



Tech Note:

# TechNote - Deploying CPPM with F5<sup>®</sup> BIG-IP<sup>®</sup> Local Traffic Manager<sup>™</sup> (LTM)

<b><u>Version</u></b>	<b><u>Date</u></b>	<b><u>Modified By</u></b>	<b><u>Comments</u></b>
0.1	July 2014	Danny Jump	Early Draft Version
0.2 / 0.3	07/11/2014	Con Stathis	Added sections on Virtual Servers and iRules®
0.5 / 0.6	07/24/2014 08/12/2014	Danny Jump	Draft Published Version for feedback. Consolidated comments and resent for final editing
0.7	08/26/2014	Con Stathis	Corrected Figures, reviewed and edited text across the document, added clarification where needed.
1.0	09/22/2014	Danny Jump	Final edited and Published Version

## Table of Contents

Overview .....	7
Why we are doing this?.....	7
Background / Introduction .....	9
Basic Load Balancing Terminology.....	10
Node, Host, Member and Server .....	10
Pool, Cluster and Farm .....	11
The Load Balancing Decision.....	13
To Load Balance or Not to Load Balance?.....	14
Everything NAT .....	16
What is SNAT in F5 BIG-IP LTM? SNAT vs. Inline. What is a NAT? .....	16
What is SNAT?.....	16
Global traffic and SNAT.....	16
Virtual Servers and SNAT.....	17
What is SNAT automap, a simple explanation .....	17
Alternative to SNAT, Inline.....	17
How do I capture my source address with SNAT? .....	17
What is a NAT?.....	18
RADIUS Packet Attributes.....	18
Additional notes to consider on SNAT v Inline.....	18
Technology Designs.....	20
F5 BIG-IP LTM Component Configuration.....	28
Building Blocks .....	28
Adding NODES .....	29
Adding MONITORS.....	30

Configure Radius Health-Check Monitor on CPPM side .....	33
Adding POOLS .....	38
Adding Virtual Servers (VS).....	41
ANY-network IP Forwarder (VS).....	41
CPPM-Subnet IP Forwarder (VS).....	43
COA-RFC3576_forwarder IP Forwarder (VS) .....	44
Standard Virtual Servers .....	46
RADIUS Virtual Server.....	46
HTTP Virtual Server .....	53
HTTPS Virtual Server.....	68
Appendix A - Adding SSL Certificates to F5 BIG-IP LTM .....	76
Appendix B - Overview of AOS 6.4 SLB Options.....	80

## Table of Figures

Figure 1 - Network-based load balancing appliances .....	9
Figure 2 – SLB comprises four concepts—virtual servers, clusters, services, and hosts.....	11
Figure 3 – Creating VIP groups on a CPPM cluster .....	20
Figure 4 - High level Setup for CPPM + F5 BIG-IP LTM network.....	21
Figure 5 – CPPM Detailed Setup.....	22
Figure 6 – Data Flow through F5 BIG-IP LTM and CPPM Network .....	23
Figure 7 - Overview of the traffic types that are Load-Balanced and those that aren't.....	24
Figure 8 – RADIUS Virtual Server Overview & Data-Flow.....	25
Figure 9 - Adding CPPM Nodes to F5 BIG-IP LTM.....	29
Figure 10 - Setting 'Node' default monitor to ICMP .....	29
Figure 11 - Nodes showing as available... .....	30

Figure 12 - Node availability detail – Time/Date last checked .....	30
Figure 13 - F5 BIG-IP LTM Health Check Active-Directory user check.....	31
Figure 14 – F5 BIG-IP LTM health checking against a local CPPM user.....	32
Figure 15 - Local CPPM User - "f5-hlthchk" .....	33
Figure 16 - CPPM Service Health Check for F5 RADIUS .....	34
Figure 17 - CPPM Access Tracker showing local and AD users.....	34
Figure 18 - RADIUS Request/Accept messages.....	35
Figure 19 - Detailed Access Tracker message for health check .....	35
Figure 20 – Adding the F5 BIG-IP LTM HTTP monitor .....	36
Figure 21 - Adding the F5 BIG-IP LTM HTTPS monitor.....	37
Figure 22 - Adding F5 BIG-IP LTM RADIUS pool .....	38
Figure 23 - Adding F5 BIG-IP LTM WEBAUTH (port 80) pool.....	39
Figure 24 - Adding F5 BIG-IP LTM WEBAUTH (port 443) pool.....	40
Figure 25 - Adding 'ANY-network' IP Forwarder VS .....	42
Figure 26 – Adding 'CPPM-network' IP Forwarder VS .....	43
Figure 27 – Defining the CoA Source address on the F5 BIG-IP LTM .....	44
Figure 28 – F5 iRule – Tracking, forwarding and persisting traffic for ports 1812/1813 ....	47
Figure 29 - Pasting F5 iRule into F5 BIG-IP LTM.....	49
Figure 30 - Adding new a UDP profile, copied from a parent 'UDP' profile.....	51
Figure 31 - Adding the RADIUS VS.....	52
Figure 32 – additional configuration for the RADIUS VS .....	52
Figure 33 – additional configuration for the RADIUS VS .....	53
Figure 34 - Adding a TCP WAN profile - part1 .....	55
Figure 35 - Adding a TCP WAN profile – part2.....	56
Figure 36 - Adding a TCP LAN profile – part1.....	58

Figure 37 - Adding a TCP LAN profile – part2.....	59
Figure 38 - Adding a HTTP profile .....	61
Figure 39 - Adding a F5 'OneConnect' profile .....	63
Figure 40 – Adding HTTP F5 iApps Application Service – part1 .....	64
Figure 41 - Adding HTTP Application Service – part2 .....	65
Figure 42 - Adding HTTP Application Service – part3 .....	65
Figure 43 - Adding HTTP Application Service – part4 .....	65
Figure 44 - Adding HTTP Application Service – part5 .....	66
Figure 45 - Adding HTTP Application Service – part6 .....	66
Figure 46 - Adding HTTP Application Service – part7 .....	67
Figure 47 – Viewing the created F5 iApps HTTP Application Service .....	67
Figure 48 – Adding a mobile TCP specific profile – part1.....	69
Figure 49 - Adding a mobile TCP specific profile – part2 .....	70
Figure 50 - Adding a mobile TCP specific profile – part3 .....	70
Figure 51 - Adding HTTPS F5 iApps Application Service .....	71
Figure 52 - Adding HTTPS Application Service – part1.....	72
Figure 53 - Adding HTTPS Application Service – part2.....	72
Figure 54 - Adding HTTPS Application Service – part3.....	73
Figure 55 - Adding HTTPS Application Service – part4.....	73
Figure 56 - Adding HTTPS Application Service – part5.....	73
Figure 57 - Adding HTTPS Application Service – part6.....	74
Figure 58 - Adding HTTPS Application Service – part7.....	74
Figure 59 - Adding HTTPS Application Service – part8.....	75
Figure 60 – Viewing the HTTPS Application Service .....	75
Figure 61 - Exporting HTTPS Server certificate from CPPM .....	76

Figure 62 - Importing the Certificate & Private Key in to F5 BIG-IP LTM .....	77
Figure 63 - Import the Certificate then the Key .....	77
Figure 64 - Example of importing the Certificate using plain text.....	78
Figure 65 - Certificate only imported .....	78
Figure 66 - Example of importing the Private Key using plain text .....	78
Figure 67 - Certificate and Key now imported .....	79
Figure 68 - Imported certificate attributes .....	79
Figure 69 - Adding multiple Auth servers and enabling Load Balancing .....	80
Figure 70 - AOS RADIUS Server timeout.....	81

## Open Source Code

Certain Aruba products include Open Source software code developed by third parties, including software code subject to the GNU General Public License (GPL), GNU Lesser General Public License (LGPL), or other Open Source Licenses. The Open Source code used can be found at this site: [http://www.arubanetworks.com/open\\_source](http://www.arubanetworks.com/open_source)

## Copyright

© 2014 Aruba Networks, Inc. Aruba Networks' trademarks include Aruba Networks®, Aruba The Mobile Edge Company® (stylized), Aruba Mobility-Defined Networks™, Aruba Mobility Management System®, People Move Networks Must Follow®, Mobile Edge Architecture®, RFProtect®, Green Island®, ETips®, ClientMatch™, Virtual Intranet Access™, ClearPass Access Management Systems™, Aruba Instant™, ArubaOSTM, xSec™, ServiceEdge™, Aruba ClearPass Access Management System™, Airmesh™, AirWave™, Aruba Central™, and "ARUBA@WORK™. F5, BIG-IP, iApps, iRules, and OneConnect are trademarks or service marks of F5 Networks, Inc., in the U.S. and other countries. All rights reserved. All other trademarks are the property of their respective owners.

## Overview

The following guide has been produced to help educate our customers and partners in deploying CPPM in conjunction with F5® BIG-IP® LTM™ application delivery controllers (ADC). F5 BIG-IP LTM enhanced level of availability and provide a more scalable solution. This guide was written to accompany the CPPM 6.4.x release, but there is no specific feature developed by Aruba in this code release specifically utilized in this guide. Going forward this guide will be updated and republished to reflect new and improved functionality and designs we deliver/develop.



**Note:** Where you see a red-chili  this signifies a ‘hot’ important point and highlights that this point is to be taken as a best-practice recommendation.

## Why we are doing this?

Why are we using F5 BIG-IP LTM to load balance clients to the CPPM Cluster? How does it benefit us and what is the limitation we are trying to overcome?

When a cluster of multiple CPPM nodes is deployed we need to be able to route traffic to these nodes with some level of control to ensure that a single node is not overwhelmed with requests and that the licenses across the cluster are used appropriately. AOS itself recently has included some limited functionality to achieve some of this with the addition in AOS 6.4, RADIUS load-balancing (discussed in Appendix B). But for enterprises where the basic level of server-group load-balancing or the AOS RADIUS load-balancing is not enough then what we discuss within this document over the next few pages should be very relevant to the reader. Another reason is so that the nodes within a cluster can be represented by a single IP address simplifying the NAS deployment across a large enterprise spanning large geographic areas where the NAS is not necessarily an Aruba device.

F5 BIG-IP LTM can be deployed in a multitude of ways, typically one-armed utilizing Secure (or source) NAT'ing (SNAT) and inline. We have chosen to use the inline method. Having the servers (CPPM) inline means they will need the F5 BIG-IP LTM to be their Gateway address. The one major disadvantage with SNAT is the obscurity of the clients source address. With an inline approach, the client's source address is preserved. This become very important for example when you want to send a CoA to an endpoint. Without the original source IP address of the RADIUS packet we are unable to send the CoA to the correct NAS supporting this endpoint. Just as important, if we don't know the exact type of NAS endpoint to send the CoA to, we'd be unable to send NAS specific vendor CoA messages.

So now we have chosen to utilize F5 BIG-IP LTM to traffic engineer our data path to the CPPM nodes. There are a couple of additional points you need to be aware of and the reasons why we have configured the F5 BIG-IP LTM in this particular fashion. Some of the configuration below is necessary to ensure that we track the user's RADIUS request to a specific CPPM node such that when we receive the radius account start and subsequent accounting messages we are able to send these to the same CPPM node to ensure that the same node has persistence of the session data for that endpoint. We accomplish this as described below in more details by the use of a radius attribute called the calling-station-id.



## Background / Introduction

As customer deployments become more complex we must ensure that we optimize and expand the solutions and design where availability and scale are critical.

This guide has been written and developed with the use of F5 BIG-IP LTM running s/w revision 11.5.

To provide an overview, I have borrowed some content from an F5 Load-Balancing 101 Nuts & Bolts doc, so we give acknowledgment to F5 for the Intro overview below.

Load balancing got its start in the form of network-based load balancing hardware. It is the essential foundation on which Application Delivery Controllers (ADCs) operate. The second iteration of purpose-built load balancing (following application-based proprietary systems) materialized in the form of network-based appliances. In essence, these devices would present a “virtual server” address to the outside world, and when users attempted to connect, they would forward the connection to the most appropriate real server doing bi-directional network address translation (NAT).

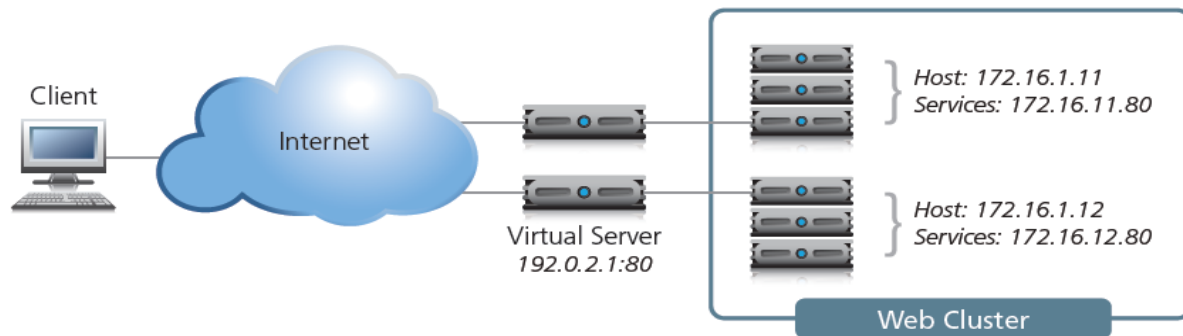


Figure 1 - Network-based load balancing appliances

## Basic Load Balancing Terminology

---

It would certainly help if everyone used the same terminology; unfortunately, every vendor of load balancing devices (and, in turn, ADCs) seems to use different terminology.

### Node, Host, Member and Server



Most ADC vendors have the concept of a **node**, **host**, **member**, or **server**; some have all four, but they typically mean different things. There are two basic concepts that they all try to express.

One concept—usually called a **node** or **server**—is the idea of the physical server itself (in our topology the CPPM appliance or CPPM VM) that will receive traffic from the ADC. This is synonymous with the IP address of the physical server and, in the absence of a ADC, would be the IP address that the server name (for example, `www.cppm-testing.com`) would resolve to. For the remainder of this paper, we will refer to this concept as the **host**.

The second concept is a **member** (sometimes, unfortunately, also called a node by some manufacturers). A **member** is usually a little more defined than a server/node in that it includes the port of the actual application that will be receiving traffic. For instance, a server named `www.cppm-testing.com` may resolve to an address of `172.16.1.10`, which represents the server/node, and may have an application (a web server) running on port `80`, making the **member** address `172.16.1.10:80`. Simply put, the **member** includes the definition of the application port as well as the IP address of the physical server. For the remainder of this paper, we will refer to the application port as the **service**.

Why all the complication? Because the distinction between a physical server and the application services running on it allows the ADC to individually interact with the applications rather than the underlying hardware. A host (`172.16.1.10`) may have more than one service available (HTTP, FTP, DNS, and so on). By defining each application uniquely (`172.16.1.10:80`, `172.16.1.10:21`, and `172.16.1.10:53`), the ADC can apply unique ADC and health monitoring (discussed later) based on the **services** instead of the **host**. However, there are still times when being able to interact with the host (like low-level health monitoring or when taking a server offline for maintenance) is extremely convenient.

Remember, most ADC uses some concept to represent the host, or physical server, and another to represent the services available on it— in this case, simply **host** and **services**.

## Pool, Cluster and Farm

ADC's allow organizations to distribute inbound traffic across multiple back-end destinations. It is therefore a necessity to have the concept of a collection of back-end destinations. Clusters, as we will refer to them herein, although also known as pools or farms, are collections of similar services available on any number of hosts. For instance, all services that offer the company web page would be collected into a cluster called "company web page" and all services that offer e-commerce services would be collected into a cluster called "e-commerce." The key element here is that all systems have a collective object that refers to "all similar services" and makes it easier to work with them as a single unit. This collective object—a cluster—is almost always made up of **services**, not hosts.

## Virtual Server

Although not always the case, today there is little dissent about the term virtual server, or virtual. It is important to note that like the definition of services, *virtual server* usually includes the application port as well as the IP address. The term "virtual service" would be more in keeping with the IP:Port convention; but because most vendors use virtual server, this paper will continue using the term "Virtual Server" as well.

## Putting it all together

Putting all of these concepts together makes up the basic steps in load balancing. The ADC presents virtual servers to the outside world. Each virtual server points to a cluster of services that reside on one or more physical hosts.

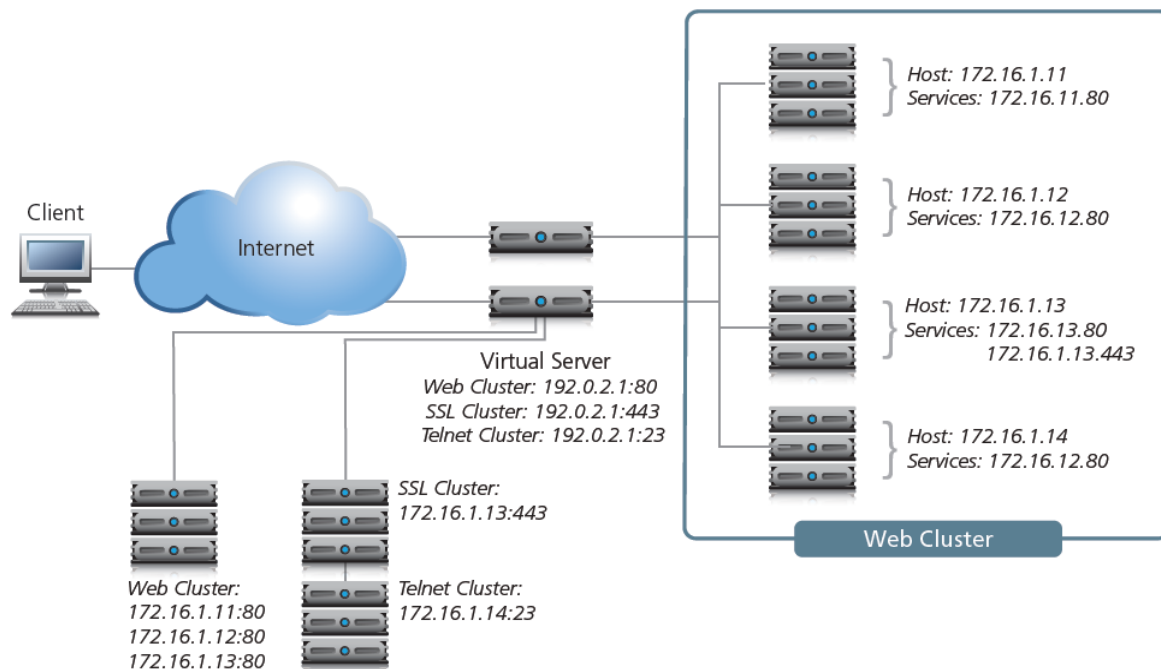


Figure 2 – SLB comprises four concepts—virtual servers, clusters, services, and hosts.

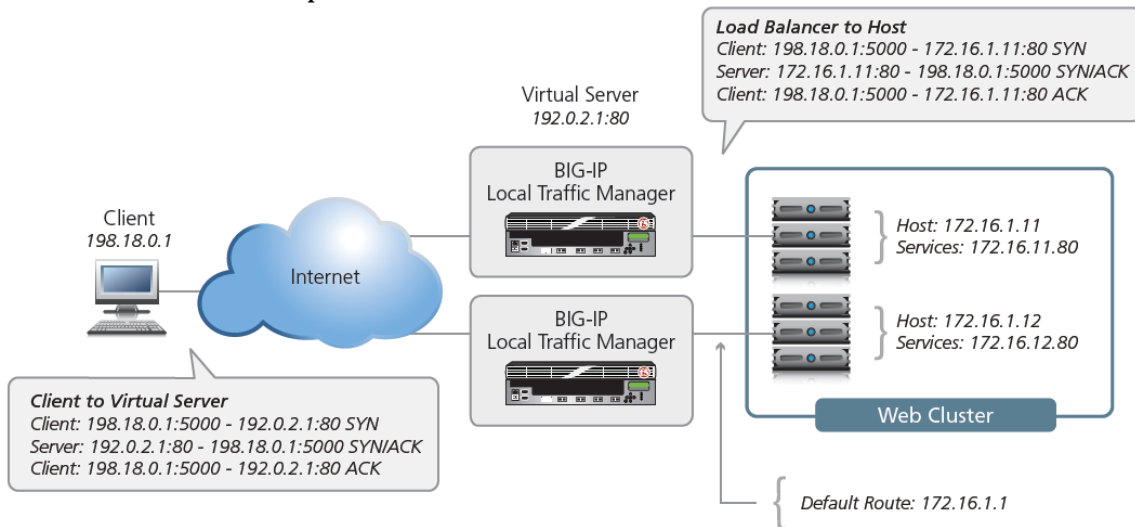
## Load Balancing Basics



With this common vocabulary established, let's examine the basic load balancing transaction. As depicted, the ADC will **typically sit in-line between the client and the hosts that provide the services the client wants to use**. As with most things in load balancing, this is not a rule, but more of a best practice in a typical deployment. Let's also assume that the ADC is already configured with a virtual server that points to a cluster consisting of two service points. In this deployment scenario, the hosts have a return route that points back to the ADC so that return traffic will be processed through it on its way back to the client.

The basic load balancing transaction is as follows:

1. The client attempts to connect with the service on the ADC.
2. The ADC accepts the connection, and after deciding which host should receive the connection, changes the destination IP (and possibly port) to match the service of the selected host (note that the source IP of the client is not touched).
3. The host accepts the connection and responds back to the original source, the client, via its default route, the ADC.
4. The ADC intercepts the return packet from the host and now changes the source IP (and possible port) to match the virtual server IP and port, and forwards the packet back to the client.
5. The client receives the return packet, believing that it came from the virtual server, and continues the process.



This very simple example is relatively straightforward, but there are a couple of key elements to take note of. Step one, as far as the client knows, it sends packets to the virtual server and the virtual server responds—simple. Step two the NAT takes place. This is where the ADC replaces the destination IP (i.e. virtual server address) sent by the client with the destination IP of the host to which it has chosen to load balance the request. Step three is the second half of this process (the part that makes the NAT “bi-directional”). The source IP of the return packet from the host will be the IP of the host; if this address were

not changed and the packet was simply forwarded to the client, the client would be receiving a packet from someone it didn't request one from, and would simply drop it. Instead, the ADC, remembering the connection, rewrites the packet so that the source IP of the return packet is that of the virtual server, thus solving this problem.

## The Load Balancing Decision

Usually at this point, two questions arise: how does the ADC decide which host to send the connection to? And what happens if the selected host isn't working?

Let's discuss the second question first. What happens if the selected host isn't working? The simple answer is that it doesn't respond to the client request and the connection attempt eventually times out and fails. This is obviously not a preferred circumstance, as it doesn't ensure high availability. That's why most ADC technology includes some level of health monitoring that determines whether a host is actually available and able to take a connection *before* attempting to send packets to it.

There are multiple levels of health monitoring, each with increasing granularity and focus. A basic monitor would simply PING the host itself. If the host does not respond to PING, it is a good assumption that any services defined on the host are probably down and the host should be removed from the cluster of available services. Unfortunately, even if the host responds to PING, it doesn't necessarily mean the service itself is working. Therefore most devices can do "service PINGs" of some kind, ranging from simple TCP connections all the way to interacting with the application via a scripted or intelligent interaction. These higher-level health monitors not only provide greater confidence in the availability of the actual services (as opposed to the host), but they also allow the ADC to differentiate between multiple services on a single host. The ADC understands that while one service might be unavailable, other services on the same host might be working just fine and should still be considered as valid destinations for user traffic.

This brings us back to the first question: How does the ADC decide which host to send a connection request to? Each virtual server has a specific dedicated cluster of services (listing the hosts that offer that service) that makes up the list of possibilities. Additionally, the health monitoring modifies that list to make a list of "currently available" hosts that provide the indicated service. It is this modified list from which the ADC chooses the host that will receive a new connection. Deciding the *exact* host depends on the load-balancing algorithm associated with that particular cluster. The most common is simple round-robin where the ADC simply goes down the list starting at the top and allocates each new connection to the next host; when it reaches the bottom of the list, it simply starts again at the top. While this is simple and very predictable, it assumes that all connections will have a similar load and duration on the back-end host, which is not always true. More advanced algorithms use things like current-connection counts, host utilization, and even real-world response times for existing traffic to the host in order to pick the most appropriate host from the available cluster services.

Sufficiently advanced ADC will also be able to synthesize health-monitoring information with load balancing algorithms to include an understanding of service dependency. This is

the case when a single host has multiple services, all of which are necessary to complete the user's request. A good analogy for CPPM would be where a CPPM host will provide both standard HTTP/s services (port 80/443 – Guest Portal) as well as RADIUS (port 1812/1813 - Authentication). In many of these circumstances, it does not matter if a user connects to a host that has only one service operational, but not the other as long as these services are similar. In other words, it is ok to send RADIUS 1812/1813 requests to a host if the HTTP/s services on that host has failed. Similarly it is ok to send HTTP/s requests to host if the RADIUS service on that host has failed.

## To Load Balance or Not to Load Balance?

Load balancing in regards to picking an available service when a client initiates a transaction request is only half of the solution. Once the connection is established, the ADC must keep track of whether the following traffic from that user should be load balanced. There are generally two specific issues with handling follow-on traffic once it has been load balanced: connection maintenance and persistence.

### *Connection maintenance*

If the user is trying to utilize a long-lived TCP connection (telnet, FTP, and more) that doesn't immediately close, the ADC must ensure that multiple data packets carried across that connection do not get load balanced to other available service hosts. This is connection maintenance and requires two key capabilities: 1) the ability to keep track of open connections and the host service they belong to; and 2) the ability to continue to monitor that connection so the connection table can be updated when the connection closes. This is rather standard fare for most ADCs.

### *Persistence*

Increasingly more common, however, is when the client uses multiple short-lived TCP connections (for example, HTTP) to accomplish a single task. In some cases, like standard web browsing, it doesn't matter and each new request can go to any of the back-end service hosts; however, there are many more instances (XML, e-commerce "shopping cart," HTTPS, and so on) where it is extremely important that multiple connections from the same user go to the same back-end service host and *not* be load balanced. This concept is called persistence, or server affinity. There are multiple ways to address this depending on the protocol and the desired results. For example, in modern HTTP transactions, the server can specify a "keep-alive" connection, which turns those multiple short-lived connections into a single long-lived connection that can be handled just like the other long-lived connections. However, this provides little relief. Even worse, as the use of web services increases, keeping all of these connections open longer than necessary would strain the resources of the entire system. In these cases, most ADCs provide other mechanisms for creating artificial server affinity.

One of the most basic forms of persistence is source-address affinity. This involves simply recording the source IP address of incoming requests and the service host they were load balanced to, and making all future transaction go to the same host. This is also an easy way

to deal with application dependency as it can be applied across all virtual servers and all services. In practice however, the wide-spread use of proxy servers on the Internet and internally in enterprise networks renders this form of persistence almost useless; in theory it works, but proxy-servers inherently hide many users behind a single IP address resulting in none of those users being load balanced after the first user's request. Today, the intelligence of ADC devices allows organizations to actually open up the data packets and create persistence tables for virtually anything within it. This enables them to use much more unique and identifiable information, such as browser cookies or user name, to maintain persistence. However, you must take care to ensure that this identifiable client information will be present in every request made, as any packets without it will not be persisted and will be load balanced again, most likely breaking the application.

## Everything NAT

---

### What is SNAT in F5 BIG-IP LTM? SNAT vs. Inline. What is a NAT?

If you're new to F5 BIG-IP LTM devices and have just started dabbling in the world of Application Delivery and SNAT, you may find yourself asking some questions about address translation. The F5 BIG-IP LTM systems can perform address translation in 3 ways, SNAT, NAT, & Virtual Servers. We'll also cover traffic F5 BIG-IP LTM handle and don't do any address translation for, which we refer to as In-Line communication.

### What is SNAT?

I've seen SNAT referenced two ways, Source Network Address Translation, and Secure Network Address Translation, both of which are correct. "Source" makes it's easier to understand, because you are translating the "source" addresses of the client initiating traffic or as the devices references it the "origin". It's "Secure" because you can't initiate traffic to a SNAT, the "translation" addresses are never known by the host initiating the traffic. In short a SNAT is made of up three components:

- **Translation – Options:** an IP address (single address), a SNAT Pool (multiple addresses), or an Automap (self IP(s) of the Local Traffic Manager). This is what the Source address of the client is translated to.
- **Origin – Options:** All addresses (everything coming in on the VLAN you specify, or an Address list (specific addresses you provide). These are indeed the source addresses of the client.
- **VLAN Traffic – Options:** All VLANs (every VLAN), Enabled on (only on the vlans specified), or Disabled on (on all VLANs except the ones you specify)

The most common misunderstanding is how SNAT can be used. Unlike a traditional NAT, you can't send traffic to a SNAT address. SNATs are either global (i.e. traffic coming through a LTM), or they can be associated with a Virtual Server. The first option is the hardest to get your head around, the second option, associating with a Virtual Server, is a lot easier to grasp and is usually everyone's first exposure to SNAT, using "SNAT automap" applied to a virtual server. In both examples SNAT is generally used to solve routing issues and can be used with a variety of mappings but not limited to, one to one, many to one, all to one, etc. Let's dive into the first option and see if we can get a better understanding of SNAT not applied to a Virtual Server, but affecting the LTM globally.

### Global traffic and SNAT

**Outbound Traffic-** Translating the source address of many hosts on an internal non-Internet routable subnet to one external Internet routable address is a common problem solved with SNAT. Think about how your home router works, it's not the same but is a similar concept. When traffic hits the F5 BIG-IP LTM the "origin" would equate to an "address list" you specify with all the hosts in it or "all addresses" for that specific VLAN, the "Translation" would be one single address (in this example). The destination addresses



now sees the “Translation” address as your new source. When traffic returns to the F5 BIG-IP LTM from the destination it is then translated back to the original origin address. It’s important to note, by default SNATs are allowed on all VLANs, but you can get more granular and split them out between multiple VLANs.

## Virtual Servers and SNAT

**Inbound Traffic-** Virtual Servers can have SNATs applied to them effectively changing the source of the Client imitating traffic to the Virtual Service. You see, in most cases, the servers you want to load balance are NOT going to have the F5 BIG-IP LTM as their gateway, so unless you translate the source address to something that belongs to the F5 BIG-IP LTM, you’re going to end up routing “around” the F5 BIG-IP LTM and not “through” the F5 BIG-IP LTM. Resulting in your VS not working and giving rise to what we call asymmetrical routing, a fancy term for traffic taking a different return path from the original request path. Asymmetrical routing is not always going to break traffic, but when dealing with a statefull device, something that maintains a connection like the F5 BIG-IP LTM, asymmetrical routing can break your communication.

## What is SNAT automap, a simple explanation

Everyone’s first exposure to SNAT is usually SNAT automap. A lot of organizations at some point just turn this on without a good understanding of SNAT. Hopefully after reading this article you have a better understanding of the inner workings of SNAT. The SNAT automap feature is going to change the source address of the communication to the physical IP or “self-ip” of the egress interface/vlan on the F5 BIG-IP LTM that can reach the pool member. *Again, this is so the communication comes back to the ADC, otherwise the destination host would route around the ADC when communicating back to the client, unless of course the servers have the F5 BIG-IP LTM as their gateway.*

## Alternative to SNAT, Inline

An Alternative to SNAT would be an Inline design. Having the servers in your pool Inline means they will need the ADC as their Gateway address. As inconvenient as this might sound vs. the “anything you can route to you can load balance” approach, there are definitely reasons why one might choose to go inline vs SNAT. The one major thing you lose with SNAT, or gain depending on your perspective is the clients source address. With an inline approach you preserve the source address. Some applications and logging systems want to see the “real” source IP of a connection.

## How do I capture my source address with SNAT?

So now you’re SNATing, you feel cool, you look cool, well you are cool! You’re load balancing anything you can route to, life is good and server administrators are happy they didn’t have to jack around their servers and change their gateway.. Until they look in their logs and are confused what happened to all the source address information! Fortunately we can still provide this information to them, it’s just going to require a little bit of

reconfiguration on their side as well as yours. Enter the Web Services XFF header option!!!!!!!!!!

The **X-Forwarded-For** header option when enabled will capture the source address of the client and place it in the HTTP header. The logging server would then need to be configured to grab this value instead of looking at the layer 3 securewirelessource address.

## What is a NAT?

NATs are a one to one mapping between addresses. Unlike SNATs and Virtual servers, NATs can be used for traffic initiated in both Directions. You can Send Traffic to the NAT address or the Origin address can send traffic to any address. NATs are not connection based like SNATs i.e. they are not tracked by the BIG-IP. A NAT is made up of two major components:

- NAT address
- Origin Address

## RADIUS Packet Attributes

Regardless of what happens to the IP headers due to SNAT/NAT/Proxy etc. the content of the RADIUS packets remains intact. So we still retain Calling-Station-IP, NAS-Identifier, Framed-IP-Address etc. The IP header manipulation techniques are there to enforce and control packet routing. Specifically to ensure that asymmetric routing is not occurring between client and server.

## Additional notes to consider on SNAT v Inline



As discussed in our introduction we have chosen a deployment strategy where the F5 BIG-IP LTM is deployed inline to ensure the source IP addresses are not amended when the RADIUS auth reaches the CPPM server. However late in the writing of this TechNote we have reason to believe that a SNAT deployment will work with a ClearPass Deployment. Having the option to deploy either inline or offpath and utilize the SNAT feature provides for an extra level of flexibility which may suit some customers networks. The following are what we believe to be the requirements to make this work successfully with SNAT.

- 1. CPPM MUST have the F5 BIG-IP LTM SNAT address (radius source address) setup as a Network Device in CPPM and set to IETF Vendor Type - This is due to the fact that all Radius will have a source IP of the F5 SNAT address and will need to match against a network device in CPPM to be allowed and to select a Radius Shared Secret to be used*
- 2. CPPM MUST have each individual NAD setup as a Network Device in CPPM using the NAD's configured NAS-IP and set to the specific Vendor Type - This is due to the fact that CoA uses NAS-IP and CPPM will need to lookup in the Network Device config the Radius Shared Secret and Vendor type.*

*3. The Radius Source-IP SHOULD NOT be used in policy to determine enforcement profile to be used, use NAS-IP or other Radius elements instead.*

*4. All Network Devices MUST have the same Radius Shared Secret - This is due to the fact that a single Network Device will be matched against 2 Network Device objects in CPPM, the F5 BIG-IP LTM one and the individual NAD one.*

We intend to verify this in the coming weeks but feel at this junction due to the demands from our customers and partners we will release this document to the field and update our finding in a later version of this document.

## Technology Designs

The following section covers configuration, design and technology of a specific scenario we have built and tested. Our testing has been performed utilizing CPPM 6.3.x code running on two CP-HW-5K appliances. The two CPPM nodes have been clustered to provide a Publisher and Subscriber pair following standard CPPM clustering configuration. We have defined the SUB to be a standby-PUB as a good deployment strategy and perform some limited CPPM failover testing. We have also defined a VIP address between the two CPPM nodes, however we are recommending that this VIP only be used for ClearPass admin traffic. Our VIP address is 10.2.100.180 (pre-empt to cppm181) on the MGMT interfaces. Our VIP address is 10.2.100.180 (pre-empt to cppm181) on the MGMT interfaces.

Virtual IP	Primary Node	Secondary Node	Status
1. 10.2.102.180	cppm181.cppm-testing.com [MGMT] <span style="color: green;">●</span>	cppm182.cppm-testing.com [MGMT]	Enabled

● Indicates current node serving Virtual IP

**Virtual IP Details -**

Virtual IP:

	Node	Interface	Subnet
Primary Node:	--select--	<input type="text"/>	
Secondary Node:	--select--	<input type="text"/>	
Enabled:	<input checked="" type="checkbox"/>		

Reset Delete Save Close

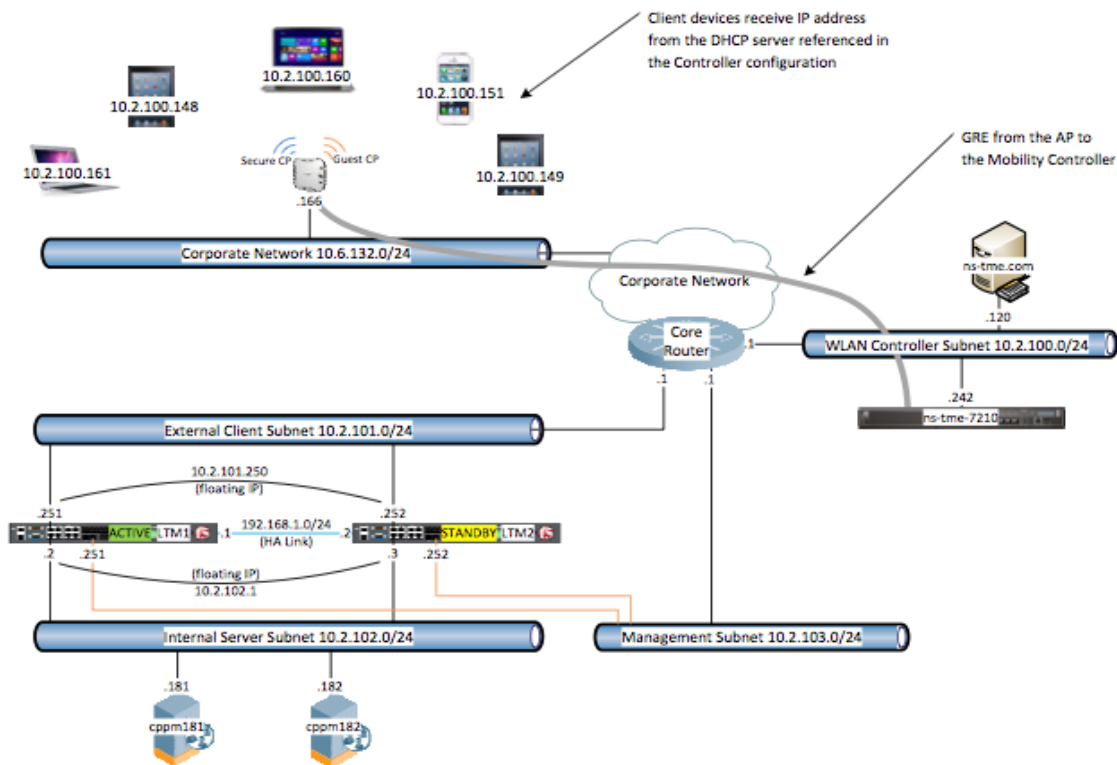
**Figure 3 – Creating VIP groups on a CPPM cluster**

**Our F5 BIG-IP LTM environment:** Our F5 BIG-IP LTM is a dual instance HA cluster of 2 x BIG-IP 3600 with 11.5.1 Build 0.0.110 and Hotfix Version 3.0.131 code, released in June 2014. Configuration of F5 BIG-IP LTM clustering is beyond the scope of this document, but it is well documented on the F5 support site.

**Note:** Multiple deployment scenarios exist for an F5 BIG-IP LTM. Simplistically an F5 BIG-IP LTM can be thought of as a router, packets/flows come in one interface/VLAN and leave on another. They can also be deployed in a L2 or L3 one-armed scenario. Deployment and integration of an F5 BIG-IP LTM into a customer network is beyond the scope of this document and is not covered. Our deployment is based on a simple routed deployment, we have an ‘internal’ server facing VLAN (10.2.102/24) which is the server VLAN or put another way where the CPPM nodes sit. Then we have an “external” client facing VLAN (10.2.101/24) that is effectively where the client traffic is originating from/via and finally we have management VLAN (10.2.103/24). In addition as seen in the earlier SLB overview there is a concept of Virtual Servers (see Page 9 + 14), these are in effect the F5 BIG-IP LTM listening’ VIP’s which receive the incoming data flows and load-balance to the server on the

internal VLAN in accordance with the load balancing algorithm as appropriate. We have used F5 iRules to traffic-engineer the behavior of traffic across the CPPM nodes.

Shown below are several pictorial diagrams representing the network we used to test and produce this TechNote, they have been added to allow the reader to follow references within this document and to provide a design reference going forward.



**Figure 4 - High level Setup for CPPM + F5 BIG-IP LTM network**

The above diagram is showing at a high-level the network we deployed. We attached an AP on our corporate network (10.6.132.x) and this was configured to use a 72xx controller (10.2.100.242) in the TME LAB. DHCP was provided to the clients on the SSID we projected to the AP from the 72xx controller. Our F5 BIG-IP LTM Virtual Servers (aka VIPs) were created on the 10.2.101.x network, this referred to above as the External Client Subnet. Finally, the actually CPPM hosts, are located on the 10.2.102.x networks, this is referred to above as the Internal Server Subnet. The two F5 BIG-IP LTM also share a Management Subnet. This is used in our environment for configuration sync and HA.

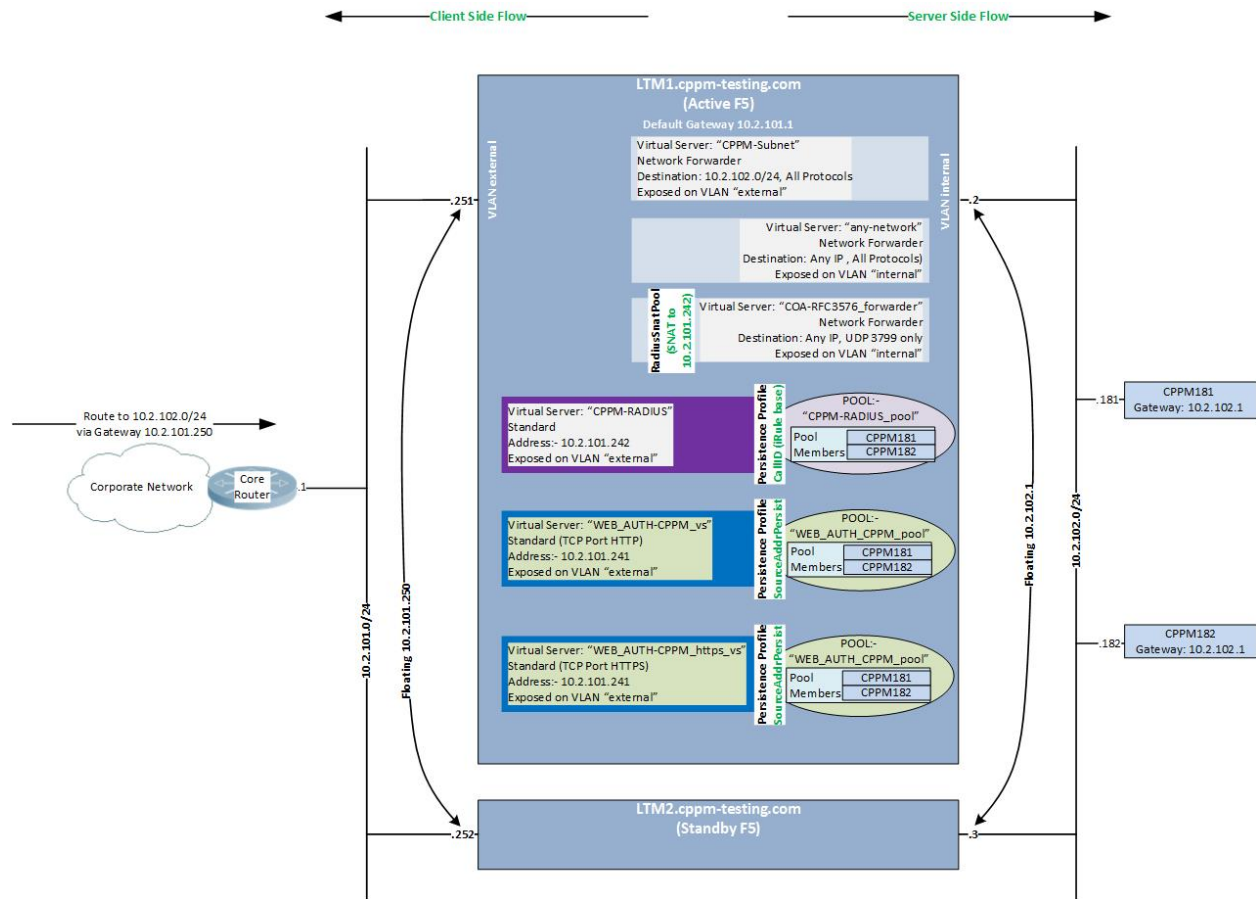


Figure 5 – CPPM Detailed Setup

The above diagram provides a detailed insight to the F5 BIG-IP LTM setup. Above you can see the logical split between Client side and Server side flows. Client to the LHS and Server (CPPM) to the RHS. Also in some detail is the Virtual Server (VS) configuration for the listening processes, RADIUS and WEB and VS forwarders for CoA.

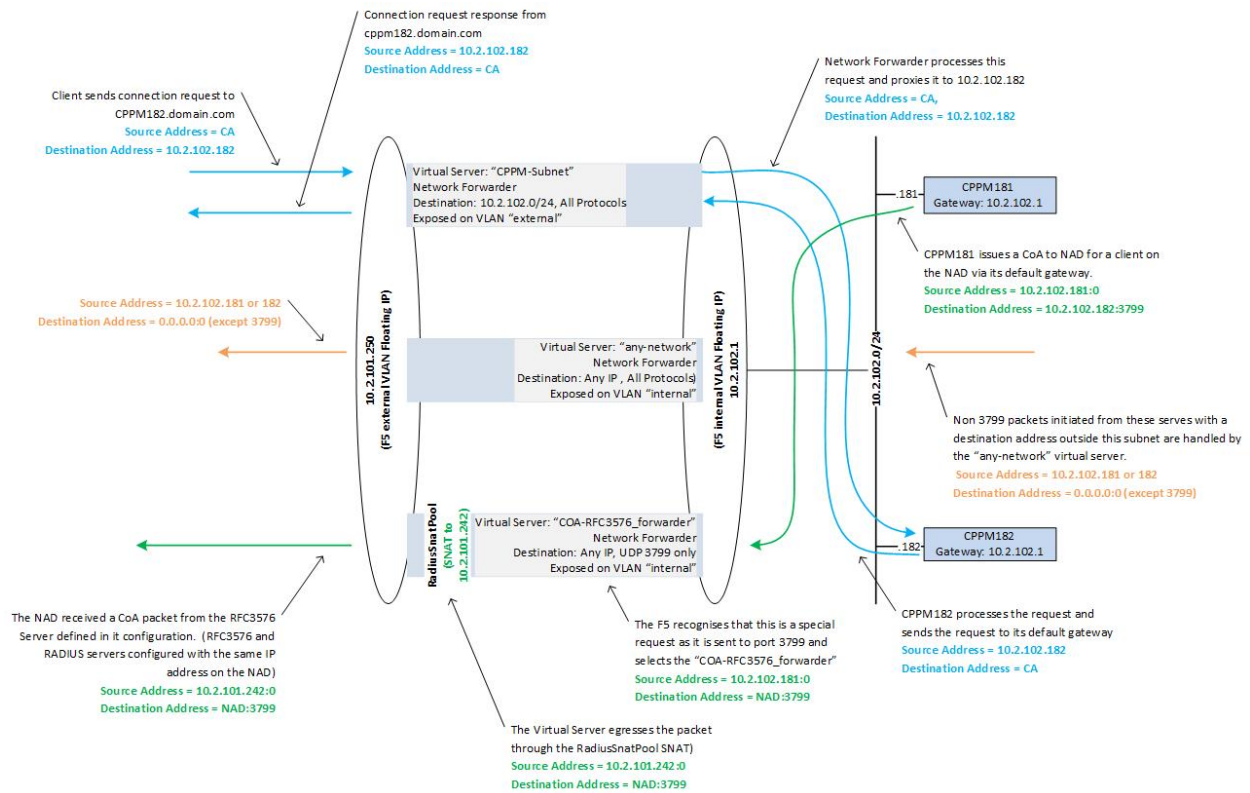


Figure 6 - Data Flow through F5 BIG-IP LTM and CPPM Network

The above schematic breaks down the data flow for the IP Forwarders configured. A client request to the actual address of a CPPM server will be addressed by the "CPPM Subnet" network forwarder. Any traffic initiated by a CPPM server will be addressed by the "any network" network forwarder **unless** this traffic is COA (UDP 3799) where this particular traffic type will be addressed by the "COA-RFC3576\_forwarder" network forwarder.

### What's In What's Out of the Client-> F5 BIG-IP LTM -> CPPM SLB flows

Below is visualization of what data flows goes through and are processed as 'interesting' traffic, by interesting we mean its intercepted and load-balanced to the backend CPPM servers, against traffic that is 'un-interesting' and is just forwarded to the ClearPass target server.

## Client (PC/NAD) <-> CPPM/BigIP

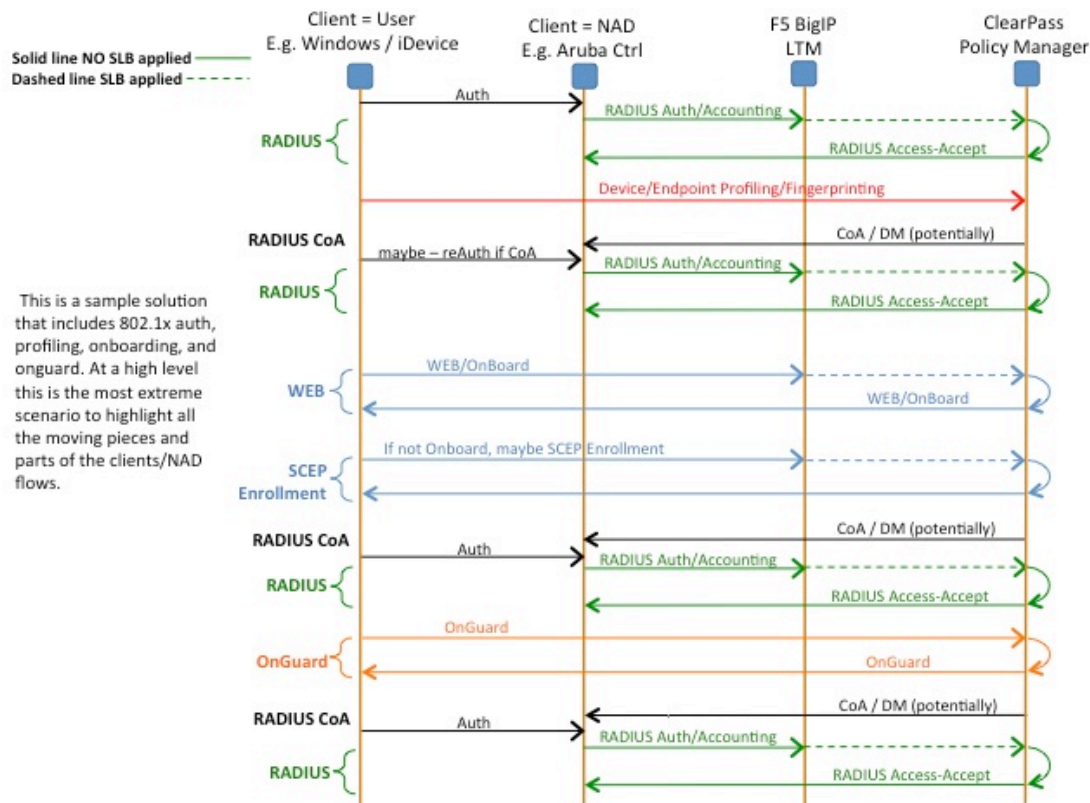


Figure 7 - Overview of the traffic types that are Load-Balanced and those that aren't



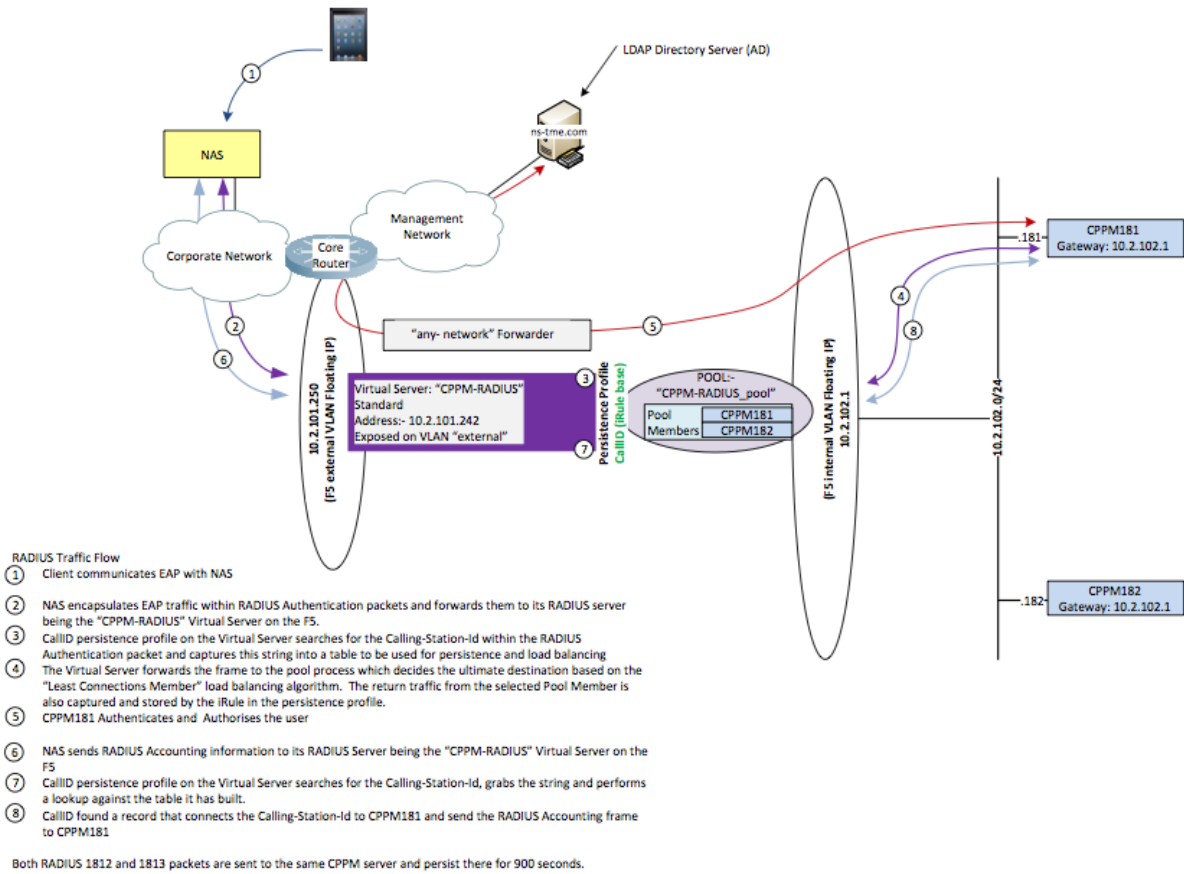


Figure 8 – RADIUS Virtual Server Overview & Data-Flow

The below sections are primarily focused on load balancing the front-end incoming client traffic, we have not documented or tested for this TechNote backend service load balancing. In reference to backend service load balancing we are referring to the deployment of load-balancing services outside of CPPM, such as a cluster of Windows Servers providing AD functionality, or a group of SQL servers that CPPM is querying for authorization.

F5 as one of the industry leaders in the ADC/SLB technology space have developed their OS to provide extremely useful wizard type configurations. By using the F5 iApps® Application Services a lot of the complicated dependencies have been consolidated making the configuration more intuitive. However we found that by utilizing the iApps wizard we were not able to fully configure the BIGIP to perform the functions we required, i.e. configuring persistency between ports 1812 (RADIUS Auth) and ports 1813 (RADIUS Accounting) to the same server. At the time of writing we found that the best practice is a manual configuration of the F5 BIG-IP LTM to achieve our objective, the exception was the configuration of HTTP(S) services through the use of F5 iApps.

## Health-Checks

Health-checks (HC) provide an important part of the overall solution when fronting CPPM nodes with an ADC/SLB device. The IP address that the clients communicate with as their destination is hosted and owned by the ADC/SLB. In effect, this IP address will always be available. It is strongly recommended that the ADC/SLB's are deployed in redundant pairs so that in the unlikely event of a failure somewhere within the SLB network infrastructure the IP address representing the CPPM services always remains UP and reachable. We recommend to treat a CPPM service failure in this part of the network as a binary AND operation. It is possible to allow some transactions (e.g. RADIUS authentication) to continue to be serviced by a CPPM node even if the HTTP web process on that node has failed (as detected by the ADC/SLB). We could then continue to direct Guest HTTP Registration/OnBoarding to a separate node that has an active HTTP web process.

This best practices document strongly recommends that if **any** HC on a node fails, be that detected by ICMP, HTTP etc. the **entire CPPM node** be taken out of service in respect of it being available to process **any** transactions.


We are recommending that a tiered number of health checks (HC) be used. Within the F5 BIG-IP LTM HC configuration you can define if ALL the checks must pass or just a partial number are successful, this partial number can be configured. We believe the best and most stable solution is to ensure that **all** HC pass and in the event of **any** HC failure the node is marked down.

### ICMP

As a minimum we can define that CPPM nodes are 'pinged' through ICMP. Unfortunately this provides little value to us except that the IP stack is active. There is little knowledge that be gained about the health or availability of CPPM's underlying processes. We strongly recommend that additional Health-Checks are used to verify the validity of the CPPM node and its ability to process transactions and that ICMP alone is not used to determine that the CPPM node is fit to be classed as 'in-service'. Below we have listed additional Health-Checks that can be utilized to better determine the validity of a CPPM node to process transactions.

### RADIUS

F5 BIG-IP LTM supports the ability to add RADIUS Health-Checks against multiple AAA services. You can add HC's for Authentication & Authorization (ports 1812/1645) and separate HC's for Accounting (ports 1813/1646).

 **Note:** Because we have restrictions in our ability to perform direct SQL HC's we recommend that two RADIUS HC's be configured. One is defined to use an external authentication source such as AD, this allows CPPM and the down-stream repository to be validated and then a separate HC defined against a local user. When we perform a HC against this local user, CPPM will make a SQL call to the underlying PostgreSQL DB to

validate the user exists. No caching is performed by the CPPM RADIUS service so a SQL call is made for every authentication therefore validating the availability and state of the PostgreSQL DB process.

## WEB

Utilizing a HC against HTTP/s process yields likely the most useful status of a node to process transactions. The HTTP Apache process is responsible for Administration, Guest Registration-Login-Self-Service. By running HC's against the CPPM web service you are checking and monitoring many features with in CPPM.

## SQL

To utilize the ability to perform SQL health checks against a CPPM node mandates that the SQL connection is SSL encrypted. This SSL encryption occurs on the PostgreSQL native port of 5432. The exposed CPPM PostgreSQL User-ID (appexternal) mandates that the connection to the DB is SSL encrypted. Unfortunately the current F5 BIG-IP LTM PostgreSQL Health-Check does not support the option of using SSL over the PostgreSQL native port.

ARUBA will investigate additional methods of supporting a SQL monitor to the CPPM PostgreSQL database. This is under investigation at this time and no commitment is made to provide this functionality.

In our discussion directly with F5, they have also commented that enhancing the SQL monitor to add encryption is also under investigation for a later release. We will continue to monitor this over time and update this document should the desired behavior change with F5 BIG-IP LTM.

# F5 BIG-IP LTM Component Configuration



Our recommendation is to manually configure the F5 BIG-IP LTM with the exception for the configuration using iApps for HTTP Services. Below the various building blocks required.

## Building Blocks

For this implementation we chose the BIG-IP Local Traffic Manager (LTM) product module of the F5 BIG-IP product family, the definition of these components are detailed below.

Building Block	Description
<b>Node</b>	Is a logical object on the F5 BIG-IP LTM system that identifies the IP address of a physical resource on the network – in this case the CPPM servers
<b>Node Default Health Monitor</b>	Is a health monitor designed to report the status of the node itself - in this case we use ICMP echo/echo-reply to tell us if the node is up.
<b>Load Balancing Pool</b>	Is a logical set of devices, such as web servers, that you group together to receive and process traffic. The load balancing pool has the responsibility of distributing traffic to its set of devices according to the load balancing method
<b>Pool Member</b>	Is a logical object defined within a pool that represents a node inclusive of the service offered by that node - In our lab, a pool member for the RADIUS pool is defined as 10.2.102.181:1812
<b>Pool Health Monitor</b>	Is a health monitor designed to report the status of the service offered by a pool member
<b>Load Balancing Method</b>	Is the method used by the F5 BIG-IP LTM system to distribute traffic across the members within a pool – we selected the load balancing method of “Least Connections Member”
<b>Virtual Server</b>	A virtual server is a traffic-management object on the BIG-IP system that is represented by an IP address and a service. The function of the virtual server is to process requests from clients consuming the service offered.
<b>IP Forwarder</b>	An IP Forwarder is nothing more than a specific virtual server accepting traffic that matches the virtual server address and forwarding it to the destination IP address that is specified in the request rather than load balancing the traffic to a pool. Address translation is disabled when you create an IP forwarding virtual server, leaving the destination and source address in the packet unchanged.
<b>iRule</b>	An F5 iRule is a powerful and flexible feature of F5 BIG-IP LTM devices applied at the Virtual Server allowing you to directly manipulate and manage any IP application traffic. F5 iRules enable you to customize how you intercept, inspect, transform, and direct inbound or outbound application traffic. F5 iRules are based on the TCL scripting language.
<b>Persistence Profile</b>	A persistence profile is a pre-configured object that automatically enables session persistence when you assign the profile to a virtual server. Persistence profiles are either pre-configured or can be created to serve a specific purpose. In this case we created a persistence profile based on an F5 iRule to persist RADIUS session traffic from a client to a specific CPPM. Persistence is based on the Calling-Station ID extracted from the RADIUS packet and not the IP address of the NAS.

## Adding NODES

First we have to add the NODES (aka CPPM Servers). Add all the CPPM nodes within your cluster, providing an appropriate name and the IP address of the node. Take special care to ensure that the Health Monitor is set to 'Node Default' when you add the nodes. All other parameter can be left at their default settings.

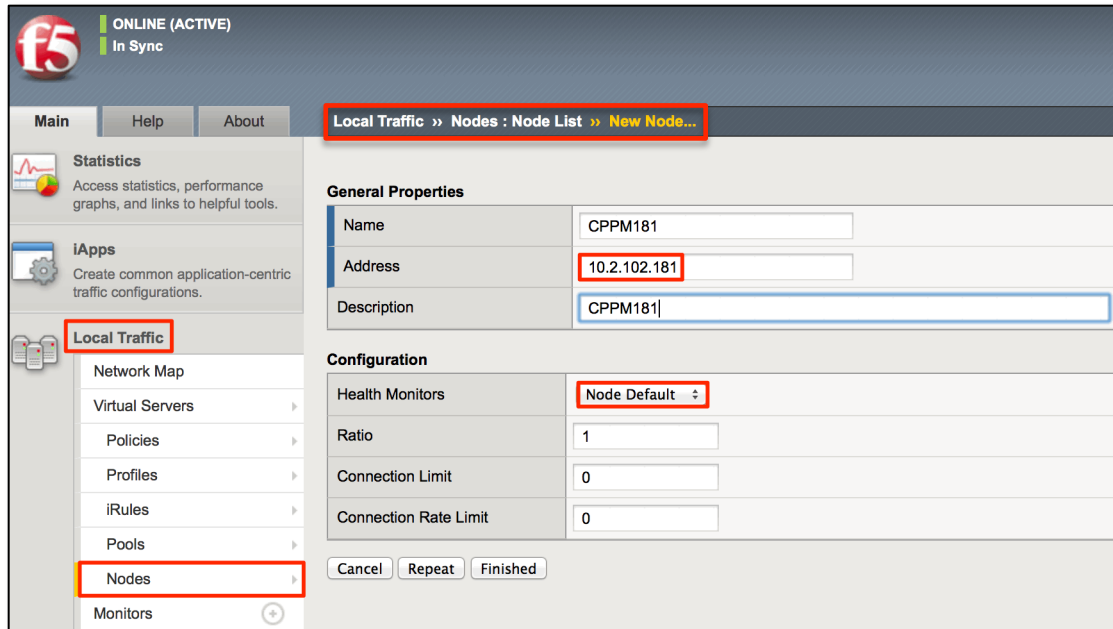


Figure 9 - Adding CPPM Nodes to F5 BIG-IP LTM

Following on from this, ensure that the Health Monitor for the Nodes is set as below. Just configure an icmp monitor at this stage. We will add additional monitor later.

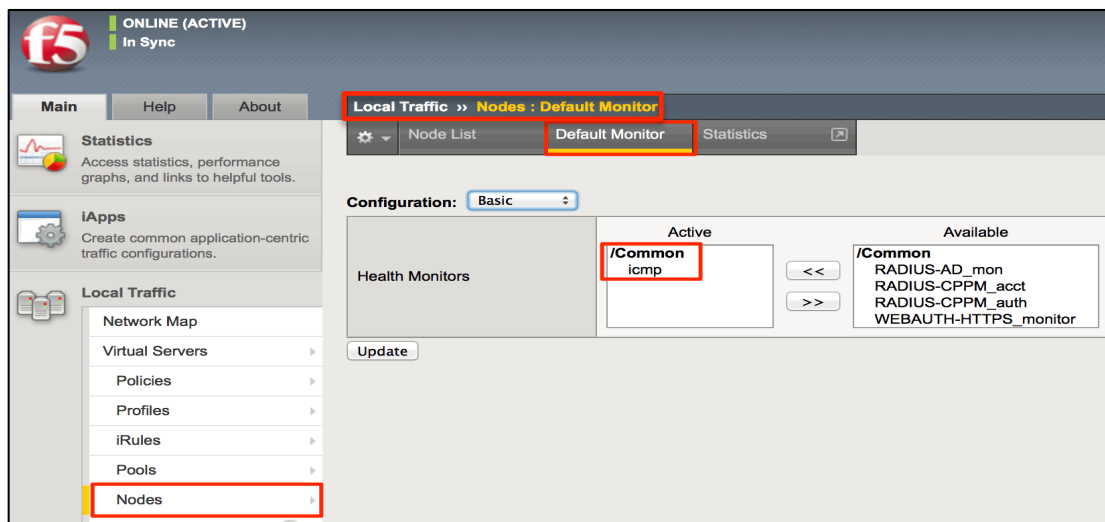



Figure 10 - Setting 'Node' default monitor to ICMP

After adding the nodes required, ensure that all the Nodes  *N* once they are powered up, connected to the network and have an IP address configured as can be seen below for the two nodes we have in our deployment.

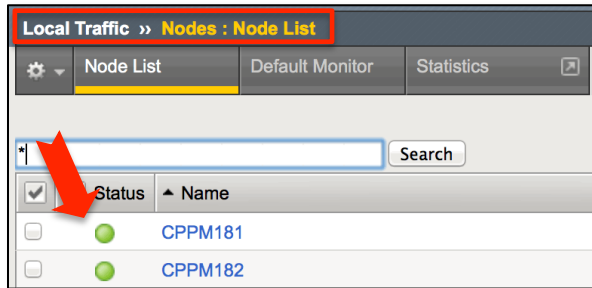


Figure 11 - Nodes showing as available...



General Properties	
Name	CPPM181
Address	10.2.102.181
Partition / Path	Common
Description	<input type="text"/>
Availability	 Available (Enabled) - Node address is available 2014-06-30 17:28:06
Health Monitors	 icmp
Monitor Logging	<input type="checkbox"/> Enable
Current Connections	0
State	<input checked="" type="radio"/> Enabled (All traffic allowed) <input type="radio"/> Disabled (Only persistent or active connections allowed) <input type="radio"/> Forced Offline (Only active connections allowed)

Figure 12 - Node availability detail - Time/Date last checked

## Adding MONITORS

Next we need to add multiple monitors to the pool that will be used to Health-Check the status of the CPPM processes going forward. We configured the following monitors

- **Radius Authentication Monitor for an Active\_Directory User**
- **Radius Authentication Monitor for a CPPM Local User**
- **Radius Accounting Monitor**
- **HTTP Monitor**
- **HTTPS Monitor**

## Radius Authentication Monitor for an Active\_Directory User

To add these monitors, go to **Local Traffic->Monitors** then 'Create' in the top RHS... first we add the RADIUS authentication monitor for the AD user. This user must be configured in Active-Directory. Take special notice of the highlighted fields below.... especially the monitor interval setting of 15 secs, Manual Resume set to No and the NAS IP Address set to 10.2.102.10.

The screenshot displays the configuration for a new RADIUS monitor. The breadcrumb navigation at the top indicates the path: **Local Traffic » Monitors » New Monitor...**. The configuration is divided into two main sections: **General Properties** and **Configuration**.

**General Properties:**

- Name: RADIUS-AD\_mon
- Description: RADIUS LDAP Authentication
- Type: RADIUS
- Parent Monitor: radius

**Configuration:** (Advanced)

- Interval: 15 seconds
- Up Interval: Disabled
- Time Until Up: 0 seconds
- Timeout: 46 seconds
- Manual Resume: No
- User Name: f5-ad (labeled as AD User)
- Password: [Redacted]
- Secret: [Redacted]
- NAS IP Address: 10.2.102.10 (labeled as Must config in CPPM devices)
- Alias Address: \* All Addresses
- Alias Service Port: 1812
- Debug: No

Figure 13 - F5 BIG-IP LTM Health Check Active-Directory user check

The NAS IP address is the Network Device configured in CPPM that the F5 BIG-IP LTM uses for the RADIUS health check

### Radius Authentication Monitor for a CPPM Local User

Next we create a monitor to authenticate against a local CPPM user.

**Local Traffic » Monitors » New Monitor...**

**General Properties**

Name	RADIUS-CPPM_auth_mon
Description	RADIUS-CPPM_auth_mon
Type	RADIUS
Parent Monitor	radius

**Configuration:** Advanced

Interval	15	seconds
Up Interval	Disabled	
Time Until Up	0	seconds
Timeout	46	seconds
Manual Resume	<input type="radio"/> Yes <input checked="" type="radio"/> No	
User Name	f5-hlthchk	Local CPPM user
Password	.....	
Secret	.....	
NAS IP Address	10.2.102.10	Must config in CPPM devices
Alias Address	* All Addresses	
Alias Service Port	1812	Other: .....
Debug	No	

Cancel Repeat Finished

Figure 14 - F5 BIG-IP LTM health checking against a local CPPM user



## Configure Radius Health-Check Monitor on CPPM side

In the above we have defined two users 'f5-hlthchk' & 'f5-ad'. The first user must exist as a **local** user on CPPM and the second user should be an **Active-Directory** user. There is extra merit and having a user defined within AD as this provides an enhanced level of availability checking in that the health-check initiated from the F5 BIG-IP LTM will have to be processed by the backend AD system, one additional 'hop' away from CPPM. This 'hop' verifies that the AD and CPPM are also in communication. When configuring the health-check service on CPPM the CPPM service definition should be configured to use both the **[Local User Repository]** and **[Active Directory]** as the authentication source.



**Note:** Because we have restrictions in our ability to perform direct SQL HC's we recommend that two RADIUS HC's be configured. One is defined to use an external authentication source such as AD, this allows CPPM and the down-stream repository to be validated and then a separate HC defined against a local user. When we perform a HC against this local user, CPPM will make a SQL call to the underlying PostgreSQL DB to validate the user exists. No caching is performed by the CPPM RADIUS service so a SQL call is made for every authentication, validating the availability and state of the PostgreSQL DB.

Configuration » Identity » Local Users

### Local Users

Filter: User ID contains [ ] + Go Clear Filter

#	<input type="checkbox"/>	User ID ▲	Name	Role
1.	<input type="checkbox"/>	danny	danny	[Contractor]
2.	<input type="checkbox"/>	f5-hlthchk	f5-hlthchk	[Employee]

Figure 15 - Local CPPM User - "f5-hlthchk"

Below we configured our CPPM health-check service to match for the DNS FQDN of the F5 BIG-IP LTM as the IETF NAS-Identifier, in our case LTM1/LTM2.cppm-testing.com, customers can select appropriate rules to match to ensure that their health-check rules match this service correctly. Notice especially that we have configured two authentication sources so as we can use a single service definition to authenticate the two users, one against the local user repository and one against AD.

Configuration » Services » Edit - f5-hlthchk

### Services - f5-hlthchk

Summary Service Authentication Roles Enforcement

**Service:**  
 Name: f5-hlthchk  
 Description: f5-hlthchk  
 Type: RADIUS Enforcement ( Generic )  
 Status: Enabled  
 Monitor Mode: Disabled  
 More Options: -

**Service Rule**  
 Match ANY of the following conditions:

Type	Name	Operator	Value
1. Radius:IETF	NAS-Identifier	CONTAINS	LTM1.cppm-testing.com
2. Radius:IETF	NAS-Identifier	CONTAINS	LTM2.cppm-testing.com

**Authentication:**  
 Authentication Methods: 1. [PAP]  
 2. [MSCHAP]  
 3. [CHAP]  
 Authentication Sources: 1. [Local User Repository] ← Local and AD Auth source  
 2. Clearpass lab AD  
 Strip Username Rules: -

**Roles:**  
 Role Mapping Policy: -

**Enforcement:**  
 Use Cached Results: Disabled  
 Enforcement Policy: [Sample Allow Access Policy]

Figure 16 - CPPM Service Health Check for F5 RADIUS

The consolidated Access Tracker below shows the two CPPM nodes in our cluster. You can see on the RHS the 15 seconds health-check duration as defined on the F5 BIG-IP LTM. It also shows 'ACCEPT' in the Login Status column. This shows that the health-check on the F5 BIG-IP LTM to CPPM is successful. You also see the two usernames 'f5-hlthchk' & 'f5-ad' being used for the local and AD health checks

Monitoring » Live Monitoring » Access Tracker

### Access Tracker Jul 01, 2014 14:56:11 PDT

[All Requests] default (2 servers) Last 1 day before Today

Filter: Service contains f5 Go Clear Filter Show 50

#	Server	Source	Username	Service	Login Status	Request Timestamp
1.	10.2.102.182	RADIUS	f5-hlthchk	f5-hlthchk	ACCEPT	2014/07/01 14:56:10
2.	10.2.102.181	RADIUS	f5-hlthchk	f5-hlthchk	ACCEPT	2014/07/01 14:56:09
3.	10.2.102.182	RADIUS	f5-hlthchk	f5-hlthchk	ACCEPT	2014/07/01 14:56:08
4.	10.2.102.181	RADIUS	f5-hlthchk	f5-hlthchk	ACCEPT	2014/07/01 14:56:07
5.	10.2.102.182	RADIUS	f5-ad	f5-hlthchk	ACCEPT	2014/07/01 14:56:06
6.	10.2.102.181	RADIUS	f5-ad	f5-hlthchk	ACCEPT	2014/07/01 14:56:05
7.	10.2.102.181	RADIUS	f5-ad	f5-hlthchk	ACCEPT	2014/07/01 14:56:04
8.	10.2.102.182	RADIUS	f5-ad	f5-hlthchk	ACCEPT	2014/07/01 14:56:03
9.	10.2.102.182	RADIUS	f5-hlthchk	f5-hlthchk	ACCEPT	2014/07/01 14:55:55
10.	10.2.102.181	RADIUS	f5-hlthchk	f5-hlthchk	ACCEPT	2014/07/01 14:55:54
11.	10.2.102.182	RADIUS	f5-hlthchk	f5-hlthchk	ACCEPT	2014/07/01 14:55:53
12.	10.2.102.181	RADIUS	f5-hlthchk	f5-hlthchk	ACCEPT	2014/07/01 14:55:52

Figure 17 - CPPM Access Tracker showing local and AD users

CPPM receives a RADIUS Access-Request and will return an RADIUS Access-Accept as shown below in the packet trace between CPPM and F5 BIG-IP LTM device.

No.	Time	Source	Destination	Protocol	Length	Info
70	3.080816	10.2.100.251	10.2.100.180	RADIUS	111	Accounting-Request(4) (id=224, l=69)
71	3.084581	10.2.100.180	10.2.100.251	RADIUS	62	Accounting-Response(5) (id=224, l=20)
294	6.082770	10.2.100.251	10.2.100.180	RADIUS	120	Access-Request(1) (id=227, l=78)
312	6.100902	10.2.100.180	10.2.100.251	RADIUS	120	Access-Accept(2) (id=227, l=78)

Figure 18 - RADIUS Request/Accept messages

Request Details		
Summary	Input	Output
Session Identifier:	R00021194-10-53b375b3	
Date and Time:	Jul 01, 2014 20:00:03 PDT	
End-Host Identifier:	-	
Username:	f5-ad	← AD user
Access Device IP/Port:	10.2.102.10:	
System Posture Status:	UNKNOWN (100)	
Policies Used -		
Service:	f5-hlthchk	
Authentication Method:	PAP	
Authentication Source:	AD:10.2.100.120	← AD server
Authorization Source:	Clearpass lab AD	
Roles:	[User Authenticated]	
Enforcement Profiles:	[Allow Access Profile]	
Service Monitor Mode:	Disabled	
Online Status:	Not Available	

Figure 19 - Detailed Access Tracker message for health check

Above shows the details behind the AD user health check messages in access tracker. This shows it is configured to use PAP as an auth method and the authentication source is our AD server. The local user health check is the same except the Authentication source is local.

## HTTP Monitor



Now we have to add two Monitors for WEB traffic, one specifically for port 80 and another for port 443 traffic. Firstly the **HTTP** (port 80) monitor. Notice that the monitor type is **'http'**, we have changed the interval monitor time to be the same frequency as the RADIUS monitor. Also we have used the Alias Service Port of HTTP (80). This is all that is required for this simple monitor to work.

The screenshot shows the configuration for a new HTTP monitor in the F5 BIG-IP LTM interface. The configuration is as follows:

Local Traffic » Monitors » New Monitor...	
<b>General Properties</b>	
Name	WEBAUTH-HTTP_monitor
Description	WEBAUTH-HTTP_monitor
Type	HTTP
Parent Monitor	http
<b>Configuration: Basic</b>	
Interval	15 seconds
Timeout	46 seconds
Send String	GET /\r\n <i>← Leave as default</i>
Receive String	
Receive Disable String	
User Name	
Password	
Reverse	<input type="radio"/> Yes <input checked="" type="radio"/> No
Transparent	<input type="radio"/> Yes <input checked="" type="radio"/> No
Alias Address	* All Addresses
Alias Service Port	80 HTTP

Figure 20 – Adding the F5 BIG-IP LTM HTTP monitor

**Note:** There is no configuration required on CPPM for this monitor to work.

## HTTPS Monitor

Next we add the **HTTPS** (port 443) monitor. Notice that the monitor type is '**https**', we have changed the interval monitor time to be the same frequency as the RADIUS monitor. We have used the Alias Service Port of HTTPS (443). This is all that is required for this monitor to work.

The screenshot shows the configuration for a new HTTPS monitor. The breadcrumb navigation is 'Local Traffic >> Monitors >> New Monitor...'. The 'General Properties' section includes:

- Name: WEBAUTH-HTTPS\_monitor
- Description: WEBAUTH-HTTPS\_monitor
- Type: HTTPS
- Parent Monitor: https

The 'Configuration' section is set to 'Basic' and includes:

- Interval: 15 seconds
- Timeout: 46 seconds
- Send String: GET /\r\n (with a red arrow pointing to it and the text 'Leave as default')
- Receive String: (empty)
- Receive Disable String: (empty)
- Cipher List: DEFAULT:+SHA:+3DES:+KEDH
- User Name: (empty)
- Password: (empty)
- Reverse:  Yes  No
- Transparent:  Yes  No
- Alias Address: \* All Addresses
- Alias Service Port: 443 (with a dropdown menu set to HTTPS)

Figure 21 - Adding the F5 BIG-IP LTM HTTPS monitor

**Note:** There is no configuration required on CPPM for this monitor to work.

## Adding POOLS

Next we need to add the Pools, remember from our F5 BIG-IP LTM introduction. Pools are basically a collection of nodes providing specific services in F5 speak. Please take special care when configuring the pools as incorrectly adding the wrong monitors to a pool could have disastrous consequences if the wrong monitors are 'monitoring' the wrong services. Below is the pool configuration for **RADIUS**. Notice specifically that we have changed the default Load Balancing to **Least Connection (member)** and have added the resource members.

**Note:** On the advanced configuration, activated by changing 'Basic' to 'Advanced' is a setting called 'Slow Ramp Time'. The default is 10 seconds, we recommend changing this to be 300 seconds. For large installation this controls the impact on a new node to ensure when its added to a pool the F5 BIG-IP LTM does not immediately start sending it data to process. This allows for the node, if its still booting/restarting, to fully complete its startup before receiving authentication/accounting data to process.

The screenshot displays the 'New Pool...' configuration page in the F5 BIG-IP LTM interface. The breadcrumb trail at the top reads 'Local Traffic » Pools : Pool List » New Pool...'. The configuration is set to 'Basic'.

**Configuration:** Basic

**Name:** CPPM-RADIUS\_pool

**Description:** Pool of CPPM servers for RADIUS Authentication and Accounting

**Health Monitors:**

Active	Available
<b>/Common</b> RADIUS-AD_mon RADIUS-CPPM_acct_mon RADIUS-CPPM_auth_mon	<b>/Common</b> WEBAUTH-HTTPS_monitor WEBAUTH-HTTP_monitor cppm-sql-hc gateway_icmp

**Resources:**

**Load Balancing Method:** Least Connections (member)

**Priority Group Activation:** Disabled

**New Members:**

- R:1 P:0 C:0 CPPM182 10.2.102.182 :1812
- R:1 P:0 C:0 CPPM181 10.2.102.181 :1812

Figure 22 - Adding F5 BIG-IP LTM RADIUS pool

**Note:** The above monitors will check RADIUS AD-user authentication, RADIUS local-user authentication and RADIUS accounting authentication, failure of any one of these monitors will mark the resource as down and take it out of the pool. Note also that the WEBAUTH

monitors are **not** included for the RADIUS pool as the RADIUS service can operate independent of WEB services

Below is the pool configuration for **WEBAUTH (port 80)**. Notice specifically that we have changed the default Load Balancing to **Least Connection (member)** and have added the resource members.

**Note:** On the advanced configuration, activated by changing 'Basic' to 'Advanced' is a setting called 'Slow Ramp Time'. The default is 10 seconds, we recommend changing this to be 300 seconds. For large installation this controls the impact on a new node to ensure when its added to a pool the F5 BIG-IP LTM does not immediately start sending it data to process. This allows for the node, if its still booting/restarting, to fully complete its startup before receiving authentication/accounting data to process.

The screenshot shows the configuration for a new pool named **WEB\_AUTH\_CPPM\_pool**. The configuration is set to **Basic**. The Name is **WEB\_AUTH\_CPPM\_pool** and the Description is **CPPM Pool for http WEB\_AUTH**. The Health Monitors section shows three monitors in the **Active** state: **RADIUS-AD\_mon**, **RADIUS-CPPM\_auth\_mon**, and **WEBAUTH-HTTP\_monitor**. The Resources section shows the Load Balancing Method set to **Least Connections (member)**, Priority Group Activation set to **Disabled**, and two new members added: **R:1 P:0 C:0 CPPM181 10.2.102.181 :80** and **R:1 P:0 C:0 CPPM182 10.2.102.182 :80**.

**Figure 23 - Adding F5 BIG-IP LTM WEBAUTH (port 80) pool**

**Note:** The above monitors will check RADIUS AD-user authentication, RADIUS local-user authentication and the WEB server on CPPM. The monitor to CPPM apache server is in this case based upon a port 80 monitor, failure of any one of these monitors will mark the resource as down and take it out of the pool.

Below is the pool configuration for **WEBAUTH (port 443)**. Notice specifically that we have changed the default Load Balancing to **Least Connection (member)** and have added the resource members.

**Note:** On the advanced configuration, activated by changing 'Basic' to 'Advanced' is a setting called 'Slow Ramp Time'. The default is 10 seconds, we recommend changing this to be 300 seconds. For large installation this controls the impact on a new node to ensure when its added to a pool the F5 BIG-IP LTM does not immediately start sending it data to process. This allows for the node, if its still booting/restarting, to fully complete its startup before receiving authentication/accounting data to process.

**Local Traffic >> Pools : Pool List >> New Pool...**

**Configuration:** Basic

**Name:** WEB\_AUTH-CPPM\_https\_pool

**Description:** CPPM Pool for https WEB\_AUTH

**Health Monitors:**

Active	Available
<b>/Common</b> RADIUS-AD_mon RADIUS-CPPM_auth_mon <b>WEBAUTH-HTTPS_monitor</b>	<b>/Common</b> RADIUS-CPPM_acct_mon WEBAUTH-HTTP_monitor cppm-sql-hc gateway_icmp

**Resources:**

**Load Balancing Method:** Least Connections (member)

**Priority Group Activation:** Disabled

**New Members:**

- R:1 P:0 C:0 CPPM181 10.2.102.181 :443
- R:1 P:0 C:0 CPPM182 10.2.102.182 :443

**Figure 24 - Adding F5 BIG-IP LTM WEBAUTH (port 443) pool**

**Note:** The above monitors will check RADIUS AD-user authentication, RADIUS local-user authentication and the WEB server on CPPM. The monitor to CPPM apache server is in this case based upon a port 80 monitor, failure of any one of these monitors will mark the resource as down and take it out of the pool.



## Adding Virtual Servers (VS)

This section is perhaps one of the most complicated sections after the configuration of F5 iRules..!!

In this section you are effectively creating the listening VIP's on F5 BIG-IP LTM that the clients communicate with. We have specific VS for specific services - RADIUS, HTTP, HTTPS & three special IP forwarding Virtual Servers, Any-Network, CPPM-Subnet and CoA.

Let's begin by configuring the IP Forwarding Virtual Servers:

### ANY-network IP Forwarder (VS)

Before you configure this IP forwarder please ensure that the customer's device does not have one already configured. This particular forwarder works like a default gateway static route in that this listener processes all traffic of unknown destination. If the customer's device has one of these then ensure that it is exposed to the VLAN that hosts the CPPM servers.

### Adding the IP Forwarder (VS)

On the left hand side navigation plane navigate to **Local Traffic > Virtual Servers** and click on the "Create..." button found in the top right of the action plane. The following configuration screen will appear. Proceed to populate the configuration screen as shown in the screen shot below:

**Local Traffic » Virtual Servers : Virtual Server List » New Virtual Server...**

**General Properties**

Name: any-network  
 Description: Default route to any-network  
 Type: Forwarding (IP)  
 Source: 0.0.0.0/0  
 Destination: Type:  Host  Network  
 Address: 0.0.0.0  
 Mask: 0.0.0.0  
 Service Port: \*, All Ports  
 Notify Status to Virtual Address:   
 State: Enabled

**Configuration:** Basic

Protocol: \* All Protocols  
 Protocol Profile (Client): fastL4  
 VLAN and Tunnel Traffic: Enabled on...  
 VLANs and Tunnels: Selected: /Common, internal; Available: /Common, HA, external, http-tunnel, socks-tunnel  
 Source Address Translation: None

**Acceleration**

Rate Class: None  
 SPDY Profile: None

**Resources**

iRules: Enabled; Available: /Common, CallIDUIE, HTTPTCPTablePersist, TIPSSecure, \_sys\_APM\_ExchangeSupport\_OA\_Basi

**Annotations:**  
 - Red box around 'any-network' in Name field.  
 - Red box around 'Default route to any-network' in Description field.  
 - Red box around 'Forwarding (IP)' in Type dropdown.  
 - Red box around '0.0.0.0/0' in Source field.  
 - Red box around '0.0.0.0' in Address field and '0.0.0.0' in Mask field.  
 - Red box around '\*', 'All Ports' in Service Port dropdown.  
 - Red box around '\* All Protocols' in Protocol dropdown.  
 - Red box around 'fastL4' in Protocol Profile dropdown.  
 - Red box around 'Enabled on...' in VLAN and Tunnel Traffic dropdown.  
 - Red box around 'internal' in Selected VLAN list.  
 - Red arrow pointing to 'internal' in Selected VLAN list.  
 - Red text: 'Move the CPPM VLAN to Selected' near Source Address Translation.  
 - Red arrow pointing to 'iRules' section in Resources.  
 - Red text: 'Leave this empty' near SPDY Profile dropdown.

Figure 25 - Adding 'ANY-network' IP Forwarder VS

Leave all additional settings after iRules in the Resources section at default. Click “Finished” at the bottom left of the action plane when the above steps are completed. You have now configured this Virtual Server.

## CPPM-Subnet IP Forwarder (VS)

This IP Forwarder forwards traffic to the subnet hosting the CPPM servers. The OnGuard agent uses accesses the CPPM servers using their actual IP address. Server management via SSH or the UI is also possible using this forwarder providing that CPPM has been configured to use a single interface.

### Adding the IP Forwarder (VS)

On the left hand side navigation plane navigate to **Local Traffic > Virtual Servers** and click on the “Create...” button found in the top right. The following configuration screen will appear. Proceed to populate the configuration screen as shown in the screen shot below:

**Local Traffic >> Virtual Servers : Virtual Server List >> New Virtual Server...**

**General Properties**

Name	CPPM-Subnet
Description	Direct Access to the CPPM Servers
Type	Forwarding (IP)
Source	0.0.0.0/0
Destination	Type: <input type="radio"/> Host <input checked="" type="radio"/> Network Address: 10.2.102.0 Mask: 255.255.255.0
Service Port	* All Ports
Notify Status to Virtual Address	<input checked="" type="checkbox"/>
State	Enabled

**Configuration: Basic**

Protocol	* All Protocols
Protocol Profile (Client)	fastL4
VLAN and Tunnel Traffic	Enabled on...
VLANs and Tunnels	Selected: /Common external Available: /Common HA, http-tunnel, internal, socks-tunnel
Source Address Translation	None

**Acceleration**

Rate Class	None
SPDY Profile	None

**Resources**

iRules	Enabled Available: /Common CallIDUIE, HTTPTCPTablePersist, TIPSSecure, sys_APM_ExchangeSupport_OA_Bas
--------	--

**Move external VLAN to Selected**

**Leave this empty**

Figure 26 – Adding 'CPPM-network' IP Forwarder VS

Leave all additional settings after iRules in the Resources section at default. Click “Finished” at the bottom left of the action plane when the above steps are completed. You have now configured this Virtual Server.

## COA-RFC3576\_forwarder IP Forwarder (VS)

This IP Forwarder is used to forward COA traffic from any CPPM server to any NAD in the network requiring COA. This forwarder has two functions:



1. Forward the COA traffic to the NAD
2. Change the source address of the COA packet destined for the NAD to be that of the Virtual Server providing the RADIUS service. The NAD must have the RFC3576 server configured with the same IP address as the RADIUS server.

Before we begin to add the IP Forwarder we first configure the SNAT pool that will be used to obscure the source address from any CPPM server to be that of the RADIUS Virtual Server. The RADIUS server used in this example is 10.2.101.242, the RFC3576 server used in this example is also 10.2.101.242.

### SNAT Pool for CoA

On the left hand side navigation plane navigate to **Local Traffic > Address Translation** and select **SNAT Pool List** from the options offered. Click on the “Create...” button found in the top right of the action plane. Complete the configuration screen as shown.

Local Traffic » Address Translation : SNAT Pool List » New SNAT Pool...

General Properties

Name: RadiusSnatPool

Configuration

Enter IP address of RADIUS Server

IP Address: 10.2.101.242

Add

Member List

10.2.101.242

Edit Delete

Cancel Repeat Finished

Navigate to **Local Traffic > Address Translation** and select **SNAT Translation List** from the options offered. Look in the Action plane and confirm that the SNAT Translation List has the RADIUS IP address as an entry and that its state is “Enabled” – see below.

Local Traffic » Address Translation : SNAT Translation List

SNAT List | SNAT Pool List | SNAT Translation List | NAT List | Statistics

Create...

State	Name	Translation Address	Application	IP Address	Partition / Path
Enabled	10.2.101.242	10.2.101.242		10.2.101.242	Common

Delete...

Confirm State = Enabled

Figure 27 – Defining the CoA Source address on the F5 BIG-IP LTM

## Adding the IP Forwarder (VS)

On the left hand side navigation plane navigate to **Local Traffic** and select **Virtual Servers** from the options offered. Click on the “Create...” button found in the top right of the action plane. Proceed to populate the configuration screen as shown in the screen shot below:

**Local Traffic >> Virtual Servers : Virtual Server List >> New Virtual Server...**

**General Properties**

Name	COA-RFC3576_forwarder
Description	COA-RFC3576 with CPPM Source NAT
Type	Forwarding (IP)
Source	0.0.0.0/0
Destination	Type: <input type="radio"/> Host <input checked="" type="radio"/> Network Address: 0.0.0.0 Mask: 0.0.0.0
Service Port	3799 (Other: )
Notify Status to Virtual Address	<input checked="" type="checkbox"/>
State	Enabled

**Configuration:** Basic

Protocol	UDP
Protocol Profile (Client)	fastL4
VLAN and Tunnel Traffic	Enabled on...
VLANs and Tunnels	Selected: /Common internal Available: /Common HA, external, http-tunnel, socks-tunnel
Source Address Translation	SNAT
SNAT Pool	RadiusSnatPool

**Acceleration**

Rate Class	None
SPDY Profile	None

**Resources**

iRules	Enabled: (empty) Available: /Common CallIDUIE, HTTPTCPTablePersist, TIPSSecure, _sys_APM_ExchangeSupport_OA_BasicAuth
--------	--

Buttons: Cancel, Repeat, **Finished**

**Note:** This IP forwarder is applied to the VLAN from which traffic originates. CoA traffic egressing the CPPM servers on UDP port 3799 is processed by the “**COA-RFC3576\_forwarder**” instead of the “**any-network**” IP forwarder as the former is a more specific match.

The “**COA-RFC3576\_forwarder**” applies the RadiusSnatPool address translation before forwarding this packet to the NAD, the result is that the NAS receives a CoA packet from its configured RADIUS/RFC3576 server IP and not from the real address of the CPPM server issuing the CoA.

## Standard Virtual Servers

The following section deals with the configuration of “Standard Virtual Servers” these VS’s are responsible for delivering the RADIUS service to the NAS and the HTTP/S services to the Client.

### RADIUS Virtual Server

The RADIUS virtual server is configured as a single service listener listening on all UDP ports. It is configured as a single IP address, load balancing RADIUS requests from any NAS to nodes defined in the “**CPPM-RADIUS\_pool**” pool. This particular virtual server is equipped with a persistence profile, the functionality of which is delivered by a specific iRule. The iRule has two functions:

1. Process only UDP 1812 and 1813 packets
2. Ensure that 1812 and 1813 packets from the same NAS and with the same ***Radius:IETF:Calling-Station-Id*** are sent and persist to the same CPPM server in the pool.

As the virtual server is open to listening on any UDP port, we leave it up to the iRule to look at the incoming request and process it only if the request is on UDP ports 1812 or 1813. In this way we have created a **single virtual server** to address **both** UDP 1812 and 1813 incoming requests. The configuration of individual virtual servers for each protocol port does not guarantee persistence of RADIUS Authentication and Accounting to the same pool member for a specific ***Radius:IETF:Calling-Station-Id***.

Before we begin to configure the RADIUS virtual server, we need to build the F5 iRule and create a custom persistence profile to action the F5 iRule. Lets start with the F5 iRule:

#### ***Persistence iRule***

An F5 iRule is a TCL script that configures the F5 BIG-IP LTM to perform a customized function. In our case we are using this F5 iRule to manipulate the flow of RADIUS traffic such that we only action interesting traffic (UDP 1812 & 1813) and we persist both RADIUS auth and RADIUS acct traffic originating from a specific Calling-Station-Id to the same CPPM server.

To implement the first action, we simply tell the F5 BIG-IP LTM to drop any packet that is not UDP 1812 or 1813. This statement forces the F5 BIG-IP LTM to accept only UDP 1812 & 1813 packets. The second action requires the F5 BIG-IP LTM to look inside the RADIUS request and search for the “*Calling-Station-Id*” string (avp31). We chose to read this variable as a string to remove the dependence on delimiters – this way we can address any value of the “*Calling-Station-Id*” variable regardless of how this variable is presented. The “*Calling-Station-Id*” string is then loaded into a table and kept for a period of 900 seconds (15 minutes). The F5 BIG-IP LTM will then pass the packet to the pool for load balancing. The load balancing algorithm will forward the request to the selected pool member.

### *iRule: CallIDUIE*

The F5 iRule created is named CallIDUIE. The following section lists the F5 iRule and provides an explanation as to its operation. You will notice a number of **“log”** statements within the F5 iRule itself. These are only used to track the operation of the F5 iRule and do not affect the operation of the F5 iRule in any way. Please ensure that these are removed in a high production environment or we will slow the system down with excessive disk writes. In extreme cases, we can fill up the disk and cause a crash.

The output from the **“log”** statements can be viewed by issuing the following command at the shell:

```
config # tail -f /var/log/ltm
```

```
when CLIENT_ACCEPTED {
  if { ([UDP::local_port] != 1812) && ([UDP::local_port] != 1813) } {
    log local0. "packet on port [UDP::local_port] dropped"
    drop
  } else {
    set CALLID [RADIUS::avp 31 string]
    persist uie $CALLID
    log local0. "persisted $CALLID"
  }
}

when CLIENT_DATA {
  if { [UDP::local_port] == 1813 } {
    set CALLID [RADIUS::avp 31 string]
    set IP [RADIUS::avp 8 ip4]
    if { $IP != "" } {
      table set $IP [LB::server addr] 900
      log local0. "Radius maps $IP to [LB::server addr] for $CALLID"
    }
  }
}

when LB_SELECTED {
  log local0. "Selected [LB::server addr] [LB::server port]"
}

when SERVER_DATA {
  persist add uie $CALLID
  log local0. "persist added for $CALLID to [LB::server addr]"
}
```

**Figure 28 – F5 iRule – Tracking, forwarding and persisting traffic for ports 1812/1813**



Before we begin explaining the operation of the F5 iRule, we must first address a few RFC and F5 specific terminologies used in this F5 iRule:

- avp 31:** is the RADIUS Attribute Value Pair as defined in RFC2865. Value **31** references the ***Calling-Station-Id*** attribute
- avp 8:** is the RADIUS Attribute Value Pair as defined in RFC2865. Value **8** references the ***Framed-IP-Address*** attribute
- uie:** is the F5 BIG-IP LTM Universal Inspection Engine (UIE) used in the implementation of universal persistence. The UIE is a set of functions that allows you to observe, direct, and persist load-balanced traffic using F5 iRules. The UIE has the ability to inspect content data within a packet in both the client request and server response.

### **F5 iRule Explanation:**

The first part of the F5 iRule begins with “*when CLIENT\_ACCEPTED {*”, this F5 iRule is triggered when a client side connection is accepted. The F5 iRule then checks to see if the client side traffic presented is **not** destined for UDP port 1812 or UDP port 1813. The packet is dropped if a match occurs. This check ensures that the only traffic type serviced is UDP 1812 or UDP 1813. Once the packet has been identified as a packet of interest, the F5 iRule looks for the Calling-Station-Id attribute (avp 31) and assigns this attribute to the CALLID variable. The Universal Inspection Engine then uses the value of the CALLID variable in its persistence table.

The second part of the F5 iRule beginning with “*when CLIENT\_DATA {*”, looks for RADIUS accounting packets that are from the same Calling-Station-Id, captures their Framed-IP-Address attribute and increases the table entry age timer from the default of 180 seconds to 900 seconds.

The third part of the F5 iRule beginning with “*when LB\_SELECTED {*”, serves to log the IP address and UDP port of the CPPM server selected for this transaction. It is informational only.

The final part of the F5 iRule beginning with “*when SERVER\_DATA {*”, looks at the return traffic from the server and adds The CALLID value to the persistence table if first seen in RESPONSE events.

### **Adding the F5 iRule**

On the left hand side navigation plane navigate to **Local Traffic > iRules** and select **iRule List** from the set of options provided. Click on the “Create...” button found in the top right of the action plane. The following configuration screen will appear. Proceed to populate the configuration screen as shown in the screen shot below:



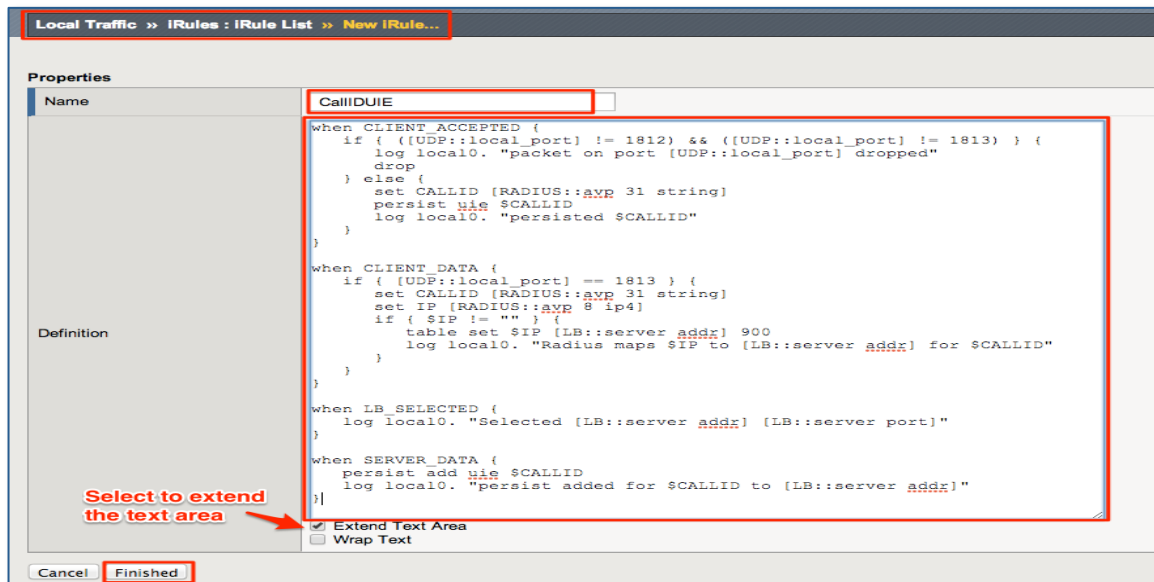


Figure 29 - Pasting F5 iRule into F5 BIG-IP LTM

The F5 iRule's has been provided previously for you to copy and paste into the "Definition" text box as shown above. The F5 iRule is purposely presented in Courier font in an effort to remove unwanted formatting. It is good practice to copy the content into a text-only editor to remove any formatting and copy from that before pasting into the F5 BIG-IP LTM text box. Don't forget to click on Finished to complete the addition of the F5 iRule.

### Adding the Custom Persistence Profile

On the left hand side navigation plane navigate to **Local Traffic > Profiles** and select **Persistence** from the set of options provided. Click on the "Create..." button found in the top right of the action plane. The following configuration screen will appear. Proceed to populate the configuration screen as shown in the screen shot below:

Local Traffic » Profiles : Persistence » New Persistence Profile...

**General Properties**

Name	CallID
Persistence Type	Universal
Parent Profile	universal

Select this checkbox to access options

**Configuration** Custom

Mirror Persistence	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Match Across Services	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Match Across Virtual Servers	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Match Across Pools	<input type="checkbox"/>	<input checked="" type="checkbox"/>
iRule	/Common/CallIDUIE	<input checked="" type="checkbox"/>
Timeout	Specify... 900 seconds	<input checked="" type="checkbox"/>
Override Connection Limit	<input type="checkbox"/>	<input checked="" type="checkbox"/>

Cancel Repeat **Finished**

Click on Finished to complete the addition of the custom persistence profile.

Now that we have configured the F5 iRule and the Custom Persistence Profile, we have to add a UDP protocol profile to use in the RADIUS Virtual Server build. This is not a special profile, it is a copy of the parent profile – we must do this, as the use of default profiles is not best practice.

### *Adding the CPPM-RADIUS custom UDP Profile*

On the left hand side navigation plane navigate to **Local Traffic > Profiles > Protocol** and select **UDP** from the set of options provided. Click on the “Create...” button found in the top right of the action plane. The following configuration screen will appear. Proceed to populate the configuration screen as shown in the screen shot below:

The screenshot shows the configuration interface for a new UDP profile. The breadcrumb navigation at the top is "Local Traffic » Profiles : Protocol : UDP » New UDP Profile...". The "General Properties" section contains a "Name" field with the value "CPPM-RADIUS" and a "Parent Profile" dropdown menu set to "udp". The "Settings" section includes a "Custom" checkbox and several configuration options: "Proxy Maximum Segment" (checkbox), "Idle Timeout" (Specify... dropdown, 60 seconds), "IP ToS" (Specify... dropdown, 0), "Link QoS" (Specify... dropdown, 0), "Datagram LB" (checkbox), and "Allow No Payload" (checkbox). At the bottom, there are three buttons: "Cancel", "Repeat", and "Finished".

General Properties	
Name	CPPM-RADIUS
Parent Profile	udp

Settings		Custom <input type="checkbox"/>
Proxy Maximum Segment	<input type="checkbox"/>	<input type="checkbox"/>
Idle Timeout	Specify... 60 seconds	<input type="checkbox"/>
IP ToS	Specify... 0	<input type="checkbox"/>
Link QoS	Specify... 0	<input type="checkbox"/>
Datagram LB	<input type="checkbox"/>	<input type="checkbox"/>
Allow No Payload	<input type="checkbox"/>	<input type="checkbox"/>

Buttons: Cancel Repeat **Finished**

**Figure 30 - Adding new a UDP profile, copied from a parent 'UDP' profile**

Click on Finished to complete the addition of the UDP profile

You are finally ready to build the RADIUS Virtual Server.

### Adding the CPPM-RADIUS Virtual Server

On the left hand side navigation plane navigate to **Local Traffic** and select **Virtual Servers** from the options offered. Click on the “Create...” button found in the top right of the action plane. The following configuration screen will appear. Proceed to populate the configuration screen as shown in the screen shots below:

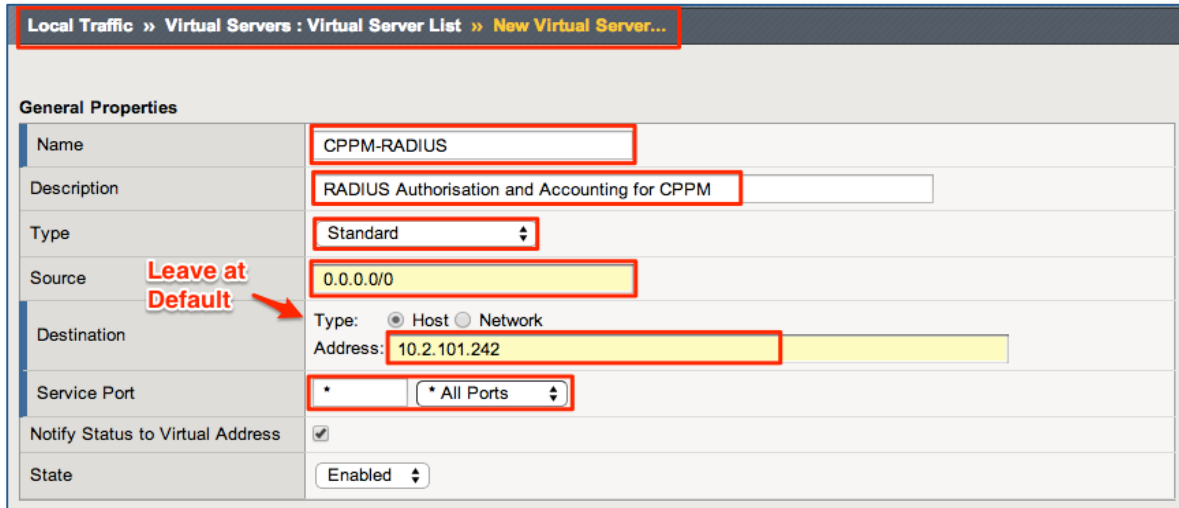


Figure 31 - Adding the RADIUS VS

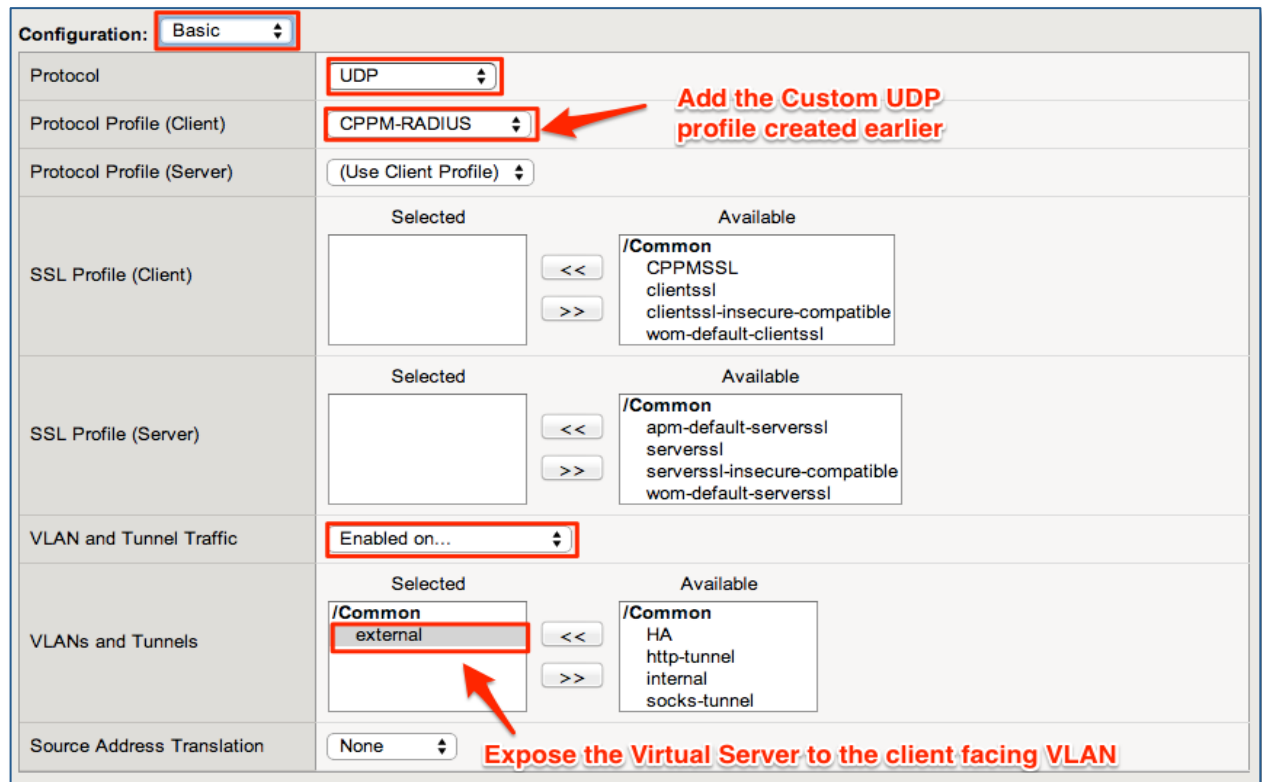


Figure 32 – additional configuration for the RADIUS VS

The screenshot displays the configuration interface for Content Rewrite. It includes sections for Content Rewrite (Rewrite Profile: None, HTML Profile: None), Acceleration (Rate Class: None), and Resources. The Resources section contains an iRules list (currently empty, with a note "No iRules added"), a Default Pool dropdown set to "CPPM-RADIUS\_pool", a Default Persistence Profile dropdown set to "CallID", and a Fallback Persistence Profile dropdown set to "None". Red arrows and text annotations highlight these specific configurations.


Figure 33 – additional configuration for the RADIUS VS

Click on Finished to complete the addition of the CPPM-RADIUS Virtual Server

**Note:** The above screen shots are components of the configuration screen used to config the CPPM-RADIUS virtual server

## HTTP Virtual Server

The HTTP virtual server is configured as a single service listener listening on TCP port 80. It is configured as a single IP address, load balancing HTTP requests from any Client to nodes defined in the “**WEB\_AUTH\_CPPM\_pool**” pool. This particular virtual server is equipped with a custom persistence profile, the functionality of which is based on Source Address Affinity persistence. The fact that we created this persistence profile gives us the option to refine the client persistence behavior for this virtual server in the future if needed.

 In addition to the persistence profile, this virtual server has two *TCP Protocol profiles*, one *HTTP profile* and a F5 “*OneConnect™*” connection pooling profile. F5 BIG-IP LTM F5’s “*OneConnect*” executes a HTTP connection to the CPPM servers and uses this single connection for many clients serviced by the F5 BIG-IP LTM. This is beneficial to ClearPass as it greatly reduces the HTTP connect load on CPPM mitigating the possibility of CPPM reaching Apache connection pool exhaustion. The following sections show how to configure the optimizers, HTTP profile and F5 OneConnect.

We configure a *tcp-wan-optimized* for client traffic and a *tcp-lan-optimized profile* for server traffic. In our best practice model, we use the WAN optimized profile to handle

client traffic. This is done, as we are unaware of the origin of client traffic and to account for client traffic over low bandwidth or high latency links. We use the LAN optimized profile to handle server traffic as we expect the servers to be in the Data Centre and directly behind the F5 BIG-IP LTM where there is very low latency and high bandwidth.

### About *tcp-wan-optimized* Profile Settings

The **tcp-wan-optimized** profile we will be creating is based on a pre-configured profile type that will be associated with the virtual server. In cases where the F5 BIG-IP LTM system is load balancing traffic over a WAN link, we enhance the performance of our wide-area TCP traffic by using the **tcp-wan-optimized** profile.

If the traffic profile is strictly WAN-based, and a standard virtual server with a TCP profile is required, you can configure your virtual server to use a **tcp-wan-optimized** profile to enhance WAN-based traffic. For example, in many cases, the client connects to the F5 BIG-IP LTM virtual server over a WAN link, which is generally slower than the connection between the F5 BIG-IP LTM system and the pool member servers. By configuring our virtual server to use **the tcp-wan-optimized** profile, the F5 BIG-IP LTM system can accept the data more quickly, allowing resources on the pool member servers to remain available. Also, use of this profile can increase the amount of data that the F5 BIG-IP LTM system buffers while waiting for a remote client to accept that data. Finally, we increase network throughput by reducing the number of short TCP segments that the F5 BIG-IP LTM system sends on the network.

A **tcp-wan-optimized** profile is similar to a TCP profile, except that the default values of certain settings vary, in order to optimize the system for WAN-originated traffic.

As best practice, we create a unique, customised **tcp-wan-optimized** profile specifying the **tcp-wan-optimized** profile as the parent profile.

### Adding the “*tcp-wan-optimized*” Profile

On the left hand side navigation plane navigate to **Local Traffic > Profiles > Protocol** and select **TCP** from the set of options provided. Click on the “Create...” button found in the top right of the action plane. The following configuration screen will appear. Proceed to populate the configuration screen as shown in the screen shot below:

Local Traffic > Profiles : Protocol : TCP > New TCP Profile...

Add the name:  
**WEB\_AUTH-CPPM\_tcp-wan-optimized**

General Properties

Name	WEB_AUTH-CPPM
Parent Profile	tcp-wan-optimized

Select the “*tcp-wan-optimized*” parent profile from the “drop-down” box. Leave all other settings at their defaults and select “Finished” at the bottom of the page to complete the addition of this profile.

These defaults should be as shown in the screen capture below:

Settings		Custom <input type="checkbox"/>
Reset On Timeout	<input checked="" type="checkbox"/> Enabled	<input type="checkbox"/>
Time Wait Recycle	<input checked="" type="checkbox"/> Enabled	<input type="checkbox"/>
Delayed Acks	<input checked="" type="checkbox"/> Enabled	<input type="checkbox"/>
Proxy Maximum Segment	<input type="checkbox"/>	<input type="checkbox"/>
Proxy Options	<input type="checkbox"/>	<input type="checkbox"/>
Proxy Buffer Low	131072 bytes	<input type="checkbox"/>
Proxy Buffer High	131072 bytes	<input type="checkbox"/>
Idle Timeout	Specify... 300 seconds	<input type="checkbox"/>
Zero Window Timeout	Specify... 20000 milliseconds	<input type="checkbox"/>
Time Wait	Specify... 2000 milliseconds	<input type="checkbox"/>
Fin Wait	Specify... 5 seconds	<input type="checkbox"/>
Close Wait	Specify... 5 seconds	<input type="checkbox"/>
Send Buffer	65535 bytes	<input type="checkbox"/>
Receive Window	65535 bytes	<input type="checkbox"/>
Keep Alive Interval	Specify... 1800 seconds	<input type="checkbox"/>
Maximum Syn Retransmissions	3	<input type="checkbox"/>
Maximum Segment Retransmissions	8	<input type="checkbox"/>
IP ToS	Specify... 0	<input type="checkbox"/>
Link QoS	Specify... 0	<input type="checkbox"/>
Selective ACKs	<input checked="" type="checkbox"/> Enabled	<input type="checkbox"/>
Selective NACK	<input type="checkbox"/>	<input type="checkbox"/>
Explicit Congestion Notification	<input type="checkbox"/>	<input type="checkbox"/>
Timestamps Extension for High Performance (RFC 1323)	<input checked="" type="checkbox"/> Enabled	<input type="checkbox"/>
Limited Transmit Recovery	<input checked="" type="checkbox"/> Enabled	<input type="checkbox"/>
Slow Start	<input checked="" type="checkbox"/> Enabled	<input type="checkbox"/>
Deferred Accept	<input type="checkbox"/>	<input type="checkbox"/>
Verified Accept	<input type="checkbox"/>	<input type="checkbox"/>
Nagle's Algorithm	<input checked="" type="checkbox"/> Enabled	<input type="checkbox"/>
Acknowledge on Push	<input checked="" type="checkbox"/> Enabled	<input type="checkbox"/>
MD5 Signature	<input type="checkbox"/>	<input type="checkbox"/>
MD5 Signature Passphrase		<input type="checkbox"/>

Ensure that the defaults set are as per this screen capture

Figure 34 - Adding a TCP WAN profile - part1

MD5 Signature Passphrase	<input type="text"/>	<input type="checkbox"/>
Congestion Control	High Speed	<input type="checkbox"/>
Congestion Metrics Cache	<input checked="" type="checkbox"/> Enabled	<input type="checkbox"/>
Appropriate Byte Counting (RFC 3465)	<input checked="" type="checkbox"/> Enabled	<input type="checkbox"/>
D-SACK (RFC 2883)	<input type="checkbox"/>	<input type="checkbox"/>
Packet Loss Ignore Rate	<input type="text" value="0"/>	<input type="checkbox"/>
Packet Loss Ignore Burst	<input type="text" value="0"/>	<input type="checkbox"/>
Initial Congestion Window Size	<input type="text" value="0"/>	<input type="checkbox"/>
Initial Receive Window Size	<input type="text" value="0"/>	<input type="checkbox"/>
Initial Retransmission Timeout Base Multiplier for SYN Retransmission	<input type="text" value="0"/> milliseconds	<input type="checkbox"/>
Delay Window Control	<input type="checkbox"/>	<input type="checkbox"/>
Hardware SYN Cookie Protection	<input checked="" type="checkbox"/> Enabled	<input type="checkbox"/>
Max Segment Size (MSS)	<input type="text" value="1460"/>	<input type="checkbox"/>
Rate Pace	<input type="checkbox"/>	<input type="checkbox"/>
Multipath TCP	<input type="checkbox"/>	<input type="checkbox"/>
MPTCP Checksum	<input type="checkbox"/>	<input type="checkbox"/>
MPTCP Checksum Verify	<input type="checkbox"/>	<input type="checkbox"/>
MPTCP Debug	<input type="checkbox"/>	<input type="checkbox"/>
MPTCP Fallback	reset	<input type="checkbox"/>
MPTCP Fast Join	<input type="checkbox"/>	<input type="checkbox"/>
MPTCP Join Max	<input type="text" value="5"/>	<input type="checkbox"/>
MTCP Make After Break	<input type="checkbox"/>	<input type="checkbox"/>
MPTCP No Join DSS ACK	<input type="checkbox"/>	<input type="checkbox"/>
MPTCP RTO Max	<input type="text" value="5"/>	<input type="checkbox"/>
MPTCP Retransmit Min	<input type="text" value="1000"/>	<input type="checkbox"/>
MPTCP Subflow Max	<input type="text" value="6"/>	<input type="checkbox"/>
MPTCP Timeout	<input type="text" value="3600"/>	<input type="checkbox"/>

Ensure that the defaults set are as per this screen capture

Cancel Repeat **Finished**

Figure 35 - Adding a TCP WAN profile - part2

**Note:** The above screen shots are components of the configuration screen used to configure the “*WEB\_AUTH-CPPM\_tcp-wan-optimized*” profile



### About *tcp-lan-optimized* profile settings

The **tcp-lan-optimized** profile we will be creating is based on a pre-configured profile type that will be associated with the virtual server. In cases where the F5 BIG-IP LTM virtual server is load balancing LAN-based or interactive traffic, we enhance the performance of our local-area TCP traffic to the CPPM servers by using the **tcp-lan-optimized** profile.

If the traffic profile is strictly LAN-based, or highly interactive, and a standard virtual server with a TCP profile is required, you can configure the virtual server to use the **tcp-lan-optimized** profile to enhance LAN-based or interactive traffic. For example, applications producing an interactive TCP data flow, such as SSH and TELNET, normally generate a TCP packet for each keystroke. A TCP profile setting such as **Slow Start** can introduce latency when this type of traffic is being processed. By configuring the virtual server to use the **tcp-lan-optimized** profile, we can ensure that the F5 BIG-IP LTM system delivers LAN-based or interactive traffic without delay.

A **tcp-lan-optimized** profile is similar to a TCP profile, except that the default values of certain settings vary, in order to optimize the system for LAN-based traffic.

As best practice, we create a unique, customised **tcp-lan-optimized** profile specifying the **tcp-lan-optimized** profile as the parent profile.

### Adding the “*tcp-lan-optimized*” Profile

On the left hand side navigation plane navigate to **Local Traffic > Profiles > Protocol** and select **TCP** from the set of options provided. Click on the “Create...” button found in the top right of the action plane. The following configuration screen will appear. Proceed to populate the configuration screen as shown in the screen shot below:

Local Traffic >> Profiles : Protocol : TCP >> New TCP Profile...

Add the name:  
**WEB\_AUTH-CPPM\_tcp-lan-optimized**

General Properties

Name	WEB_AUTH-CPPM
Parent Profile	tcp-lan-optimized

Select the “*tcp-lan-optimized*” parent profile from the “drop-down” box. Leave all other settings at their defaults and select “Finished” at the bottom of the page to complete the addition of this profile.

These defaults should be as shown in the screen capture below:

Settings		Custom <input type="checkbox"/>
Reset On Timeout	<input checked="" type="checkbox"/> Enabled	<input type="checkbox"/>
Time Wait Recycle	<input checked="" type="checkbox"/> Enabled	<input type="checkbox"/>
Delayed Acks	<input checked="" type="checkbox"/> Enabled	<input type="checkbox"/>
Proxy Maximum Segment	<input type="checkbox"/>	<input type="checkbox"/>
Proxy Options	<input type="checkbox"/>	<input type="checkbox"/>
Proxy Buffer Low	98304 bytes	<input type="checkbox"/>
Proxy Buffer High	131072 bytes	<input type="checkbox"/>
Idle Timeout	Specify... 300 seconds	<input type="checkbox"/>
Zero Window Timeout	Specify... 20000 milliseconds	<input type="checkbox"/>
Time Wait	Specify... 2000 milliseconds	<input type="checkbox"/>
Fin Wait	Specify... 5 seconds	<input type="checkbox"/>
Close Wait	Specify... 5 seconds	<input type="checkbox"/>
Send Buffer	65535 bytes	<input type="checkbox"/>
Receive Window	65535 bytes	<input type="checkbox"/>
Keep Alive Interval	Specify... 1800 seconds	<input type="checkbox"/>
Maximum Syn Retransmissions	3	<input type="checkbox"/>
Maximum Segment Retransmissions	8	<input type="checkbox"/>
IP ToS	Specify... 0	<input type="checkbox"/>
Link QoS	Specify... 0	<input type="checkbox"/>
Selective ACKs	<input checked="" type="checkbox"/> Enabled	<input type="checkbox"/>
Selective NACK	<input type="checkbox"/>	<input type="checkbox"/>
Explicit Congestion Notification	<input type="checkbox"/>	<input type="checkbox"/>
Timestamps Extension for High Performance (RFC 1323)	<input checked="" type="checkbox"/> Enabled	<input type="checkbox"/>
Limited Transmit Recovery	<input checked="" type="checkbox"/> Enabled	<input type="checkbox"/>
Slow Start	<input type="checkbox"/>	<input type="checkbox"/>
Deferred Accept	<input type="checkbox"/>	<input type="checkbox"/>
Verified Accept	<input type="checkbox"/>	<input type="checkbox"/>
Nagle's Algorithm	<input type="checkbox"/>	<input type="checkbox"/>
Acknowledge on Push	<input checked="" type="checkbox"/> Enabled	<input type="checkbox"/>
MD5 Signature	<input type="checkbox"/>	<input type="checkbox"/>
MD5 Signature Passphrase		<input type="checkbox"/>

**Ensure that the defaults set are as per this screen capture**

Figure 36 - Adding a TCP LAN profile - part1

MD5 Signature Passphrase	<input type="text"/>	<input type="checkbox"/>
Congestion Control	High Speed	<input type="checkbox"/>
Congestion Metrics Cache	<input checked="" type="checkbox"/> Enabled	<input type="checkbox"/>
Appropriate Byte Counting (RFC 3465)	<input checked="" type="checkbox"/> Enabled	<input type="checkbox"/>
D-SACK (RFC 2883)	<input type="checkbox"/>	<input type="checkbox"/>
Packet Loss Ignore Rate	<input type="text" value="0"/>	<input type="checkbox"/>
Packet Loss Ignore Burst	<input type="text" value="0"/>	<input type="checkbox"/>
Initial Congestion Window Size	<input type="text" value="0"/>	<input type="checkbox"/>
Initial Receive Window Size	<input type="text" value="0"/>	<input type="checkbox"/>
Initial Retransmission Timeout Base Multiplier for SYN Retransmission	<input type="text" value="0"/> milliseconds	<input type="checkbox"/>
Delay Window Control	<input type="checkbox"/>	<input type="checkbox"/>
Hardware SYN Cookie Protection	<input checked="" type="checkbox"/> Enabled	<input type="checkbox"/>
Max Segment Size (MSS)	<input type="text" value="1460"/>	<input type="checkbox"/>
Rate Pace	<input type="checkbox"/>	<input type="checkbox"/>
Multipath TCP	<input type="checkbox"/>	<input type="checkbox"/>
MPTCP Checksum	<input type="checkbox"/>	<input type="checkbox"/>
MPTCP Checksum Verify	<input type="checkbox"/>	<input type="checkbox"/>
MPTCP Debug	<input type="checkbox"/>	<input type="checkbox"/>
MPTCP Fallback	reset	<input type="checkbox"/>
MPTCP Fast Join	<input type="checkbox"/>	<input type="checkbox"/>
MPTCP Join Max	<input type="text" value="5"/>	<input type="checkbox"/>
MTCP Make After Break	<input type="checkbox"/>	<input type="checkbox"/>
MPTCP No Join DSS ACK	<input type="checkbox"/>	<input type="checkbox"/>
MPTCP RTO Max	<input type="text" value="5"/>	<input type="checkbox"/>
MPTCP Retransmit Min	<input type="text" value="1000"/>	<input type="checkbox"/>
MPTCP Subflow Max	<input type="text" value="6"/>	<input type="checkbox"/>
MPTCP Timeout	<input type="text" value="3600"/>	<input type="checkbox"/>

**Ensure that the defaults set are as per this screen capture**

Cancel Repeat **Finished**

Figure 37 - Adding a TCP LAN profile - part2

**Note:** The above screen shots are components of the configuration screen used to configure the “*WEB\_AUTH-CPPM\_tcp-wan-optimized*” profile

## About HTTP Profile Settings

F5 BIG-IP LTM offers several features that we can use to intelligently control our application layer traffic. Examples of these features are the insertion of headers into HTTP requests and the compression of HTTP server responses.

These features are available through various configuration profiles. A profile is a group of settings, with values, that correspond to HTTP traffic defining the way that you want the F5 BIG-IP LTM system to manage HTTP traffic.

To manage HTTP traffic, we can use any of these profile types:

- HTTP (Hypertext Transfer Protocol)
- HTTP Compression
- Web Acceleration

In addition to these profiles, F5 BIG-IP LTM includes other features to help manage our application traffic, such as health monitors for checking the health of HTTP and HTTPS services, and F5 iRules for querying or manipulating header or content data.

### HTTP profile type

We configure a HTTP profile to ensure that HTTP traffic management suits our specific needs. You can configure the profile settings when you create a profile, or after profile creation by modifying the profile's settings.

For these profile settings, we can specify values where none exist, or modify any default values to suit our needs.

We can also use the default HTTP profile, named `http`, as is, if you do not want to create a custom HTTP profile, however, this profile may be used to service many Virtual Servers. In the interest of adhering to best practice, we create a copy of the `http` profile and call it `WEB_AUTH-CPPM_http` but using the `http` parent profile settings. This way we have a unique HTTP profile for use with CPPM WEB\_AUTH hence any changes made will only impact this profile and not the parent.

**Note:** The `http` profile is considered a default profile because it does not inherit setting values from a parent profile.

### Adding the "HTTP" Profile

On the left hand side navigation plane navigate to **Local Traffic > Profiles > Protocol** and select **TCP** from the set of options provided. Click on the "Create..." button found in the top right of the action plane. The following configuration screen will appear. Proceed to populate the configuration screen as shown in the screen shot below:

**Local Traffic » Profiles : Services : HTTP » New HTTP Profile...**

**General Properties**

Name	WEB_AUTH-CPPM <span style="color: red;">← Add the name: WEB_AUTH-CPPM_http</span>
Proxy Mode	Reverse
Parent Profile	http

**Settings** Custom

Basic Auth Realm		<input type="checkbox"/>
Fallback Host		<input type="checkbox"/>
Fallback on Error Codes		<input type="checkbox"/>
Request Header Erase		<input type="checkbox"/>
Request Header Insert		<input type="checkbox"/>
Response Headers Allowed		<input type="checkbox"/>
Request Chunking	Preserve	<input type="checkbox"/>
Response Chunking	Selective	<input type="checkbox"/>
OneConnect Transformations	<input checked="" type="checkbox"/> Enabled	<input type="checkbox"/>
Redirect Rewrite	None	<input type="checkbox"/>
Encrypt Cookies		<input type="checkbox"/>
Cookie Encryption Passphrase	*****	<input type="checkbox"/>
Confirm Cookie Encryption Passphrase	*****	<input type="checkbox"/>
Insert X-Forwarded-For	Disabled	<input type="checkbox"/>
LWS Maximum Columns	80	<input type="checkbox"/>
LWS Separator		<input type="checkbox"/>
Maximum Requests	0	<input type="checkbox"/>
Send Proxy Via Header In Request	Preserve	<input type="checkbox"/>
Send Proxy Via Header In Response	Preserve	<input type="checkbox"/>
Accept XFF	<input type="checkbox"/>	<input type="checkbox"/>
XFF Alternative Names		<input type="checkbox"/>
Server Agent Name	BigIP	<input type="checkbox"/>

**Enforcement**

Allow Truncated Redirect	Disabled	<input type="checkbox"/>
Maximum Header Size	32768 bytes	<input type="checkbox"/>
Maximum Header Count	64	<input type="checkbox"/>
Pipeline Action	Allow	<input type="checkbox"/>
Unknown Method	Allow	<input type="checkbox"/>

**sFlow**

Polling Interval	Default	<input type="checkbox"/>
Sampling Rate	Default	<input type="checkbox"/>

Cancel Repeat **Finished**

Figure 38 - Adding a HTTP profile

**Note:** Leave the Settings, Enforcement and sFlow at their default settings as shown in the screen capture above - remember to click on **“Finished”** to complete the addition of this profile.

## About F5 OneConnect profiles

The F5 OneConnect profile type implements the F5 BIG-IP LTM OneConnect feature. This feature can increase network throughput by efficiently managing connections created between the F5 BIG-IP LTM system and back-end pool members. The F5 OneConnect feature can be used with any TCP-based protocol, such as HTTP or RTSP.

### How does F5 OneConnect work?

The F5 **OneConnect** feature works with request headers to keep the existing server-side connections open and available for reuse by other clients. When a client makes a new connection to a virtual server configured with a F5 OneConnect profile, the F5 BIG-IP LTM system parses the request, selects a server using the load-balancing method defined in the pool, and creates a connection to that server. When the client's initial request is complete, the F5 BIG-IP LTM system temporarily holds the connection open and makes the idle TCP connection to the pool member available for reuse.

When another connection is subsequently initiated to the virtual server, if an existing server-side flow to the pool member is open and idle, the F5 BIG-IP LTM system applies the F5 OneConnect source mask to the IP address in the request to determine whether the request is eligible to reuse the existing idle connection. If the request is eligible, the F5 BIG-IP LTM system marks the connection as non-idle and sends a client request over that connection. If the request is not eligible for reuse, or an idle server-side flow is not found, the F5 BIG-IP LTM system creates a new server-side TCP connection and sends client requests over the new connection.

### A Note About Client Source IP addresses:



The standard address translation mechanism on the F5 BIG-IP LTM system translates only the destination IP address in a request and not the source IP address (that is, the client node's IP address). However, when the F5 OneConnect feature is enabled, allowing multiple client nodes to re-use a server-side connection, **the source IP address in the header of each client node's request is always the IP address of the client node that initially opened the server-side connection.** of server system output.

### Adding the F5 OneConnect Profile

On the left hand side navigation plane navigate to **Local Traffic > Profiles > Other** and select **OneConnect** from the set of options provided. Click on the "Create..." button found in the top right of the action plane. The following configuration screen will appear. Proceed to populate the configuration screen as shown in the screen shot below:

Local Traffic >> Profiles : Other : OneConnect >> New OneConnect Profile...

**Add the name:  
WEB\_AUTH-CPPM\_oneconnect**

**General Properties**

Name	WEB_AUTH-CPPM
Parent Profile	oneconnect

**Settings** Custom

Source Mask	0.0.0.0	<input type="checkbox"/>
Maximum Size	10000 connections	<input type="checkbox"/>
Maximum Age	86400 seconds	<input type="checkbox"/>
Maximum Reuse	1000	<input type="checkbox"/>
Idle Timeout Override	Disabled	<input type="checkbox"/>

Cancel Repeat **Finished**

**Figure 39 - Adding a F5 'OneConnect' profile**

Remember to click on “**Finished**” to complete the addition of the OneConnect profile.

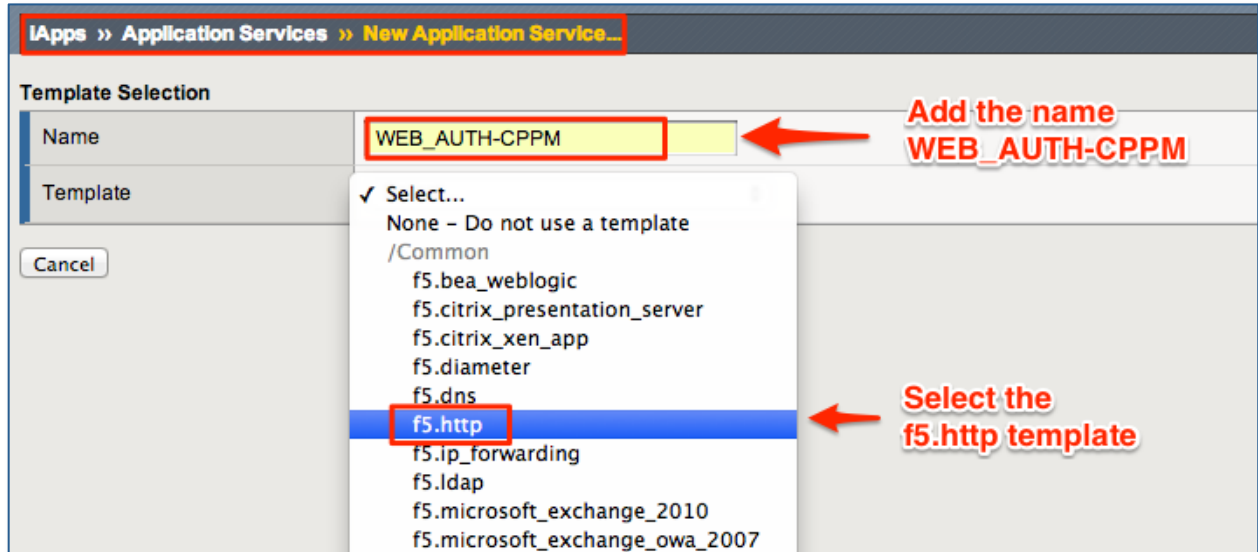
**Note on the F5 OneConnect profile:**

We have left the Source Mask setting at the default of 0.0.0.0, this means that the session created by one source address will be available for use by many. However, the server will see all flows using this open connection as coming from the source address of the device responsible for the initial connection. We may need to tune this parameter to only re-use this open connection for the device that originated the connection – this has not been tested as we did not have the scale of devices to perform this testing and will constitute one of the subjects planned for an updated version of this Tech Note.

Finally we can commence the addition of the HTTP Virtual Server now that we have added all of the required subcomponents. We will make use of the F5 BIG-IP LTM iApps function using the profiles, pools and nodes we have configured to deploy the HTTP Virtual Server

### Adding the WEB\_AUTH-CPPM vs Virtual Server:

On the left hand side navigation plane, click on **iApps** and then on **Application Services**. Click on the “Create...” button found in the top right of the action plane. The following configuration screen will appear. Proceed to populate the configuration screen as shown in the screen shot below:



This will bring up the f5.http template. The “**Template Selection:**” should be set to “**Basic**” and from the template options, select “**Advanced – Configure advanced options**”. We select this option as this Virtual Server is built using the components configured earlier.

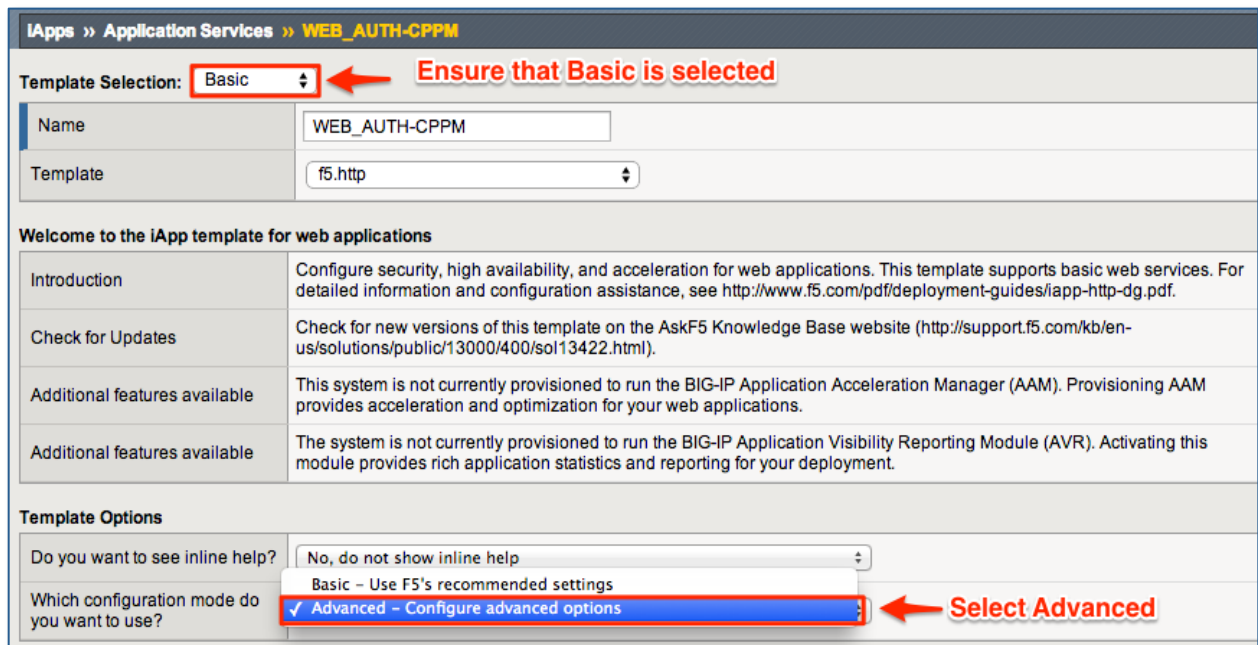


Figure 40 – Adding HTTP F5 iApps Application Service – part1



In the Network section on the same page ensure that the only VLAN where traffic is enabled is on the Client facing VLAN – in our case this VLAN is called “**external**”. By default, this template places all VLANs in the “**Selected**” box, we need to move all unwanted VLANs from that “**Selected**” box to the “**Options**” box leaving the Client facing VLAN selected. In the same section, select the “**Servers have a route through the F5 BIG-IP LTM system**” option from the routing section. See below:

**Network**

What type of network connects clients to the BIG-IP system? Wide area network (WAN)

Do you want to restrict client traffic to specific VLANs? Yes, enable traffic only on the VLANs I specify

On which VLANs should traffic be enabled or disabled?

Selected: /Common external

Options: /Common HA, internal

What type of network connects servers to the BIG-IP system? Local area network (LAN)

Where will the virtual servers be in relation to the web servers? BIG-IP virtual server IP and web servers are on different subnets

How have you configured routing on your web servers?  Servers have a route to clients through the BIG-IP system

Figure 41 - Adding HTTP Application Service - part2

The SSL Encryption section should be set to “**Plaintext to and from both clients and servers**”, if not please select this option.

**SSL Encryption**

How should the BIG-IP system handle SSL traffic? Plaintext to and from both clients and servers

Figure 42 - Adding HTTP Application Service - part3

The next section configures the Virtual Server and Pools, here is where we make use of all the pre-configuration from the previous section. The notes below are self-explanatory.

**Virtual Server and Pools**

What IP address do you want to use for the virtual server? 10.2.101.241 **Add the IP address of the Virtual Server**

If using a network virtual address, what is the IP mask? **Leave blank as the Virtual Server is a host not a subnet**

What port do you want to use for the virtual server? 80

Do you want to enable connection and persistence mirroring? Do not enable connection/persistence mirroring

What FQDNs will clients use to access the servers? Host webauth.cppm-testing.com **Add FQDN**

Which HTTP profile do you want to use? WEB\_AUTH\_CPPM\_http **Select HTTP profile**

Which persistence profile do you want to use? SourceAddrPersist **Select Persistence Profile**

Do you want to create a new pool or use an existing one? WEB\_AUTH\_CPPM\_pool **Select Pool**

Figure 43 - Adding HTTP Application Service - part4

**Note:** You can add as many FQDNs as needed to represent the Virtual Server created. Remember that these FQDNs must be resolvable by DNS in your network.

The next section is to be configured as shown. Please disable caching and compression as both of these have not been tested and could produce unwanted behavior. Set the client-side connection optimizer to be the TCP optimizer configured earlier.

Delivery Optimization		
Which Web Acceleration profile do you want to use for caching?	Do not use caching	Do not use caching
Which compression profile do you want to use?	Do not compress HTTP responses	Do not use compression
How do you want to optimize client-side connections?	WEB_AUTH-CPPM_tcp-wan-optimized	Select TCP profile created

Figure 44 - Adding HTTP Application Service - part5

The next section deals with Server Offload. In this section we configure the behavior of the server side connections. The screen shot below should be self-explanatory.

Server Offload		
Which OneConnect profile do you want to use?	WEB_AUTH-CPPM_oneconnect	Select OneConnect profile
Which NTLM profile do you want to use?	Do not use NTLM (recommended)	
How do you want to optimize server-side connections?	WEB_AUTH-CPPM_tcp-lan-optimized	Select TCP profile created
Should the BIG-IP system queue TCP requests?	No, do not enable TCP request queuing (recommended)	
Use a Slow Ramp time for newly added servers?	Yes, use Slow Ramp (recommended)	
How many seconds should Slow Ramp time last?	300	

Figure 45 - Adding HTTP Application Service - part6

All other settings should be left at default – please confirm that the default settings are as seen in the screen capture above and if not, please set them to be as per the screen capture above.

The following section shows the F5 iRule and the Statistics and Logging configurations. There are no F5 iRules required for this service and Statistics and Logging is optional and depends on the customer. For our deployment Statistics and logging is not required. Refer to screen capture below:

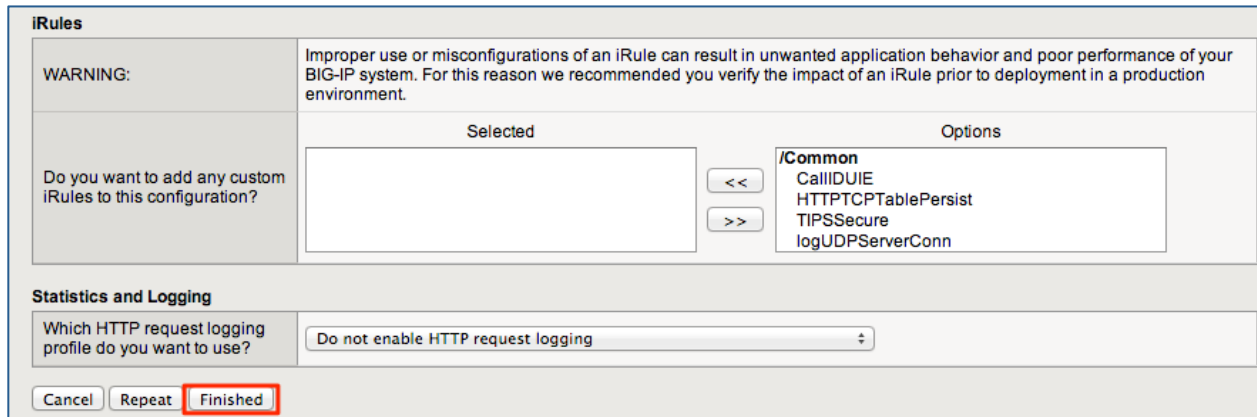


Figure 46 - Adding HTTP Application Service - part7

Click on “Finished” to complete the addition of the WEB\_AUTH-CPPM\_vs Virtual Server. The following Component listing should be visible once this template is executed. This can be viewed by navigating to **iApps > Application Services** and clicking on the WEB\_AUTH-CPPM iApps. This should bring up the Components view of the virtual server as shown below.

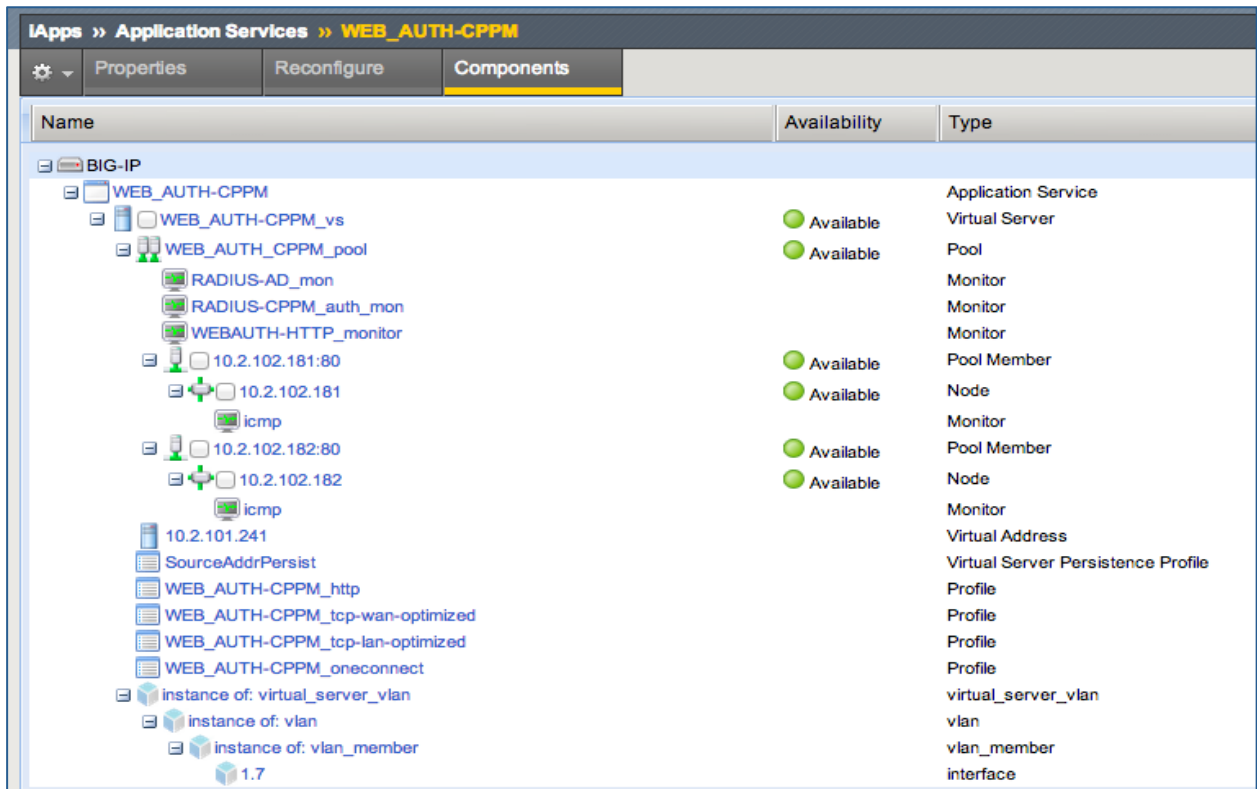


Figure 47 - Viewing the created F5 iApps HTTP Application Service

## HTTPS Virtual Server

The HTTPS virtual server is configured as a single service listener listening on TCP port 443. It is configured as a single IP address, load balancing HTTPS requests from any Client to nodes defined in the “**WEB\_AUTH\_CPPM\_https\_pool**” pool. This particular virtual server is equipped with a custom persistence profile, the functionality of which is based on Source Address Affinity persistence. The fact that we created this persistence profile gives us the option to refine the client persistence behavior for this virtual server in the future if needed.

In addition to the persistence profile, this virtual server has one *TCP protocol profile*, one *HTTP profile* and a F5 “*OneConnect*” connection pooling profile. F5’s “*OneConnect*” executes a HTTP connection to the CPPM servers and uses this single connection for many clients serviced by the F5 BIG-IP LTM. This is beneficial to ClearPass as it greatly reduces the HTTP connect load on CPPM mitigating the possibility of CPPM reaching Apache connection pool exhaustion. The following section shows how to configure the TCP protocol profile as the HTTP and F5 OneConnect profiles have been covered previously.

The TCP protocol profile created for this deployment is based on a Multi Path TCP mobile optimized tcp profile or “*mptcp-mobile-optimized*” profile.

### **About mptcp-mobile-optimized profile settings:**

The mptcp-mobile-optimized profile is a profile type for use in reverse proxy and enterprise environments for mobile applications that are front-ended by an F5 BIG-IP LTM. This profile uses newer congestion control algorithms and a newer TCP stack, and is generally better for files that are larger than 1 MB. Specific options in the pre-configured profile are set to optimize traffic for most mobile users in this environment, these settings can be tuned to accommodate our network.

**Note:** Although the pre-configured settings produced the best results in the test lab, network conditions are extremely variable. For the best results, start with the default settings and then experiment to find out what works best in your network.

The enabled Multipath TCP (MPTCP) option provides more bandwidth and higher network utilization. It allows multiple client-side flows to connect to a single server-side flow. MPTCP automatically and quickly adjusts to congestion in the network, moving traffic away from congested paths and toward uncongested paths.

The Congestion Control setting includes delay-based and hybrid algorithms, which may better address TCP performance issues better than fully loss-based congestion control algorithms in mobile environments.

The enabled Rate Pace option mitigates “bursty” behavior in mobile networks and other configurations. It can be useful on high latency or high BDP (bandwidth-delay product) links, where packet drop is likely to be a result of buffer overflow rather than congestion.

An mptcp-mobile-optimized profile is similar to a TCP profile, except that the default values of certain settings vary, in order to optimize the system for mobile traffic.

### Adding the “tcp-protocol” Profile

On the left hand side navigation plane navigate to **Local Traffic > Profiles > Protocol** and select **TCP** from the set of options provided. Click on the “Create...” button found in the top right of the action plane. The following configuration screen will appear. Proceed to populate the configuration screen as shown in the screen shot below:

**Local Traffic >> Profiles : Protocol : TCP >> New TCP Profile...**

**General Properties**

Name	WEB_AUTH-CPPM	← Add name: WEB_AUTH-CPPM_mptcp-mobile-optimized
Parent Profile	tcp	← Select tcp parent profile

**Settings** Custom

Reset On Timeout	<input checked="" type="checkbox"/> Enabled	<input checked="" type="checkbox"/>
Time Wait Recycle	<input checked="" type="checkbox"/> Enabled	<input checked="" type="checkbox"/>
Delayed Acks	<input checked="" type="checkbox"/> Enabled	<input checked="" type="checkbox"/>
Proxy Maximum Segment	<input type="checkbox"/>	<input type="checkbox"/>
Proxy Options	<input type="checkbox"/>	<input type="checkbox"/>
Proxy Buffer Low	131072 bytes	<input checked="" type="checkbox"/>
Proxy Buffer High	131072 bytes	<input checked="" type="checkbox"/>
Idle Timeout	Specify... 300 seconds	<input type="checkbox"/>
Zero Window Timeout	Specify... 20000 milliseconds	<input type="checkbox"/>
Time Wait	Specify... 2000 milliseconds	<input type="checkbox"/>
Fin Wait	Specify... 5 seconds	<input type="checkbox"/>
Close Wait	Specify... 5 seconds	<input type="checkbox"/>
Send Buffer	262144 bytes	<input checked="" type="checkbox"/>
Receive Window	131072 bytes	<input checked="" type="checkbox"/>
Keep Alive Interval	Specify... 1800 seconds	<input type="checkbox"/>
Maximum Syn Retransmissions	3	<input type="checkbox"/>
Maximum Segment Retransmissions	8	<input type="checkbox"/>
IP ToS	Specify... 0	<input type="checkbox"/>
Link QoS	Specify... 0	<input type="checkbox"/>
Selective ACKs	<input checked="" type="checkbox"/> Enabled	<input checked="" type="checkbox"/>
Selective NACK	<input type="checkbox"/>	<input type="checkbox"/>
Explicit Congestion Notification	<input type="checkbox"/>	<input type="checkbox"/>

Ensure that the settings highlighted are as shown in this screen capture

Figure 48 – Adding a mobile TCP specific profile – part1

Explicit Congestion Notification	<input type="checkbox"/>		<input type="checkbox"/>
Timestamps Extension for High Performance (RFC 1323)	<input checked="" type="checkbox"/> Enabled		<input checked="" type="checkbox"/>
Limited Transmit Recovery	<input checked="" type="checkbox"/> Enabled		<input checked="" type="checkbox"/>
Slow Start	<input checked="" type="checkbox"/> Enabled		<input checked="" type="checkbox"/>
Deferred Accept	<input type="checkbox"/>		<input type="checkbox"/>
Verified Accept	<input type="checkbox"/>		<input type="checkbox"/>
Nagle's Algorithm	<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>
Acknowledge on Push	<input checked="" type="checkbox"/> Enabled		<input type="checkbox"/>
MD5 Signature	<input type="checkbox"/>		<input type="checkbox"/>
MD5 Signature Passphrase	<input type="text"/>		<input type="checkbox"/>
Congestion Control	High Speed		<input checked="" type="checkbox"/>
Congestion Metrics Cache	<input checked="" type="checkbox"/> Enabled		<input checked="" type="checkbox"/>
Appropriate Byte Counting (RFC 3465)	<input checked="" type="checkbox"/> Enabled		<input checked="" type="checkbox"/>
D-SACK (RFC 2883)	<input type="checkbox"/>		<input checked="" type="checkbox"/>
Packet Loss Ignore Rate	0		<input checked="" type="checkbox"/>
Packet Loss Ignore Burst	0		<input checked="" type="checkbox"/>
Initial Congestion Window Size	0		<input checked="" type="checkbox"/>
Initial Receive Window Size	0		<input type="checkbox"/>
Initial Retransmission Timeout Base Multiplier for SYN Retransmission	0 milliseconds		<input type="checkbox"/>
Delay Window Control	<input type="checkbox"/>		<input checked="" type="checkbox"/>
Hardware SYN Cookie Protection	<input checked="" type="checkbox"/> Enabled		<input checked="" type="checkbox"/>
Max Segment Size (MSS)	1460		<input checked="" type="checkbox"/>

**Ensure that the settings highlighted are as shown in this screen capture**

Figure 49 - Adding a mobile TCP specific profile - part2

Max Segment Size (MSS)	1460	<input checked="" type="checkbox"/>
Rate Pace	<input type="checkbox"/>	<input type="checkbox"/>
Multipath TCP	<input type="checkbox"/>	<input type="checkbox"/>
MPTCP Checksum	<input type="checkbox"/>	<input type="checkbox"/>
MPTCP Checksum Verify	<input type="checkbox"/>	<input type="checkbox"/>
MPTCP Debug	<input type="checkbox"/>	<input type="checkbox"/>
MPTCP Fallback	reset	<input type="checkbox"/>
MPTCP Fast Join	<input type="checkbox"/>	<input type="checkbox"/>
MPTCP Join Max	5	<input type="checkbox"/>
MTCP Make After Break	<input type="checkbox"/>	<input type="checkbox"/>
MPTCP No Join DSS ACK	<input type="checkbox"/>	<input type="checkbox"/>
MPTCP RTO Max	5	<input type="checkbox"/>
MPTCP Retransmit Min	1000	<input type="checkbox"/>
MPTCP Subflow Max	6	<input type="checkbox"/>
MPTCP Timeout	3600	<input type="checkbox"/>

**This value may need to be reduced to ensure that no packet fragmentation. For now set this value to 1460**

Cancel Repeat **Finished**

Figure 50 - Adding a mobile TCP specific profile - part3

Click on **“Finished”** to complete the addition of this TCP profile.

## Adding the HTTPS Virtual Server

This particular virtual server terminates client side SSL connections and creates server side SSL connections to connect to the HTTPS services on the ClearPass servers. The creation of certificates and the importing of certificates to the F5 BIG-IP LTM can be found in the APPENDIX A titled “**Adding SSL Certificates to F5 BIG-IP LTM**” later in this document. Outside the SSL components, all other configuration of this Virtual Server is the same as that for the HTTP Virtual Server described earlier. As before, we will make use of the F5 iApps function using the imported certificates, profiles, pools and nodes we have configured to deploy the HTTPS Virtual Server.

**Note:** If you have *not* added ClearPass SSL certificates to the F5 BIG-IP LTM, please stop now and do this before proceeding. If you have added SSL certificates to the F5 BIG-IP LTM please continue. (**Appendix A** below discusses the process of adding certificates to F5 BIG-IP LTM)

On the left hand side navigation plane, click on **iApps** and then on **Application Services**. Click on the “Create...” button found in the top right of the action plane. The following configuration screen will appear. Proceed to populate the configuration screen as shown in the screen shot below:

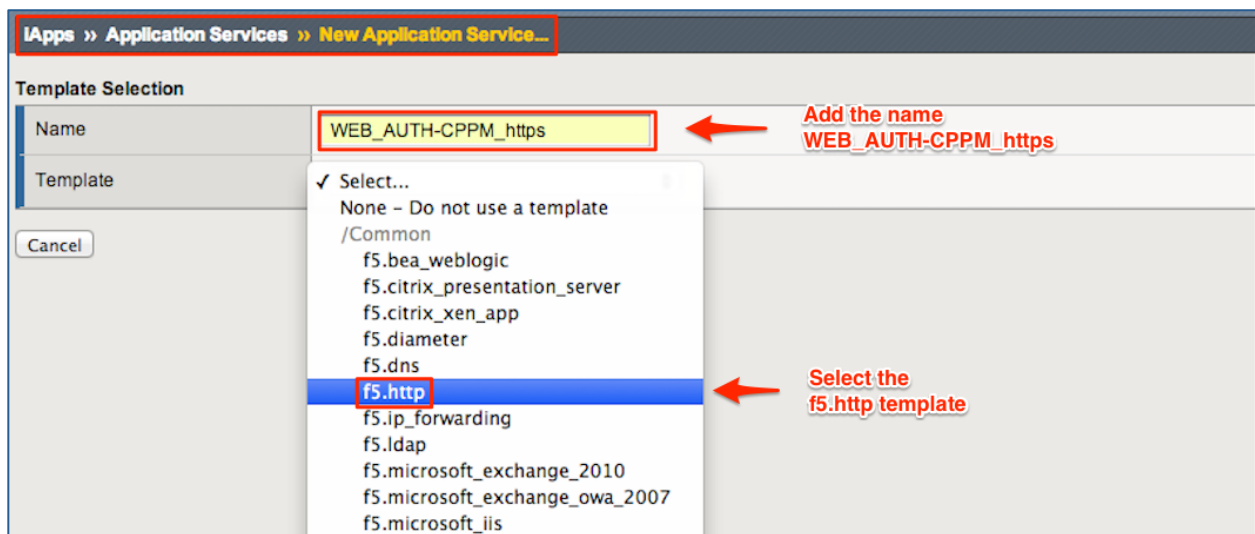


Figure 51 - Adding HTTPS F5 iApps Application Service

This will bring up the f5.http template. The “**Template Selection:**” should be set to “**Basic**” and from the template options, select “**Advanced – Configure advanced options**”. We select this option as this Virtual Server is built using the components configured earlier.

**iApps » Application Services » WEB\_AUTH-CPPM\_https**

Template Selection: **Basic** ← Ensure that Basic is selected

Name: WEB\_AUTH-CPPM\_https  
Template: f5.http

**Welcome to the iApp template for web applications**

Introduction	Configure security, high availability, and acceleration for web applications. This template supports basic web services. For detailed information and configuration assistance, see <a href="http://www.f5.com/pdf/deployment-guides/iapp-http-dg.pdf">http://www.f5.com/pdf/deployment-guides/iapp-http-dg.pdf</a> .
Check for Updates	Check for new versions of this template on the AskF5 Knowledge Base website ( <a href="http://support.f5.com/kb/en-us/solutions/public/13000/400/sol13422.html">http://support.f5.com/kb/en-us/solutions/public/13000/400/sol13422.html</a> ).
Additional features available	This system is not currently provisioned to run the BIG-IP Application Acceleration Manager (AAM). Provisioning AAM provides acceleration and optimization for your web applications.
Additional features available	The system is not currently provisioned to run the BIG-IP Application Visibility Reporting Module (AVR). Activating this module provides rich application statistics and reporting for your deployment.

**Template Options**

Do you want to see inline help? No, do not show inline help

Which configuration mode do you want to use?  
 Basic - Use F5's recommended settings  
 **Advanced - Configure advanced options** ← Select Advanced

**Figure 52 - Adding HTTPS Application Service - part1**

In the Network section on the same page ensure that the only VLAN where traffic is enabled is on the Client facing VLAN – in our case this VLAN is called “**external**”. By default, this template places all VLANs in the “**Selected**” box, we need to move all unwanted VLANs from that “**Selected**” box to the “**Options**” box leaving the Client facing VLAN selected. In the same section, select the “**Servers have a route through the BIG-IP system**” option from the routing section. See below:

**Network**

What type of network connects clients to the BIG-IP system? Wide area network (WAN)

Do you want to restrict client traffic to specific VLANs? Yes, enable traffic only on the VLANs I specify

On which VLANs should traffic be enabled or disabled?

Selected	Options
/Common <b>external</b>	/Common HA internal

What type of network connects servers to the BIG-IP system? Local area network (LAN)

Where will the virtual servers be in relation to the web servers? BIG-IP virtual server IP and web servers are on different subnets

How have you configured routing on your web servers?  
 **Servers have a route to clients through the BIG-IP system**  
 Servers do not have a route to clients through the BIG-IP system

**Figure 53 - Adding HTTPS Application Service - part2**

Here the SSL Encryption section should be set to “**Terminate SSL from clients, re-encrypt to servers (SSL Bridging)**”, if not please select this option. This option terminates client



side SSL connections and establishes a new SSL connection from the F5 BIG-IP LTM to CPPM. See screen capture below:

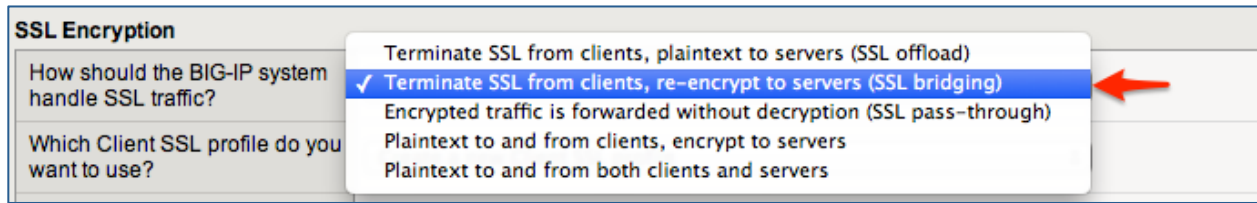


Figure 54 - Adding HTTPS Application Service - part3

Select the Client SSL profile previously created for client side traffic, select the Server SSL profile created for server side traffic as shown in the screen capture below.

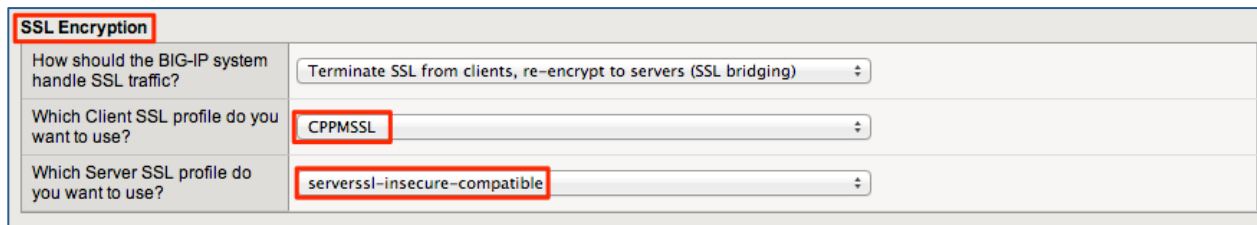


Figure 55 - Adding HTTPS Application Service - part4

The next section configures the Virtual Server and Pools. Here is where we make use of all the pre-configuration effort done in the previous section. The notes on the screen capture are self-explanatory.

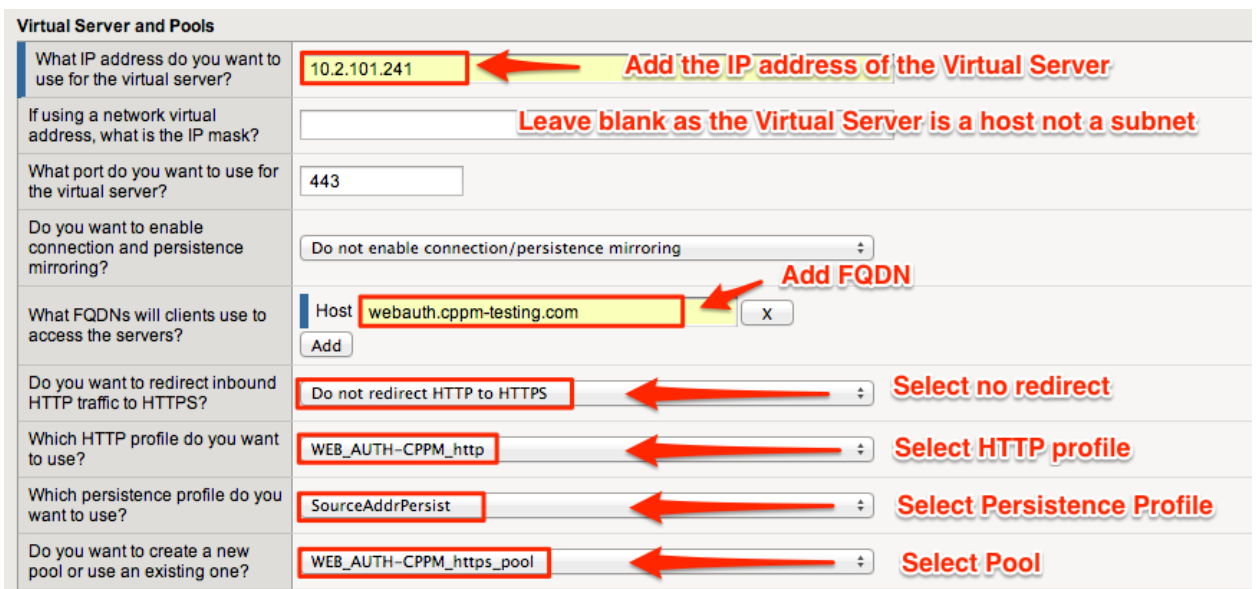


Figure 56 - Adding HTTPS Application Service - part5

**Note:** You can add as many FQDNs as needed to represent the Virtual Server created. Remember that these FQDNs must be resolvable by DNS in your network.

The next section is to be configured as shown. Please disable caching and compression as both of these have not been tested and could produce unwanted behavior. Set the client-side connection optimizer to be the MultiPath TCP optimizer configured earlier.

Delivery Optimization		
Which Web Acceleration profile do you want to use for caching?	Do not use caching	Do not use caching
Which compression profile do you want to use?	Do not compress HTTP responses	Do not use compression
How do you want to optimize client-side connections?	WEB_AUTH-CPPM_mptcp-mobile-optimized	Select TCP profile created

Figure 57 - Adding HTTPS Application Service - part6

The next section deals with Server Offload. In this section we configure the behavior of the server side connections. The screen shot below should be self-explanatory.

Server Offload		
Which OneConnect profile do you want to use?	WEB_AUTH-CPPM_oneconnect	Select the OneConnect profile
Which NTLM profile do you want to use?	Do not use NTLM (recommended)	
How do you want to optimize server-side connections?	WEB_AUTH-CPPM_tcp-lan-optimized	Select TCP profile created
Should the BIG-IP system queue TCP requests?	No, do not enable TCP request queuing (recommended)	
Use a Slow Ramp time for newly added servers?	Yes, use Slow Ramp (recommended)	
How many seconds should Slow Ramp time last?	300	

Figure 58 - Adding HTTPS Application Service - part7

All other settings should be left at default – please confirm that the default settings are as seen in the screen capture above and if not, please set them as the screen capture above.

The following section shows the F5 iRule and the Statistics and Logging configurations. There are no F5 iRules required for this service and Statistics and Logging is optional and depends on the customer. For our deployment Statistics and logging is not required. Refer to screen capture below:

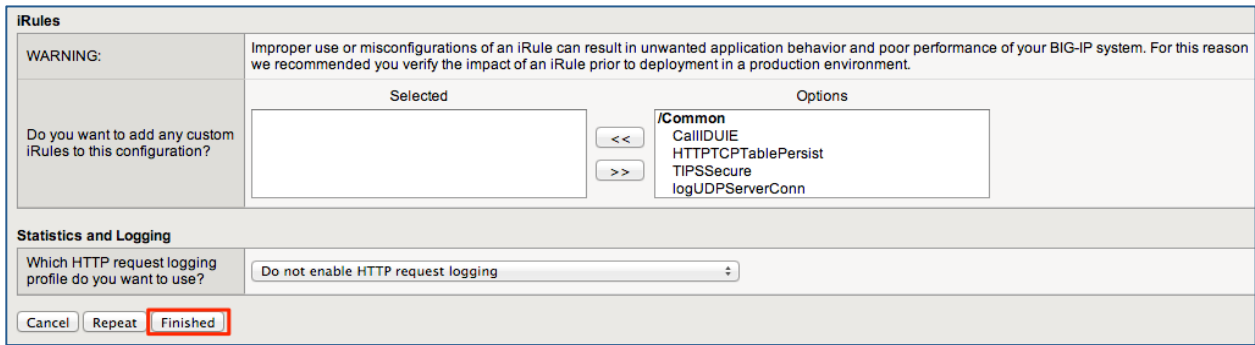


Figure 59 - Adding HTTPS Application Service – part8

Click on “Finished” to complete the addition of the WEB\_AUTH-CPPM\_vs Virtual Server.

The following Component listing should be visible once this template is executed. This can be viewed by navigating to **iApps > Application Services** and clicking on the WEB\_AUTH-CPPM iApps. This should bring up the Components view of the virtual server as shown below.

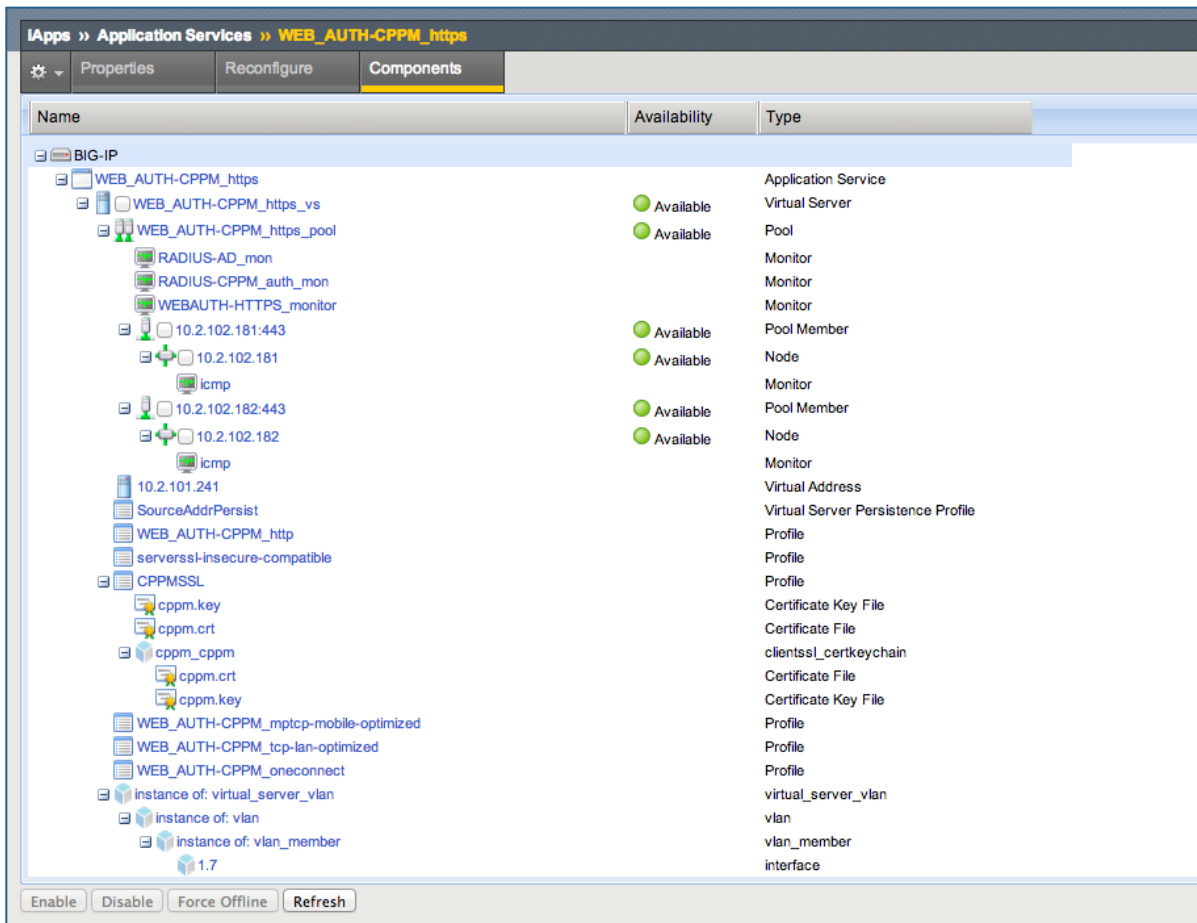


Figure 60 – Viewing the HTTPS Application Service

# Appendix A - Adding SSL Certificates to F5 BIG-IP LTM

The first thing required to get SSL termination set up is to install the CPPM SSL certificates onto the F5 BIG-IP LTM. We will assume you have already processed your Certificate Signing Request (CSR), sent this for signing and have your HTTPS server certificate plus private key now installed on your CPPM Cluster. First, you'll need to export the HTTPS certificate plus private key from the CPPM system. This is done under Administration, Certificate, Server Certificate. Ensure that you export the HTTPS certificate. This will download a file (folder on MAC) with two files, the certificate and the private key, these files will be required later for importing in F5 BIG-IP LTM. **Note:** Take special care of the private key, it's the crown jewels.

Administration » Certificates » Server Certificate

Server Certificate

Select Server:  Select Type:

[Create Self-Signed Certificate](#)  
[Create Certificate Signing Request](#)  
[Import Server Certificate](#)  
[Export Server Certificate](#)

Subject:	CN=vip.cppm-testing.com
Issued by:	EMAILADDRESS=support@cacert.org, CN=CA Cert Signing Authority, OU=http://www.cacert.org, O=Root CA
Issue Date:	Apr 10, 2014 12:24:35 PDT
Expiry Date:	Oct 07, 2014 12:24:35 PDT
Validity Status:	Valid
Details:	<a href="#">View Details</a>

**Root CA Certificate:**

Subject:	EMAILADDRESS=support@cacert.org, CN=CA Cert Signing Authority, OU=http://www.cacert.org, O=Root CA
Issued by:	EMAILADDRESS=support@cacert.org, CN=CA Cert Signing Authority, OU=http://www.cacert.org, O=Root CA
Issue Date:	Mar 30, 2003 04:29:49 PST
Expiry Date:	Mar 29, 2033 05:29:49 PDT
Validity Status:	Valid
Details:	<a href="#">View Details</a>

**Figure 61 - Exporting HTTPS Server certificate from CPPM**

Please take time to consider planning the generation of the CSR. Consider that this certificate will potentially represent multiple systems and interfaces, in the planning process think specifically about the Virtual Server on the F5 BIG-IP LTM and the VIP on the data/mgmt interfaces on CPPM if these are indeed also grouped for HA, if not then the physical interfaces. The certificate configuration in either the Common-Name or Subject Alternate Name must match specifically the FQDN/IP address from the clients' browser.

**Note:** CSR design is beyond the scope of this document, however another document 'CPPM TechNote - PKI 101' available on the Aruba support site discusses this at length.

Some issues were experienced in loading and using the exported CPPM certificates on the F5 BIG-IP LTM. In working through the certificate / private-key import we had to utilize the

openssl tools that are fortunately a part of OS X but can be installed for Windows from SourceForge.... <http://gnuwin32.sourceforge.net/packages/openssl.htm>

Adding the certificates to the F5 BIG-IP LTM is done by navigating to **Local Traffic -> SSL Certificates -> Import**. You must import the **.key** and the **.crt** files obtained from your CPPM separately but with the same "Name" property. So give your certificate and key a common alias, for example matching the domain, host or service the certificate will represent, such as "cppm-testing.com". Upload the **.key** file as a Key, and the **.crt** file as a Certificate; remember both files must use the same value in the alias field.

Under **System, File Management, SSL Certificate List**, click on **Import** to get going....

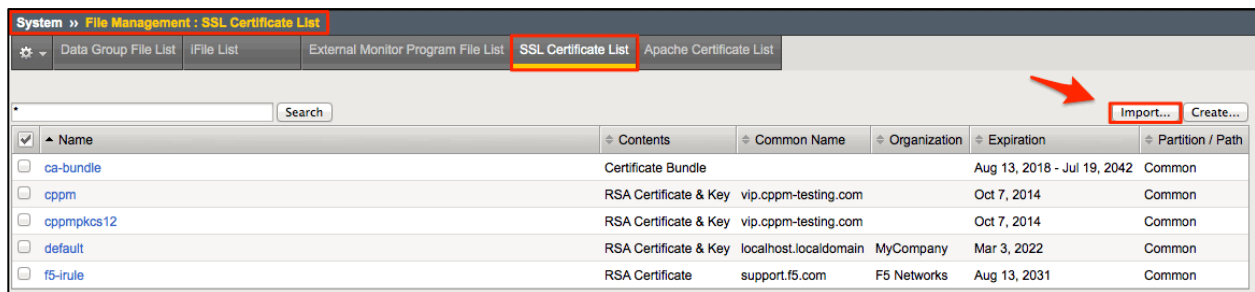


Figure 62 - Importing the Certificate & Private Key in to F5 BIG-IP LTM

From the Import menu, multiple items exist. We will import a Certificate and a Private Key.

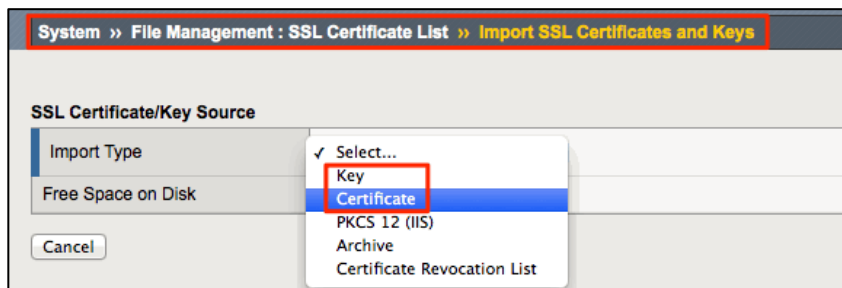


Figure 63 - Import the Certificate then the Key

Choose **Certificate**, provide the item with an alias name, and remember to use the same name/alias for the Certificate as the Key later. We then pasted in the text from the CPPM certificate; previously exported out of CPPM.

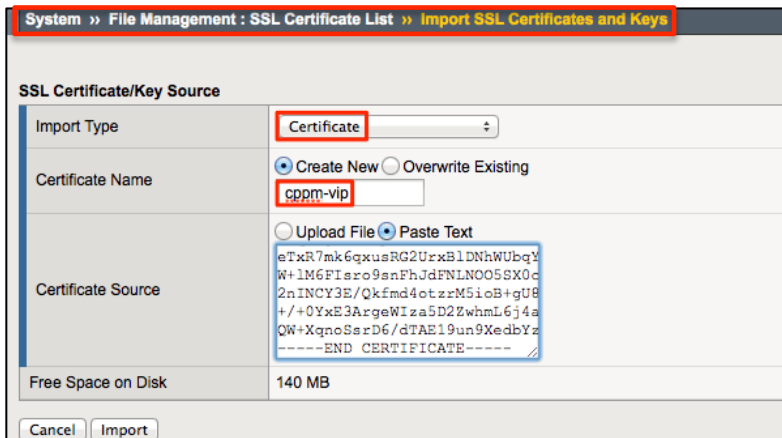


Figure 64 - Example of importing the Certificate using plain text

Following the import of the certificate you see below for 'cppm-vip' that only the Certificate has been imported, for the other items in the list you can see a Certificate/Key pair.

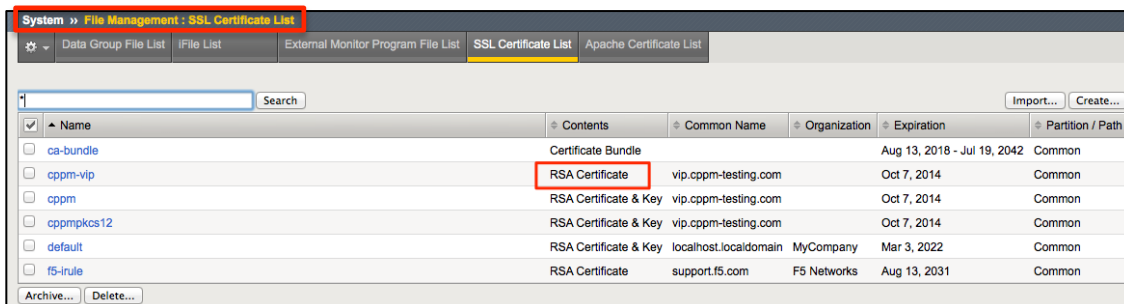


Figure 65 - Certificate only imported

Following the import of the Certificate, we follow a similar process for the Private Key.

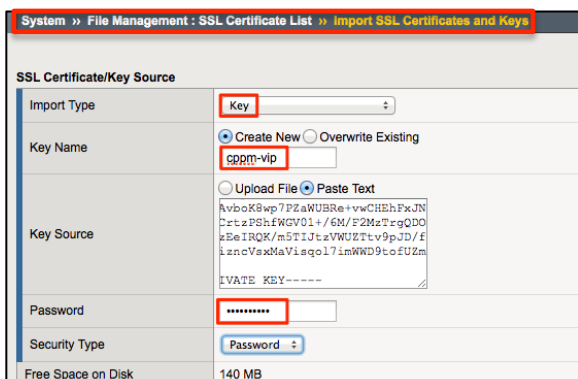


Figure 66 - Example of importing the Private Key using plain text

Again we pasted in the text from the private key, exported previously from CPPM and in this case supplied the password for the private key that came out of the original CPPM CSR.

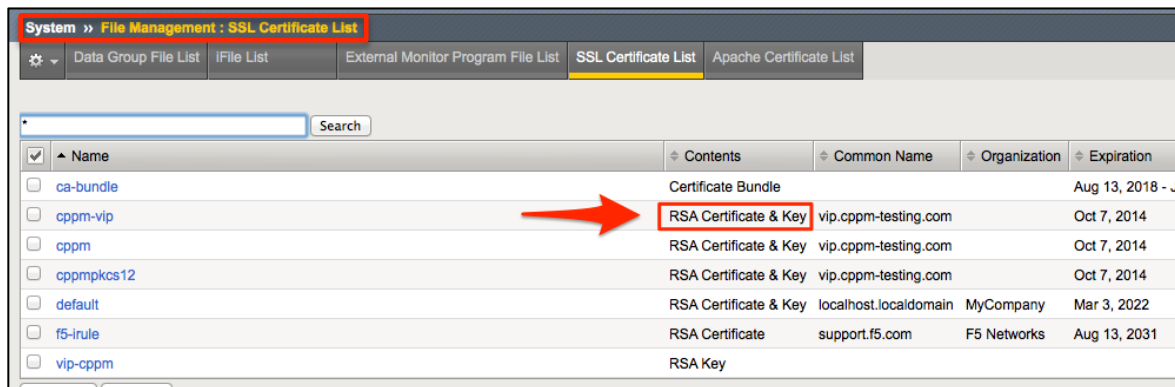


Figure 67 - Certificate and Key now imported

After completing the installation of the certificate and pkey, the list of SSL Certificates on F5 BIG-IP LTM should include your certificate and key in the list as a single entry, meaning they're associated with each other, in our example you can see that 'cppm-vip' now includes both a Certificate and a Key. If you have not entered the cert and key with the same name/alias you will have two separate items. One being a certificate one being a private key. As an example above you can see an orphaned key 'vip-cppm', this does not yet have its certificate installed or it was installed with a different name/alias than its certificate.

Following the certificate import, it's a good idea and sanity check to ensure the certificate includes the FQDN's you expect this certificate to represent, CN and SAN entries. Click on the certificate name/alias and then on the Certificate tab as shown below.

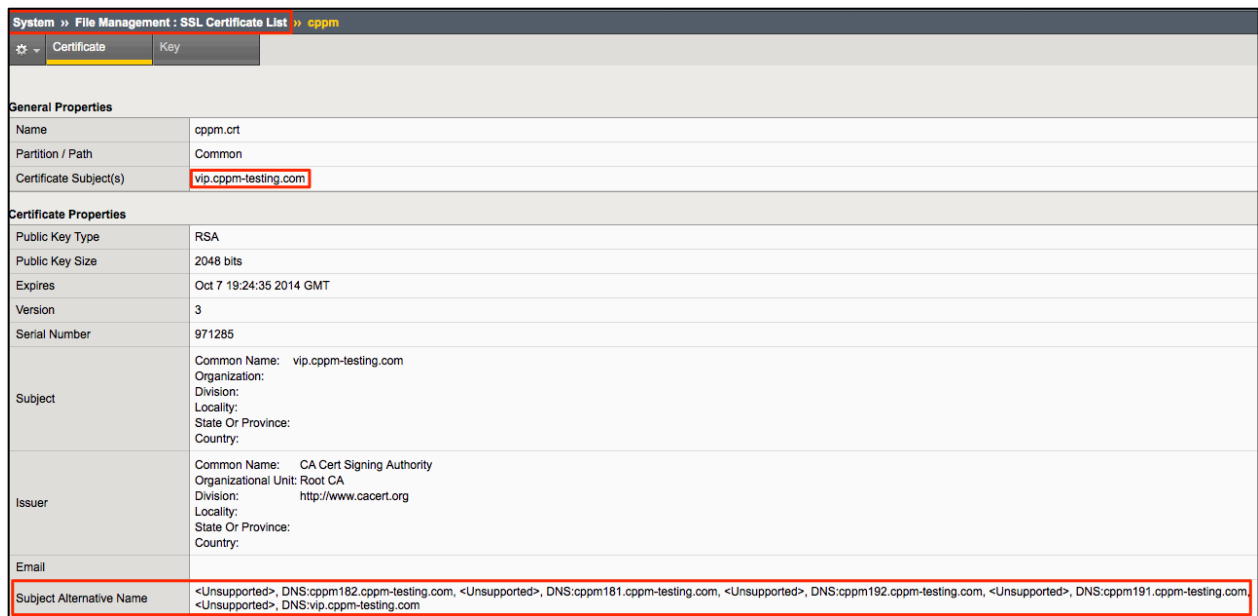


Figure 68 - Imported certificate attributes

## Appendix B - Overview of AOS 6.4 SLB Options

Starting in AOS 6.4 we introduced load balancing into the wireless controllers. This amounts to authentication only load-balancing. Load-balancing of requests to external servers RADIUS/LDAP etc. of which ClearPass is an external server. Prior to AOS 6.4, the AOS controllers allow you to configure multiple external authentication servers within a server group, however the controller will always use the first server in the group unless this server is down, then the controller will sequentially work through the list configured. In essence this provided RADIUS survivability via fail-over / fail-through not load balancing.

The load-balancing algorithm computes the expected time taken to authenticate a new client for each authentication server and chooses that authentication server for which the expected authentication time is least. Authentication time, in this context, means the total time spent back and forth between controller and the authentication server. For example, in the case of RADIUS, the authentication time would be sum of time elapsed between sending **Access-Request** to receiving **Access-Challenge** and time elapsed between sending client's response for **Access-Challenge** to receiving **Access-Accept**.

The below screen shot shows we have defined multiple RADIUS servers, 10.2.100.181 & 10.2.100.182. Then we have created a Server Group called **danny-test**. Within the Server Group is the new option where you can enable 'Load Balancing' as shown below.

The screenshot shows the configuration for a Server Group named 'danny-test'. The 'Load Balance' checkbox is checked, and the 'Fail Through' checkbox is unchecked. The 'Servers' table lists two RADIUS servers:

Name	Server-Type	
10.2.102.182	Radius	No
10.2.102.181	Radius	No

Figure 69 - Adding multiple Auth servers and enabling Load Balancing

Note that the currently AOS does not provide for any health-checks within its configuration. The ability to define with the Server Group 'Fail Through' does though allow for a highly available solution. The fail-through will work on protocol timeout of the primary RADIUS server not responding to request.



<ul style="list-style-type: none"><li>Server Group<ul style="list-style-type: none"><li>RADIUS Server<ul style="list-style-type: none"><li><b>10.2.100.180</b></li><li>10.2.100.190</li></ul></li><li>LDAP Server</li><li>Internal DB</li><li>Tacacs Accounting Server</li><li>TACACS Server</li><li>RFC 2576 Server</li></ul></li></ul>	<b>RADIUS Server &gt; 10.2.100.180</b>	
	Host	<input type="text" value="10.2.100.180"/>
	Key	<input type="password" value="....."/> Retype: <input type="password" value="....."/>
	Auth Port	<input type="text" value="1812"/>
	Acct Port	<input type="text" value="1813"/>
	Retransmits	<input type="text" value="3"/>
	Timeout	<input type="text" value="5"/> sec

Figure 70 - AOS RADIUS Server timeout