

*2016 /
2017*

Guide d'installation et d'administrateur de Qwirk.eu

Auteurs :
Romain GARCIA
Dorian GAILLETON
Mathieu CALEMARD-DU
GARDIN
Kassem DAOUSSI
Nicolas MAITREJEAN

[Nom de la société]

2016 / 2017

Table des matières :

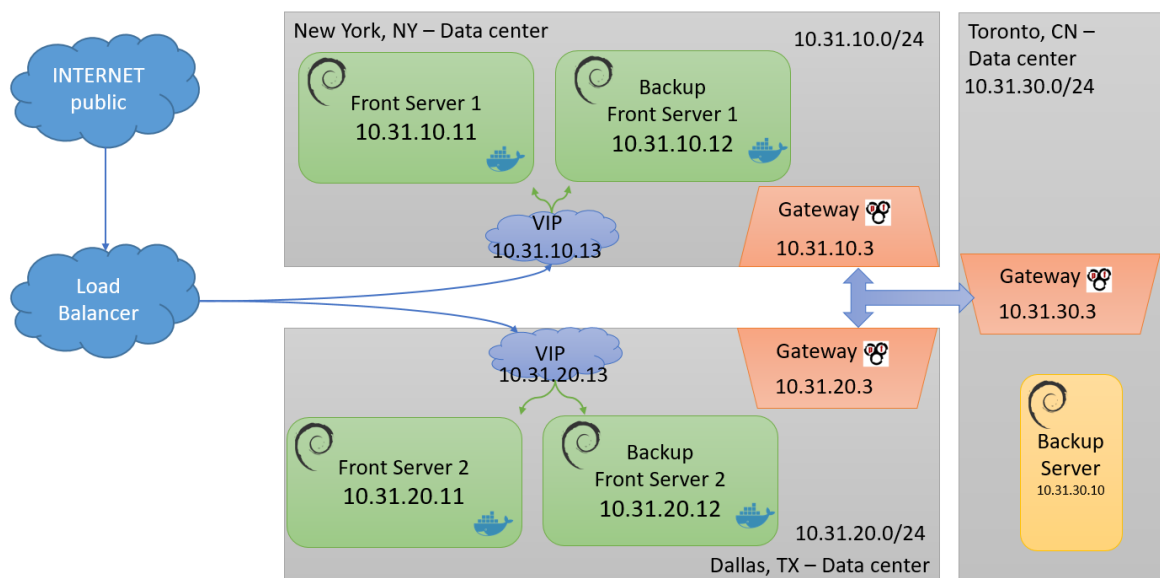
I.	Résumé	2
II.	Front servers	2
1)	Installation	2
2)	Cluster actif/passif data Center NY	3
3)	Cluster actif/passif data Center TX	6
4)	Mise en place du site web (4 serveurs)	7
5)	Installation Sockage MongoDB	9
6)	MongoDB Replica Set	10
III.	Gateway	12
1)	Network Gateway NY	12
2)	IP SEC Gateway NY	14
3)	Redirection de port	16
•	Gateway CN et TX	16
IV.	Load Balancer	16
V.	Backup Server	17



I. Résumé

Dans un premier temps nous allons récapituler le schéma de l'infrastructure machine par machine.

Physical network topology



En résumé nous avons 9 machines :

- Load balancer
- Front server 1 NY
- Front server 2 TX
- Front server backup 1 NY
- Front server backup 2 CN
- Gateway NY
- Gateway TX
- Gateway CN
- Backup Sever CN

Pour commencer nous allons voir l'installation des 4 Front servers.

II. Front servers

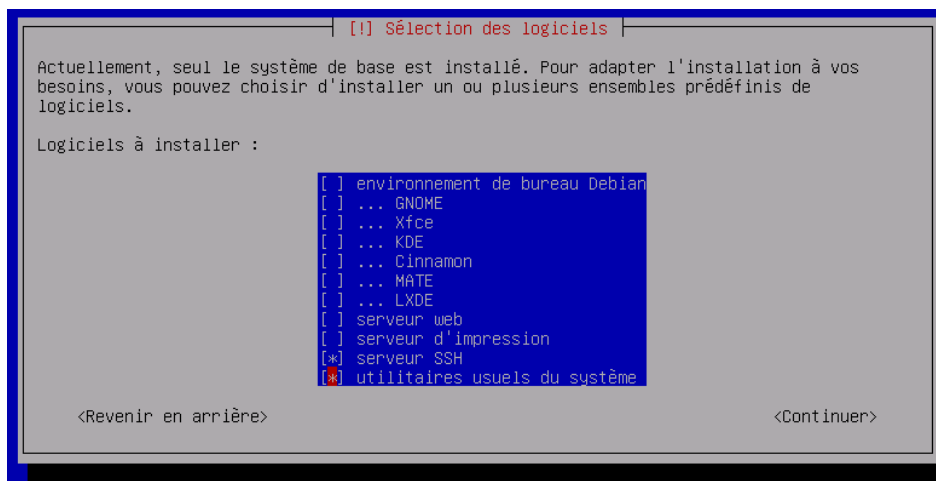
1) Installation

Pour la distribution nous avons choisi d'utiliser Debian version 8.6 car cette distribution nous permettait d'utiliser tous les services dont nous avons besoin pour réaliser le projet mais également car c'est une distribution dont nous avons l'habitude d'utiliser et de travailler avec.



Lancer l'installation sans interface graphique.

L'installation est une installation standard, nous n'avons pas besoin de paramètres particuliers :



Configurer les interfaces réseaux comme dans le schéma résumé de l'infrastructure.

Puis nous allons configurer le cluster actif/passif pour les deux data centers.
Pour cela nous allons utiliser « Corosync » et « Pacemaker ».

Sur les 4 front servers nous allons exécuter ces commandes :

Pour faire une jessie-backports afin d'avoir accès aux packages nécessaire :

```
cat > /etc/apt/sources.list.d/jessie-backports.list << "EOF"
deb http://http.debian.net/debian jessie-backports main
EOF
```

Puis on lance l'installation :

```
# apt-get update
# apt-get install -t jessie-backports pacemaker crmsh
```

Installer nginx pour notre server web :

```
# apt-get install nginx
```

Nous allons commencer la configuration par data center.

2) Cluster actif/passif data Center NY

Recapitulons les configurations du réseau :

Network : 10.31.10.0 /24

Front Server : 10.31.10.11



Front Server 1 Backup : 10.31.10.12

Dans le fichier /etc/corosync/corosync.conf de Font Server 1

*# This is a *partial* config file to show a unicast setup.*

```
totem {  
    version: 2  
  
    cluster_name: debian  
    transport: udpu  
  
    token: 3000  
    token_retransmits_before_loss_const: 10  
  
    clear_node_high_bit: yes  
  
    crypto_cipher: aes256 # default was 'none'  
    crypto_hash: sha1     # default was 'none'  
  
    interface {  
        ringnumber: 0  
  
        # set address of the network here; default was '127.0.0.1'  
        bindnetaddr: 10.31.10.0  
  
        mcastaddr: 239.255.1.1  
        mcastport: 5405  
  
        ttl: 1  
    }  
}  
  
logging {  
    fileline: off  
  
    to_stderr: no  
    to_logfile: no  
    to_syslog: yes  
  
    syslog_facility: daemon  
    debug: off  
  
    timestamp: on  
    logger_subsys {  
        subsys: QUORUM  
        debug: off  
    }  
}  
  
quorum {  
    provider: corosync_votequorum  
    two_node: 1 # value added
```



```
    expected_votes: 2
  }
  nodelist {
    node {
      ring0_addr: 10.31.10.11
    }
    node {
      ring0_addr: 10.31.10.12
    }
  }
}
```

On va ensuite générer des clés d'authentications toujours sur Front Server 1 :

```
root@frontserver1:~# corosync-keygen
```

Puis nous allons copier toutes les configurations dans le front server 1 backup :

```
root@frontserver1:~# scp /etc/corosync/corosync.conf
root@frontserver1backup:~#/etc/corosync/corosync.conf
root@frontserver1:~# scp /etc/corosync/authkey
root@frontserver1backup:~#/etc/corosync/authkey
```

Activer maintenant les services sur les deux serveurs :

```
service corosync start
```

```
service pacemaker start
```

Regarder l'état de votre cluster :

```
crm status
```

Il devrait ressembler à l'affichage ci-dessous :

```
Last updated: Tue Mar 21 17:30:41 2017
Last change: Tue Mar 21 17:30:32 2017 by root via crm_attribute on
frontserver1backup
Stack: corosync
Current DC: node02 (version 1.1.14-70404b0) - partition with quorum
2 nodes and 0 resources configured
```

```
Online: [ frontserver1 frontserver1backup]
```

Une fois le cluster ok, il faut paramétrer la ressource nginx sur le front server1.
Exécuter la commande :

```
root@frontserver1:~# crm configure
```



Puis configurer la ressource comme ceci :

(L'IP 10.31.10.13 est notre IP virtuelle, contenant la ressource partagée entre les deux nodes du cluster)

```
crm(live)configure#
    property stonith-enabled=no
    property no-quorum-policy=ignore
    property default-resource-stickiness=100
    primitive IP-nginx ocf:heartbeat:IPaddr2 \
        params ip="10.31.10.13" nic="eth0" cidr_netmask="24" \
        meta migration-threshold=2 \
        op monitor interval=20 timeout=60 on-fail=restart
    primitive Nginx-rsc ocf:heartbeat:nginx \
        meta migration-threshold=2 \
        op monitor interval=20 timeout=60 on-fail=restart
    colocation lb-loc inf: IP-nginx Nginx-rsc
    order lb-ord inf: IP-nginx Nginx-rsc
    commit
```

Le « crm status » devrait maintenant ressembler à ceci :

```
crm status
```

```
Last updated: Tue Apr 12 14:53:26 2016      Last change: Tue Apr 12 14:52:45 2016
by root via cibadmin on node02
Stack: corosync
Current DC: node02 (version 1.1.14-70404b0) - partition with quorum
2 nodes and 2 resources configured
```

```
Online: [ frontserver1 frontserver1backup]
```

```
Full list of resources:
```

```
IP-nginx    (ocf::heartbeat:IPaddr2): Started frontserver1
Nginx-rsc   (ocf::heartbeat:nginx): Started frontserver1
```

La configuration est également présente sur le front server 1 backup comme il s'agit de la configuration de la ressource et qu'elle est partagée entre les deux serveurs.

Exécuter ces commandes sur les deux serveurs pour activer le cluster automatiquement lors du démarrage des machines :

```
# systemctl enable nginx
# systemctl enable pacemaker
# systemctl enable corosync
```

3) Cluster actif/passif data Center TX

Reproduire toute la configuration du data center NY en remplacement les adresses IP (En rouges) par les IP correspondantes :

Network : 10.31.20.0/24



Front Server 2 : 10.31.20.11

Front Server 2 Backup : 10.31.20.12

Nos clusters sont maintenant fonctionnels.

4) Mise en place du site web (4 serveurs)

Installer meteor :

Apt-get Install meteor

Télécharger le répertoire du site dans un dossier, puis dans ce dossier installer les dépendances :

Meteor Npm install

Puis créer le fichier /etc/nginx/sites-available/qwirk comme ci-dessous:

```
server {  
    listen 80;  
    server_name qwirk.eu;  
    rewrite ^ https://qwirk.eu$request_uri permanent;  
}  
  
server {  
    listen 443 ssl;  
    server_name qwirk.eu;  
  
    access_log /var/log/nginx/qwirk.access;  
    error_log /var/log/nginx/qwirk.error;  
  
    ssl_certificate /etc/letsencrypt/live/qwirk.eu/fullchain.pem;  
    ssl_certificate_key /etc/letsencrypt/live/qwirk.eu/privkey.pem;  
  
    include snippets/ssl-params.conf;  
  
    # Tell Nginx and Passenger where your app's 'public' directory is
```




```
root /var/www/qwirk/bundle/public;

# Turn on Passenger

passenger_enabled on;

# Tell Passenger that your app is a Meteor app

passenger_app_type node;

passenger_startup_file main.js;

# Tell your app where MongoDB is

passenger_env_var MONGO_URL mongodb://localhost:25017/qwirk;

# Tell your app what its root URL is

passenger_env_var ROOT_URL https://qwirk.eu;

passenger_env_var MAIL_URL
smtp://noreply%40qwirk.eu:nPm7n4ms7DI@thaekcorp.com:587;
}
```

Noud activons la conf sur /etc/nginx/sites-enabled/qwirk avec la commande `ln -s`

Qwirk nécessite également un serveur peerjs qui fait relay pour les appels audio/video.

```
server {

    listen 443 ssl;

    server_name peer.qwirk.eu;

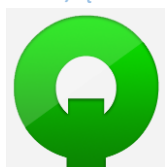
    access_log /var/log/nginx/peer.access;

    error_log /var/log/nginx/peer.error;

    ssl_certificate /etc/letsencrypt/live/peer.qwirk.eu/fullchain.pem;
    ssl_certificate_key /etc/letsencrypt/live/peer.qwirk.eu/privkey.pem;

    include snippets/ssl-params.conf;

    location / {
```



```
proxy_pass http://localhost:9000;

proxy_http_version 1.1;

proxy_set_header Upgrade $http_upgrade;
proxy_set_header Connection 'upgrade';
proxy_set_header Host $host;
proxy_cache_bypass $http_upgrade;
}

proxy_read_timeout 86400s;
proxy_send_timeout 86400s;
}
```

Information complémentaire ici :

<https://www.phusionpassenger.com/library/walkthroughs/deploy/meteor/>

5) Installation Sockage MongoDB

Installer Docker sur les 4 serveurs :

```
$ sudo apt-get install \
  apt-transport-https \
  ca-certificates \
  curl \
  gnupg2 \
  software-properties-common
```

```
$ curl -fsSL https://download.docker.com/linux/debian/gpg | sudo apt-key add -
```

```
$ sudo apt-key fingerprint 0EBFCD88
```

```
$ sudo add-apt-repository \
  "deb [arch=amd64] https://download.docker.com/linux/debian \
  $(lsb_release -cs) \
  stable"
```

```
$ sudo apt-get update
```

```
$ sudo apt-get install docker-ce
```

```
$ apt-cache madison docker-ce
```

```
docker-ce | 17.03.0~ce-0~debian-jessie | https://download.docker.com/linux/debian
jessie/stable amd64 Packages
```

```
$ sudo apt-get install docker-ce=17.03.2-rc1
```



6) MongoDB Replica Set

Sur Front Server 1, configurer le cluster mongo comme ceci :

```
root@node*:/# mkdir -p /home/core
root@node*:/# cd /home/core
root@node*:/# openssl rand -base64 741 > mongodb-keyfile
root@node*:/# chmod 600 mongodb-keyfile
root@node*:/# sudo chown 999 mongodb-keyfile
```

```
root@node1:/# docker run --name mongo \
-v /home/core/mongo-files/data:/data/db \
-v /home/core/mongo-files:/opt/keyfile \
--hostname="node1.example.com" \
-p 27017:27017 \
-d mongo:2.6.5 --smallfiles
```

```
root@node1:/# docker exec -it mongo /bin/bash
```

Lancer mongo :

```
root@node1:/# mongo
```

```
> use admin
```

```
> db.createUser( {
  user: "siteUserAdmin",
  pwd: "password",
  roles: [ { role: "userAdminAnyDatabase", db: "admin" } ]
});
```

```
> db.createUser( {
  user: "siteRootAdmin",
  pwd: "password",
  roles: [ { role: "root", db: "admin" } ]
});
```

```
> exit
```

```
root@node1:/# docker stop mongo
```

```
root@node1:/# docker rm mongo
root@node1:/# docker run \
--name mongo \
-v /home/core/mongo-files/data:/data/db \
-v /home/core/mongo-files:/opt/keyfile \
--hostname=" node1.ny.qwirk.eu " \
```



```
--add-host node1.ny.qwirk.eu:10.31.10.11} \  
--add-host node2.ny.qwirk.eu:10.31.10.12} \  
--add-host node1.tx.qwirk.eu:10.31.20.11} \  
--add-host node2.tx.qwirk.eu:10.31.20.12} \  
-p 27017:27017 -d mongo:2.6.5 \  
--smallfiles \  
--keyFile /opt/keyfile/mongodb-keyfile \  
--replSet "rs0"
```

```
root@node1:/# docker exec -it mongo /bin/bash  
root@node1:/# mongo
```

```
> use admin
```

```
> db.auth("siteRootAdmin", "password");
```

```
> rs.initiate()
```

```
rs0:PRIMARY> rs.conf()
```

Puis sur Front Server 1 Backup :

```
root@node1:/# docker run \  
--name mongo \  
-v /home/core/mongo-files/data:/data/db \  
-v /home/core/mongo-files:/opt/keyfile \  
--hostname=" host node2.ny.qwirk.eu " \  
--add-host node1.ny.qwirk.eu:10.31.10.11} \  
--add-host node2.ny.qwirk.eu:10.31.10.12} \  
--add-host node1.tx.qwirk.eu:10.31.20.11} \  
--add-host node2.tx.qwirk.eu:10.31.20.12} \  
-p 27017:27017 -d mongo:2.6.5 \  
--smallfiles \  
--keyFile /opt/keyfile/mongodb-keyfile \  
--replSet "rs0"
```

Sur Front Server 2 :

```
root@node1:/# docker run \  
--name mongo \  
-v /home/core/mongo-files/data:/data/db \  
-v /home/core/mongo-files:/opt/keyfile \  
--hostname=" node1.tx.qwirk.eu:10.31.20.11" \  
--add-host node1.ny.qwirk.eu:10.31.10.11} \  
--add-host node2.ny.qwirk.eu:10.31.10.12} \  
--add-host node1.tx.qwirk.eu:10.31.20.11} \  
--add-host node2.tx.qwirk.eu:10.31.20.12} \  
-p 27017:27017 -d mongo:2.6.5 \  

```



```
--smallfiles \  
--keyFile /opt/keyfile/mongodb-keyfile \  
--replSet "rs0"
```

Et sur Front Server 2 Backup :

```
--name mongo \  
-v /home/core/mongo-files/data:/data/db \  
-v /home/core/mongo-files:/opt/keyfile \  
--hostname=" node2.tx.qwirk.eu:10.31.20.12" \  
--add-host node1.ny.qwirk.eu:10.31.10.11} \  
--add-host node2.ny.qwirk.eu:10.31.10.12} \  
--add-host node1.tx.qwirk.eu:10.31.20.11} \  
--add-host node2.tx.qwirk.eu:10.31.20.12} \  
-p 27017:27017 -d mongo:2.6.5 \  
--smallfiles \  
--keyFile /opt/keyfile/mongodb-keyfile \  
--replSet "rs0"
```

Repasser sur Front Server 1 :

```
rs0:PRIMARY> rs.add("node2.ny.qwirk.eu")  
rs0:PRIMARY> rs.add("node1.tx.qwirk.eu:")  
rs0:PRIMARY> rs.add("node2.tx.qwirk.eu:")
```

III. Gateway

Pour les gateways de nos data centers nous avons utilisé la distribution PfSense 2.3.4.

Chaque PfSense aura deux interfaces réseaux, une LAN, local au data center et une WAN entre les gateways et le load balancer comme ceci :

Network WAN : 192.168.200.0 /24

Gateway NY LAN : 10.31.10.3

Gateway NY WAN : 192.168.200.10

Gateway TX LAN : 10.31.20.3

Gateway TX WAN : 192.168.200.20

Gateway CN LAN : 10.31.30.3

Gateway CN WAN : 192.168.200.40

1) Network Gateway NY

Installer le PfSense avec deux interfaces réseaux et assigner-les avec l'option 2 comme ci-dessous :



```

Bootup complete

FreeBSD/amd64 (pfSense.ny.qwirk.eu) (ttyv0)

*** Welcome to pfSense 2.3.4-RELEASE (amd64 full-install) on pfSense ***

WAN (wan)      -> em0      -> v4: 192.168.200.10/24
LAN (lan)      -> em1      -> v4: 10.31.10.3/24

0) Logout (SSH only)          9) pfTop
1) Assign Interfaces          10) Filter Logs
2) Set interface(s) IP address 11) Restart webConfigurator
3) Reset webConfigurator password 12) PHP shell + pfSense tools
4) Reset to factory defaults  13) Update from console
5) Reboot system              14) Enable Secure Shell (sshd)
6) Halt system                15) Restore recent configuration
7) Ping host                  16) Restart PHP-FPM
8) Shell

Enter an option:
Message from syslogd@pfSense at Jun  8 22:46:29 ...
pfSense php-fpm[2841]: /services_unbound.php: Successful login for user 'admin' from: 10.31.10.6

```

Puis aller sur un navigateur sur une machine présente dans le même réseau LAN que le PfSense et accéder à l'interface web de configuration sur l'adresse IP LAN 10.31.10.3.

Commencer à configurer les paramètres généraux dans « System /General Setup » comme ci-dessous :

System / General Setup

System

Hostname
Name of the firewall host, without domain part

Domain
Do not use 'local' as a domain name. It will cause local hosts running mDNS (avahi, bonjour, etc.) to be unable to resolve local hosts not running mDNS.

DNS Server Settings

DNS Servers	Address	Gateway	Action
<input type="text" value="8.8.8.8"/>	<input type="text" value="none"/>	<input type="button" value="Delete"/>	
<input type="text" value="8.8.4.4"/>	<input type="text" value="none"/>	<input type="button" value="Delete"/>	
<input type="text" value="192.168.200.20"/>	<input type="text" value="none"/>	<input type="button" value="Delete"/>	

Puis commencer à configurer le DNS du PfSense pour les 4 serveurs comme l'exemple ci-dessous qui représente le front server 1:

Services / DNS Resolver / General Settings / Edit Host Override

Host Override Options

Host
Name of the host, without the domain part
e.g.: "myhost"









Domain
Domain of the host
e.g.: "example.com"

IP Address
IP address of the host
e.g.: 192.168.100.100 or fd00:abcd::1

Description
A description may be entered here for administrative reference (not parsed).



Reproduire l'exemple ci-dessus pour les 3 autres serveurs pour avoir la configuration comme ci-dessous :

Host Overrides				
Host	Domain	IP	Description	Actions
node1	ny.qwirk.eu	10.31.10.11		 
node1	tx.qwirk.eu	10.31.20.11		 
node2	tx.qwirk.eu	10.31.20.12		 
node2	ny.qwirk.eu	10.31.10.12		 

Le résumé du pfSense devrait être comme ci-dessous :

Status / [Dashboard](#)

System Information

Name	pfSense.ny.qwirk.eu
System	<p>pfSense</p> <p>Serial: 4478f767-4c8b-11e7-bb87-000c29ce205e</p> <p>Netgate Unique ID: 580a947cb85877f21a9f</p>
BIOS	<p>Vendor: Phoenix Technologies LTD</p> <p>Version: 6.00</p> <p>Release Date: 07/02/2015</p>
Version	<p>2.3.4-RELEASE (amd64)</p> <p>built on Wed May 03 15:13:29 CDT 2017</p> <p>FreeBSD 10.3-RELEASE-p19</p> <p>The system is on the latest version.</p>
Platform	pfSense
CPU Type	Intel(R) Core(TM) i7-6700HQ CPU @ 2.60GHz
Uptime	01 Hour 12 Minutes 18 Seconds
Current date/time	Thu Jun 8 23:56:33 CEST 2017
DNS server(s)	<ul style="list-style-type: none"> 127.0.0.1 8.8.8.8 8.8.4.4 192.168.200.20
Last config change	Wed Jun 7 23:12:35 CEST 2017

Interfaces

WAN	↑	1000baseT <full-duplex>	192.168.200.10
LAN	↑	1000baseT <full-duplex>	10.31.10.3

2) IP SEC Gateway NY

Pour commencer la configuration du tunnel IP SEC, configurer le CA (Certificate authorities) comme ci-dessous :

System / Certificate Manager / CAs

CAs

Certificates

Certificate Revocation

Certificate Authorities

Name	Internal	Issuer	Certificates	Distinguished Name	Actions
Qwirk_CA	✓	self-signed	3	emailAddress=203697@supinfo.com, ST=France, OU=Supinfo, O=Supinfo, L=Lyon, CN=qwirk-ca, C=FR Valid From: Tue, 06 Jun 2017 13:37:51 +0200 Valid Until: Fri, 04 Jun 2027 13:37:51 +0200	<div><div></div><div></div><div></div><div></div></div>

+ Add









Puis créer les certificats des 3 gateways :



System / Certificate Manager / Certificates

CAs Certificates Certificate Revocation

Certificates

Name	Issuer	Distinguished Name	In Use	Actions
webConfigurator default (592af5bcbdbf3) Server Certificate CA: No, Server: Yes	self-signed	emailAddress=admin@pfSense.localdomain, ST=State, O=pfSense webConfigurator Self-Signed Certificate, L=Locality, CN=pfSense-592af5bcbdbf3, C=US Valid From: Sun, 28 May 2017 18:07:24 +0200 Valid Until: Fri, 18 Nov 2022 17:07:24 +0100	webConfigurator	 
NY crt Server Certificate CA: No, Server: Yes	Qwirk_CA	emailAddress=203697@supinfo.com, ST=France, OU=Supinfo, O=Supinfo, L=Lyon, CN=192.168.200.10, C=FR Valid From: Tue, 06 Jun 2017 14:00:31 +0200 Valid Until: Fri, 04 Jun 2027 14:00:31 +0200	IPsec Tunnel	 
TX crt Server Certificate CA: No, Server: Yes	Qwirk_CA	emailAddress=203697@supinfo.com, ST=France, OU=Supinfo, O=Supinfo, L=Lyon, CN=192.168.200.20, C=FR Valid From: Tue, 06 Jun 2017 14:01:05 +0200 Valid Until: Fri, 04 Jun 2027 14:01:05 +0200		 
CN crt Server Certificate CA: No, Server: Yes	Qwirk_CA	emailAddress=203697@supinfo.com, ST=France, OU=Supinfo, O=Supinfo, L=Lyon, CN=192.168.200.40, C=FR Valid From: Tue, 06 Jun 2017 14:01:51 +0200 Valid Until: Fri, 04 Jun 2027 14:01:51 +0200		 

Configurer ensuite les routes des tunnels en utilisant les certificats précédents :

VPN / IPsec / Tunnels

Tunnels

Mobile Clients

Pre-Shared Keys

Advanced Settings

IPsec Tunnels

IKE	Remote Gateway	Mode	P1 Protocol	P1 Transforms	P1 Description	Actions
<div><div><input type="checkbox"/></div><div>Disable</div></div> V2	WAN 192.168.200.20		AES (256 bits)	SHA512	IPSEC TX	<div><div></div><div></div></div>
Mode	Local Subnet	Remote Subnet	P2 Protocol	P2 Transforms	P2 Auth Methods	P2 actions
<div><div><input type="checkbox"/></div><div>Disable</div></div> tunnel	10.31.10.0/24	10.31.20.0/24	ESP	AES (256 bits)	SHA512	<div><div></div><div></div></div>
<div><div></div><div>Add P2</div></div>						

<div><div><input type="checkbox"/></div><div>Disable</div></div> V2	WAN 192.168.200.40		AES (256 bits)	SHA512	IPSEC CN	<div><div></div><div></div></div>
Mode	Local Subnet	Remote Subnet	P2 Protocol	P2 Transforms	P2 Auth Methods	P2 actions
<div><div><input type="checkbox"/></div><div>Disable</div></div> tunnel	LAN	10.31.30.0/24	ESP	AES (256 bits)	SHA512	<div><div></div><div></div></div>
<div><div></div><div>Add P2</div></div>						

Add P1


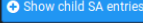

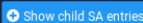
Delete P1s

Une fois la configuration terminée pour les 3 gateways, vos deux tunnels sont connectés comme ci-dessous :

Status / IPsec / Overview

Overview Leases SADs SPDs

IPsec Status

Description	Local ID	Local IP	Remote ID	Remote IP	Role	Reauth	Algo	Status	
IPSEC TX	192.168.200.10	192.168.200.10	192.168.200.20	192.168.200.20	IKEV2 initiator	24972 seconds (06:56:12)	AES_CBC HMAC_SHA2_512_256 PRF_HMAC_SHA2_512 MODP_2048	ESTABLISHED 2644 seconds (00:44:04) ago	
									
IPSEC CN	192.168.200.10	192.168.200.10	192.168.200.40	192.168.200.40	IKEV2 initiator	27503 seconds (07:38:23)	AES_CBC HMAC_SHA2_512_256 PRF_HMAC_SHA2_512 MODP_2048	ESTABLISHED 10 seconds (00:00:10) ago	
									



3) Redirection de port

Rediriger le port 80 pour que lorsque l'on contact le pfSense sur ce port, on soit redirigé sur l'adresse IP Virtuelle du cluster actif/passif du data center comme ci-dessous :

Firewall / NAT / Port Forward										
Port Forward										
Rules										
	Interface	Protocol	Source Address	Source Ports	Dest. Address	Dest. Ports	NAT IP	NAT Ports	Description	Actions
<input checked="" type="checkbox"/>	WAN	TCP	*	*	WAN address	80 (HTTP)	10.31.10.13	80 (HTTP)		

4) Gateway CN et TX

Reproduire la configuration la configuration du Gateway NY sur les deux autres Gateway en changeants les adresses IP nécessaire et les noms nécessaires.

Néanmoins seul la Gateway TX doit avoir la redirection de port sur l'adresse virtuelle du cluster de son data center.

Le CA est quant à lui à importer du PfSense NY pour l'appliquer aux 2 autres.

IV. Load Balancer

Installer la même distribution que pour les serveurs (debian) avec une interface ouverte au public et une interface en 192.168.200.30 pour communiquer avec la gateways.

Pour mettre en place le cluster actif/actif, installer nginx :

Apt-get install nginx

Puis créer le fichier /etc/nginx/sites-available/qwirk comme ci-dessous :

```
upstream qwirk.eu {
    server 192.168.200.10;
    server 192.168.200.20;
}

server {
    listen 80;

    location / {
        proxy_pass http://qwirk.eu;
    }
}
```



```
}
```

Ainsi que /etc/nginx/sites-enabled/qwirk

```
upstream qwirk.eu {
    server 192.168.200.10;
    server 192.168.200.20;
}

server {
    listen 80;

    location / {
        proxy_pass http://qwirk.eu;
    }
}
```

Ainsi le load balancer répartira la charge entre les deux data centers.

V. Backup Server

Nous créons un script de backup :

```
#!/bin/sh
DIR=`date +%m%d%y`
DEST=/db_backups/$DIR
mkdir $DEST
mongodump -h 10.31.10.11 -d qwirk -u <username> -p <password> -
o $DEST
```

Nous l'exécutons via cron sur un intervalle régulier :

```
45 1 * * * ../scripts/db_backup.sh
```

De cette manière nous créons un backup de la base mongodb chaque nuit à 1h45.



VI. Annexes

Corosync.txt : Configuration de corosync

Pfsense.ny.qwirk.eu.txt : Fichier de configuration du pfsense en xml

Pfsense.tx.qwirk.eu.txt : Fichier de configuration du pfsense en xml

Pfsense.cn.qwirk.eu.txt : Fichier de configuration du pfsense en xml

LogicalTopology.png : Schéma Logical Network Topology

PhysicalTotpology.png : Schéma Physical Network Topology

