Route Not Found Troubleshooting

Initial Issue

Cannot access the machine pod from outside the server.

This problem is caused by the built in kong that is included with the RIC Installation inside namespace ricplt.

```
kubectl get svc -n ricplt
```

Output:

```
⋮
⋮
r4-infrastructure-kong-manager          NodePort       10.99.249.248     <none>       8002:32003/TCP,8445:31632/TCP   9d
r4-infrastructure-kong-proxy            LoadBalancer   10.104.222.44     <pending>    80:32080/TCP,443:32443/TCP      9d
⋮
⋮
```

Problems:

- The proxy route is not working
- The kong manager web dashboard is not responsive

Solution

Installing new kong in docker container, outside ricplt.

Run the instalation script provided

```
cd kong-setup/
```

Output:

```
• Deploying Kong Gateway to Docker...
i Debugging info logged to:
    /tmp/kong/quickstart/kong-quickstart.log
▯ Downloading Docker images... ✔
▯ Destroying previous kong-quickstart containers... ✔
▯ Starting database... ✔
▯ Starting Kong Gateway... ✔
▯ Validating kong state... ✔

☻ Kong Gateway Ready ☻


=======================================================
 ⚒                           Using Kong Gateway
=======================================================

• Kong Gateway Data Plane endpoints:
    HTTP  = http://localhost:8000
    HTTPS = https://localhost:8443

• This script has written an environment file you can
  source to make connecting to Kong Gateway easier.
  Run this command to source these variables into your
  environment:

→ source /tmp/kong/quickstart/kong.env

• Now you can make requests to your Kong Gateway using
  variables from the file, for example:

→ curl -s $KONG_PROXY/mock


=======================================================
 ⚒                           Administer Kong Gateway
=======================================================

• Kong Gateway Admin API endpoint:
    HTTP = http://localhost:8001

• To administer the gateway with curl:

→ curl -s https://localhost:8001

• To stop the gateway run:

→ curl -s https://get.konghq.com/quickstart | \
    bash -s -- -d -a kong-quickstart

Or remove the Docker container directly.
```

Verifying Testing the newly installed kong

Verify Docker container

```
docker ps -a
```

Output:

```
CONTAINER ID   IMAGE                          COMMAND                  CREATED          STATUS                    PORTS
df8b68a17bd5   kong/kong-gateway:3.7.0.0      "/entrypoint.sh kong…"   About a minute ago   Up About a minute (healthy)   0.0.0.0:
da77a869bc95   postgres:13                    "docker-entrypoint.s…"   About a minute ago   Up About a minute         5432/tcp
```

Verify the kong is running

```
curl --head localhost:8001
```

Output:

```
HTTP/1.1 200 OK
Date: Thu, 18 Jul 2024 01:54:26 GMT
Content-Type: application/json; charset=utf-8
Connection: keep-alive
Access-Control-Allow-Origin: *
X-Kong-Admin-Request-ID: 97fbe95dc76f0123958cf2242b3bef34
Content-Length: 20990
X-Kong-Admin-Latency: 241
Server: kong/3.7.0.0-enterprise-edition
```

Configuring the proxy and route

Create gateway proxy

```
#Create proxy that links to a1mediator service in http://10.96.102.138:10000
curl -i -X POST http://localhost:8001/services/ \
>    --data name=a1mediator-service \
>    --data url=http://10.96.102.138:10000
```

Output:

```
HTTP/1.1 201 Created
Date: Thu, 18 Jul 2024 02:05:24 GMT
Content-Type: application/json; charset=utf-8
Connection: keep-alive
Access-Control-Allow-Origin: *
X-Kong-Admin-Request-ID: b4fb24165636582f8866585f0128ae0a
Content-Length: 384
X-Kong-Admin-Latency: 17
Server: kong/3.7.0.0-enterprise-edition

{"connect_timeout":60000,"path":null,"id":"b58a0be9-5964-4b8d-a287-de8a47db575d","tls_verify":null,"tls_verify_depth":null,"ca_certi
```
◄ ▬▬▬▬▬▬▬▬▬▬▬ ►

Create Route

```
#Create route that access that newly created proxy that can be accessed from localhost:8000/a1mediator
curl -i -X POST http://localhost:8001/services/a1mediator-service/routes \
>    --data 'paths[]=/a1mediator'
```

Output:

```
HTTP/1.1 201 Created
Date: Thu, 18 Jul 2024 02:05:30 GMT
Content-Type: application/json; charset=utf-8
Connection: keep-alive
Access-Control-Allow-Origin: *
X-Kong-Admin-Request-ID: 6949b55e48b749283dda6174f5fd282e
Content-Length: 480
X-Kong-Admin-Latency: 19
Server: kong/3.7.0.0-enterprise-edition

{"hosts":null,"paths":["/a1mediator"],"request_buffering":true,"response_buffering":true,"snis":null,"tags":null,"created_at":172126
```
◄ ▬▬▬▬▬▬▬▬▬▬▬ ►

The newly configured kong

```
#Using A1-P healthcheck API here
curl -i -X GET http://localhost:8000/a1mediator/A1-P/v2/healthcheck
```

Output:

```
HTTP/1.1 200 OK
Content-Length: 0
Connection: keep-alive
Date: Thu, 18 Jul 2024 02:06:05 GMT
X-Kong-Upstream-Latency: 1
X-Kong-Proxy-Latency: 2
Via: kong/3.7.0.0-enterprise-edition
X-Kong-Request-Id: b5913904e93e7e3498744b253612da1d
```

If the response is 200 OK, it means the kong proxy is already running correctly/

```
  #Using A1-P healthcheck API here
  curl -i -X GET http://localhost:8000/a1mediator/A1-P/v2/healthcheck
```

```
  HTTP/1.1 200 OK
  Content-Length: 0
  Connection: keep-alive
  Date: Thu, 18 Jul 2024 02:06:05 GMT
  X-Kong-Upstream-Latency: 1
  X-Kong-Proxy-Latency: 2
  Via: kong/3.7.0.0-enterprise-edition
  X-Kong-Request-Id: b5913904e93e7e3498744b253612da1d
```