ALOHA Load-Balancer

Microsoft Exchange 2010 deployment guide

Document version:	v1.4
ALOHA version concerned:	v4.2 and above
Microsoft Exchange Server:	2010 RTM, SP1, SP2, SP3
Last update date:	November 6th, 2013

1. Introduction 3
1.1. About Exceliance 3
1.2. About ALOHA Load-balancer 3
1.3. About this guide 3
1.4. Appliance supported 4
1.5. Aloha firmware versions supported 4
1.6. Microsoft Exchange version supported 4
1.7. Document history 4
1.8. Disclaimer 4
2. Introduction to Microsoft Exchange 2010 5
2.1. Exchange 2010 architecture 5
2.2. Client Access Services 6
2.3. Ports and protocols 7
2.4. Server affinity 7
2.5. Why using a load-balancer in an Exchange 2010 platform 7
3. ALOHA configuration for CAS servers 9
3.1. Configuration without SSL offloading 9
3.2. Configuration with SSL offloading 13
3.2.1. SSL configuration 13
3.2.2. Load-balancing configuration 15
4. Microsoft Exchange 2010 procedures 21
4.1. Set static TCP Port for MS Exchange RPC Client Access service 21
4.1.1. Procedure for Exchange 2010 21
4.1.2. Procedure for Exchange 2010 SP1 and SP2 21
4.2. Set static TCP Port for MS Exchange Address Book service 21
4.2.1. Procedure for Exchange 2010 21
4.2.2. Procedure for Exchange 2010 SP1 and SP2 22

1. Introduction

1.1. About Exceliance

Exceliance is a software company, editing the Application Delivery Controller called **ALOHA Load-Balancer**.

Headquartered in Jouy-en-Josas (Yvelines), Exceliance is part of the EXOSEC group. All its team (including R&D and technical support) are based in France.

Exceliance currently has around 100 customers in the banking, retail groups, energy and e-commerce industries and the public sector. Exceliance solutions are also used by many hosting providers.

The ALOHA Load-balancer is designed to improve performance, guarantee quality of service and ensure the availability of critical business applications, by dynamically balancing flows and queries on the company's various servers.

1.2. About ALOHA Load-balancer

The ALOHA Load-Balancer is designed to load-balance at layer 4 and control application delivery at layer 7. It is designed to **integrate simply** and quickly into any environment or architecture.

The main ALOHA benefits:

- Guarantee high availability of your applications and services as well as improve web application performance
- Makes infrastructures scalable and reliable
- Simplifies architecture: IPv6 frontend, SSL offloading, layer 7 routing

Developed using HAProxy open source load balancing software, Exceliance solutions are known for their processing performance, reliability and wealth of features. Offered at more affordable prices than other commercial solutions, they are easy to deploy and to administer.

ALOHA Load-balancer is available either as a **physical appliance** or as a **Virtual Appliance**. The Virtual appliance can run on top of **any hypervisor** available on the market.

1.3. About this guide

This guide first explains in the main lines how Exchange 2010 is designed and why a Load-Balancer makes sense with such platforms.

The guide also provides configuration templates to setup the ALOHA Load-Balancer for Microsoft Exchange 2010 for the two most common architectures: with or without SSL offloading.

The lastest version of this guide can be downloaded from Exceliance website: http://www.exceliance.fr/.

1.4. Appliance supported

All ALOHA Load-Balancers appliances can be used with Microsoft Exchange 2010.(physical and virtual ones).

1.5. Aloha firmware versions supported

ALOHA 4.2 and above are supported to load-Balance Microsoft Exchange 2010.

1.6. Microsoft Exchange version supported

ALOHA load-balancer can be used with the following versions of Microsoft Exchange:

- Microsoft exchange 2010
- Microsoft exchange 2010 SP1
- Microsoft exchange 2010 SP2
- Microsoft exchange 2010 SP3

1.7. Document history

Version	Date	Changes summary
V1.4	November, 6th, 2013	- Exchange 2010 SP3 support
V1.3	August, 5th, 2013	- turned listen sections into frontend / backend for better compatibility with ALOHA GUI - improved RPC services monitoring
V1.2	September, 27th 2012	- Link to CAS array deployment - minor updates in Aloha templates
V1.0	March 01, 2012	Initial release

1.8. Disclaimer

The Exchange 2010 configuration tips provided in this guide are purely informational. For more information about Microsoft Exchange 2010 tools and how to use them, please refer to Microsoft web site which is fully and properly documented.

This guide does not provide information on how to setup an Exchange cluster.

2. Introduction to Microsoft Exchange 2010

Microsoft Exchange provides businesses with email, calendar and contacts on the PC, phone and web.

One of the most interesting point of Microsoft Exchange 2010 is that you can now dedicates **roles** to servers. This new way of working allows administrator to build redundant platforms, using a load-balancer to allow clients to get connected on the services.

Thanks to its new design, Microsoft exchange is now **scalable**.

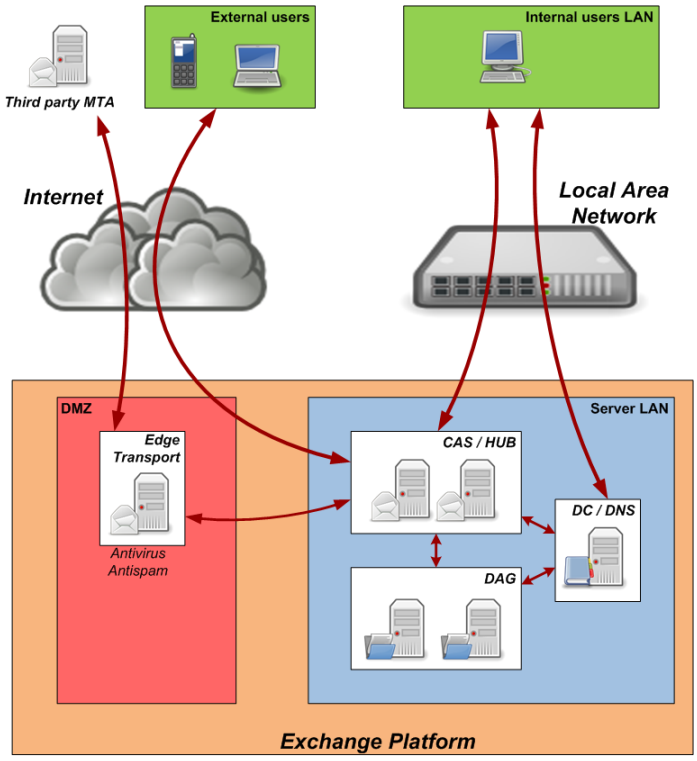
There are 5 roles: Mailbox, Client Access, Hub Transport, Unified Messaging and Edge Transport Server.

Role	Purpose
Client Access	Frontend servers on which client will get connected to access their emails, contacts and agenda
Edge Transport Server	handles the internet facing mail flow, with security features (anti-virus and anti-spam)
Hub Transport	Exchange 2010 mail router, within the organization
Mailbox	Servers hosting mails (in mailboxes) public folders
Unified Messaging	Enables the ability to deliver fax and voicemail to Outlook 2010 clients

The ALOHA Load-balancer can balance services from Client Access and Edge Transport Server.

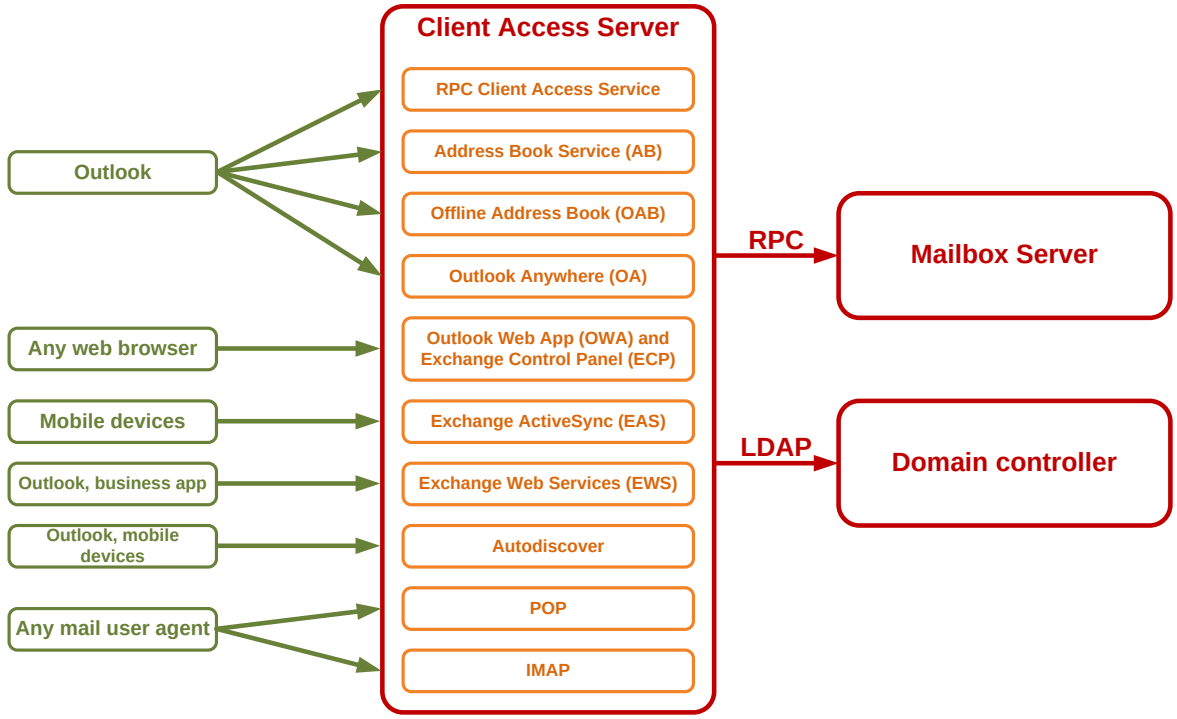
2.1. Exchange 2010 architecture

The diagram below shows how the different roles are used on an Exchange platform:



2.2. Client Access Services

The diagram below shows the services hosted by the CAS server and the interaction with both Active Directory and mailbox server. It also shows the client type per service.



Basically, the ALOHA Load-Balancer will stand between the **clients** and the **Client Access Servers**.

2.3. SMTP load-balancing

2.3.1. Using DNS

SMTP load-balancing can be achieved by setting up two or more DNS MX (Mail eXchanger) entries, each one pointing to an Exchange HUB server.

A SMTP client would use first the MX record with the lowest preference, then try the next higher preference.

2.3.2. Using a load-balancer

A load-balancer can be used to load-balance SMTP. You need a single MX entry, pointing to the load-balancer.

The load-balancer would balance requests among SMTP servers configured behind it.

Of course, you we can combine both solutions.

2.4. Ports and protocols

The table below summarizes the different ports and protocol involved on the Client Access servers:

TCP port	Protocol	CAS Service
80 and 443	HTTP	- Autodiscover (AS) - Exchange ActiveSync (EAS) - Exchange Control Panel (ECP) - Offline Address Book (OAB) - Outlook Anywhere (OA) - Outlook Web App (OWA)
110 and 995	POP3	POP3
135	TCP	RPC EndPoint Mapper (EPM)
143 and 993	IMAP4	IMAP4
60000	TCP	Static port for RPC Client Access Service
60001	TCP	Static port Address Book Service

The static ports for both RPC Client Access and Address Book service are chosen randomly by default. Microsoft recommends that any port within the range 59531 to 60554 should be used, and that the same ports should be used on all Client Access Servers within the same AD site.

Read Chapter 4 of this guide for more details on how to configure static ports on Microsoft Exchange CAS servers.

2.5. Server affinity

Affinity depends on the service. The table below summarizes the affinity requirements per service:

Persistence required	Persistence recommended	No persistence required
Exchange Control Panel (ECP)	Address Book Service (AB)	AutoDiscover (AD)
Exchange Web Service (EWS)	Exchange ActiveSync (EAS)	Offline Address Book (OAB)
Outlook Web App (OWA)	Outlook Anywhere (OA)	POP3
RPC Client Access Service	Remote PowerShell	IMAP4

2.6. Why using a load-balancer in an Exchange 2010 platform

First of all, even if Exchange 2010 provides services arrays, to ensure high-availability, it does not provide any load balancing mechanism.
That mean we need a third party appliance to balance traffic across Client Access Servers and services.

The services that can be load-balanced are the ones hosted by the Client Access Servers as well as SMTP for Edge Transport Servers.

Using a load-balancer to load-balance Microsoft Exchange 2010 will bring some benefits:

- **Application aware health checking**
  A load-balancer provides application layer health check which provides the status of the service itself and are more efficient than a simple ping.

- **Granular persistence methods**
  Depending on Exchange service, client software and architecture, different persistence methods must be applied.
- **SSL offloading**
  A load-balancer can handle SSL connection for the CAS array servers. That way, CAS servers can focus on their jobs.
- **Scale up**
  Building an architecture with a load-balancer allows scale up
- **Scale out**
  Splitting services on the load-balancer side, at the cost of more VIP and IP used, brings the ability to scale out the CAS array, dedicating some servers to services.

2.7. Exchange 2010 configuration

In order to ensure your CAS array is compatible with a HLB, follow the instructions provided by Microsoft:

- http://technet.microsoft.com/en-us/library/ee332317.aspx

Some excellent blog post also describes the procedure:

- http://telnet25.wordpress.com/2012/04/16/how-to-set-client-access-server-array/
- http://www.more2know.nl/2010/04/23/setup-exchange-2010-cas-array-to-load-balance-mapi/

3. ALOHA configuration for CAS servers

There are many architectures possible with the ALOHA about a Microsoft Exchange 2010 platform.

The two examples provided in this guide shows the two most implemented:

1. Basic load-balancing of both clear and encrypted services, without SSL offloading
2. Advanced load-balancing with SSL offloading on the ALOHA Load-Balancer

Of course the ALOHA can be used in more complicated scenarios, but we can't cover all of them in this guide.

3.1. Configuration without SSL offloading

On the GUI, browse the **LB Layer7** tab, then copy/paste the areas below, updating the part in **blue**, to meet your architecture needs, following the supplied instructions.

```
######## Default values for all entries till next defaults section
defaults
  mode tcp
  log global
  option tcplog
  balance leastconn
  option  dontlognull
  option  redispatch
  option  contstats
  timeout server 600s
  timeout client 600s
  timeout connect 5s
  timeout http-request 15s
  timeout http-keep-alive 15s
  timeout queue 60s
  retries 3
  default-server inter 15s rise 2 fall 2
  backlog 10000

# Persistence tables
backend sourceaddr
  stick-table size 10k type ip

# CAS 1 dedicated monitoring (for RPC services)
listen chk_CAS1
 bind 127.0.0.1:1001
 mode http
 monitor-uri /check
 monitor fail if { nbsrv lt 3 }
 server CAS1_epm 10.0.0.15:135   check
 server CAS1_ca  10.0.0.15:60000 check
 server CAS1_ab  10.0.0.15:60001 check
```

```
# CAS 2 dedicated monitoring (for RPC services)
listen chk_CAS2
 bind 127.0.0.1:1002
 mode http
 monitor-uri /check
 monitor fail if { nbsrv lt 3 }
 server CAS2_epm 10.0.0.16:135   check
 server CAS2_rpc 10.0.0.16:60000 check
 server CAS2_ab  10.0.0.16:60001 check


# Redirection to SSL frontend
frontend web
  bind 0.0.0.0:80
  mode http
  option httplog
  log global
  redirect prefix https://mail.xlc.local:443


# WEB configuration without SSL acceleration
backend bk_web_ssl
  balance leastconn
  mode tcp
  option tcplog
  log global
  stick on src table sourceaddr
  server CAS1 10.0.0.15:443 check
  server CAS2 10.0.0.16:443 check


frontend ft_web_ssl
  bind 0.0.0.0:443 name https
  mode tcp
  option tcplog
  log global
  default_backend bk_web_ssl
```

IP address where clients will get connected to

FQDN hosting your Outlook Web access service

label and IP address of CAS servers

IP address where clients will get connected to

```
# exchange end point mapper configuration
backend bk_exchange_epm
  balance leastconn
  mode tcp
  log global
  option tcplog
  stick on src table sourceaddr
  option httpchk HEAD /check HTTP/1.0
  timeout server 600s
  timeout connect 5s
  default-server inter 1s rise 1 fall 1 on-marked-down shutdown-sessions
  server CAS1 10.0.0.15:135 check addr 127.0.0.1 port 1001 observe layer4
  server CAS2 10.0.0.16:135 check addr 127.0.0.1 port 1002 observe layer4
```

label and IP address of CAS servers

dedicated server monitoring (points to the appropriate chk_CAS bind IP and port)

```
frontend ft_exchange_epm
  bind 0.0.0.0:135 name epm
  mode tcp
  log global
  option tcplog
  timeout client 600s
  default_backend bk_exchange_epm
```

IP address where clients will get connected to

```
# exchange client access
backend bk_exchange_ca
  balance leastconn
  mode tcp
  log global
  option tcplog
  stick on src table sourceaddr
  option httpchk HEAD /check HTTP/1.0
  timeout server 600s
  timeout connect 5s
  default-server inter 1s rise 1 fall 1 on-marked-down shutdown-sessions
  server CAS1 10.0.0.15:60000 weight 10 check addr 127.0.0.1 port 1001 observe layer4
  server CAS2 10.0.0.16:60000 weight 10 check addr 127.0.0.1 port 1002 observe layer4
```

label and IP address of CAS servers

dedicated server monitoring (points to the appropriate chk_CAS bind IP and port)

```
frontend ft_exchange_ca
  bind 0.0.0.0:60000 name ca
  mode tcp
  log global
  option tcplog
  timeout client 600s
  default_backend bk_exchange_ca

# exchange address book
backend bk_exchange_ab
  balance leastconn
  mode tcp
  log global
  option tcplog
  stick on src table sourceaddr
  option httpchk HEAD /check HTTP/1.0
  timeout server 600s
  timeout connect 5s
  default-server inter 1s rise 1 fall 1 on-marked-down shutdown-sessions
  server CAS1 10.0.0.15:60001 weight 10 check addr 127.0.0.1 port 1001 observe layer4
  server CAS2 10.0.0.16:60001 weight 10 check addr 127.0.0.1 port 1002 observe layer4
```

IP address where clients will get connected to

label and IP address of CAS servers

dedicated server monitoring (points to the appropriate chk_CAS bind IP and port)

```
frontend ft_exchange_ab
  bind 0.0.0.0:60001 name ab
  mode tcp
  log global
  option tcplog
  timeout client 600s
  default_backend bk_exchange_ab


# exchange IMAP
backend bk_exchange_imap
  balance leastconn
  mode tcp
  log global
  option tcplog
  option httpchk HEAD /check HTTP/1.0
  timeout server 600s
  timeout connect 5s
  default-server inter 1s rise 2 fall 3 on-marked-down shutdown-sessions
  server CAS1 10.0.0.15 weight 10 check port 143 observe layer4
  server CAS2 10.0.0.16 weight 10 check port 143 observe layer4
```

IP address where clients will get connected to

label and IP address of CAS servers

```
frontend ft_exchange_imap
  bind 0.0.0.0:143 name imap
  bind 0.0.0.0:993 name imaps
  mode tcp
  log global
  option tcplog
  timeout client 600s
  default_backend bk_exchange_imap


# exchange POP
backend bk_exchange_pop
  balance leastconn
  mode tcp
  log global
  option tcplog
  timeout server 600s
  timeout connect 5s
  default-server inter 1s rise 2 fall 3 on-marked-down shutdown-sessions
  server CAS1 10.0.0.15 weight 10 check port 110 observe layer4
  server CAS2 10.0.0.16 weight 10 check port 110 observe layer4



frontend ft_exchange_pop
  bind 0.0.0.0:110 name pop
  bind 0.0.0.0:995 name pops
  mode tcp
  log global
  option tcplog
  timeout client 600s
  default_backend bk_exchange_pop
```

IP address where clients will get connected to

label and IP address of CAS servers

IP address where clients will get connected to

3.2. Configuration with SSL offloading

3.2.1. SSL configuration

1. Certificate integration

On the GUI, browse the **SSL** tab, scroll down and click on **New** in the **SSL certificates** section, then fill the form as below:

- choose a SSL certificate name, usually this is the fqdn

- check the box "upload a private key"
- paste your private key
- click on upload
- check the box "upload certificate"
- paste your certificate
- click on upload

2. HTTPS offloading configuration

On the GUI, browse the **SSL** tab, then copy/paste the lines below, updating the parameters to your needs:

```
[ssl_exchange_http]
client = no
key =  /etc/ssl/frontends/mail.xlc.local/key.pem
cert = /etc/ssl/frontends/mail.xlc.local/crt.pem
accept  = 10.0.0.17:443
connect = /ssl:exchange_http
sendproxy = yes
```

SSL certificate name

IP address where clients will get connected to

3. IMAPS offloading

On the GUI, browse the **SSL** tab, then copy/paste the lines below, updating the parameters to your needs:

```
[ssl_exchange_imap]
client = no
key =  /etc/ssl/frontends/mail.xlc.local/key.pem
cert = /etc/ssl/frontends/mail.xlc.local/crt.pem
accept  = 10.0.0.17:993
connect = /ssl:exchange_imap
```

SSL certificate name

IP address where clients will get connected to

4. POPS offloading

On the GUI, browse the **SSL** tab, then copy/paste the lines below, updating the parameters to your needs:

```
[ssl_exchange_pop]
client = no
key =  /etc/ssl/frontends/mail.xlc.local/key.pem
cert = /etc/ssl/frontends/mail.xlc.local/crt.pem
accept  = 10.0.0.17:995
connect = /ssl:exchange_pop
```

SSL certificate name

IP address where clients will get connected to

3.2.2. Load-balancing configuration

On the GUI, browse the **LB Layer7** tab, then copy/paste the areas below, updating the part in **blue**, to meet your architecture needs, following the supplied instructions.

```
# Default values applied to all other section
# unless being locally overridden
defaults
  mode tcp
  log global
  option tcplog
  balance leastconn
  option  dontlognull
  option  redispatch
  option  contstats
  timeout server 600s
  timeout client 600s
  timeout connect 5s
  timeout http-request 15s
  timeout http-keep-alive 15s
  timeout queue 60s
  retries 3
  default-server inter 15s rise 2 fall 2
  backlog 10000

# Persistence tables
backend sourceaddr
  stick-table size 10k type ip

backend authorization
  stick-table size 10k expire 1h type string
```

```
# WEB configuration.
frontend ft_web
  bind /ssl:web accept-proxy name https
  bind 10.0.0.17:80 name http
  mode http
  option httplog
  log global
  option forwardfor

  redirect prefix https://mail.xlc.local if { dst_port 80 }

  redirect location /owa  if { path / }
  use_backend bk_activesync if { path_beg -i /microsoft-server-activesync }
  use_backend bk_oa if { path_beg -i /rpc/rpcproxy.dll }
  use_backend bk_autodiscover if { path_beg -i /autodiscover }
  use_backend bk_ews if { path_beg -i /ews }
  use_backend bk_oab if { path_beg -i /oab }
  default_backend bk_owa_ecp


# Activesync services with basic Auth
backend bk_activesync
  mode http
  option httplog
  stick on hdr(Authorization) table authorization
  option httpchk GET /Microsoft-Server-ActiveSync/ HTTP/1.1\r\nHost:\ mail.xlc.local
  http-check expect rstatus (2..|3..|401)
  server CAS1 10.0.0.15:80 check
  server CAS2 10.0.0.16:80 check

# Outlook anywhere services
backend bk_oa
  mode http
  option httplog
  stick on src table sourceaddr
  option httpchk RPC_IN_DATA /rpc/rpcproxy.dll?mail.xlc.local:6001 HTTP/1.1\r\nUser-Agent:\
MSRPC\r\nHost:\ mail.xlc.local
  http-check expect rstatus (2..|3..|401)

  server CAS1 10.0.0.15:80 check
  server CAS2 10.0.0.16:80 check

# Autodiscover services
backend bk_autodiscover
  mode http
  option httplog
  option httpchk GET /Autodiscover/Autodiscover.xml HTTP/1.1\r\nUser-Agent:\
Mozilla/5.0\r\nHost:\ mail.xlc.local
  http-check expect rstatus (2..|3..|401)

  server CAS1 10.0.0.15:80 check
  server CAS2 10.0.0.16:80 check
```

IP address where clients will get connected to

hostname hosting the services

hostname hosting the services

label and IP address of CAS servers

hostname hosting the services

label and IP address of CAS servers

hostname hosting the services

label and IP address of CAS servers

```
# Exchange web services
backend bk_ews
  mode http
  option httplog
  stick on src table sourceaddr
  server CAS1 10.0.0.15:80 track bk_owa_ecp/CAS1
  server CAS2 10.0.0.16:80 track bk_owa_ecp/CAS2


# Outlook address book
backend bk_oab
  mode http
  option httplog

  server CAS1 10.0.0.15:80 track bk_owa_ecp/CAS1
  server CAS2 10.0.0.16:80 track bk_owa_ecp/CAS2


# Outlook Web Application and Exchange Control panel services
backend bk_owa_ecp
  mode http
  option httplog
  cookie ALBWA insert indirect nocache
  option httpchk GET /owa/auth/logon.aspx?url=http://mail.xlc.local/owa/&reason=0
HTTP/1.1\r\nUser-Agent:\ Mozilla/5.0\r\nHost:\ mail.xlc.local
  server CAS1 10.0.0.15:80 cookie CAS1 check
  server CAS2 10.0.0.16:80 cookie CAS2 check



# exchange end point mapper configuration
backend bk_exchange_epm
  balance leastconn
  mode tcp
  log global
  option tcplog
  stick on src table sourceaddr
  option httpchk HEAD /check HTTP/1.0
  timeout server 600s
  timeout connect 5s
  default-server inter 1s rise 1 fall 1 on-marked-down shutdown-sessions
  server CAS1 10.0.0.15:135 check addr 127.0.0.1 port 1001 observe layer4
  server CAS2 10.0.0.16:135 check addr 127.0.0.1 port 1002 observe layer4
```

label and IP address of CAS servers

server label from the owa_ecp farm to track status

label and IP address of CAS servers

server label from the owa_ecp farm to track status

hostname hosting the services

label, cookie and IP address of CAS servers

label and IP address of CAS servers

dedicated server monitoring (points to the appropriate chk_CAS bind IP and port)

```
frontend ft_exchange_epm
  bind 0.0.0.0:135 name epm
  mode tcp
  log global
  option tcplog
  timeout client 600s
  default_backend bk_exchange_epm


# exchange client access
backend bk_exchange_ca
  balance leastconn
  mode tcp
  log global
  option tcplog
  stick on src table sourceaddr
  option httpchk HEAD /check HTTP/1.0
  timeout server 600s
  timeout connect 5s
  default-server inter 1s rise 1 fall 1 on-marked-down shutdown-sessions
  server CAS1 10.0.0.15:60000 weight 10 check addr 127.0.0.1 port 1001 observe layer4
  server CAS2 10.0.0.16:60000 weight 10 check addr 127.0.0.1 port 1002 observe layer4
```

IP address where clients will get connected to

label and IP address of CAS servers

dedicated server monitoring (points to the appropriate chk_CAS bind IP and port)

```
frontend ft_exchange_ca
  bind 0.0.0.0:60000 name ca
  mode tcp
  log global
  option tcplog
  timeout client 600s
  default_backend bk_exchange_ca

# exchange address book
backend bk_exchange_ab
  balance leastconn
  mode tcp
  log global
  option tcplog
  stick on src table sourceaddr
  option httpchk HEAD /check HTTP/1.0
  timeout server 600s
  timeout connect 5s
  default-server inter 1s rise 1 fall 1 on-marked-down shutdown-sessions
  server CAS1 10.0.0.15:60001 weight 10 check addr 127.0.0.1 port 1001 observe layer4
  server CAS2 10.0.0.16:60001 weight 10 check addr 127.0.0.1 port 1002 observe layer4
```

IP address where clients will get connected to

label and IP address of CAS servers

dedicated server monitoring (points to the appropriate chk_CAS bind IP and port)

```
frontend ft_exchange_ab
  bind 0.0.0.0:60001 name ab
  mode tcp
  log global
  option tcplog
  timeout client 600s
  default_backend bk_exchange_ab


# exchange IMAP
backend bk_exchange_imap
  balance leastconn
  mode tcp
  log global
  option tcplog
  option httpchk HEAD /check HTTP/1.0
  timeout server 600s
  timeout connect 5s
  default-server inter 1s rise 2 fall 3 on-marked-down shutdown-sessions
  server CAS1 10.0.0.15 weight 10 check port 143 observe layer4
  server CAS2 10.0.0.16 weight 10 check port 143 observe layer4


frontend ft_exchange_imap
  bind 0.0.0.0:143 name imap
  bind 0.0.0.0:993 name imaps
  mode tcp
  log global
  option tcplog
  timeout client 600s
  default_backend bk_exchange_imap


# exchange POP
backend bk_exchange_pop
  balance leastconn
  mode tcp
  log global
  option tcplog
  timeout server 600s
  timeout connect 5s
  default-server inter 1s rise 2 fall 3 on-marked-down shutdown-sessions
  server CAS1 10.0.0.15 weight 10 check port 110 observe layer4
  server CAS2 10.0.0.16 weight 10 check port 110 observe layer4
```

IP address where clients will get connected to

label and IP address of CAS servers

IP address where clients will get connected to

label and IP address of CAS servers

```
frontend ft_exchange_pop
  bind 0.0.0.0:110 name pop
  bind 0.0.0.0:995 name pops
  mode tcp
  log global
  option tcplog
  timeout client 600s
  default_backend bk_exchange_pop
```

IP address where clients will get connected to

4. Microsoft Exchange 2010 procedures

4.1. Set static TCP Port for MS Exchange RPC Client Access service

More information about this procedure can be found here:

http://social.technet.microsoft.com/wiki/contents/articles/configure-static-rpc-ports-on-an-exchange-2010-client-access-server.aspx

4.1.1. Procedure for Exchange 2010

On your CAS servers, do:

1. Open regedit
2. Browse to the key **HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\services\MSExchangeRPC**
3. Create a new key named **ParametersSystem**
4. Under this key create a **REG_DWORD** named **TCP/IP Port**
5. **Right click** then **Modify** on the TCP/IP Port REG_DWORD
6. Fill the form: **Value** with the TCP port number and check **Decimal** base.
7. Click on **OK**

Microsoft recommends using a port from the range **59531 to 60554**, and the same one on all the CAS servers.

Bear in mind that the port number must be different for **RPC Client Access** and **Address Book service**.

4.1.2. Procedure for Exchange 2010 SP1 and SP2

This is the same procedure than Exchange 2010. Please read paragraph 4.1.1.

4.2. Set static TCP Port for MS Exchange Address Book service

4.2.1. Procedure for Exchange 2010

On your CAS servers, do:

1. Open an explorer window
2. Browse to **C:\Program Files\Microsoft\Exchange Server\V14\Bin** directory
3. Open the file named **microsoft.exchange.addressbook.service.exe.config** with notepad
4. Find the string **RpcTcpPort**
5. Replace the Value by the TCP port number you want
6. Close the file

Microsoft recommends using a port from the range **59531 to 60554**, and the same one on all the CAS servers.

4.2.2. Procedure for Exchange 2010 SP1 and SP2

On your CAS servers, do:

1. Open regedit
2. Browse to the key **HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\services\MSExchangeAB\Parameters**
3. Under this key create a **REG_SZ** named **RpcTcpPort**
4. **Right click** then **Modify** on the RpcTcpPort REG_SZ
5. Fill the field **Value** with the TCP port number you want
6. Click on **OK**

Microsoft recommends using a port from the range **59531 to 60554**, and the same one on all the CAS servers.

When upgradring Exchange 2010 to SP1, it is recommended to create this registry key before the upgrade, that way, once the CAS server would reboot, there won't be any outage.