Annexe B

Infrastructure Docker OpenZiti N2

Cette annexe est une marche à suivre pour reproduire l'installation de la deuxième infrastructure OpenZiti déployée avec Docker (section 10.2).

https://github.com/MehSalhi/TB-ZeroTrust-OpenZiti/tree/master/Network/N2

B.1 Composants

- contrôleur OpenZiti
- initialisateur du contrôleur
- edge router OpenZiti Cloud
- edge router OpenZiti Base
- edge router OpenZiti GCS
- console web OpenZiti
- drone
- gcs
- base
- police fédérale
- serveur vidéo qui reçoit le flux et le retransmet à des clients

B.2 Pré-requis

Les logiciels suivants doivent être installés :

- Docker
- Docker compose

B.3 Marche à suivre

B.3.1 Environnement

Créer un fichier suivant :

(Fichier disponible sur Github dans le répertoire ‘n2/‘)

Code source B.3.1: docker-compose.yaml

```yaml
# Auteur      : Mehdi Salhi
# Auteur      : OpenZiti documentation
# Sujet       : Travail de Bachelor Zero Trust OpenZiti
# But         : Déploie une infrastructure OpenZiti
# No          : n2
# Description : 1 contrôleur, 3 routeur, 1 initialisteur, 1 serveur vidéo, 1
#               drone, 1 gcs, 1 base. Tous les fichiers sont sauvegardés dans
#         le répertoire local "vol", et monté dans le répertoire "persistent"
#         sur les containers
version: '2.4'
services:
  # openziti controller
  ziti-controller:
    image: "${ZITI_IMAGE}:${ZITI_VERSION}"
    env_file:
      - ./.env
    ports:
      - ${ZITI_EDGE_CONTROLLER_PORT:-1280}:${ZITI_EDGE_CONTROLLER_PORT:-1280}
      - ${ZITI_CTRL_PORT:-6262}:${ZITI_CTRL_PORT:-6262}
    environment:
      - ZITI_EDGE_IDENTITY_ENROLLMENT_DURATION=${ZITI_EDGE_IDENTITY_ENROLLMENT_DURATION}
      - ZITI_EDGE_ROUTER_ENROLLMENT_DURATION=${ZITI_EDGE_ROUTER_ENROLLMENT_DURATION}
    volumes:
      - type: bind
        source: ./vol
        target: /persistent
    entrypoint:
      - "/var/openziti/scripts/run-controller.sh"
    networks:
      internet:
         aliases:
           - ziti-edge-controller


  # controller init
  ziti-controller-init-container:
    image: "${ZITI_IMAGE}:${ZITI_VERSION}"
    depends_on:
      - ziti-controller
    environment:
      - ZITI_CONTROLLER_RAWNAME="${ZITI_CONTROLLER_RAWNAME}"
      - ZITI_EDGE_CONTROLLER_RAWNAME="${ZITI_EDGE_CONTROLLER_RAWNAME}"
    env_file:
      - ./.env
    networks:
      - internet
    volumes:
      - type: bind
        source: ./vol
        target: /persistent
    entrypoint:
      - "/var/openziti/scripts/run-with-ziti-cli.sh"
    command:
      - "/var/openziti/scripts/access-control.sh"

###################################
# Routers
###################################

  # edge router cloud
  ziti-edge-router-cloud:
    image: "${ZITI_IMAGE}:${ZITI_VERSION}"
    hostname: ziti-edge-router-cloud
    depends_on:
      - ziti-controller
    environment:
      - ZITI_CONTROLLER_RAWNAME="${ZITI_CONTROLLER_RAWNAME}"
      - ZITI_EDGE_CONTROLLER_RAWNAME="${ZITI_EDGE_CONTROLLER_RAWNAME}"
      - ZITI_EDGE_ROUTER_RAWNAME=${ZITI_EDGE_ROUTER_RAWNAME:-ziti-edge-router-cloud}
      - ZITI_EDGE_ROUTER_ROLES=public
    ports:
      - ${ZITI_EDGE_ROUTER_PORT:-3020}:${ZITI_EDGE_ROUTER_PORT:-3022}
      - 10082:10080
    networks:
      internet:
        aliases:
          - ziti-edge-router-cloud
    volumes:
      - type: bind
        source: ./vol
        target: /persistent
    entrypoint: ["bash", "-c", sleep 5 && /var/openziti/ziti-bin/ziti-router run /persistent/ziti-edge-router-cloud.yaml]
    #stdin_open: true
    #tty: true

  # edge router gcs
  ziti-edge-router-gcs:
    image: "${ZITI_IMAGE}:${ZITI_VERSION}"
    hostname: ziti-edge-router-gcs
    depends_on:
      - ziti-controller
    environment:
      - ZITI_CONTROLLER_RAWNAME="${ZITI_CONTROLLER_RAWNAME}"
      - ZITI_EDGE_CONTROLLER_RAWNAME="${ZITI_EDGE_CONTROLLER_RAWNAME}"
      - ZITI_EDGE_ROUTER_RAWNAME=${ZITI_EDGE_ROUTER_RAWNAME:-ziti-edge-router-gcs}
      - ZITI_EDGE_ROUTER_ROLES=public
    ports:
      - ${ZITI_EDGE_ROUTER_PORT:-3022}:${ZITI_EDGE_ROUTER_PORT:-3022}
      - 10080:10080
    networks:
      internet:
        aliases:
          - ziti-edge-router-gcs
      gcs:
        aliases:
          - ziti-edge-router-gcs
    volumes:
      - type: bind
        source: ./vol
        target: /persistent
    entrypoint: ["bash", "-c", sleep 5 && /var/openziti/ziti-bin/ziti-router run /persistent/ziti-edge-router-gcs.yaml]
    #stdin_open: true
    #tty: true

  # edge router base
  ziti-edge-router-base:
    image: "${ZITI_IMAGE}:${ZITI_VERSION}"
    hostname: ziti-edge-router-base
    depends_on:
      - ziti-controller
    environment:
      - ZITI_CONTROLLER_RAWNAME="${ZITI_CONTROLLER_RAWNAME}"
      - ZITI_EDGE_CONTROLLER_RAWNAME="${ZITI_EDGE_CONTROLLER_RAWNAME}"
      - ZITI_EDGE_ROUTER_RAWNAME=${ZITI_EDGE_ROUTER_RAWNAME:-ziti-edge-router-base}
      - ZITI_EDGE_ROUTER_ROLES=public
    ports:
      - ${ZITI_EDGE_ROUTER_PORT:-3021}:${ZITI_EDGE_ROUTER_PORT:-3022}
      - 10081:10080
    networks:
      internet:
        aliases:
          - ziti-edge-router-base
      base:
        aliases:
          - ziti-edge-router-base

    volumes:
      - type: bind
        source: ./vol
        target: /persistent
    entrypoint: ["bash", "-c", sleep 5 && /var/openziti/ziti-bin/ziti-router run /persistent/ziti-edge-router-base.yaml]
    #stdin_open: true
    #tty: true

    # console
  ziti-console:
    image: openziti/zac
    environment:
      - ZAC_SERVER_CERT_CHAIN=/persistent/pki/${ZITI_EDGE_CONTROLLER_HOSTNAME:-ziti-controller}-intermediate/certs/${ZITI_EDGE_CONTROLLER_HOSTNAME:-ziti-controller}-server.cert
      - ZAC_SERVER_KEY=/persistent/pki/${ZITI_EDGE_CONTROLLER_HOSTNAME:-ziti-controller}-intermediate/keys/${ZITI_EDGE_CONTROLLER_HOSTNAME:-ziti-controller}-server.key
      - PORTTLS=8443
    ports:
      - 1408:1408
      - 8443:8443
    working_dir: /usr/src/app
    volumes:
      - type: bind
        source: ./vol
        target: /persistent
    networks:
      - internet

    ##### machines
    # video server
  video-server:
    image: mehdi/rtmp-hls_server
    build:
      dockerfile: ./rtmp-hls-server/Dockerfile
      context: .
      network: host
    depends_on:
      - ziti-controller
      - ziti-edge-router-gcs
      - ziti-edge-router-cloud
      - ziti-edge-router-base
    cap_add:
      - NET_ADMIN
    ports:
```

```
    - 8080:8080
    - 1935:1935
  volumes:
    - type: bind
      source: ./vol
      target: /persistent
  healthcheck:
    test: curl --fail http://localhost:8080 || exit 1
  entrypoint:
    - "/persistent/tunnel-server.sh"
  networks:
    internet:
      aliases:
        - video-server


# drone
drone:
  image: debian-bullslim
  build:
    dockerfile: ./Dockerfile_drone
    context: .
    network: host
  depends_on:
    - video-server
  cap_add:
    - NET_ADMIN
  volumes:
    - type: bind
      source: ./vol
      target: /persistent
  entrypoint:
    - "/persistent/tunnel-drone.sh"
  privileged: true
  stdin_open: true
  tty: true
  networks:
    - internet


# base
base:
  image: debian-bullslim
  build:
    dockerfile: ./Dockerfile_base
    context: .
    network: host
  depends_on:
    - ziti-controller
```

```
      - ziti-edge-router-gcs
      - ziti-edge-router-cloud
      - ziti-edge-router-base
    cap_add:
      - NET_ADMIN
    volumes:
      - type: bind
        source: ./vol
        target: /persistent
    entrypoint:
      - "/persistent/tunnel-base.sh"
    privileged: true
    stdin_open: true
    tty: true
    networks:
      - base
      - internet


  # GCS
  gcs:
    image: debian-bullslim
    build:
      dockerfile: ./Dockerfile_gcs
      context: .
      network: host
    depends_on:
      - ziti-controller
      - ziti-edge-router-gcs
      - ziti-edge-router-cloud
      - ziti-edge-router-base
    cap_add:
      - NET_ADMIN
    volumes:
      - type: bind
        source: ./vol
        target: /persistent
    entrypoint:
      - "/persistent/tunnel-gcs.sh"
    privileged: true
    stdin_open: true
    tty: true
    networks:
      - gcs
      - internet

# networks
networks:
  internet:
    driver: bridge
  gcs:
    driver: bridge
  base:
    driver: bridge
```

Dans le même répertoire que le fichier docker-compose.yaml, mettre le fichier .env suivant : (Fichier disponible sur Github dans le répertoire ‘n2/‘)

Code source B.3.2: .env

```
# OpenZiti Variables
ZITI_IMAGE=openziti/quickstart
ZITI_VERSION=latest

# The duration of the enrollment period (in minutes), default if not set
# shown - 7days
ZITI_EDGE_IDENTITY_ENROLLMENT_DURATION=10080
ZITI_EDGE_ROUTER_ENROLLMENT_DURATION=10080

# controller address/port information
ZITI_CONTROLLER_RAWNAME=ziti-controller
#ZITI_CONTROLLER_HOSTNAME=advertised.address
#ZITI_CTRL_PORT=8440

ZITI_EDGE_CONTROLLER_RAWNAME=ziti-edge-controller
#ZITI_EDGE_CONTROLLER_HOSTNAME=advertised.address
#ZITI_EDGE_CONTROLLER_PORT=8441
#ZITI_EDGE_CONTROLLER_IP_OVERRIDE=172.17.0.1

# router address/port information
#ZITI_EDGE_ROUTER_RAWNAME=advertised.address
#ZITI_EDGE_ROUTER_PORT=8442
#ZITI_EDGE_ROUTER_IP_OVERRIDE=172.17.0.1
```

Commande pour créer les différents services, configurations, accès, etc. :

```
# se connecter au contrôleur et ajouter les routeurs
docker exec -it n2-ziti-controller-1 bash
ziti edge login
```

```
####### Routeurs ########

# créer routeurs cloud, gcs, base
ziti create config router edge --routerName  ziti-edge-router-cloud \
                                --output ziti-edge-router-cloud.yaml
ziti edge create edge-router ziti-edge-router-cloud --jwt-output-file
 ↪  ziti-edge-router-cloud.jwt --tunneler-enabled -a public

ziti create config router edge --routerName  ziti-edge-router-base \
                                --output ziti-edge-router-base.yaml
ziti edge create edge-router ziti-edge-router-base --jwt-output-file
 ↪  ziti-edge-router-base.jwt --tunneler-enabled -a public

ziti create config router edge --routerName  ziti-edge-router-gcs \
                                --output ziti-edge-router-gcs.yaml
ziti edge create edge-router ziti-edge-router-gcs --jwt-output-file
 ↪  ziti-edge-router-gcs.jwt --tunneler-enabled -a public

# inscrire chaque routeur depuis une console respective
ziti-router enroll ziti-edge-router-cloud.yaml --jwt
 ↪  ziti-edge-router-cloud.jwt
ziti-router enroll ziti-edge-router-base.yaml --jwt
 ↪  ziti-edge-router-base.jwt
ziti-router enroll ziti-edge-router-gcs.yaml --jwt
 ↪  ziti-edge-router-gcs.jwt
```

```
####### Drone ########

# identité drone
ziti edge create identity user drone.ziti -a "drone-video" -o
↪  drone.ziti.jwt

# créer configurations pour service echo
ziti edge create config echo.host.v1 host.v1 '{"protocol":"tcp",
↪  "address":"'"localhost"'", "port":9999}'

ziti edge create config echo.intercept.v1 intercept.v1
↪  '{"protocols":["tcp"],"addresses":["drone.ziti"],
↪  "portRanges":[{"low":9999, "high":9999}]}'

# service echo
ziti edge create service echo.svc --configs
↪  echo.intercept.v1,echo.host.v1

# créer politiques dial/bind pour service echo
ziti edge create service-policy echo.policy.dial Dial --service-roles
↪  "@echo.svc" --identity-roles '#echo'

ziti edge create service-policy echo.policy.bind Bind --service-roles
↪  '@echo.svc' --identity-roles "@drone.ziti"

# depuis le drone, s'inscrire au contrôleur
ziti-edge-tunnel enroll --jwt drone.ziti.jwt --identity drone.ziti.json
```

```
####### Base ########

# créer identitié base avec accès service echo
ziti edge create identity user base.ziti -a "echo" -o base.ziti.jwt

# depuis la base, s'inscrire auprès du contrôleur
ziti-edge-tunnel enroll --jwt base.ziti.jwt --identity base.ziti.json
```

```
####### GCS ########

# identité gcs
ziti edge create identity user gcs.ziti -a 'echo, drone-ssh' -o
↪  gcs.ziti.jwt

# service, configurations et politique ssh
ziti edge create config drone.ssh.host.v1 host.v1 '{"protocol":"tcp",
↪  "address":"'"localhost"'", "port":22}'
ziti edge create config drone-ssh.intercept.v1 intercept.v1
↪  '{"protocols":["tcp"],"addresses":["drone.ziti"],
↪  "portRanges":[{"low":22, "high":22}]}'
ziti edge create service drone-ssh.svc --configs
↪  drone-ssh.intercept.v1,drone-ssh.host.v1
ziti edge create service-policy drone-ssh.policy.dial Dial
↪  --service-roles "@drone-ssh.svc" --identity-roles '#drone-ssh'
ziti edge create service-policy drone-ssh.policy.bind Bind
↪  --service-roles '@drone-ssh.svc' --identity-roles "@drone.ziti"

# inscription depuis le drone
ziti-edge-tunnel enroll --jwt ./gcs.ziti.jwt --identity ./gcs.json

####### FederalPolice ########
ziti edge create identity user federalpolice.ziti -a 'echo' -o
↪  federalpolice.ziti.jwt
```