

Installing APISIX

1. Open a terminal on your VM.
2. Make a folder for APISIX:

```
sudo mkdir -p /opt/apisix
```

3. Create the file `/opt/apisix/apisix.yaml`:

```
sudo nano /opt/apisix/apisix.yaml
```

4. Copy and paste exactly this content into `apisix.yaml`. Replace `YOUR_BASE64_DEMO_TOKEN` and `YOUR_BASE64_LIVE_TOKEN` with your real Base64(API_KEY:SECRET_KEY) strings:

```
apisix:
  node_listen: 9080 # APISIX will listen on port 9080 for HTTP

consumers:
  - username: finmars # a name for this client
    plugins:
      basic-auth:
        username: foo # user name you want to use for APISIX basic auth
        password: bar # password you want to use for APISIX basic auth

upstreams:
  - id: 1
    nodes:
      api-demo.exante.eu:443: 1
    scheme: https
    pass_host: node
    type: roundrobin

  - id: 2
    nodes:
      api-live.exante.eu:443: 1
    scheme: https
    pass_host: node
    type: roundrobin

plugin_configs:
```

```
- id: 1
  plugins:
    basic-auth: {}
    proxy-rewrite:
      headers:
        set:
          # Replace this with your Base64(API_KEY:SECRET_KEY) for the demo
environment
          Authorization: "Basic %TOKEN%"
      regex_uri:
        - "^/demo/(.*)"
        - "/$1"

- id: 2
  plugins:
    basic-auth: {}
    proxy-rewrite:
      headers:
        set:
          # Replace this with your Base64(API_KEY:SECRET_KEY) for the live
environment
          Authorization: "Basic %TOKEN%"
      regex_uri:
        - "^/live/(.*)"
        - "/$1"

routes:
  - uris:
    - /demo/md/*/accounts
    - /demo/md/*/symbols/*
    - /demo/md/*/summary/*
    - /demo/md/*/ohlc/*
    - /demo/md/*/transactions
    - /demo/md/*/crossrates/*
    - /demo/trade/*/orders/*
  upstream_id: 1
  plugin_config_id: 1

- uris:
```

```
- /live/md/*/accounts
- /live/md/*/symbols/*
- /live/md/*/summary/*
- /live/md/*/ohlc/*
- /live/md/*/transactions
- /live/md/*/crossrates/*
- /live/trade/*/orders/*

upstream_id: 2
plugin_config_id: 2

deployment:
  role: data_plane
  role_data_plane:
    config_provider: yaml

#END
```

5. Save and close the file:

- Press `Ctrl+O`, then `Enter` to save.
- Press `Ctrl+X` to exit Nano.

2. Create a restart script for APISIX

1. In the terminal, make a new script:

```
nano /opt/apisix/restart_apisix.sh
```

2. Copy and paste this into `restart_apisix.sh`:

```
#!/bin/bash

# If a container named "apache-apisix" exists, stop and remove it
if docker ps -a --format '{{.Names}}' | grep -Eq "^apache-apisix\$"; then
  echo "Stopping and removing existing apache-apisix container..."
  docker stop apache-apisix
  docker rm apache-apisix
fi
```

```
echo "Starting a new APISIX container..."
docker run -d \
  --name apache-apisix \
  -p 9080:9080 \
  -e APISIX_STAND_ALONE=true \
  -v /opt/apisix/apisix.yaml:/usr/local/apisix/conf/config.yaml \
  -v /opt/apisix/apisix.yaml:/usr/local/apisix/conf/apisix.yaml \
  apache/apisix

echo "APISIX container is now running."
```

3. Save and close:

- Press `Ctrl+O`, then `Enter`.
- Press `Ctrl+X`.

4. Make the script executable:

```
chmod +x /opt/apisix/restart_apisix.sh
```

3. Run APISIX for the first time

1. In the terminal, run:

```
./restart_apisix.sh
```

2. Wait a few seconds. APISIX will start in Docker, listening on port 9080, reading your `/opt/apisix/apisix.yaml` as both `config.yaml` and `apisix.yaml`.

3. Check that it is running:

```
sudo docker ps
```

You should see a line for `apache/apisix` with `0.0.0.0:9080->9080/tcp`.

4. Look at the logs to ensure no errors:

```
sudo docker logs apache-apisix
```

You should see messages like “config file ... reloaded” and no errors.

4. Test APISIX locally with curl (no Nginx yet)

1. In the terminal, run:

```
curl -u foo:bar http://127.0.0.1:9080/demo/md/3.0/accounts
```

- `-u foo:bar` sends your Basic Auth.
- The path `/demo/md/3.0/accounts` matches your route pattern `/demo/md/*/accounts`.

2. If everything is correct, you will see JSON returned from Exante.

If you see `{"error_msg": "404 Route Not Found"}`, double-check that the path matches exactly and that your tokens are correct.

Revision #5

Created 2025-06-03 15:46:07 UTC by Sergei Zhitenev

Updated 2025-06-04 09:38:27 UTC by Sergei Zhitenev