



HAProxy

Reverse proxy w/ HA – Capitoul – 22/06/2017

HAProxy, c'est quoi ?

- C'est un « reverse proxy » sur couches 4 & 7 (et 6)
 - Performant, même sur VM
 - Stable
 - Logique « frontend » et « backend »
 - Open source (GPL & LGPL)

HAProxy, ça fait quoi ?

- « statefull » High-Availability
 - Gestion statuts serveurs
 - Checks réseaux & applicatifs
 - Protection DDoS
- Load-balancing & service-balancing
 - Répartition des flux (least-conn, RR)
 - Serveurs conditionnels (sticky cookie, SNI, règles URL, en-têtes)
 - Interface de gestion des services
- Reverse proxy
 - HTTP: R/W des en-têtes & réponses
 - SSL Off-loading
- Ce n'est PAS un serveur WEB

Interface d'administration

SCOUT_FRONTAUX_PROD_HTTP

		Queue			Session rate			Sessions					Bytes		Denied		Errors			Warnings		Server									
		Cur	Max	Limit	Cur	Max	Limit	Cur	Max	Limit	Total	LbTot	Last	In	Out	Req	Resp	Req	Conn	Resp	Retr	Redis	Status	LastChk	Wght	Act	Bck	Chk	Dwn	Dwntme	Thr
<input type="checkbox"/>	scout-f1	0	0	-	2	353		7	60	-	6 675 220	763 031	1s	17 134 299 329	117 268 222 919		0		148	115	486	0	10m41s UP	L4OK in 0ms	1	Y	-	3	1	29s	-
<input type="checkbox"/>	scout-f2	0	0	-	6	322		2	56	-	8 426 038	763 526	1s	18 073 706 806	124 764 597 240		0		289	93	994	0	12m45s UP	L4OK in 0ms	1	Y	-	5	2	25s	-
<input type="checkbox"/>	scout-f3	0	0	-	0	384		0	82	-	7 316 024	762 910	2s	20 637 336 235	119 113 680 953		0		303	162	1014	0	10m31s UP	L4OK in 0ms	1	Y	-	6	2	28s	-
<input checked="" type="checkbox"/>	scout-f4	0	0	-	9	300		2	52	-	5 676 550	763 504	0s	18 485 226 957	102 152 590 476		0		205	124	741	0	12m16s UP	L4OK in 0ms	1	Y	-	6	2	43s	-
<input type="checkbox"/>	scout-f5	0	0	-	13	318		2	58	-	6 229 155	763 207	1s	17 584 003 311	113 299 601 264		0		176	96	571	0	10m47s UP	L4OK in 0ms	1	Y	-	4	1	20s	-
	Backend	0	0		33	527		13	154	800	34 319 216	3 816 178	0s	91 914 572 638	576 598 692 852	0	0		1 121	590	3806	0	20d2h UP		5	5	0		0	0s	-

Choose the action to perform on the checked servers :

- Set state to READY
- Set state to DRAIN
- Set state to MAINT
- Health: disable checks
- Health: enable checks
- Health: force UP
- Health: force NOLB
- Health: force DOWN
- Agent: disable checks
- Agent: enable checks
- Agent: force UP
- Agent: force DOWN
- Kill Sessions

Apply

SCOUT_FRONTAUX_PPROD_HTTP

		Queue			Session rate					
		Cur	Max	Limit	Cur	Max	Limit	Cur	Max	Limit
<input type="checkbox"/>	scout-f1-preprod	0	0	-	0	195		0	11	
	Backend	0	0		0	195		0	11	

Choose the action to perform on the checked servers :

		Bytes		Denied		Errors			Warnings		Server								
		In	Out	Req	Resp	Req	Conn	Resp	Retr	Redis	Status	LastChk	Wght	Act	Bck	Chk	Dwn	Dwntme	Thr
	scout-f1-preprod	13 427 787	4 554 489 002		0		6	0	18	0	10m59s UP	L4OK in 0ms	1	Y	-	4	2	46s	-
	Backend	13 517 287	4 559 103 722	0	0		374	0	18	0	10m59s UP		1	1	0		2	46s	-

Apply

SCOUT_ONLYOFFICE_PROD_HTTP

		Queue			Session rate					
		Cur	Max	Limit	Cur	Max	Limit	Cur	Max	Limit
<input type="checkbox"/>	onlyoffice	0	0	-	0	72		3	20	
	Backend	0	0		0	72		3	20	200

Choose the action to perform on the checked servers :

		Bytes		Denied		Errors			Warnings		Server								
		In	Out	Req	Resp	Req	Conn	Resp	Retr	Redis	Status	LastChk	Wght	Act	Bck	Chk	Dwn	Dwntme	Thr
	onlyoffice	470 256	2 947 771 594		0		0	480	12	0	8m46s UP	L4OK in 0ms	1	Y	-	2	1	55s	-
	Backend	515 265	2 948 661 934	0	0		71	480	12	0	8m46s UP		1	1	0		1	55s	-

Apply

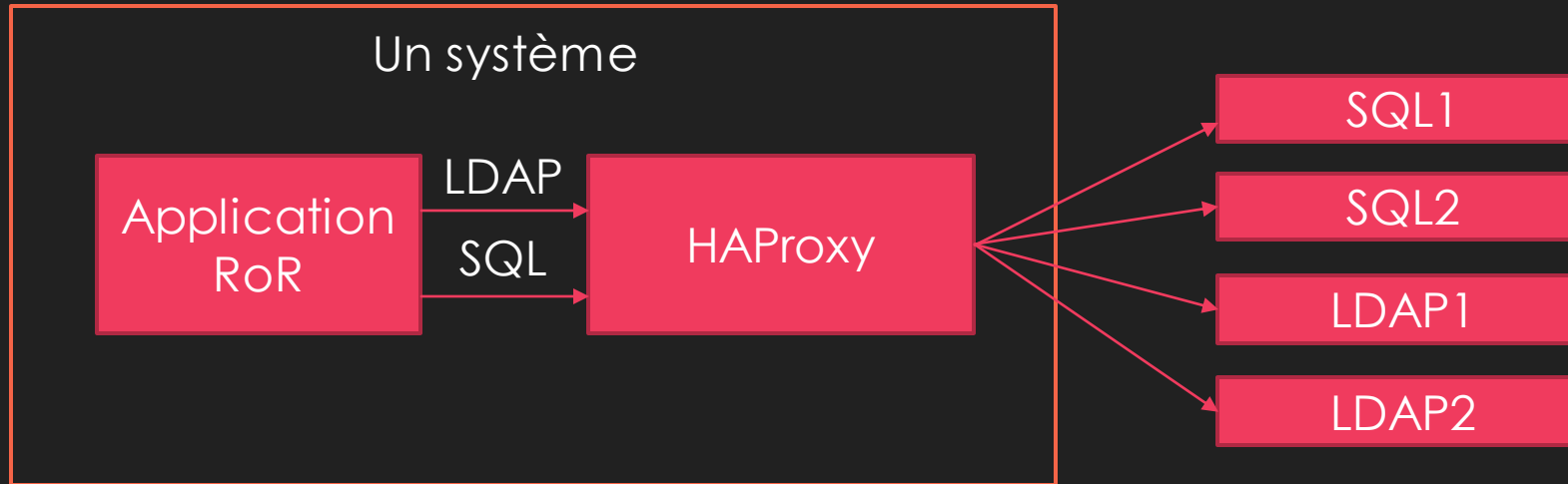
Usages

- Gestion d'un service entier
 - Exemple du SCOUT: HA, R. charge, SSL Offloading sur HTTP + Webdav, SMTP, IMAP, XMPP ...
 - Infra Web: SSL Offloading et point d'entrée unique pour les sites => flux réseaux et configuration simplifiés



- Faire de la « HA client », passer en « statefull »
 - Exemple des LDAP UNR: openldap fait très mal de la HA, on intercale un HAProxy
 - SSL & timeout
 - cluster MariaDB Galera (alternative à MaxScale)
 - Chaque client doit écrire sur le même serveur
 - CAS UNR: mises à jour applicatives

Usages (2)



Configuration minimaliste

```
# ----- # Git/SSH - # -----
```

```
listen GITSSH
```

```
    bind      comue-lb:22  
    mode      tcp  
    option    tcplog  
    server    comue-forge 1.3.3.7:22
```

Configuration avancée

```
default-server error-limit 10 rise 15 on-marked-down shutdown-sessions maxconn 125 maxqueue 125
```

```
listen UT
```

```
bind /var/run/haproxy/etab_UT.sock mode 666
```

```
mode tcp
```

```
timeout server 2m
```

```
timeout client 2m
```

```
timeout check 7s
```

```
option ldap-check
```

```
balance leastconn
```

```
server A ldap1.univ-toulouse.fr:636 check check-ssl ssl on-error mark-down
```

```
server B ldap2.univ-toulouse.fr:636 check check-ssl ssl on-error mark-down
```

```
server backup ldap-backup.univ-toulouse.fr:636 check check-ssl ssl backup
```


Conclusion

- Très rapide à mettre en place (simplicité de la configuration)
- Très bonne documentation
- Adapté à très grande échelle
- Peut être facilement redondé (Pacemaker)

- A envisager plutôt que des solutions nginx ou Apache, moins performantes