

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
- initialiseur 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

```
1 # Auteur      : Mehdi Salhi
2 # Auteur      : OpenZiti documentation
3 # Sujet       : Travail de Bachelor Zero Trust OpenZiti
4 # But         : Déploie une infrastructure OpenZiti
5 # No          : n2
6 # Description : 1 contrôleur, 3 routeur, 1 initialisateur, 1 serveur vidéo, 1
7 #              drone, 1 gcs, 1 base. Tous les fichiers sont sauvegardés dans
8 #              le répertoire local "vol", et monté dans le répertoire "persistent"
9 #              sur les containers
10 version: '2.4'
11 services:
12   # openziti controller
13   ziti-controller:
14     image: "${ZITI_IMAGE}:${ZITI_VERSION}"
15     env_file:
16       - ./env
17     ports:
18       - ${ZITI_EDGE_CONTROLLER_PORT:-1280}:${ZITI_EDGE_CONTROLLER_PORT:-1280}
19       - ${ZITI_CTRL_PORT:-6262}:${ZITI_CTRL_PORT:-6262}
20     environment:
21       - ↪ ZITI_EDGE_IDENTITY_ENROLLMENT_DURATION=${ZITI_EDGE_IDENTITY_ENROLLMENT_DURATION}
22       - ZITI_EDGE_ROUTER_ENROLLMENT_DURATION=${ZITI_EDGE_ROUTER_ENROLLMENT_DURATION}
23     volumes:
24       - type: bind
25         source: ./vol
26         target: /persistent
27     entrypoint:
28       - "/var/openziti/scripts/run-controller.sh"
29     networks:
30       internet:
31         aliases:
32           - ziti-edge-controller
33
34   # controller init
35   ziti-controller-init-container:
36     image: "${ZITI_IMAGE}:${ZITI_VERSION}"
37     depends_on:
```

```

39     - ziti-controller
40   environment:
41     - ZITI_CONTROLLER_RAWNAME="${ZITI_CONTROLLER_RAWNAME}"
42     - ZITI_EDGE_CONTROLLER_RAWNAME="${ZITI_EDGE_CONTROLLER_RAWNAME}"
43   env_file:
44     - ./env
45   networks:
46     - internet
47   volumes:
48     - type: bind
49       source: ./vol
50       target: /persistent
51   entrypoint:
52     - "/var/openziti/scripts/run-with-ziti-cli.sh"
53   command:
54     - "/var/openziti/scripts/access-control.sh"
55
56 #####
57 # Routers
58 #####
59
60 # edge router cloud
61 ziti-edge-router-cloud:
62   image: "${ZITI_IMAGE}:${ZITI_VERSION}"
63   hostname: ziti-edge-router-cloud
64   depends_on:
65     - ziti-controller
66   environment:
67     - ZITI_CONTROLLER_RAWNAME="${ZITI_CONTROLLER_RAWNAME}"
68     - ZITI_EDGE_CONTROLLER_RAWNAME="${ZITI_EDGE_CONTROLLER_RAWNAME}"
69     - ZITI_EDGE_ROUTER_RAWNAME=${ZITI_EDGE_ROUTER_RAWNAME:-ziti-edge-router-cloud}
70     - ZITI_EDGE_ROUTER_ROLES=public
71   ports:
72     - ${ZITI_EDGE_ROUTER_PORT:-3020}:${ZITI_EDGE_ROUTER_PORT:-3022}
73     - 10082:10080
74   networks:
75     internet:
76       aliases:
77         - ziti-edge-router-cloud
78   volumes:
79     - type: bind
80       source: ./vol
81       target: /persistent
82   entrypoint: ["bash", "-c", sleep 5 && /var/openziti/ziti-bin/ziti-router run
↪ /persistent/ziti-edge-router-cloud.yaml]
83   #stdin_open: true
84   #tty: true
85
86 # edge router gcs

```

```

87  ziti-edge-router-gcs:
88    image: "${ZITI_IMAGE}:${ZITI_VERSION}"
89    hostname: ziti-edge-router-gcs
90    depends_on:
91      - ziti-controller
92    environment:
93      - ZITI_CONTROLLER_RAWNAME="${ZITI_CONTROLLER_RAWNAME}"
94      - ZITI_EDGE_CONTROLLER_RAWNAME="${ZITI_EDGE_CONTROLLER_RAWNAME}"
95      - ZITI_EDGE_ROUTER_RAWNAME=${ZITI_EDGE_ROUTER_RAWNAME:-ziti-edge-router-gcs}
96      - ZITI_EDGE_ROUTER_ROLES=public
97    ports:
98      - ${ZITI_EDGE_ROUTER_PORT:-3022}:${ZITI_EDGE_ROUTER_PORT:-3022}
99      - 10080:10080
100   networks:
101     internet:
102       aliases:
103         - ziti-edge-router-gcs
104     gcs:
105       aliases:
106         - ziti-edge-router-gcs
107   volumes:
108     - type: bind
109       source: ./vol
110       target: /persistent
111   entrypoint: ["bash", "-c", sleep 5 && /var/openziti/ziti-bin/ziti-router run
↪ /persistent/ziti-edge-router-gcs.yaml]
112   #stdin_open: true
113   #tty: true
114
115   # edge router base
116   ziti-edge-router-base:
117     image: "${ZITI_IMAGE}:${ZITI_VERSION}"
118     hostname: ziti-edge-router-base
119     depends_on:
120       - ziti-controller
121     environment:
122       - ZITI_CONTROLLER_RAWNAME="${ZITI_CONTROLLER_RAWNAME}"
123       - ZITI_EDGE_CONTROLLER_RAWNAME="${ZITI_EDGE_CONTROLLER_RAWNAME}"
124       - ZITI_EDGE_ROUTER_RAWNAME=${ZITI_EDGE_ROUTER_RAWNAME:-ziti-edge-router-base}
125       - ZITI_EDGE_ROUTER_ROLES=public
126     ports:
127       - ${ZITI_EDGE_ROUTER_PORT:-3021}:${ZITI_EDGE_ROUTER_PORT:-3022}
128       - 10081:10080
129     networks:
130       internet:
131         aliases:
132           - ziti-edge-router-base
133       base:
134         aliases:

```

```

135         - ziti-edge-router-base
136
137     volumes:
138         - type: bind
139           source: ./vol
140           target: /persistent
141     entrypoint: ["bash", "-c", sleep 5 && /var/openziti/ziti-bin/ziti-router run
↪ /persistent/ziti-edge-router-base.yaml]
142     #stdin_open: true
143     #tty: true
144
145     # console
146     ziti-console:
147         image: openziti/zac
148         environment:
149             - ZAC_SERVER_CERT_CHAIN=/persistent/pki/${ZITI_EDGE_CONTROLLER_HOSTNAME:-ziti-
↪ controller}-intermediate/certs/${ZITI_EDGE_CONTROLLER_HOSTNAME:-ziti-controller}
↪ -server.cert
150             -
↪ ZAC_SERVER_KEY=/persistent/pki/${ZITI_EDGE_CONTROLLER_HOSTNAME:-ziti-controller}
↪ -intermediate/keys/${ZITI_EDGE_CONTROLLER_HOSTNAME:-ziti-controller}-server.key
151             - PORTTLS=8443
152     ports:
153         - 1408:1408
154         - 8443:8443
155     working_dir: /usr/src/app
156     volumes:
157         - type: bind
158           source: ./vol
159           target: /persistent
160     networks:
161         - internet
162
163     ##### machines
164     # video server
165     video-server:
166         image: mehdi/rtmp-hls_server
167         build:
168             dockerfile: ./rtmp-hls-server/Dockerfile
169             context: .
170             network: host
171         depends_on:
172             - ziti-controller
173             - ziti-edge-router-gcs
174             - ziti-edge-router-cloud
175             - ziti-edge-router-base
176         cap_add:
177             - NET_ADMIN
178     ports:

```

```
179     - 8080:8080
180     - 1935:1935
181     volumes:
182     - type: bind
183       source: ./vol
184       target: /persistent
185     healthcheck:
186       test: curl --fail http://localhost:8080 || exit 1
187     entrypoint:
188       - "/persistent/tunnel-server.sh"
189     networks:
190       internet:
191         aliases:
192         - video-server
193
194
195     # drone
196     drone:
197       image: debian-bullslim
198       build:
199         dockerfile: ./Dockerfile_drone
200         context: .
201         network: host
202       depends_on:
203       - video-server
204       cap_add:
205       - NET_ADMIN
206       volumes:
207       - type: bind
208         source: ./vol
209         target: /persistent
210       entrypoint:
211       - "/persistent/tunnel-drone.sh"
212       privileged: true
213       stdin_open: true
214       tty: true
215       networks:
216       - internet
217
218
219     # base
220     base:
221       image: debian-bullslim
222       build:
223         dockerfile: ./Dockerfile_base
224         context: .
225         network: host
226       depends_on:
227       - ziti-controller
```

```

228     - ziti-edge-router-gcs
229     - ziti-edge-router-cloud
230     - ziti-edge-router-base
231   cap_add:
232     - NET_ADMIN
233   volumes:
234     - type: bind
235       source: ./vol
236       target: /persistent
237   entrypoint:
238     - "/persistent/tunnel-base.sh"
239   privileged: true
240   stdin_open: true
241   tty: true
242   networks:
243     - base
244     - internet
245
246
247   # GCS
248   gcs:
249     image: debian-bullslim
250     build:
251       dockerfile: ./Dockerfile_gcs
252       context: .
253       network: host
254     depends_on:
255       - ziti-controller
256       - ziti-edge-router-gcs
257       - ziti-edge-router-cloud
258       - ziti-edge-router-base
259     cap_add:
260       - NET_ADMIN
261     volumes:
262       - type: bind
263         source: ./vol
264         target: /persistent
265     entrypoint:
266       - "/persistent/tunnel-gcs.sh"
267     privileged: true
268     stdin_open: true
269     tty: true
270     networks:
271       - gcs
272       - internet
273
274   # networks
275   networks:
276     internet:

```

```
277     driver: bridge
278   gcs:
279     driver: bridge
280   base:
281     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

```
1  # OpenZiti Variables
2  ZITI_IMAGE=openziti/quickstart
3  ZITI_VERSION=latest
4
5  # The duration of the enrollment period (in minutes), default if not set
6  # shown - 7days
7  ZITI_EDGE_IDENTITY_ENROLLMENT_DURATION=10080
8  ZITI_EDGE_ROUTER_ENROLLMENT_DURATION=10080
9
10 # controller address/port information
11 ZITI_CONTROLLER_RAWNAME=ziti-controller
12 #ZITI_CONTROLLER_HOSTNAME=advertised.address
13 #ZITI_CTRL_PORT=8440
14
15 ZITI_EDGE_CONTROLLER_RAWNAME=ziti-edge-controller
16 #ZITI_EDGE_CONTROLLER_HOSTNAME=advertised.address
17 #ZITI_EDGE_CONTROLLER_PORT=8441
18 #ZITI_EDGE_CONTROLLER_IP_OVERRIDE=172.17.0.1
19
20 # router address/port information
21 #ZITI_EDGE_ROUTER_RAWNAME=advertised.address
22 #ZITI_EDGE_ROUTER_PORT=8442
23 #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 identité 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

```

