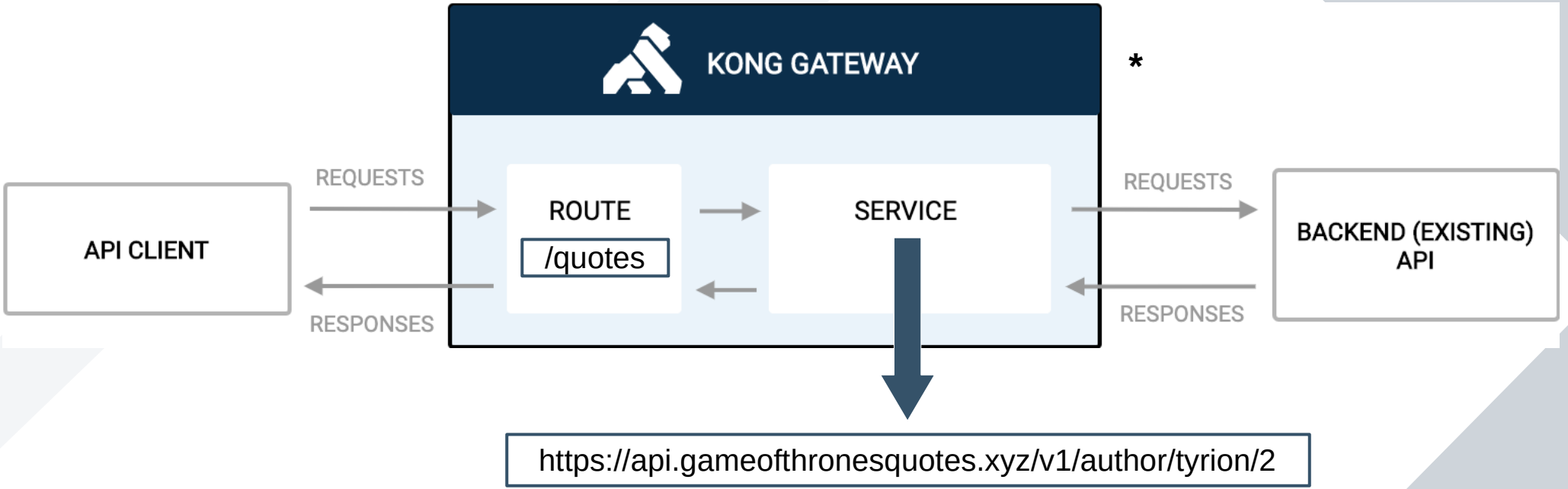


20. Se valida el consumo de la API empleando la clave generada en el paso No. 17:

curl -i http://localhost:8000/quotes -H 'apikey:UT6Lr2huYG8v6qo0UMS3dhjxqzLDZN8W'

```
dev@dev-ThinkPad-T530:~$ curl -i http://localhost:8000/quotes -H 'apikey:UT6Lr2huYG8v6qo0UMS3dhjxqzLDZN8W'
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
Content-Length: 395
Connection: keep-alive
Access-Control-Allow-Origin: *
Age: 0
Cache-Control: public, max-age=0, must-revalidate
Date: Sun, 17 Dec 2023 07:27:34 GMT
Etag: W/"18b-OER9tyccV03UwsRpgZNEdiuy19w"
Server: Vercel
Strict-Transport-Security: max-age=63072000
X-Powered-By: Express
X-Vercel-Cache: MISS
X-Vercel-Id: iad1::iad1::dwcwd-1702798054509-a385b011d696
X-Kong-Upstream-Latency: 351
X-Kong-Proxy-Latency: 4
Via: kong/2.0.4

[{"sentence":"Death is so terribly final, while life is full of possibilities.","character":{"name":"Tyrion Lannister","slug":"tyrion","house"
:{"name":"House Lannister of Casterly Rock","slug":"lannister"}}},{"sentence":"That's what I do: I drink and I know things.","character":{"nam
e":"Tyrion Lannister","slug":"tyrion","house":{"name":"House Lannister of Casterly Rock","slug":"lannister"}}}]dev@dev-ThinkPad-T530:~$
```

Diagrama de alto nivel de cómo están desplegados los componentes



* Esquema basado en la imagen https://docs.konghq.com/assets/images/products/gateway/getting-started-guide/route-and-service.png

Repositorio de GitHub donde se encuentra esta presentación:

https://github.com/BambooThink/BootcampDevOps2023/tree/main/Solucion_Desafio_11