

Note d'Application

Bascule d'ALPHA via injection de route en BGP

Version du document : v1.2

Dernière mise à jour : 8 novembre 2013



Objectif

Cette note décrit comment créer une plateforme hautement disponible en utilisant le protocole de routage **BGP** pour choisir un ALOHA Load-Balancer disponible.

Limitations

Pour le moment, l'ALOHA Load-Balancer peut annoncer uniquement sa propre disponibilité, sans tenir compte de l'état des fermes de serveurs. Cela signifie que cette procédure peut être utilisée pour déclencher une bascule en fonction de la disponibilité d'un ALOHA, mais pas en fonction de la disponibilité ou de la capacité de traitement d'une seule ferme de serveurs.

Ce genre de fonctionnalité, Injection de route pour une IP de service en fonction de la capacité ou de la disponibilité d'une ferme arrivera plus tard dans l'ALOHA.

Ceci dit, il est possible d'écrire son propre script hébergé sur l'ALOHA pour maintenir à jour la configuration **BGP** en fonction de la capacité des fermes.

Difficulté



Versions concernées

– Aloha 4.2 et supérieur

Changelog

Version	Description
1.2	Ajout de la configuration pour routeur Extreme Networks
1.1	Ajout de la configuration pour routeur Brocade
1.0	Initial release

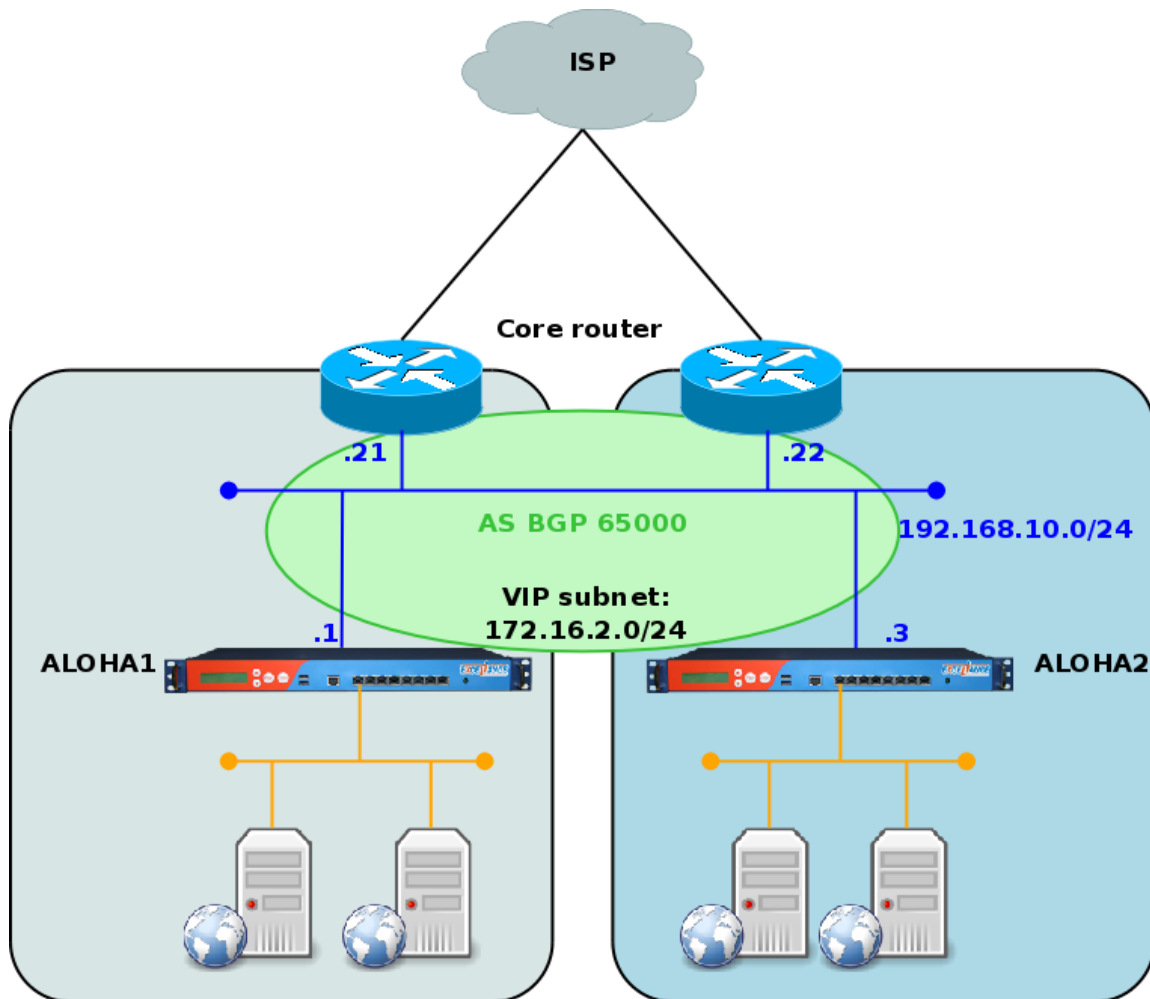
Contexte

Habituellement, ce genre d'architecture correspond à un usage "multi-datacenter", connectés via un MAN ou un WAN. Mais il peut être utilisé dans un même datacenter, sur le LAN.

Le principe est assez simple : construire un **Système Autonome BGP** (BGP Autonomous System ou AS) où l'ALOHA peut injecter des routes dans votre coeur de réseau. Le coeur de réseau sera configuré pour choisir un ALOHA en priorité s'il est disponible ou basculer sur le second : c'est une infrastructure **Active/Passive**.

Schéma

Le schéma ci-dessous montre comment les éléments sont interconnectés :



- Le **coeur de réseau** sera configuré pour envoyer le trafic vers **ALOHA1** et basculer sur **ALOHA2**.
- Le numéro d'AS BGP est le 65000, les routes injectées par les **ALOHAs** sont les IPs de services : 172.16.2.0/24.
- Le **coeur de réseau** et les **ALOHAs** peuvent communiquer via le réseau 192.168.10.0/24.



Le IPs de services pourraient être des IPs publiques



Dans cette note, nous ne fournirons que la configuration du routeur coeur #1 (192.168.10.21)

BGP configuration ALOHA


Dans l'ALOHA, le nom du service de routage dynamique est **bird**.

Démarrage de bird

Dans la GUI de l'ALOHA, cliquer sur l'onglet **Services** puis descendre tout en bas de la page et cliquer sur le lien **mode avancé**. Cliquer sur **OK**. *Les services Avancés sont affichés en rouge.*

Cliquer sur l'icône **edition** du service **bird** :  .

Ensuite, commenter ou supprimer la ligne **no autostart**.

Maintenant vous pouvez démarrer **bird** en cliquant sur l'icône **Démarrage** :  .

Répéter cette opération sur les deux ALOHAs.

Configuration de bird pour l'injection de route

Ci-dessous, la configuration **BGP** pour les deux **ALOHA**s, correspondant au diagram ci-dessus. Cette configuration demeure la même, peu importe le routeur BGP utilisé sur le coeur de réseau. Pour modifier la configuration de **bird**, cliquer sur l'icône d'**édition** du service **bird** dans l'onglet **Services**.

Configuration **ALOHA1** :

```
log syslog all;
router id 192.168.10.1;

protocol device {
  scan time 10;
}

protocol static VIPs {
  route 172.16.2.11/32 via 192.168.10.1;
  route 172.16.2.12/32 via 192.168.10.1;
  route 172.16.2.13/32 via 192.168.10.1;
}

protocol bgp {
  import none;
  export filter {
    if proto = "VIPs" then accept;
    reject;
  };
  local as 65000;
  source address 192.168.10.1;
  neighbor 192.168.10.21 as 65000;
}
```

Configuration **ALOHA2** :

```
log syslog all;
router id 192.168.10.3;

protocol device {
  scan time 10;
}

protocol static VIPs {
  route 172.16.2.11/32 via 192.168.10.3;
  route 172.16.2.12/32 via 192.168.10.3;
  route 172.16.2.13/32 via 192.168.10.3;
}

protocol bgp {
  import none;
  export filter {
    if proto = "VIPs" then accept;
    reject;
  };
  local as 65000;
  source address 192.168.10.3;
  neighbor 192.168.10.21 as 65000;
}
```

Une fois que la configuration de bird a été effectuée, vous devez le recharger en cliquant sur l'icône **reload** :



Configuration BGP coeur de réseau

Ce chapitre présente différents routeurs **BGP** avec leur configuration, correspondant au schéma en place.



Ces configuration sont des exemples basiques et peuvent nécessiter des modifications pour votre environnement.

bird

bird est un logiciel opensource qui peut être utilisé sur un coeur de réseau. Ci-dessous, la configuration **bird** pour accepter les annonces BGP des **ALOHA**s :

```
# Configure logging
log syslog { info, remote, warning, error, auth, fatal, bug };

router id 192.168.10.21;

filter aloha_vip {
  if net ~ 172.16.2.0/24 then accept;
  else reject;
}

protocol kernel {
  scan time 10;
  import none;
  export all;
}

protocol device {
  scan time 10;
}

protocol bgp aloha1 {
  local as 65000;
  export none;
  import filter aloha_vip;
  source address 192.168.10.21;
  neighbor 192.168.10.1 as 65000;
  default bgp_local_pref 300;
}

protocol bgp aloha2 {
  local as 65000;
  export none;
  import filter aloha_vip;
  source address 192.168.10.21;
  neighbor 192.168.10.3 as 65000;
  default bgp_local_pref 200;
```

```
}
```

Le poids (**bgp_local_pref**) est plus grand pour **ALOHA1**, il sera donc choisi en priorité s'il est disponible.

La table d'information de routage de **bird** devrait ressembler à ceci :

```
# birdc show route
BIRD 1.2.5 ready.
172.16.2.11/32   via 192.168.10.1 on eth0 [aloha1 16:35] * (100) [i]
                 via 192.168.10.3 on eth0 [aloha2 16:35] (100) [i]
172.16.2.13/32   via 192.168.10.1 on eth0 [aloha1 16:35] * (100) [i]
                 via 192.168.10.3 on eth0 [aloha2 16:35] (100) [i]
172.16.2.12/32   via 192.168.10.1 on eth0 [aloha1 16:35] * (100) [i]
                 via 192.168.10.3 on eth0 [aloha2 16:35] (100) [i]
```

La route primaire est celle indiquée par l'étoile * et **bird** l'utilisera en priorité.

Pour confirmer ceci, il suffit de regarder la table de routage du serveur hébergeant le coeur de réseau :

```
# ip route
172.16.2.13 via 192.168.10.1 dev eth0 proto bird
172.16.2.12 via 192.168.10.1 dev eth0 proto bird
172.16.2.11 via 192.168.10.1 dev eth0 proto bird
```

D'un point de vue du kernel, une seule route est connue.

- Si **ALOHA1** est indisponible, alors **bird** mettra à jour la table de routage du routeur avec l'IP de l'**ALOHA2** IP pour toutes les IPs de services.
- Si **ALOHA1** arrête d'annoncer une route, alors le **bird** mettra à jour la table de routage du routeur avec l'IP d'**ALOHA2** comme destination pour cette IP de service particulière.

Brocade

Cisco est l'un des équipementier réseau leader du marché.

La configuration ci-dessous montre comment configurer un router BGP **Brocade** pour accepter les injections de route des **ALOHAs** :

```
ip prefix-list aloha_vip deny 0.0.0.0/0
ip prefix-list aloha_vip permit 172.16.2.0/24 le 32

router bgp
local-as 65000
neighbor aloha peer-group
neighbor aloha remote-as 65000
neighbor 192.168.10.1 peer-group aloha
neighbor 192.168.10.1 description aloha1
neighbor 192.168.10.3 peer-group aloha
neighbor 192.168.10.3 description aloha3
!
address-family ipv4
neighbor 192.168.10.1 activate
neighbor 192.168.10.1 route-map in local_pref_300
neighbor 192.168.10.1 prefix-list aloha_vip in
neighbor 192.168.10.3 activate
neighbor 192.168.10.3 route-map in local_pref_100
neighbor 192.168.10.3 prefix-list aloha_vip in
exit-address-family
exit

route-map local_pref_300 permit 10
set local-preference 300

route-map local_pref_100 permit 10
set local-preference 100
```

(Désolé, pas d'exemple de table de routage disponible)

Cisco

Cisco est l'un des équipementier réseau leader du marché.

La configuration ci-dessous montre comment configurer un router BGP **Cisco** pour accepter les injections de route des **ALOHA**s :

```
!  
configure terminal  
!  
ip prefix-list aloha_vip deny 0.0.0.0/0  
ip prefix-list aloha_vip permit 172.16.2.0/24 le 32  
!  
router bgp 65000  
  bgp router-id 192.168.10.21  
  bgp log-neighbor-changes  
  neighbor aloha peer-group  
  neighbor aloha remote-as 65000  
  neighbor 192.168.10.1 peer-group aloha  
  neighbor 192.168.10.1 description aloha1  
  neighbor 192.168.10.3 peer-group aloha  
  neighbor 192.168.10.3 description aloha3  
  !  
  address-family ipv4  
    neighbor 192.168.10.1 activate  
    neighbor 192.168.10.1 localpref 300  
    neighbor 192.168.10.1 prefix-list aloha_vip in  
    neighbor 192.168.10.3 activate  
    neighbor 192.168.10.3 localpref 200  
    neighbor 192.168.10.3 prefix-list aloha_vip in  
    no auto-summary  
    no synchronization  
  exit-address-family  
!  
exit  
exit  
!
```

Maintenant, regardons la table de routage du routeur :

```
Router#sh ip bgp  
BGP table version is 4, local router ID is 192.168.10.21  
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,  
              r RIB-failure, S Stale  
Origin codes: i - IGP, e - EGP, ? - incomplete  
  
   Network        Next Hop           Metric LocPrf Weight Path  
* 172.16.2.11/32  192.168.10.3       100    200 i  
*>i              192.168.10.1       100    300 i  
* 172.16.2.12/32  192.168.10.3       100    200 i  
*>i              192.168.10.1       100    300 i  
* 172.16.2.13/32  192.168.10.3       100    200 i  
*>i              192.168.10.1       100    300 i
```

La table de routage des équipements **Cisco** est assez complète : on peut y voir le poids et le choix des routes. Nous pouvons clairement voir aussi que les routes ont été apprises via le protocole **iBGP**.

- Si **ALOHA1** est indisponible, alors le **routeur Cisco** mettra à jour sa table de routage avec l'IP d'**ALOHA2** comme destination pour toutes les IPs de services.

- Si **ALOHA1** cesse d'annoncer l'une des routes, alors le **routeur Cisco** mettra à jour sa table de routage avec l'IP de l'**ALOHA2** comme destination pour cette IP de service.

Extreme Networks

La configuration ci-dessous montre comment configurer un router BGP **Extreme Networks** pour accepter les injections de route des **ALOHA**s :

```
# bgp configuration
configure bgp AS-number 65000
configure bgp routerid 192.168.10.21
configure bgp local-preference 300

create bgp peer-group aloha
configure bgp peer-group aloha remote-AS-number 65000
configure bgp peer-group aloha route-policy in aloha_vip
enable bgp peer-group aloha

create bgp neighbor 192.168.10.1 peer-group "aloha"
configure bgp neighbor 192.168.10.1 description "aloha1"
configure bgp neighbor 192.168.10.1 peer-group aloha acquire-all
enable bgp neighbor 192.168.10.1

create bgp neighbor 192.168.10.3 peer-group "aloha"
configure bgp neighbor 192.168.10.3 description "aloha2"
configure bgp neighbor 192.168.10.3 peer-group aloha acquire-all
enable bgp neighbor 192.168.10.3

enable bgp

# policy configuration
edit policy aloha1_vip
entry filter1 {
  if match all {
    nlri 0.0.0.0/0 exact;
  } then {
    deny;
  }
}
entry filter2 {
  if match all {
    nlri 172.16.2.0/24;
  } then {
    permit;
    local-preference 300;
  }
}

edit policy aloha2_vip
entry filter1 {
  if match all {
    nlri 0.0.0.0/0 exact;
  } then {
    deny;
  }
}
```

```
entry filter2 {  
  if match all {  
    nlri 172.16.2.0/24;  
  } then {  
    permit;  
    local-preference 200;  
  }  
}
```

(Désolé, pas d'exemple de table de routage disponible)

OpenBGPD

OpenBGPD est le service de routage BGP fourni avec le système d'exploitation **OpenBSD**.

La configuration ci-dessous montre comment configurer **OpenBGPD** pour accepter les injections de routes de la part des **ALOHA**s :

```
AS 65000
router-id 192.168.10.21

log updates

group aloha {
  remote-as 65000
  neighbor 192.168.10.1 {
    descr "aloha1"
    set localpref 300
    announce none
  }
  neighbor 192.168.10.3 {
    descr "aloha2"
    set localpref 200
    announce none
  }
}

deny from any
allow from group aloha inet prefixlen 24 - 32
```

Maintenant, regardons la table de routage du routeur :

```
# bgpctl show rib
flags: * = Valid, > = Selected, I = via IBGP, A = Announced
origin: i = IGP, e = EGP, ? = Incomplete

flags destination          gateway          lpref   med aspath origin
I*>  172.16.2.11/32        192.168.10.1    300     0  i
I*   172.16.2.11/32        192.168.10.3    200     0  i
I*>  172.16.2.12/32        192.168.10.1    300     0  i
I*   172.16.2.12/32        192.168.10.3    200     0  i
I*>  172.16.2.13/32        192.168.10.1    300     0  i
I*   172.16.2.13/32        192.168.10.3    200     0  i
```

La base d'information de route (RIB) d'**OpenBGPD** donne beaucoup d'informations : nous pouvons voir le poids (**lpref**) des routes ainsi que celle actuellement choisie. Nous pouvons aussi clairement voir que les routes ont été apprises via le protocole **iBGP**.

- Si **ALOHA1** est indisponible, alors **OpenBGPD** met à jour la table de routage du routeur avec l'IP d'**ALOHA2** pour toutes les IPs de services.
- Si **ALOHA1** arrête d'annoncer une route, alors **OpenBGPD** mettra à jour la table de routage du routeur avec l'IP d'**ALOHA2** pour cette IP de service particulière.