Configuring TWC with a proxy

On this page:

nGinx Scripts

install_proxy_nginx_tcp.sh

nginx.conf.template.tcp

stream.conf.template.tcp

install_proxy_nginx_https.sh

nginx.conf.template.https

stream.conf.template.https

https.conf.template

HAProxy Scripts

install_proxy_haproxy15_tcp.sh

install_proxy_haproxy18_tcp.sh

haproxy.cfg.template.tcp

install_proxy_haproxy15_https.sh

install_proxy_haproxy18_https.sh

haproxy.cfg.template.https

⚠ Please note that this information is provided as a courtesy only and support services are not offered for any of the features described in this article.

Introduction

There are environments in which the TWCloud applications need to be fronted by a proxy. The most widespread use case for this is port restrictions whereby the native ports cannot be exposed. Typically, all external traffic is restricted to a single port, such as 443, which is allowed to traverse corporate firewalls or proxies.

In order to configure a proxy, we first need to understand the traffic flows, since each traffic flow needs to be addressed. Teamwork Cloud is composed of 3 services (Webapp, Authserver, Teamwork Cloud), which need to expose 4 traffic flows (or port bindings) to function.

- Webapp (native port 8443 - http/s)
- Authserver (native port 8555 - http/s)
- Teamwork Cloud
  - REST API (native port 8111 - http/s)
  - Client Communication (native port 3579 -TCP cleartext or native port 10002 - TLS/TCP encrypted) - 10002 is the default port used by the client and is configured in the TWCloud Admin Settings page

Additionally, if the FlexNet license server is running on the same instance with the same port constraints, a TCP proxy must be created for it, forwarding to our cameo vendor daemon (native port 1101 - TCP).

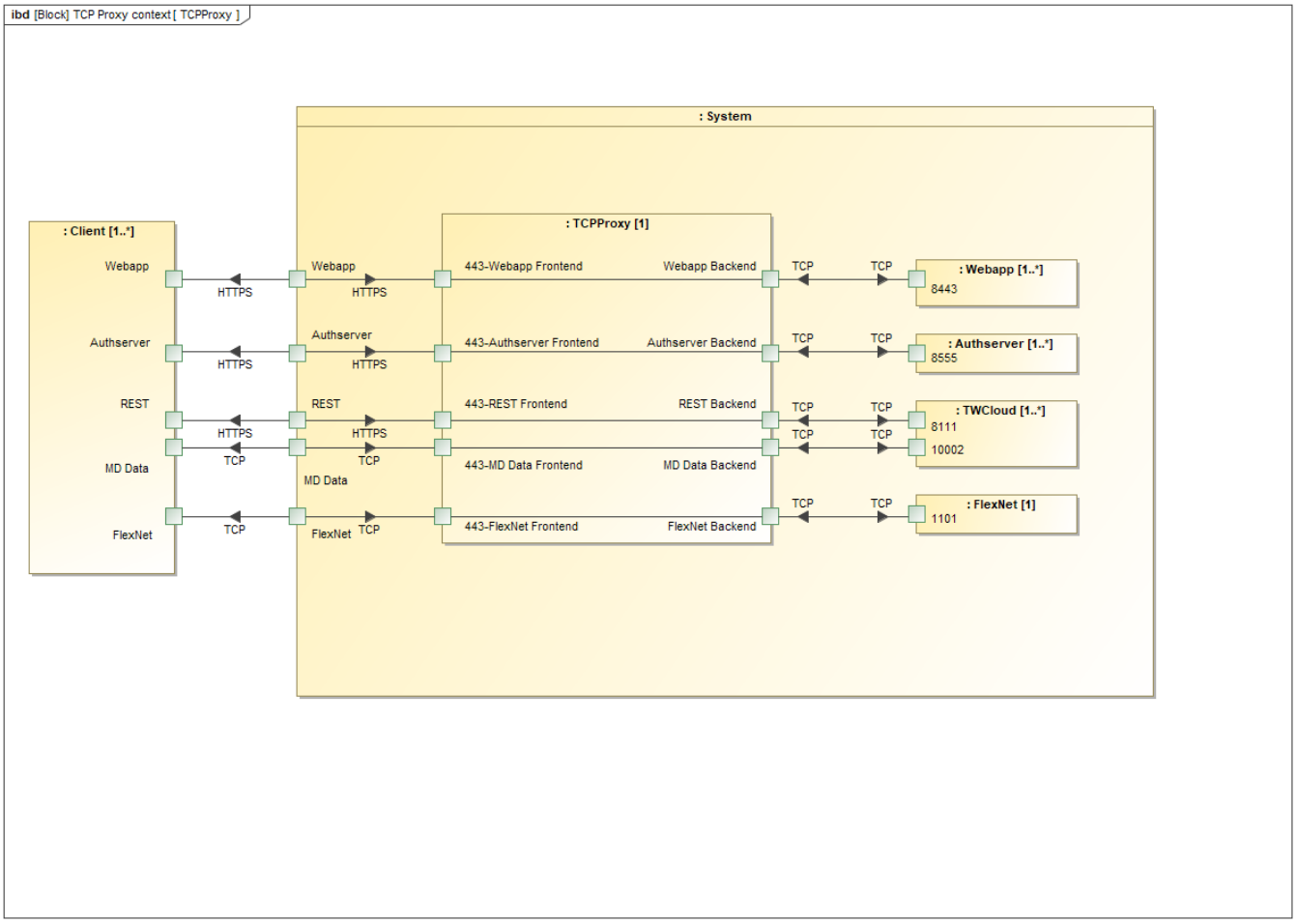
Since in TCP/IP you can only bind a single instance of a port to an IP address, the instance will need to have multiple IP addresses in which to bind each traffic flow.

Traffic flows are tied to frontends (the part of the proxy that is exposed to the external world, which receives the requests and forwards to the backends) and backends (handling of the actual requests).

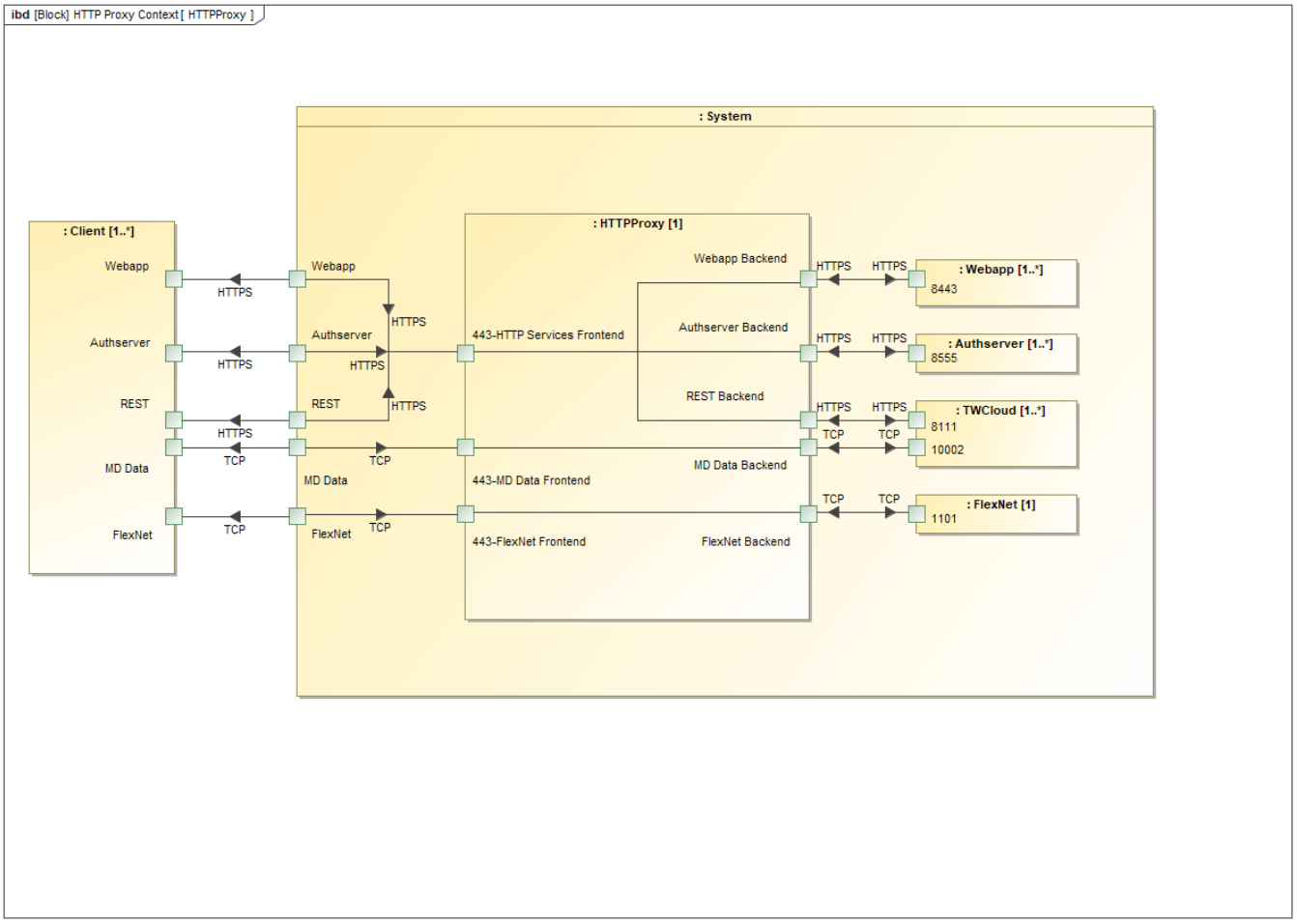
The number of IP addresses required depends on the type of proxying which we configure.

There are 2 types of proxying - TCP (in which case the proxying is done at Layer 4 of the OSI model) or HTTP (in which case the proxying is done at Layer 7 of the OSI).

Layer 4 proxying is the lighter-weight of the 2 methods since it simply forwards incoming packets from the frontend to its associated backend. In this case, we need 4 IP addresses (5 if proxying FlexNet as well) in order to bind to each data flow.



Layer 7 proxying inspects the actual content of the data coming through the proxy. Therefore, SSL termination takes place at the proxy. In doing so, we can now manipulate the data coming through and take specific actions. Since each of the HTTPS services exposes a path (/webapp for the TWCloud Admin Console, /osmc for the REST API, and /authentication for the authserver), we can treat the incoming data as a single flow (a single frontend), and have the proxy send the request to the respective backend based on the path of the request.



There are 2 classes of proxies/load balancers. Hardware (such as F5 Big-IP, Cirtix NetScaler) which are external to the application instance, and software (such as nGinx, HAProxy) which can be run externally (on a dedicated instance) or on the same instance as the application.

Proxy configuration

This document will outline how to deploy both Layer 4 (TCP) and Layer 7 (HTTPS) proxies utilizing both nGinx and HAProxy. The Layer 7 configuration is a hybrid configuration in that it also includes Layer 4 proxying for the MD client data stream and FlexNet server.

When we initially deploy Teamwork Cloud, we specified a local IP address. This will become the IP address for the Webapp frontend. Keep in mind that the native ports (8443, 8111, 8555, MDTWCloud port, and the cameo vendor daemon for FlexNet) bind to all interfaces.

For this example, we have an instance with 5 IP addresses - 10.254.254.31 (Webapp), 10.254.254.32 (REST), 10.254.254.33 (Authserver), 10.254.254.34 (MDTWC), and 10.254.254.35 (FlexNet).

The configurations are the most basic configurations in order for the system to operate. You may add features or change behaviors by modifying these files.

It is noteworthy that in order to simplify all of the aspects of certificates, you should either use a wildcard certificate or one which contains SAN's for all of the public fqdn's and/or IP addresses.

In order to facilitate the deployment, we have prepackaged installation scripts and template files which will install the proxies from the respective repositories as well as generate the basic configuration files.

nGinx

Layer 4 Proxy

As depicted in the TCP Proxy Context IBD, we have 5 frontends listening on port 443. Therefore, the instance must have 5 local IP addresses to which to bind port 443.

After deploying the Proxy, we need to make some changes to **authserver.properties** in order to allow the proxied TWCloud Admin console access.

authserver.properties

```
#-------------------------------------------------
# Host name or IP of the server instance.
#-------------------------------------------------
# server.public.host needs to be modified to reflect the authserver frontend IP address or FQDN
server.public.host=10.254.254.33
# server.public.port is set to the port number on which the authserver frontend listeens to
server.public.port=443

# authentication.redirect.uri.whitelist needs to have the Webapp's URL appened to the string.  If using port
443 as the public port for webapp, the port number is ommitted
authentication.redirect.uri.whitelist=https://10.254.254.31:8443/webapp/,https://10.254.254.31:8111/,
https://md_redirect,https://10.254.254.31/webapp/
```

install_proxy_nginx_tcp.sh

```
#!/bin/bash
#
# nGinx Proxy/LB depoyment and configuration script
#
# Benjamin Krajmalnik (benjamin.krajmalnik@3ds.com)
#
NGINX_CONF=/etc/nginx/nginx.conf
STREAM_CONF=/etc/nginx/conf.d/stream.conf

echo "This script will deploy and configure nGinx as a local proxy where all services are bound to port 443"
echo ""
echo "--------------------------------------------------------------------------------"
echo ""
read -e -p "Please enter the port number for MD->TWCloud Communications (Settings/Secured Connection in
TWCloud Admin):  "  -i "10002"  SSL_PORT
echo ""
read -e -p "Please enter the IP Address for the Webapp Proxy:  "  -i ""  WEBAPP_IP
echo ""
read -e -p "Please enter the IP Address for the REST API Proxy:  "  -i ""  REST_IP
echo ""
read -e -p "Please enter the IP Address for the Authserver Proxy:  "  -i ""  AUTH_IP
echo ""
read -e -p "Please enter the IP Address for the MD->TWCloud Proxy:  "  -i ""  MD_IP
echo ""
read -e -p "Please enter the IP Address for the FlexNET Proxy:  "  -i ""  FLEX_IP
echo ""
echo "--------------------------------------------------------------------------------"
echo ""
OS=$(cat /etc/redhat-release | cut -f 1 -d " ");
if [ $OS = 'CentOS' ]
then
        echo "Installing epel-release for CentOS"
        yum -y -q install epel-release
else
        echo "Installing epel-release for RHEL"
        rpm -ivh https://dl.fedoraproject.org/pub/epel/epel-release-latest-7.noarch.rpm
        yum -y -q update
fi
echo ""
echo "Installing nGinx"
yum -y -q install nginx
echo ""
echo "Reconfiguring SELinux for http connections"
echo ""
setsebool httpd_can_network_connect on -P
echo "Copying nGinx configuration templates"
\cp -f nginx.conf.template.tcp $NGINX_CONF
\cp -f stream.conf.template.tcp $STREAM_CONF
echo ""
echo "Applying configuration changes"
```

```
echo ""

sed -i "s/:SSL_PORT/:$SSL_PORT/g" $STREAM_CONF
sed -i "s/WEBAPP_IP:/$WEBAPP_IP:/g" $STREAM_CONF
sed -i "s/REST_IP:/$REST_IP:/g" $STREAM_CONF
sed -i "s/AUTH_IP:/$AUTH_IP:/g" $STREAM_CONF
sed -i "s/MD_IP:/$MD_IP:/g" $STREAM_CONF
sed -i "s/FLEX_IP:/$FLEX_IP:/g" $STREAM_CONF
echo ""
echo "Configuring firewall rule to allow port 443"
FWZONE=$(firewall-cmd --get-default-zone)
echo "Discovered firewall zone $FWZONE"
cat <<EOF | tee /etc/firewalld/services/nginx.xml
<?xml version="1.0" encoding="utf-8"?>
<service version="1.0">
    <short>nginx</short>
    <description>nginx</description>
    <port port="443" protocol="tcp"/>
</service>
EOF
sleep 5
firewall-cmd --zone=$FWZONE --add-service=nginx --permanent
firewall-cmd --reload
echo "nGinx Local Proxy Deployment Complete"
```

nginx.conf.template.tcp

```
# For more information on configuration, see:
#   * Official English Documentation: http://nginx.org/en/docs/
#   * Official Russian Documentation: http://nginx.org/ru/docs/

user nginx;
worker_processes auto;
error_log /var/log/nginx/error.log;
pid /run/nginx.pid;

# Load dynamic modules. See /usr/share/nginx/README.dynamic.
include /usr/share/nginx/modules/*.conf;

events {
    worker_connections 1024;
}

#  stream.conf contains the configuration for our TWCloud Proxy/Load balancer

include conf.d/stream.conf;
```

stream.conf.template.tcp

```
# stream core module provides TCP/UDP proxying and load balancing

stream {

        #
        # By default, SELinux only allows the http process access to the following ports (semanage ports -l
| grep http)
        #        http_port_t                    tcp      80, 81, 443, 488, 8008, 8009, 8443, 9000
        # In order for the proxy to connect to other ports, they need to either be added, or a simpler
solution is to allow
        #        http to connect to all ports, by issuing the following command
        #                sudo setsebool httpd_can_network_connect on -P
        #
        # application.conf in TWCloud needs to be configured to turn off the internal load balancing in
TWCloud on all nodes
```

```
        #     setting:  load_balancer = false (default is true) - make sure setting is uncommented
        #
        #
        # Additional configuration in TWC
        #
        #        Create a separate keystore for webapp, authserver, and twc, based on the common names for
each server
        #        i.e.  twcadmin.domain (webapp - 8443), twc.domain (twc - 8111), twcauth.domain (authserver
- 8555)
        #
        #        These keystores are used an all nodes, since we will be proxying and end user systems will
not see the individual nodes
        #
        # Notation in template
        #
        # 127.0.0.1 is the private (local) IP addresses for this node
        # SSL_PORT is the user defined port to which TWCloud will bind for TLS communications between the MD
clients and TWC
        #        If you do not want to use SSL, then you can use the default of 3579
        #
        # WEBAPP_IP is the local IP address on this machine to which we will bind access to TWCloud Admin
(8443)
        # REST_IP is the local IP address on this machine to which we will bind access to the REST API (8111)
        # AUTH_IP is the local IP address on this machine to which we will bind access to the Authserver
(8555)
        # MD_IP is the local IP address on this machine to which we will bind access to the MD->TWCloud data
stream (SSL_PORT)
        # FLEX_IP is the local IP address on this machine to which we will bind access to the cameo daemon
on the FLEX server (1101)
        #
        # Each of these IP's would be resolvable to the service FQDN (and common name of TWCloud Admin,
REST, and Authserver).
        #
        #

        # define our proxied server groups
        # hash directive is a mechanism for traffic from a given client to go to a given server
        # For a single node proxy it is not necessary, but is in the configuration for consistency when the
server group consists of a load balanced group
        # the upstream servers are the nodes running TWC


        upstream webapp_servers {
                hash $remote_addr consistent;
                server 127.0.0.1:8443;
        }

        upstream auth_servers {
                hash $remote_addr consistent;
                server 127.0.0.1:8555;
        }

        upstream rest_servers {
                hash $remote_addr consistent;
                server 127.0.0.1:8111;
        }

        upstream md_servers {
                hash $remote_addr consistent;
                server 127.0.0.1:SSL_PORT;
        }

        # define the proxy listeners

        server {
                listen WEBAPP_IP:443;
                proxy_buffer_size 16k;
                proxy_pass webapp_servers;
        }

        server {
```

```
                listen AUTH_IP:443;
                proxy_buffer_size 16k;
                proxy_pass auth_servers;
        }

        server {
                listen REST_IP:443;
                proxy_buffer_size 16k;
                proxy_pass rest_servers;
        }

        server {
                listen MD_IP:443;
                proxy_buffer_size 16k;
                proxy_pass md_servers;
        }

        # in this particular configuration, the FLEXNet server is running on the same server as the proxy
        # therefore, proxy_pass is pointing to localhost, since the cameo daemon binds to all ports
        # If Flex is running on a seprate instance, specify that instance's IP address

        server {
                listen FLEX_IP:443;
                proxy_buffer_size 16k;
                proxy_pass 127.0.0.1:1101;
        }
}
```

Layer 7 Proxy

As depicted in the HTTP Proxy Context IBD, we have 3 frontends listening on port 443. Therefore, the instance must have 3 local IP addresses to which to bind port 443.

After deploying the Proxy, we need to make some changes to **authserver.properties** in order to allow the proxied TWCloud Admin console access. Unlike the TCP Proxy where each stream was bound to its own IP address, all 3 HTTPS streams are handled by a single frontend. Therefore, the **serverver.public.host**, in this case, is the primary IP address.

⚠ In a Layer 7 HTTPS proxy, SSL termination takes place at the proxy. For this to take place, you will need a PEM encoded file containing the private key and certificates for the server, including the full certificate chain.

authserver.properties

```
#-------------------------------------------------
# Host name or IP of the server instance.
#-------------------------------------------------
# server.public.host needs to be modified to reflect the authserver frontend IP address or FQDN
server.public.host=10.254.254.31
# server.public.port is set to the port number on which the authserver frontend listeens to
server.public.port=443

# authentication.redirect.uri.whitelist needs to have the Webapp's URL appened to the string.  If using port
443 as the public port for webapp, the port number is ommitted
authentication.redirect.uri.whitelist=https://10.254.254.31:8443/webapp/,https://10.254.254.31:8111/,
https://md_redirect,https://10.254.254.31/webapp/
```

install_proxy_nginx_https.sh

```
#!/bin/bash
#
# nGinx Proxy/LB depoyment and configuration script
#
# Benjamin Krajmalnik (benjamin.krajmalnik@3ds.com)
#
```

```
NGINX_CONF=/etc/nginx/nginx.conf
STREAM_CONF=/etc/nginx/conf.d/stream.conf
HTTPS_CONF=/etc/nginx/conf.d/https.conf

echo "This script will deploy and configure nGinx as a local proxy where all services are bound to port 443"
echo ""
echo "--------------------------------------------------------------------------------"
echo ""
read -e -p "Please enter the port number for MD->TWCloud Communications (Settings/Secured Connection in
TWCloud Admin):  "  -i "10002"  SSL_PORT
echo ""
read -e -p "Please enter the IP Address for the https proxy (Webapp/AUthserver/REST):  "  -i ""  WEBAPP_IP
echo ""
read -e -p "Please enter the IP Address for the MD->TWCloud Proxy:  "  -i ""  MD_IP
echo ""
read -e -p "Please enter the IP Address for the FlexNET Proxy:  "  -i ""  FLEX_IP
echo ""
read -e -p "Please enter the full path to your server certificate (PEM which includes certs and key):  "  -i
""  CERT_PATH
echo ""
echo "--------------------------------------------------------------------------------"
echo ""
OS=$(cat /etc/redhat-release | cut -f 1 -d " ");
if [ $OS = 'CentOS' ]
then
        echo "Installing epel-release for CentOS"
        yum -y -q install epel-release
else
        echo "Installing epel-release for RHEL"
        rpm -ivh https://dl.fedoraproject.org/pub/epel/epel-release-latest-7.noarch.rpm
        yum -y -q update
fi
echo ""
echo "Installing nGinx"
yum -y -q install nginx
echo ""
echo "Reconfiguring SELinux for http connections"
echo ""
setsebool httpd_can_network_connect on -P
echo "Copying nGinx configuration templates"
\cp -f nginx.conf.template.https $NGINX_CONF
\cp -f stream.conf.template.https $STREAM_CONF
\cp -f https.conf.template $HTTPS_CONF
echo ""
echo "Applying configuration changes"
echo ""

sed -i "s/:SSL_PORT/:$SSL_PORT/g" $STREAM_CONF
sed -i "s/WEBAPP_IP:/$WEBAPP_IP:/g" $HTTPS_CONF
sed -i "s/MD_IP:/$MD_IP:/g" $STREAM_CONF
sed -i "s/FLEX_IP:/$FLEX_IP:/g" $STREAM_CONF
sed -i "s#CERT_PATH#$CERT_PATH#g" $HTTPS_CONF
echo ""
echo "Configuring firewall rule to allow port 443"
FWZONE=$(firewall-cmd --get-default-zone)
echo "Discovered firewall zone $FWZONE"
cat <<EOF | tee /etc/firewalld/services/nginx.xml
<?xml version="1.0" encoding="utf-8"?>
<service version="1.0">
    <short>nginx</short>
    <description>nginx</description>
    <port port="443" protocol="tcp"/>
</service>
EOF
sleep 5
firewall-cmd --zone=$FWZONE --add-service=nginx --permanent
firewall-cmd --reload
echo "nGInx Local Proxy Deployment Complete"
```

nginx.conf.template.https

```
# For more information on configuration, see:
#   * Official English Documentation: http://nginx.org/en/docs/
#   * Official Russian Documentation: http://nginx.org/ru/docs/

user nginx;
worker_processes auto;
error_log /var/log/nginx/error.log;
pid /run/nginx.pid;

# Load dynamic modules. See /usr/share/nginx/README.dynamic.
include /usr/share/nginx/modules/*.conf;

events {
    worker_connections 1024;
}

#  stream.conf contains the configuration for our TWCloud Proxy/Load balancer

include conf.d/stream.conf;

include conf.d/https.conf;
```

stream.conf.template.https

```
# stream core module provides TCP/UDP proxying and load balancing

stream {

        #
        # By default, SELinux only allows the http process access to the following ports (semanage ports -l
| grep http)
        #        http_port_t                    tcp      80, 81, 443, 488, 8008, 8009, 8443, 9000
        # In order for the proxy to connect to other ports, they need to either be added, or a simpler
solution is to allow
        #        http to connect to all ports, by issuing the following command
        #                sudo setsebool httpd_can_network_connect on -P
        #
        # application.conf in TWCloud needs to be configured to turn off the internal load balancing in
TWCloud on all nodes
        #     setting:  load_balancer = false (default is true) - make sure setting is uncommented
        #
        #
        #
        # Notation in template
        #
        # 127.0.0.1 is the private (local) IP addresses for this node
        # SSL_PORT is the user defined port to which TWCloud will bind for TLS communications between the MD
clients and TWC
        #        If you do not want to use SSL, then you can use the default of 3579
        #
        # MD_IP is the local IP address on this machine to which we will bind access to the MD->TWCloud data
stream (SSL_PORT)
        # FLEX_IP is the local IP address on this machine to which we will bind access to the cameo daemon
on the FLEX server (1101)
        #
        # Each of these IP's would be resolvable to the service FQDN (and common name of TWCloud Admin,
REST, and Authserver).
        #
        #
        # define our proxied server groups
        # hash directive is a mechanism for traffic from a given client to go to a given server
        # For a single node proxy it is not necessary, but is in the configuration for consistency when the
server group consists of a load balanced group
        # the upstream servers are the nodes running TWC


        upstream md_servers {
                hash $remote_addr consistent;
                server 127.0.0.1:SSL_PORT;
        }

        # define the proxy listeners

        server {
                listen MD_IP:443;
                proxy_buffer_size 16k;
                proxy_pass md_servers;
        }


        server {
                listen FLEX_IP:443;
                proxy_buffer_size 16k;
                proxy_pass 127.0.0.1:1101;
        }
}
```

https.conf.template

```
http {

    ssl_certificate                 CERT_PATH;
    ssl_certificate_key             CERT_PATH;
    ssl_session_timeout             1d;
    ssl_session_cache               shared:TWCSSL:10m;
    ssl_protocols                   TLSv1.2 TLSv1.3;
    ssl_ciphers                     ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-RSA-AES128-GCM-SHA256:ECDHE-ECDSA-
AES256-GCM-SHA384:ECDHE-RSA-AES256-GCM-SHA384:ECDHE-ECDSA-CHACHA20-POLY1305:ECDHE-RSA-CHACHA20-POLY1305:DHE-
RSA-AES128-GCM-SHA256:DHE-RSA-AES256-GCM-SHA384;
    ssl_prefer_server_ciphers       off;
    proxy_set_header                Host $host;


    upstream webapp {
        ip_hash;
        server 127.0.0.1:8443;
    }

    upstream authserver {
        ip_hash;
        server 127.0.0.1:8555;
    }

    upstream rest {
            ip_hash;
            server 127.0.0.1:8111;
    }


    server {
        listen WEBAPP_IP:443 ssl;
        location /webapp {
            proxy_pass https://webapp;
        }
        location /authentication {
            proxy_pass https://authserver;
        }
        location /osmc {
            proxy_pass https://rest;
        }
    }
}
```

HAProxy

The default package for HAProxy is 1.5, which is EOL. A package for HAProxy 1.8 is also available in an alternate repository.

The provided scripts contain versions to install both the default (1.5) and the alternate (1.8).

Layer 4 Proxy

As depicted in the TCP Proxy Context IBD, we have 5 frontends listening on port 443. Therefore, the instance must have 5 local IP addresses to which to bind port 443.

After deploying the Proxy, we need to make some changes to **authserver.properties** in order to allow the proxied TWCloud Admin console access.

authserver.properties

```
#------------------------------------------------
# Host name or IP of the server instance.
#------------------------------------------------
# server.public.host needs to be modified to reflect the authserver frontend IP address or FQDN
server.public.host=10.254.254.33
# server.public.port is set to the port number on which the authserver frontend listeens to
server.public.port=443

# authentication.redirect.uri.whitelist needs to have the Webapp's URL appened to the string.  If using port
443 as the public port for webapp, the port number is ommitted
authentication.redirect.uri.whitelist=https://10.254.254.31:8443/webapp/,https://10.254.254.31:8111/,
https://md_redirect,https://10.254.254.31/webapp/
```

install_proxy_haproxy15_tcp.sh

```
#!/bin/bash
#
# HAProxy Proxy/LB depoyment and configuration script
#
# Benjamin Krajmalnik (benjamin.krajmalnik@3ds.com)
#
HAPROXY_CONF=/etc/haproxy/haproxy.cfg


echo "This script will deploy and configure HAProxy as a local proxy where all services are bound to port
443"
echo ""
echo "--------------------------------------------------------------------------------"
echo ""
read -e -p "Please enter the port number for MD->TWCloud Communications (Settings/Secured Connection in
TWCloud Admin):  "  -i "10002"  SSL_PORT
echo ""
read -e -p "Please enter the IP Address for the Webapp Proxy:  "  -i ""  WEBAPP_IP
echo ""
read -e -p "Please enter the IP Address for the REST API Proxy:  "  -i ""  REST_IP
echo ""
read -e -p "Please enter the IP Address for the Authserver Proxy:  "  -i ""  AUTH_IP
echo ""
read -e -p "Please enter the IP Address for the MD->TWCloud Proxy:  "  -i ""  MD_IP
echo ""
read -e -p "Please enter the IP Address for the FlexNET Proxy:  "  -i ""  FLEX_IP
echo ""
echo "--------------------------------------------------------------------------------"
echo ""
OS=$(cat /etc/redhat-release | cut -f 1 -d " ");
if [ $OS = 'CentOS' ]
then
        echo "Installing epel-release for CentOS"
        yum -y -q install epel-release
else
        echo "Installing epel-release for RHEL"
        rpm -ivh https://dl.fedoraproject.org/pub/epel/epel-release-latest-7.noarch.rpm
        yum -y -q update
fi
echo ""
echo "Installing HAProxy"
yum -y -q install haproxy
echo ""
echo "Reconfiguring SELinux for http connections"
echo ""
setsebool -P haproxy_connect_any 1 -P
echo "Copying HAProxy configuration template"
\cp -f haproxy.cfg.template.tcp $HAPROXY_CONF
echo ""
echo "Applying configuration changes"
```

```
echo ""

sed -i "s/:SSL_PORT/:$SSL_PORT/g" $HAPROXY_CONF
sed -i "s/WEBAPP_IP:/$WEBAPP_IP:/g" $HAPROXY_CONF
sed -i "s/REST_IP:/$REST_IP:/g" $HAPROXY_CONF
sed -i "s/AUTH_IP:/$AUTH_IP:/g" $HAPROXY_CONF
sed -i "s/MD_IP:/$MD_IP:/g" $HAPROXY_CONF
sed -i "s/FLEX_IP:/$FLEX_IP:/g" $HAPROXY_CONF
echo ""
echo "Configuring firewall rule to allow port 443"
FWZONE=$(firewall-cmd --get-default-zone)
echo "Discovered firewall zone $FWZONE"
cat <<EOF | tee /etc/firewalld/services/haproxy.xml
<?xml version="1.0" encoding="utf-8"?>
<service version="1.0">
    <short>haproxy</short>
    <description>haproxy</description>
    <port port="443" protocol="tcp"/>
</service>
EOF
sleep 5
firewall-cmd --zone=$FWZONE --add-service=haproxy --permanent
firewall-cmd --reload
echo "HAProxy Local Proxy Deployment Complete"
```

install_proxy_haproxy18_tcp.sh

```
#!/bin/bash
#
# HAProxy Proxy/LB depoyment and configuration script
#
# Benjamin Krajmalnik (benjamin.krajmalnik@3ds.com)
#
HAPROXY_CONF=/etc/opt/rh/rh-haproxy18/haproxy/haproxy.cfg


echo "This script will deploy and configure HAProxy as a local proxy where all services are bound to port
443"
echo ""
echo "--------------------------------------------------------------------------------"
echo ""
read -e -p "Please enter the port number for MD->TWCloud Communications (Settings/Secured Connection in
TWCloud Admin):  "  -i "10002"  SSL_PORT
echo ""
read -e -p "Please enter the IP Address for the Webapp Proxy:  "  -i ""  WEBAPP_IP
echo ""
read -e -p "Please enter the IP Address for the REST API Proxy:  "  -i ""  REST_IP
echo ""
read -e -p "Please enter the IP Address for the Authserver Proxy:  "  -i ""  AUTH_IP
echo ""
read -e -p "Please enter the IP Address for the MD->TWCloud Proxy:  "  -i ""  MD_IP
echo ""
read -e -p "Please enter the IP Address for the FlexNET Proxy:  "  -i ""  FLEX_IP
echo ""
echo "--------------------------------------------------------------------------------"
echo "Installing Software Collections (SCL)"
yum install centos-release-scl
echo ""
echo "Installing HAProxy 1.8"
yum -y -q install rh-haproxy18-haproxy rh-haproxy18-haproxy-syspaths
echo ""
echo "Reconfiguring SELinux for http connections"
echo ""
setsebool -P haproxy_connect_any 1 -P
echo "Copying HAProxy configuration template"
\cp -f haproxy.cfg.template.tcp $HAPROXY_CONF
echo ""
echo "Applying configuration changes"
echo ""

sed -i "s/:SSL_PORT/:$SSL_PORT/g" $HAPROXY_CONF
sed -i "s/WEBAPP_IP:/$WEBAPP_IP:/g" $HAPROXY_CONF
sed -i "s/REST_IP:/$REST_IP:/g" $HAPROXY_CONF
sed -i "s/AUTH_IP:/$AUTH_IP:/g" $HAPROXY_CONF
sed -i "s/MD_IP:/$MD_IP:/g" $HAPROXY_CONF
sed -i "s/FLEX_IP:/$FLEX_IP:/g" $HAPROXY_CONF
echo ""
echo "Configuring firewall rule to allow port 443"
FWZONE=$(firewall-cmd --get-default-zone)
echo "Discovered firewall zone $FWZONE"
cat <<EOF | tee /etc/firewalld/services/haproxy.xml
<?xml version="1.0" encoding="utf-8"?>
<service version="1.0">
    <short>haproxy</short>
    <description>haproxy</description>
    <port port="443" protocol="tcp"/>
</service>
EOF
sleep 5
firewall-cmd --zone=$FWZONE --add-service=haproxy --permanent
firewall-cmd --reload
echo "HAProxy Local Proxy Deployment Complete"
```

haproxy.cfg.template.tcp

```
#---------------------------------------------------------------------
# HAProxy Local Proxy for TWCloud
# Benjamin Krajmalnik (benjamin.krajmalnik@3ds.com)
#---------------------------------------------------------------------

#---------------------------------------------------------------------
# Global settings
#---------------------------------------------------------------------
global

    log 127.0.0.1 local2

    chroot      /var/lib/haproxy
    pidfile     /var/run/haproxy.pid
    maxconn     4000
    user        haproxy
    group       haproxy
    daemon

    # turn on stats unix socket
    stats socket /var/lib/haproxy/stats

#---------------------------------------------------------------------
# common defaults that all the 'listen' and 'backend' sections will
# use if not designated in their block
#---------------------------------------------------------------------
defaults
    mode                    tcp
    balance                 source
    log                     global
    option                  tcplog
    option                  dontlognull
    option http-server-close
    option                  redispatch
    retries                 3
    timeout http-request    10s
    timeout queue           1m
    timeout connect         10s
    timeout client          1m
    timeout server          1m
    timeout http-keep-alive 10s
    timeout check           10s
    maxconn                 3000

#---------------------------------------------------------------------
# frontends  which proxy to the backends
#---------------------------------------------------------------------


frontend            webapp
    bind            WEBAPP_IP:443
    mode            tcp
    default_backend webapp


frontend            authserver
    bind            AUTH_IP:443
    mode            tcp
    default_backend authserver


frontend            rest
    bind            REST_IP:443
    mode            tcp
    default_backend rest


frontend            md
```

```
    bind            MD_IP:443
    mode            tcp
    default_backend md


frontend            flex
    bind            FLEX_IP:443
    mode            tcp
    default_backend flex


#---------------------------------------------------------------------
#  backends for proxying services
#---------------------------------------------------------------------
backend webapp
    mode tcp
    server      static 127.0.0.1:8443

backend authserver
    mode tcp
    server      static 127.0.0.1:8555

backend rest
    mode tcp
    server      static 127.0.0.1:8111

backend md
    mode tcp
    server      static 127.0.0.1:SSL_PORT

backend flex
    mode tcp
    server      static 127.0.0.1:1101
```

Layer 7 Proxy

As depicted in the HTTP Proxy Context IBD, we have 3 frontends listening on port 443. Therefore, the instance must have 3 local IP addresses to which to bind port 443.

After deploying the Proxy, we need to make some changes to **authserver.properties** in order to allow the proxied TWCloud Admin console access. Unlike the TCP Proxy where each stream was bound to its own IP address, all 3 HTTPS streams are handled by a single frontend. Therefore, the **serverver.public.host**, in this case, is the primary IP address.

⚠ In a Layer 7 HTTPS proxy, SSL termination takes place at the proxy. For this to take place, you will need a PEM encoded file containing the private key and certificates for the server, including the full certificate chain.

authserver.properties

```
#-------------------------------------------------
# Host name or IP of the server instance.
#-------------------------------------------------
# server.public.host needs to be modified to reflect the authserver frontend IP address or FQDN
server.public.host=10.254.254.31
# server.public.port is set to the port number on which the authserver frontend listeens to
server.public.port=443

# authentication.redirect.uri.whitelist needs to have the Webapp's URL appened to the string.  If using port
443 as the public port for webapp, the port number is ommitted
authentication.redirect.uri.whitelist=https://10.254.254.31:8443/webapp/,https://10.254.254.31:8111/,
https://md_redirect,https://10.254.254.31/webapp/
```

install_proxy_haproxy15_https.sh

```
#!/bin/bash
#
# HAProxy Proxy/LB depoyment and configuration script - SSL Termination
#
# Benjamin Krajmalnik (benjamin.krajmalnik@3ds.com)
#
HAPROXY_CONF=/etc/haproxy/haproxy.cfg


echo "This script will deploy and configure HAProxy as a local proxy with TLS termination where all services
are bound to port 443"
echo ""
echo "--------------------------------------------------------------------------------"
echo ""
read -e -p "Please enter the port number for MD->TWCloud Communications (Settings/Secured Connection in
TWCloud Admin):  "  -i "10002"  SSL_PORT
echo ""
read -e -p "Please enter the IP Address for the https proxy (Webapp/Authserver/REST):  "  -i ""  WEBAPP_IP
echo ""
read -e -p "Please enter the IP Address for the MD->TWCloud Proxy:  "  -i ""  MD_IP
echo ""
read -e -p "Please enter the IP Address for the FlexNET Proxy:  "  -i ""  FLEX_IP
echo ""
read -e -p "Please enter the full path to your server certificate (PEM which includes certs and key):  "  -i
""  CERT_PATH
echo ""

echo "--------------------------------------------------------------------------------"
echo ""
OS=$(cat /etc/redhat-release | cut -f 1 -d " ");
if [ $OS = 'CentOS' ]
then
        echo "Installing epel-release for CentOS"
        yum -y -q install epel-release
else
        echo "Installing epel-release for RHEL"
        rpm -ivh https://dl.fedoraproject.org/pub/epel/epel-release-latest-7.noarch.rpm
        yum -y -q update
fi
echo ""
echo "Installing HAProxy"
yum -y -q install haproxy
echo ""
echo "Reconfiguring SELinux for http connections"
echo ""
setsebool -P haproxy_connect_any 1 -P
echo "Copying HAProxy configuration template"
\cp -f haproxy.cfg.template.https $HAPROXY_CONF
echo ""
echo "Applying configuration changes"
echo ""

sed -i "s/:SSL_PORT/:$SSL_PORT/g" $HAPROXY_CONF
sed -i "s/WEBAPP_IP:/$WEBAPP_IP:/g" $HAPROXY_CONF
sed -i "s/MD_IP:/$MD_IP:/g" $HAPROXY_CONF
sed -i "s/FLEX_IP:/$FLEX_IP:/g" $HAPROXY_CONF
sed -i "s#CERT_PATH#$CERT_PATH#g" $HAPROXY_CONF


echo ""
echo "Configuring firewall rule to allow port 443"
FWZONE=$(firewall-cmd --get-default-zone)
echo "Discovered firewall zone $FWZONE"
cat <<EOF | tee /etc/firewalld/services/haproxy.xml
<?xml version="1.0" encoding="utf-8"?>
<service version="1.0">
    <short>haproxy</short>
    <description>haproxy</description>
    <port port="443" protocol="tcp"/>
</service>
EOF
```

```
sleep 5
firewall-cmd --zone=$FWZONE --add-service=haproxy --permanent
firewall-cmd --reload
echo "HAProxy Local Proxy Deployment Complete"
```

install_proxy_haproxy18_https.sh

```
#!/bin/bash
#
# HAProxy 1.8 Proxy/LB depoyment and configuration script - SSL Termination
#
# Benjamin Krajmalnik (benjamin.krajmalnik@3ds.com)
#
HAPROXY_CONF=/etc/opt/rh/rh-haproxy18/haproxy/haproxy.cfg


echo "This script will deploy and configure HAProxy as a local proxy with TLS termination where all services
are bound to port 443"
echo ""
echo "--------------------------------------------------------------------------------"
echo ""
read -e -p "Please enter the port number for MD->TWCloud Communications (Settings/Secured Connection in
TWCloud Admin):  "  -i "10002"  SSL_PORT
echo ""
read -e -p "Please enter the IP Address for the https proxy (Webapp/Authserver/REST):  "  -i ""  WEBAPP_IP
echo ""
read -e -p "Please enter the IP Address for the MD->TWCloud Proxy:  "  -i ""  MD_IP
echo ""
read -e -p "Please enter the IP Address for the FlexNET Proxy:  "  -i ""  FLEX_IP
echo ""
read -e -p "Please enter the full path to your server certificate (PEM which includes certs and key):  "  -i
""  CERT_PATH
echo ""

echo "--------------------------------------------------------------------------------"
echo ""
echo "Installing Software Collections (SCL)"
yum install centos-release-scl
echo ""
echo "Installing HAProxy 1.8"
yum -y -q install rh-haproxy18-haproxy rh-haproxy18-haproxy-syspaths
echo ""
echo "Reconfiguring SELinux for http connections"
echo ""
setsebool -P haproxy_connect_any 1 -P
echo "Copying HAProxy configuration template"
echo "Reconfiguring SELinux for tcp connections"
echo ""
setsebool -P haproxy_connect_any 1 -P
echo "Copying HAProxy configuration template"
\cp -f haproxy.cfg.template.https $HAPROXY_CONF
echo ""
echo "Applying configuration changes"
echo ""

sed -i "s/:SSL_PORT/:$SSL_PORT/g" $HAPROXY_CONF
sed -i "s/WEBAPP_IP:/$WEBAPP_IP:/g" $HAPROXY_CONF
sed -i "s/MD_IP:/$MD_IP:/g" $HAPROXY_CONF
sed -i "s/FLEX_IP:/$FLEX_IP:/g" $HAPROXY_CONF
sed -i "s#CERT_PATH#$CERT_PATH#g" $HAPROXY_CONF


echo ""
echo "Configuring firewall rule to allow port 443"
FWZONE=$(firewall-cmd --get-default-zone)
echo "Discovered firewall zone $FWZONE"
cat <<EOF | tee /etc/firewalld/services/haproxy.xml
<?xml version="1.0" encoding="utf-8"?>
<service version="1.0">
```

```
    <short>haproxy</short>
    <description>haproxy</description>
    <port port="443" protocol="tcp"/>
</service>
EOF
sleep 5
firewall-cmd --zone=$FWZONE --add-service=haproxy --permanent
firewall-cmd --reload
echo "HAProxy Local Proxy Deployment Complete"
```

haproxy.cfg.template.https

```
#---------------------------------------------------------------------
# HAProxy Local hybrid Proxy for TWCloud (https/tcp)
# Benjamin Krajmalnik (benjamin.krajmalnik@3ds.com)
#---------------------------------------------------------------------

#---------------------------------------------------------------------
# Global settings
#---------------------------------------------------------------------
global

    log 127.0.0.1 local2

    chroot      /var/lib/haproxy
    pidfile     /var/run/haproxy.pid
    maxconn     4000
    user        haproxy
    group       haproxy
    daemon

    # turn on stats unix socket
    stats socket /var/lib/haproxy/stats

    #enforce TLS >= 1.2 and strong ciphers
    ssl-default-bind-options ssl-min-ver TLSv1.2
    ssl-default-bind-ciphers ECDHE-ECDSA-AES256-GCM-SHA384:ECDHE-RSA-AES256-GCM-SHA384:ECDHE-ECDSA-CHACHA20-
POLY1305:ECDHE-RSA-CHACHA20-POLY1305:ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-RSA-AES128-GCM-SHA256:ECDHE-ECDSA-
AES256-SHA384:ECDHE-RSA-AES256-SHA384:ECDHE-ECDSA-AES128-SHA256:ECDHE-RSA-AES128-SHA256
    ssl-server-verify none
#---------------------------------------------------------------------
# common defaults that all the 'listen' and 'backend' sections will
# use if not designated in their block
#---------------------------------------------------------------------
defaults
    mode                    http
    balance                 source
    log                     global
    option                  httplog
    option                  dontlognull
    option http-server-close
    option                  redispatch
    retries                 3
    timeout http-request    10s
    timeout queue           1m
    timeout connect         10s
    timeout client          1m
    timeout server          1m
    timeout http-keep-alive 10s
    timeout check           10s
    maxconn                 3000

#---------------------------------------------------------------------
# frontends  which proxy to the backends
#---------------------------------------------------------------------


frontend            https_services
    mode            http
```

```
    bind            WEBAPP_IP:443 ssl crt CERT_PATH
    acl             webapp-acl            path_beg /webapp
    use_backend     webapp if webapp-acl

    acl             authserver-acl        path_beg /authentication
    use_backend     authserver if authserver-acl

    acl             rest-acl              path_beg  /osmc
    use_backend     rest if rest-acl

frontend            md
    bind            MD_IP:443
    mode            tcp
    option          tcplog
    default_backend md


frontend            flex
    bind            FLEX_IP:443
    mode            tcp
    option          tcplog
    default_backend flex




#---------------------------------------------------------------------
#  backends for proxying services
#---------------------------------------------------------------------
backend webapp
    server      static 127.0.0.1:8443 ssl

backend authserver
     server      static 127.0.0.1:8555 ssl

backend rest
    server      static 127.0.0.1:8111 ssl

backend md
    mode tcp
    server      static 127.0.0.1:3579

backend flex
    mode tcp
    server      static 127.0.0.1:1101
```