Signaling Delivery Controller

Bare Metal System Maintenance Guide

5.1

Catalog Number: RG-016-51-33 Ver. 12

Publication Date: May 2021

Proprietary and Confidential Information of F5 Networks

Legal Information

Copyright

© 2005-2021 F5, Inc. All rights reserved.

F5, Inc. (F5) believes the information it furnishes to be accurate and reliable. However, F5 assumes no responsibility for the use of this information, nor any infringement of patents or other rights of third parties which may result from its use. No license is granted by implication or otherwise under any patent, copyright, or other intellectual property right of F5 except as specifically described by applicable user licenses. F5 reserves the right to change specifications at any time without notice.

Trademarks

AskF5, F5, F5 [DESIGN], F5, OpenBloX, OpenBloX (design), Rosetta Diameter Gateway, Signaling Delivery Controller, SDC, Traffix, and Traffix [DESIGN] are trademarks or service marks of F5, Inc., in the U.S. and other countries, and may not be used without F5's express written consent.

All other product and company names herein may be trademarks of their respective owners.

Patents

This product may be protected by one or more patents indicated at: *http://www.f5.com/about/guidelines-policies/patents*

Confidential and Proprietary

The information contained in this document is confidential and proprietary to F5. The information in this document may be changed at any time without notice.

About F5

F5 (NASDAQ: FFIV) makes the connected world run better. F5 helps organizations meet the demands and embrace the opportunities that come with the relentless growth of voice, data, and video traffic, mobile workers, and applications—in the data center, the network, and the cloud. The world's largest businesses, service providers, government entities, and consumer brands rely on F5's intelligent services framework to deliver and protect their applications and services while ensuring people stay connected. For more information, visit *www.F5.com* or contact us at *Tfx_info@f5.com*.

About this Document

Document Name: F5 Signaling Delivery Controller Bare Metal System Maintenance Guide

Catalog Number: RG-016-51-33 Ver. 12

Publication Date: May 2021

Document Objectives

This document describes the maintenance procedures that are supported in this release for bare metal deployments.


Note: In this document, “server” and “machine” are used interchangeably.

Document History

Revision Number	Change Description	Change Location
Version 2 – February 2017	Added procedure for adding or deleting routes with a script.	*Adding and Deleting Routes via Script*
Version 3 – March 2017	Edited restoring a Site’s data procedure.	*Restoring a Site's Data*
Version 4 – May 2017	Added a parameter to the adding or deleting routes script. Added note about replacing a minion server.	*Adding and Deleting Routes via Script, Installing a New Server*
Version 5 – September 2017	Added note about replacing a master server. Migrating traffic from one FEP to another FEP Added procedure to remove SDC site	*Replacing a Server, Migrating Traffic from a FEP, Removing an SDC Site*
Version 6 – January 2018	Added removing a FEP procedure	*Removing a FEP Component*

Proprietary and Confidential Information of F5 Networks

Revision Number	Change Description	Change Location
Version 7- June 2018	Added add/delete network procedure. Added adding DNS server procedure. Added Warning to addRoutes/deleteRoutes.	*Configuring Networks, Adding DNS Servers*
Version 8 - December 2018	Updated Adding a FEP procedure	*Adding a FEP*
Version 9 – January 2019	Updated required parameter: listenInterfaceName	*Adding a FEP, Adding a CPF*
Version 10 – April 2019	Updated Replacing a Server: Uploading SDC Components to the New Server procedure- updated command text and removed note	*Uploading SDC Components to the New Server*
Version 11 – November 2020	Update replacing Splunk with ELK	*Throughout document*
Version 12 – May 2021	Remove space from traffix.changeNtp.	*changeNtp API Request*

Conventions

The style conventions used in this document are detailed in Table 1.

Table 1: Conventions

Convention	Use
Normal Text Bold	Names of menus, commands, buttons, user-initiated CLI commands and other elements of the user interface
Normal Text Italic	Links to figures, tables, and sections in the document, as well as references to other documents
`Script`	Language scripts
`Courier`	File names

 Proprietary and Confidential Information of F5 Networks

Convention	Use
Note:	Notes which offer an additional explanation or a hint on how to overcome a common problem
Warning:	Warnings which indicate potentially damaging user operations and explain how to avoid them

Table of Contents

 Proprietary and Confidential Information of F5 Networks

 Proprietary and Confidential Information of F5 Networks

 Proprietary and Confidential Information of F5 Networks

List of Figures

List of Tables

Proprietary and Confidential Information of F5 Networks

1. Introduction

In this release, you can perform the following maintenance procedures in bare metal deployments:

- *Adding a FEP to a New Server*
- *Adding a FEP to an Existing Server*
- *Removing a FEP Component*
- *Adding a CPF to a New* Server
- *Adding a CPF to an Existing Server*
- *Removing a CPF*
- *Migrating an NMS Agent*
- *Removing a Server*
- *Migrating Traffic from a FEP*
- *Backing up and Restoring a Site*
- *Backing up and Restoring a Cassandra Database*
- *Removing an SDC Site*
- *Configuring SDC-Server Routes*
- *Monitoring a Site's Status*
- *Recovering an ELK Master Node*
- *Adding DNS Servers*

1.1 General Prerequisites

This document assumes that you have a comprehensive understanding of:

- Installation and process as documented in the *F5 SDC Bare Metal Installation Guide*

Proprietary and Confidential Information of F5 Networks

- Positioning of the SDC in and/or between networks including the relevant IP and network (i.e. port) information needed for your site
- SDC and EMS deployments
- SDC architecture
- SDC pipeline

Note: From 5.1 CF 30, EMS deployments will use ELK components, instead of Splunk components, to manage all SDC reporting functionalities. This change is reflected in version 11 and higher of the *Bare Metal System Maintenance Guide*. This document assumes that you have upgraded to 5.1 CF 30.

1.2 Using REST API Requests

In this release, the maintenance actions are performed by using REST API requests. The APIs are used to query the SDC master Installer to execute a request, a SALT API request, such as adding a FEP, to the relevant server. These requests between the master Installers and the other servers are based on a standard Salt API interface.

The generic flow process is:

1. An http request is sent to the master Installer. The master Installer is identified as //<master_IP_address>:8000/ in the API requests.

2. While the request is pending, you can send another request to check its status.

3. When a status request has succeeded or failed, the request flow is completed.

1.3 Generating an Authentication Token

Prior to sending any API requests, you must have a valid authentication token. You need to send a request to the master Installer to generate an authentication token.

Note: An authentication token expires after ten hours.

Proprietary and Confidential Information of F5 Networks

To generate an authentication token:

1. Send the following API request to the master Installer that is identified by the <master_IP_address> parameter:

```
curl -ksi https://<master_IP_address>:8000/login -H "Accept:
application/json" -d username='saltuser' -d password='traffix' -d
eauth='pam'
```

Note: For all API requests, you need to use the minus sign, for example "-d" and not the N-dash "-". If you copy–paste the API request, you may have to type in the "-d" again with the minus sign to avoid syntax conversion errors.

1.3.1 Authentication Request Status Codes

The following are the possible return codes for the authentication API request

Table 2: Authentication Return Status Codes

Return Code	Description
200	success
401	authentication required
406	requested Content-Type not available

Proprietary and Confidential Information of F5 Networks

2. Adding a FEP

You can add a FEP component to a new or existing server in a site. These actions are performed by using Salt API requests.

Note: Verify that all network interfaces are up and running before adding a FEP. Refer to *Add Networks* for more information.

2.1 Adding a FEP to a New Server

If you want to add a FEP component to a new server, you first need to install a new minion server and then add the new FEP component.

These are the major phases for this action:

- Creating a new Topology XML file (snippet) for the new FEP component
- Installing a new minion server from the ISO
- Defining the new server's role
- Uploading the FEP component to the new server

2.1.1 Prerequisites

2.1.1.1 Installing the New Server Hardware

Install (and verify the successful installation) of the required hardware for the new server.

2.1.1.2 Verifying the Authentication Token

Adding a FEP component to a new server is performed with an API request. To apply the REST API, you need to have a valid authentication token that is not expired.

If the authentication token is expired, you need to request a new one (see *Generating an Authentication Token*).

Proprietary and Confidential Information of F5 Networks

2.1.2 Updating the Site Topology File with New FEP on a New Server

The site topology file must be updated to include definitions for the added server, the interfaces it uses, and the specific configuration elements for the new FEP component. This is done by creating a new Topology XML file (a snippet) that is then uploaded to the Cassandra database and merged with the Site Topology file, once the addFep API flow request is completed. The API addFep request refers to the XML file (@add_fep.xml.), see *addFepRequest* for more information.

To create a new Topology XML file:

1. Fill in the following parameters:
   - vm name

Note: The vm name is the name of the newly added server.

   - Interface network
   - applicationInstance type
     - listenInterfaceName

Refer to the *F5 SDC Bare Metal Installation Guide: Site Topology File* for more information about these parameters.

The following is an example of a Topology XML file for adding a FEP on a new server:

```
hosts=<?xml version="1.0" encoding="UTF-8"?>
<topology>
        <vms>
                <vm name="sdcvm221-01">
                        <interfaces>
                                                        <interface
network="net-1" ip4="10.2.81.1" ip6="" dev="bond3" bondDev="eth5,eth6"
bondingOpts="miimon=100 mode=1 num_grat_arp=20 updelay=5000
primary=eth5" name="SDC-NET-TCP-1"/>
```


Proprietary and Confidential Information of F5 Networks

```
                              <interface network="net-1" ip4="10.2.81.3"
ip6="" dev="bond3" bondDev="eth5,eth6" bondingOpts="miimon=100 mode=1
num_grat_arp=20 updelay=5000 primary=eth5" name="SDC-NET-TCP-1-VIP">

                                                          <route
name="sdcvm221-02-sdc-si-1" net4="10.3.2.128" ip4sub="27"
gateway="10.2.81.6"/>
</interface>
                        </interfaces>
                        <applicationInstances>
                           <applicationInstance type="fep" name="SDC-
NET-TCP-1" IPv="v4" listenInterfaceName="SDC-NET-TCP-1-VIP">
<fep>

<fepType>diameter</fepType>
                                          </fep>
                            </applicationInstance>
                            <applicationInstance type="vip" name="SDC-
NET-TCP-1-VIP" IPv="v4" listenInterfaceName="SDC-NET-TCP-1-VIP">
                                          <vip>
                                                  <routerID>3</routerID>

<transportProtocol>tcp</transportProtocol>

<applicationInstanceName>SDC-NET-TCP-1</applicationInstanceName>
                                          </vip>
                            </applicationInstance>
                        </applicationInstances>
                </vm>
                <vm name="sdcvm221-02">
                         <interfaces>
                                                          <interface
network="net-1" ip4="10.2.81.2" ip6="" dev="bond3" bondDev="eth5,eth6"
bondingOpts="miimon=100 mode=1 num_grat_arp=20 updelay=5000
primary=eth5" name="SDC-NET-TCP-1"/>
```


Proprietary and Confidential Information of F5 Networks

```
                              <interface network="net-1" ip4="10.2.81.3"
ip6="" dev="bond3" bondDev="eth5,eth6" bondingOpts="miimon=100 mode=1
num_grat_arp=20 updelay=5000 primary=eth5" name="SDC-NET-TCP-1-VIP">

                                                              <route
name="sdcvm221-02-sdc-si-1" net4="10.3.2.128" ip4sub="27"
gateway="10.2.81.6"/>
</interface>
                        </interfaces>
                        <applicationInstances>
                           <applicationInstance type="fep" name="SDC-
NET-TCP-1" IPv="v4" listenInterfaceName="SDC-NET-TCP-1-VIP">
<fep>

<fepType>diameter</fepType>
                                          </fep>
                           </applicationInstance>
                           <applicationInstance type="vip" name="SDC-
NET-TCP-1-VIP" IPv="v4" listenInterfaceName="SDC-NET-TCP-1-VIP">
                                          <vip>
                                                  <routerID>3</routerID>

<transportProtocol>tcp</transportProtocol>

<applicationInstanceName>SDC-NET-TCP-1</applicationInstanceName>
                                          </vip>
                           </applicationInstance>
                     </applicationInstances>
                </vm>
        </vms>
</topology>
```

2.1.3 Installing a New Server

You need to install the operating system on the new server. The operating system is installed from the existing ISO image that is already uploaded to the site.

Proprietary and Confidential Information of F5 Networks

To load the ISO image:

1. Retrieve the ISO image from the master Installer repository: */opt/repo/iso/image.iso*
2. Start the installed site machine from the ISO image.

   The **Welcome To F5 Traffix SDC Install Menu** is displayed.
3. Under the **Welcome To F5 Traffix SDC Install Menu**, select **Install Traffix F5 EL from cdrom for bare metal**.

   The GRUB boot loader page appears.

2.1.4 Defining the New Server's Role

You need to define the new server as a minion server to host the FEP component. This is done by configuring the GRUB boot parameters. There are mandatory parameters and optional parameters that are only required if relevant for the server deployment.

To configure the GRUB boot parameters:

1. In the prompt line, after **F5-TRAFFIX_SDC:traffix/kickstart/kickstart.cfg**, add the parameters, as follows:

Note: You can press the **TAB** key to enable editing.

Table 3: Mandatory Parameters

Name	Value Description
Server	master/minion Note: When adding an SDC component (CPF/FEP), you can only select minion
Hostname	The server's hostname Note: The hostname must be identical (case sensitive) to the value defined for the vm element in the Site Topology file

Proprietary and Confidential Information of F5 Networks

Name	Value Description
master0	The IP address on the management network that the first vInstaller uses.
master1	The IP address on the management network that the second vInstaller uses.
ip	The IP address is from the management network interface for minion and master Installer servers
Netmask	netmask for the ip address defined above (only needed for IPv4, CIDR is not supported)
Device	eth device the ip address defined above Note: On some HP blades eth8 is mgmt

Table 4: Optional Parameters

Name	Value Description
Vlan	vlan number for interface (if vlan defined)
Gw	default gateway (need to be mandatory if server = master)
Debug	debug=yes enable salt log with debug
Dns	DNS

The new server is now installed with an Operating System and has a defined role (minion).

2.1.5 Adding the FEP Component

Once the new minion server is up and running, you need to send an addFep API Request to the master Installer to install the FEP application to the minion server. The API addFep request refers to the new XML snippet file (@add_fep.xml) and by doing so the newly created XML is saved and merged with the Site Topology file in Cassandra.

2.1.5.1 addFepRequest

The following is the addFep API request:

Proprietary and Confidential Information of F5 Networks

```
# curl -ksi https://<master_IP_address>:8000  -H "Accept:
application/x-yaml" -H "X-Auth-Token:<Token>" -d client="runner" -d
fun="traffix.addFep" -X POST -d @fepadd.xml
```

2.1.5.2 addFep API Request Example

The following is an example of the request and answer:

```
# curl -ksi https://<master_IP_address>:8000 -H "Accept: application/x-
yaml" -H "X-Auth-Token:aa6790aa67ae5ce87715b66bf5bd58fc3ea4bdb5" -d
client="runner" -d fun="traffix.addFep" -X POST -d @fepadd.xml

HTTP/1.1 200 OK
Content-Length: 292
Access-Control-Expose-Headers: GET, POST
Access-Control-Allow-Credentials: true
Vary: Accept-Encoding
Server: CherryPy/3.2.2
Allow: GET, HEAD, POST
Cache-Control: private
Date: Tue, 20 Oct 2015 17:46:10 GMT
Access-Control-Allow-Origin: *
Content-Type: application/x-yaml
Set-Cookie: session_id=3b3f4aec793bc67d5b562d8129609bcf7045158b;
expires=Wed, 21 Oct 2015 03:46:10 GMT; Path=/

return:
- 0
- Add FEP flow successfully started
```

2.1.5.3 Post- installation

After installation is completed, the keepalived service needs to be restarted in order to load the new configuration. Run the following command:

monit restart <keepalivedSDC host name>

Proprietary and Confidential Information of F5 Networks

Note: Restarting keepalived has a service impact as it causes all traffic to stop and should be performed during a maintenance window.

2.2 Adding a FEP to an Existing Server

If you want to add a FEP component to an existing server, you need to do the following:

- Create a new Topology XML file (snippet) for the new FEP component
- Upload the FEP component to the existing server

2.2.1 Prerequisites

2.2.1.1 Verifying the Authentication Token

Adding a FEP component to a new server is performed with an API request. To apply the REST API, you need to have a valid authentication token that is not expired.

If the authentication token is expired, you need to request a new one (see *Generating an Authentication Token*).

2.2.2 Updating the Site Topology File with Added FEP on an Existing Server

The site topology file must be updated to include definitions for the specific configuration elements for the new FEP component. This is done by creating a new Topology XML file (a snippet) that is then uploaded to the Cassandra database and merged with the Site Topology file, once the addFep API flow request is completed. The API addFep request refers to the XML file (@add_fep.xml.), see *addFepRequest* for more information.

To create a new Topology XML file:

1. Fill in the following parameters:
   - vm name
   - interface network

Proprietary and Confidential Information of F5 Networks

Note: The vm name is the name of the server that is configured to host the added FEP component.

- applicationInstance type
- listenInterfaceName

Refer to the *F5 SDC Bare Metal Installation Guide: Site Topology File* for more information about these parameters.

The following is an example of a Topology file for adding a FEP on an existing server:

```
hosts=<?xml version="1.0" encoding="UTF-8"?>
<topology>
        <vms>
                <vm name="sdclab005-16-fep-1" >
                        <interfaces>
                                <interface network="ic"
ip4="10.1.80.164" ip6="" dev="eth1"/>
                                <interface network="sig-1"
ip4="10.1.81.164" ip6="" dev="eth2" name="sig-1-ip1">
                                        <route name="fep1-vip1-route1"
net4="10.1.69.0" net6="" ip4sub="255.255.255.0" gateway="10.1.81.1"/>
                                        <route name="fep1-vip1-route2"
net4="10.1.70.0" ip4sub="24" gateway="10.1.81.1"/>
                                </interface>
                        </interfaces>
                        <applicationInstances>
                                <applicationInstance type="fep"
name="fep7" IPv="v4" listenInterfaceName="sig-1-ip1"/>
                        </applicationInstances>
                </vm>
        </vms>
</topology>
```


Proprietary and Confidential Information of F5 Networks

2.2.3 Adding the FEP Component

You need to send an addFep API Request to the master Installer to install the FEP application to the minion server. The API addFep request refers to the new XML snippet file (@add_fep.xml) and by doing so the newly created XML is saved and merged with the Site Topology file in Cassandra.

2.2.3.1 addFep API Request

The following is the addFep API request:

```
# curl -ksi https://<master_IP_address>:8000 -H "Accept: application/x-
yaml" -H "X-Auth-Token:<Token>" -d client="runner" -d
fun="traffix.addFep" -X POST -d @fepadd.xml
```

2.2.3.2 addFep API Request Example

The following is an example of the request and answer

```
# curl -ksi https://<master_IP_address>:8000 -H "Accept: application/x-
yaml" -H "X-Auth-Token:aa6790aa67ae5ce87715b66bf5bd58fc3ea4bdb5" -d
client="runner" -d fun="traffix.addFep" -X POST -d @fepadd.xml

HTTP/1.1 200 OK
Content-Length: 292
Access-Control-Expose-Headers: GET, POST
Access-Control-Allow-Credentials: true
Vary: Accept-Encoding
Server: CherryPy/3.2.2
Allow: GET, HEAD, POST
Cache-Control: private
Date: Tue, 20 Oct 2015 17:46:10 GMT
Access-Control-Allow-Origin: *
Content-Type: application/x-yaml
Set-Cookie: session_id=3b3f4aec793bc67d5b562d8129609bcf7045158b;
expires=Wed, 21 Oct 2015 03:46:10 GMT; Path=/
```


Proprietary and Confidential Information of F5 Networks

```
return:
- 0
- Add FEP flow successfully started
```

2.3 Monitoring Adding a FEP

You can check if the FEP component was successfully installed with the appStatus API Request. This API request checks the status of a specific server. The response includes the relevant status codes for successfully installed applications. In addition, as with all other API requests, there are related command execution codes.

2.3.1 appStatus API Request

```
curl -ksi https://<master_IP_address>:8000 -H "Accept: application/x-yaml" -H "X-Auth-Token:<Token>" -d client="runner" -d fun="traffix.appStatus" -d tgt="*"-d apps=True (optional for apps list)
```

While adding a FEP component to an existing system, with the addFep API request, the server that is selected to host the added FEP, will have a state of PENDING_FEP_ADD. Once the FEP component is installed and started, the server status will change to SUCCESS.

2.3.1.1.1 Command Execution Codes for addFEP API Request

Table 5: addFEP Command Error Codes

Exit Code	Description
-90	addFep is not allowed

2.3.1.2 Post-Installation

After installation is completed, the keepalived service needs to be restarted in order to load the new configuration. Run the following command:

monit restart <keepalivedSDC host name>

Proprietary and Confidential Information of F5 Networks

 Note: Restarting keepalived has a service impact as it causes all traffic to stop and should be performed during a maintenance window.

Proprietary and Confidential Information of F5 Networks

3. Removing a FEP Component

You can remove a FEP component after a faulty installation or for maintenance reasons.

Note: This procedure is only supported on Bare Metal environments.

Removing a FEP is done by deleting a defined FEP from the Topology file. The following commands are to be performed on the SDC site in which you are removing the FEP component.

To remove a FEP component from a server:

1. Log in to Cassandra on one of master Installer servers from the SDC site in which you want to remove the FEP with the following command:

/opt/cassandra/bin/cqlsh <mgmt ip>

2. Run the following command:

SELECT * from topology.fep;

The following is an example of the displayed output:

Figure 1: Example of Topology FEP Elements Output

```
siteId              | vmName                | name          | IPv  | fepType  | index | listenInterfaceName | status
--------------------+-----------------------+---------------+------+----------+-------+---------------------+--------
sdclab010-09-site-1 |              *default |      *default | null | diameter |  null |                null |   null
sdclab010-09-site-1 | sdclab010-09-site-1-1 |   feptcpin-Gx |   v4 | diameter |     6 |                sig1 | active
```

3. From the displayed table, identify the line with the FEP name you want to delete and run the following command, as shown with values from the table, on one of the master Installer servers:

DELETE from topology.fep WHERE "name" = '<FEP name>' AND "siteID" = '<siteID>' AND "vmName" = '<vmName> ';

For example:

DELETE from topology.fep WHERE "name"='feptcpin-Gx' AND "siteId"='sdclab010-09-site-1' AND "vmName"='sdclab010-09-site-1-1';

Proprietary and Confidential Information of F5 Networks

Note: If using a VIP, then run the following command to delete the relevant FEP component from the site topology file for each VM:

DELETE from topology.vip WHERE "siteId" = '<site name>' AND "vmName" = '<VM name>' AND "name" = '<FEP component name> ';

For example:

DELETE from topology.vip WHERE "siteId" = = 'Burbank' AND "vmName" = 'brbnca-f5dra01-c7000-02' AND "name" = 'feptcpout-vip-Gx';

Validate that the removed FEP no longer appears in the site topology file by running the following command:

SELECT * from topology.vip

4. Verify that the selected FEP was removed by running the following command:

   SELECT * from topology.fep;

5. Run the following commands on each master Installer server (in the relevant SDC site) to rebuild the traffix pillar:

   cd /srv/traffix/

   echo 0 > .lastRev

   monit restart <vmName_vnf>

   salt '*' saltutil.refresh_pillar

6. Back up the old configuration on the server that hosted the removed FEP:

   a. Back up to another location all files under */etc/sysconfig/network*-scripts that start with ifcfg-eth, ifcfg-bond and route.

Note: If using a VIP, then also back up the */etc/keepalived* file and run this command **rm -rf /etc/keepalived/include.d/***

Proprietary and Confidential Information of F5 Networks

7. Clean-up the old configuration on any server that hosted the removed FEP:

   rm -rf /etc/sysconfig/network-scripts/ifcfg-eth*

   rm -rf /etc/sysconfig/network-scripts/ifcfg-bond*

   rm -rf /etc/sysconfig/network-scripts/route*

   rm -rf /etc/monit.d/*

   rm -rf /etc/keepalived/include.d/*

   rm -rf /opt/traffix/components/<fepName>

   rm -rf /data/backup/<site name>/<hostname>_traffix_config_mgr-config1/configuration/<Server hostname>_<FEP Component Name>

   rm -rf /etc/rsyslog.d/< Server hostname>_< FEP Component Name>

   rm -rf /opt/traffix/sdc<version>/config/sysconfig/traffix_fep-<Server hostname>_< FEP Component Name>

8. Go to one of master Installer servers and run:

   salt "*" state.highstate queue=True

   The networking files are now restarted.

9. Clear any alarms that were generated during this procedure that are related to the removed FEP component. Run the following commands on the SDC site server or EMS server that is hosting the OAM component:

   a. Connect to a Cassandra: */opt/cassandra/bin/cqlsh <OAM IP address>*

   b. Run this command:

      CONSISTENCY QUORUM

   c. For each relevant host site and the EMS site from which you are removing the FEP component, run these commands:

 # truncate table nms_<site name>_ems.stateful_alarms;

Proprietary and Confidential Information of F5 Networks

 # truncate table nms_<EMS site name>_ems.stateful_alarms;

 # truncate table nms_<site name>_ems.internal_connections;

 # truncate table nms_<site name>_ems.external_connections;

 # truncate table nms_<site name>_ems.component_state;

10. Reboot any server that hosted the removed FEP component.

Note: After rebooting, no alarms that are related to the removed FEP component should be generated.

Proprietary and Confidential Information of F5 Networks

4. Adding a CPF

You can add a CPF component to a new or existing server in a site. These actions are performed by using Salt API requests.

4.1 Adding a CPF to a New Server

If you want to add a CPF component to a new server, you first need to install a new minion server and then add the new CPF component.

These are the major phases for this action:

- Creating a new Topology XML file (snippet) for the new CPF component
- Installing a new minion server from the ISO
- Defining the new server's role
- Uploading the CPF component to the new server

4.1.1 Prerequisites

4.1.1.1 Installing the New Server Hardware

Install (and verify the successful installation) of the required hardware for the new server.

4.1.1.2 Verifying the Authentication Token

Adding a CPF component to a new server is performed with an API request. To apply the REST API, you need to have a valid authentication token that is not expired.

If the authentication token is expired, you need to request a new one (see *Generating an Authentication Token*).

4.1.2 Updating the Site Topology File with New CPF on a New Server

The site topology file must be updated to include definitions for the added server, the interfaces it uses, and the specific configuration elements for the new CPF component. This is done by creating a new Topology XML file (a snippet) that is then uploaded to the Cassandra database and merged with the Site Topology file, once the addCpf API flow

Proprietary and Confidential Information of F5 Networks

request is completed. The API addCpf request refers to the XML file (@add_cpf.xml.), see *addCpf Request* for more information.

To create a new Topology XML file:

1. Fill in the following parameters:
   - vm name

Note: The vm name is the name of the newly added server.

   - Interface network
   - applicationInstance type
     - listenInterfaceName

Refer to the *F5 SDC Bare Metal Installation Guide: Site Topology File Structure* for more information about these parameters.

The following is an example of a topology file for adding a CPF on a new server:

```
hosts=<?xml version="1.0" encoding="UTF-8"?>
<topology>
        <vms>
                <vm name="sdclab005-12-cpf-1"
defaultGateway="10.240.9.1">
                        <interfaces>
                                <interface network="ic"
ip4="10.1.80.164" ip6="" dev="eth1"/>
                                <interface network="sig-1"
ip4="10.1.81.164" ip6="" dev="eth2" name="sig-1-ip1">
                        </interfaces>
                        <applicationInstances>
                            <applicationInstance type="cpf"
name="cpf11" IPv="v4"/>
                        </applicationInstances>
                </vm>
        </vms>
```


Proprietary and Confidential Information of F5 Networks

```
</topology>
```

4.1.2.1 Updating the Site Topology File with the ELK Forwarder

If you are adding a new server, then you need to also update the topology xml to include the ELK Forwarder. The following is an example of a topology file for adding the ELK Forwarder:

```
hosts=<?xml version="1.0" encoding="UTF-8"?>
<topology>
<vms>
        <vm name="sdclab010-12-site1-2" defaultGateway="10.240.32.1">
            <applicationInstances>
                <applicationInstance IPv="v4" name="elk" type="elk"
listenInterfaceName="mgmt.eth0" />
            </applicationInstances>
        </vm>
</vms>
</topology>
```

4.1.3 Installing a New Server

You need to install the operating system on the new server. The operating system is installed from the existing ISO image that is already uploaded to the site.

To load the ISO image:

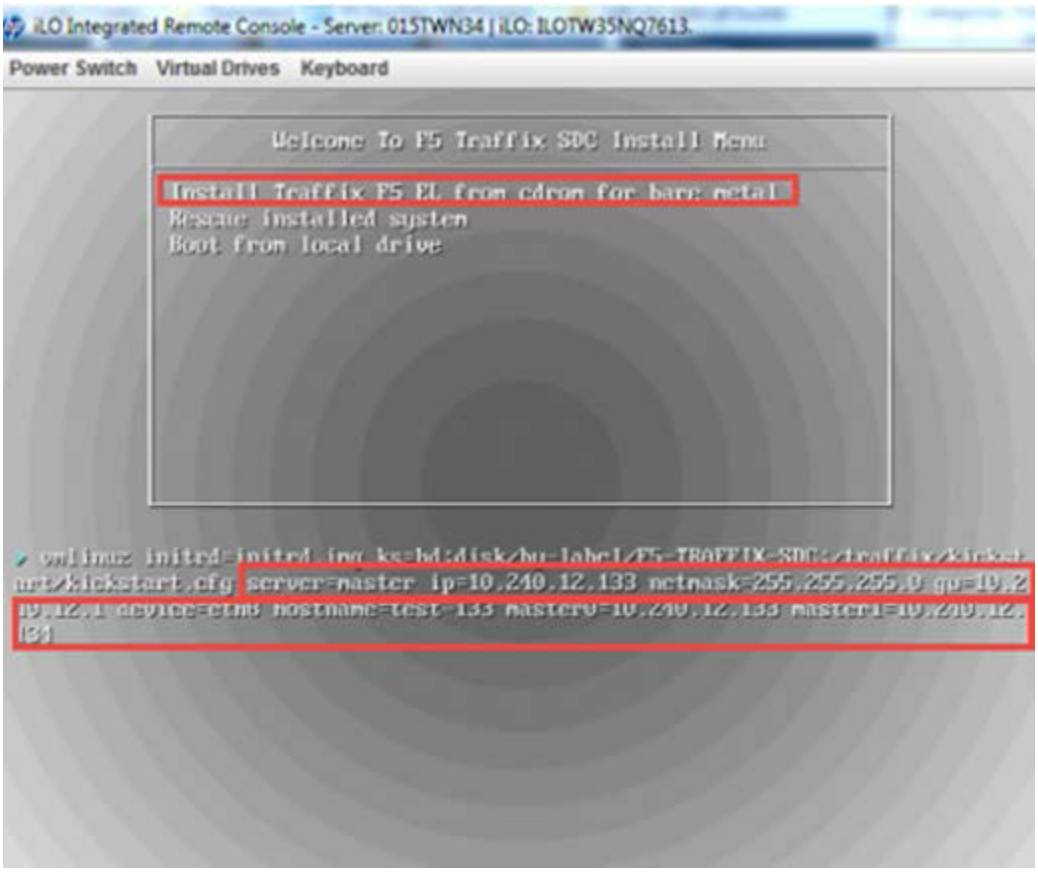
1. Retrieve the ISO image from the master Installer repository.
2. Start the installed site machine from the ISO image.

   The **Welcome To F5 Traffix SDC Install Menu** is displayed.
3. Under the **Welcome To F5 TRaffix SDC Install Menu**, select **Install Traffix F5 EL from cdrom for bare metal**.

   The GRUB boot loader page displays.

Proprietary and Confidential Information of F5 Networks

Figure 2: GRUB Boot Loader Welcome Page



4.1.4 Defining the New Server's Role

You need to define the new server as a minion server to host the CPF component. This is done by configuring the GRUB boot parameters. There are mandatory parameters and optional parameters that are only required if relevant for the server deployment.

To configure the GRUB boot parameters:

1. In the prompt line, after **F5-TRAFFIX_SDC:traffix/kickstart/kickstart.cfg**, add the parameters, as follows:

Note: You can press the TAB key to enable editing.

Proprietary and Confidential Information of F5 Networks

Table 6: Mandatory Parameters

Name	Value Description
server	master/minion Note: When adding an SDC component (CPF/FEP), you can only select minion.
hostname	the server's hostname Note: The hostname must be identical (case sensitive) to the value defined for the vm element in the Site Topology file
master0	The master (Installer) IP address
master1	ip address of the second master vInstaller
ip	The IP address is from the management network interface for minion and master Installer servers
netmask	netmask for the ip address defined above (only needed for IPv4, CIDR is not supported)
device	eth device the ip address defined above Note: On some HP blades eth8 is mgmt

Table 7: Optional Parameters

Name	Value Description
vlan	vlan number for interface (if vlan defined)
gw	default gateway (need to be mandatory if server = master)
debug	debug=yes enable salt log with debug
dns	DNS

The new server is now installed with an Operating System and has a defined role (minion).

Proprietary and Confidential Information of F5 Networks

4.1.5 Adding the CPF Component

Once the new minion server is up and running, you need to send an addCpf API Request to the master Installer to install the CPF application to the minion server. The API addCpf request refers to the new XML snippet file (@add_cpf.xml) and by doing so the newly created XML is saved and merged with the Site Topology file in Cassandra.

4.1.5.1 addCpf Request

The following is the addCpf API request:

```
# curl -ksi https://<master_IP_address>:8000 -H "Accept: application/x-
yaml" -H "X-Auth-Token: <Token>" -d client="runner" -d
fun="traffix.addCpf" -X POST -d @add_cpf.xml
```

4.1.5.2 addCPF API Request Example

The following is an example of the request and answer:

```
# curl -ksi https://<master_IP_address>:8000 -H "Accept: application/x-
yaml" -H "X-Auth-Token:4a77e8a32e3e7f2bfda5ba20b69a04c830871521" -d
client="runner" -d fun="traffix.addCpf" -X POST -d @add_cpf.xml
HTTP/1.1 200 OK
Content-Length: 52
Access-Control-Expose-Headers: GET, POST
Access-Control-Allow-Credentials: true
Vary: Accept-Encoding
Server: CherryPy/3.2.2
Allow: GET, HEAD, POST
Cache-Control: private
Date: Wed, 24 Feb 2016 14:14:39 GMT
Access-Control-Allow-Origin: *
Content-Type: application/x-yaml
Set-Cookie: session_id=4a77e8a32e3e7f2bfda5ba20b69a04c830871521;
expires=Thu, 25 Feb 2016 00:14:39 GMT; Path=/

return:
- - 0
```


Proprietary and Confidential Information of F5 Networks

```
  - Add CPF flow successfully started
```

4.1.5.3 ELK Forwarder Request

You need to send an ELK Forwarder request to the master Installer to install the ELK Forwarder on the new minion server. This API request refers to the updated XML snippet file (forwarder.xml) and by doing so the newly created XML is saved and merged with the Site Topology file in Cassandra.

```
curl -ksi https://<master_IP_address>::8000 -H "Accept: application/x-
yaml" -H "X-Auth-Token: <Token>" -d client="runner" -d
fun="traffix.addApplication" -X POST -d cName='elk' -d @forwarder.xml
```

4.2 Adding a CPF to an Existing Server

If you want to add a CPF component to an existing server, you need to do the following:

- Create a new Topology XML file (snippet) for the new CPF component
- Upload the CPF component to the existing server

4.2.1 Prerequisites

4.2.1.1 Verifying the Authentication Token

Adding a FEP component to another server is performed with an API request. To apply the REST API, you need to have a valid authentication token that is not expired.

If the authentication token is expired, you need to request a new one (see *Generating an Authentication Token*).

4.2.2 Updating the Site Topology File with Added CPF on an Existing Server

The site topology file must be updated to include definitions for the specific configuration elements for the new CPF component. This is done by creating a new Topology XML file (a snippet) that is then uploaded to the Cassandra database and merged with the Site Topology file, once the addCpf API flow request is completed. The API addCpf request refers to the XML file (@add_cpf.xml.), see *addCpf API Request* for more information.

Proprietary and Confidential Information of F5 Networks

To create a new Topology XML file:

1. Fill in the following parameters:
   - vm name
   - interface network

Note: The vm name is the name of the server that is configured to host the added CPF component.

   - applicationInstance type
     - listenInterfaceName

Refer to the *F5 SDC Bare Metal Installation Guide: Site Topology File Structure* for more information about these parameters.

The following is an example of a Topology file for adding a CPF on an existing server:

```
hosts=<?xml version="1.0" encoding="UTF-8"?>
<topology>
        <vms>
                <vm name="sdclab005-12-cpf-1"
defaultGateway="10.240.9.1">
                        <interfaces>
                                <interface network="ic"
ip4="10.1.80.164" ip6="" dev="eth1"/>
                                <interface network="sig-1"
ip4="10.1.81.164" ip6="" dev="eth2" name="sig-1-ip1">
                        </interfaces>
                        <applicationInstances>
                            <applicationInstance type="cpf"
name="cpf11" IPv="v4"/>
                        </applicationInstances>
                </vm>
        </vms>
</topology>
```


Proprietary and Confidential Information of F5 Networks

4.2.3 Adding the CPF Component

You need to send an addCpf API Request to the master Installer to install the CPF application to the minion server. The API addCpf request refers to the new XML snippet file (@add_cpf.xml) and by doing so the newly created XML is saved and merged with the Site Topology file in Cassandra.

4.2.3.1 addCpf API Request

The following is the addCpf API request:

```
# curl -ksi https://<master_IP_address>:8000 -H "Accept: application/x-
yaml" -H "X-Auth-Token:<Token>" -d client="runner" -d
fun="traffix.addCpf" -X POST -d @add_cpf.xml
```

4.2.3.2 addCPF API Request Example

The following is an example of the request and answer

```
# curl -ksi https://<master_IP_address>:8000 -H "Accept: application/x-
yaml" -H "X-Auth-Token:4a77e8a32e3e7f2bfda5ba20b69a04c830871521" -d
client="runner" -d fun="traffix.addCpf" -X POST -d @add_cpf.xml
HTTP/1.1 200 OK
Content-Length: 52
Access-Control-Expose-Headers: GET, POST
Access-Control-Allow-Credentials: true
Vary: Accept-Encoding
Server: CherryPy/3.2.2
Allow: GET, HEAD, POST
Cache-Control: private
Date: Wed, 24 Feb 2016 14:14:39 GMT
Access-Control-Allow-Origin: *
Content-Type: application/x-yaml
Set-Cookie: session_id=4a77e8a32e3e7f2bfda5ba20b69a04c830871521;
expires=Thu, 25 Feb 2016 00:14:39 GMT; Path=/

return:
- - 0
```


Proprietary and Confidential Information of F5 Networks

```
- Add CPF flow successfully started
```

4.3 Monitoring Adding a CPF

You can check if the CPF component was successfully installed with the appStatus API Request. This API request checks the status of a specific server. The response includes the relevant status codes for successfully installed applications. In addition, as with all other API requests, there are related command execution codes.

4.3.1 appStatus API Request

```
curl -ksi https://< master_IP_address>: 8000 -H "Accept: application/x-yaml" -H "X-Auth-Token:<Token>" -d client="runner" -d fun="traffix.appStatus" -d tgt="*"-d apps=True (optional for apps list)
```

While adding a CPF component to an existing system, with the addCpf API request, the server that is selected to host the added CPF, will have a state of PENDING_CPF_ADD. Once the CPF component is installed and started, the server status will change to SUCCESS.

4.3.1.1.1 Command Execution Codes for addCpf Request

Table 8: addCpf Command Error Codes

Exit Code	Description
-90	addCpf is not allowed

Proprietary and Confidential Information of F5 Networks

5. Removing a CPF

You can remove a CPF component using the scale-in API request. You can specify which CPF to remove from which server or, alternatively, you do not have to specify which CPF will be removed. In this case, the CPF component is removed in the order in which the CPFs were added. There are two different scale-in API requests depending on whether you want to specify which CPF to remove. The Site Topology file is automatically updated with the changed CPF data.

If you are removing a specified CPF from a server in order to free-up a specific server, and the server also hosts another component, such as the NMS Agent, then you also need to migrate the other component so that the server will be available for other applications. In the case that the Fluentd Forwarder is also hosted on the server, it will be automatically removed. If you need to migrate the NMS Agent, then you need to update the Site Topology File.

In the case, that you want to remove the server that was hosting the NMS Agent and the CPF, you need to remove it from the Site Topology file. See Removing a Server for more information.

 Note: If after removing a CPF, there is only one remaining CPF, your deployment will no longer support active-active high availability for CPFs.

5.1 Prerequisites

The following are the prerequisites for removing a CPF.

5.1.1 Verifying the Authentication Token

Removing a CPF component is performed with an API request. To apply the REST API, you need to have a valid authentication token that is not expired.

If the authentication token is expired, you need to request a new one (see *Generating an Authentication Token*).

Proprietary and Confidential Information of F5 Networks

5.1.2 Migrating an NMS Agent

Prior to removing a specified CPF from a server that also hosts the NMS Agent, you need to migrate the NMS component to another existing server.

You need to send a migrateNMS API Request to the master Installer to move the NMS component from the source server (<source_server_name>) to the destination server. The destination server's configurations are defined in the new XML snippet file. The API migrateNMS request refers to the new XML snippet file (@migrate_nms.xml) and by doing so the newly created XML is saved and merged with the Site Topology file in Cassandra.

Note: If the newly migrated NMS Agent remains ready to connect to the removed server, then restart both NMS Agent components.

5.1.2.1 Updating the Site Topology File with the Migrated NMS Agent

The site topology file must be updated to include definitions for the specific configuration elements for the destination server in which the NMS agent is being migrated to. This is done by creating a new Topology XML file (a snippet) that is then uploaded to the Cassandra database and merged with the Site Topology file, once the migrateNMS API flow request is completed. The API request refers to the XML file (@migrate_nms.xml.), as shown in the *migrateNMS API Request*.

To create a new Topology XML file:

1. Fill in the following parameters for the destination server (dst) to which the NMS Agent is being migrated to:

- vm name
- interface network

Note: The vm name is the name of the destination server that is configured to host the migrated NMS component.

- applicationInstance type
  - listenInterfaceName

Refer to the *F5 SDC Bare Metal Installation Guide: Site Topology File Structure* for more information about these parameters.

The following is an example of a Site Topology file for migrating an NMS Agent from one server to another:

```
# cat migrate_nms.xml

dst=<?xml version="1.0" encoding="UTF-8"?>
<topology>
        <vms>
                <vm name="sdclab005-12-master-1"
defaultGateway="10.240.9.1">
                        <interfaces>
                                <interface network="mgmt"
ip4="10.240.9.160" ip6="" dev="eth0"/>
                                <interface network="ic"
ip4="10.1.79.160" ip6="" dev="eth1"/>
                        </interfaces>
                        <applicationInstances>
                                <applicationInstance type="nms"
name="NmsAgent1" IPv="v4" listenInterfaceName="ic"/>
                        </applicationInstances>
                </vm>
        </vms>
</topology>
```

5.1.2.2 migrateNMS API Request

The following is the migrateNMS API request:

```
# curl -ksi https://<master_IP_address>:8000 -H "Accept: application/x-
yaml" -H "X-Auth-Token:<Token>" -d client="runner" -d fun=
```


Proprietary and Confidential Information of F5 Networks

```
"traffix.migrateNms" -X POST -d src="<source_server_name>" -d
nms_name="<app name>" -d @migrate_nms.xml
```

Note: Verify that the API request is accurate and refers to the correct server and nms name. In the event that it is not, and while the API request may be validated without an error, its status in the Cassandra database will be pending and the request to migrate the NMS Agent to another server will not be executed. In order to execute the API request, change the request status from pending to idle:

1. Connect to the Casandra CLI (cqlsh): /opt/apache-cassandra-2.2.4/bin/cqlsh

2. View the current status: cqlsh> SELECT * FROM statusflow.flow ;

3. Change status to idle: cqlsh> UPDATE statusflow.flow SET "flowState"='idle' WHERE "siteId"='KC' AND "flowType"='statusApi' ;

5.1.2.2.1 migrateNMS API Request Example

The following is an example of the request and answer:

```
# curl -ksi https://localhost:8000 -H "Accept: application/x-yaml" -H
"X-Auth-Token:aacb2ad10afff55bab6a17fc88dd62c1b9a53cf2" -d
client="runner" -d fun="traffix.migrateNms" -X POST -d src="sdclab005-
12-master-2" -d nms_name="NmsAgent1" -d @migrate_nms.xml
HTTP/1.1 200 OK
Content-Length: 113
Access-Control-Expose-Headers: GET, POST
Access-Control-Allow-Credentials: true
Vary: Accept-Encoding
Server: CherryPy/3.2.2
Allow: GET, HEAD, POST
Cache-Control: private
Date: Wed, 01 Jun 2016 15:53:51 GMT
Access-Control-Allow-Origin: *
Content-Type: application/x-yaml
```


Proprietary and Confidential Information of F5 Networks

```
Set-Cookie: session_id=aacb2ad10afff55bab6a17fc88dd62c1b9a53cf2;
expires=Thu, 02 Jun 2016 01:53:51 GMT; Path=/
```

5.2 Removing a Specified CPF Component

To remove a specified CPF from a specified server, you need to specify the <server hostname> and the <cpf app name> in the scale-in API request.

5.2.1 Scale-in Specified CPF API Request

The following is the API request:

```
curl -ksi https://<master_IP_address>:8000 -H "Accept: application/x-
yaml" -H "X-Auth-Token: <Token>" -d client="runner" -d
fun="traffix.scale" -d role='in' -d bare_metal=True -d hosts="<server
hostname>" -d app="<cpf app name>"
```

5.2.1.1 Scale-in Specified CPF API Request Example

The following is an example of the request and answer:

```
curl -ksi https://localhost:8000 -H "Accept: application/x-yaml" -H "X-
Auth-Token:73856b5ec0587490ea09aadf5c8f1d5524552110" -d client="runner"
-d fun="traffix.scale" -d role='in' -d bare_metal=True -d
hosts="sdclab005-12-cpf-4" -d specific_app="cpf1"

HTTP/1.1 200 OK
Content-Length: 105
Access-Control-Expose-Headers: GET, POST
Access-Control-Allow-Credentials: true
Vary: Accept-Encoding
Server: CherryPy/3.2.2
Allow: GET, HEAD, POST
Cache-Control: private
Date: Sun, 15 May 2016 10:56:47 GMT
Access-Control-Allow-Origin: *
Content-Type: application/x-yaml
```


Proprietary and Confidential Information of F5 Networks

```
Set-Cookie: session_id=73856b5ec0587490ea09aadf5c8f1d5524552110;
expires=Sun, 15 May 2016 20:56:47 GMT; Path=/

return:
- - 0
  - 'initiateVmShutDown: Graceful shutdown request sent to
[u''sdclab005-12-cpf-4_cpf1'']'
```

5.3 Removing any CPF Component

To remove a CPF without specifying which CPF, use the following scale-in API request. This API request requires that you define from which server (<server hostname>) the CPF will be removed. If there is more than one CPF installed on the specified host server, the CPF that was added first will be removed.

5.3.1 Scale-in CPF API Request

The following is the API request:

```
curl -ksi https://<master_IP_address>:8000 -H "Accept: application/x-
yaml" -H "X-Auth-Token: <Token>" -d client="runner" -d
fun="traffix.scale" d role='in' -d bare_metal=True -d hosts='<host
name>'
```

5.4 Removing a Server

In the case that you want to remove the server after you have migrated the NMS Agent and removed the CPF, you need to run a script on one of the master Installers to remove the server from the Site Topology file.

To remove the server from the Site Topology file:

1. Enter to scripts folder:

   # cd /opt/traffix/scripts

2. Run the remove host from topology script:

   # python remove_host_from_topology.py <hostname>

Proprietary and Confidential Information of F5 Networks

The <hostname> is the name of the server that you want to remove.

5.5 Monitoring Removing a CPF

You can monitor if the CPF component was successfully removed from a server in a few ways. First, the command error codes for the Scale-in Host Server API Request indicate if the request was executed as expected (see *Command Execution Codes for Scale-in Host Server API Request*). Second, you can send a traffix.scaleInStatus request to see if the earmarked CPF application has been removed from the Site Topology file (see traffix.scaleInStatus API Request). Third, you can send a siteStatus API Request and verify that the earmarked CPF is no longer listed as an installed application (see *siteStatus API Request*).

Note: When sending any API request, you must have a current authentication token. Verify that the authentication token is valid and has not expired. For more information, see *Verifying the Authentication Token*.

5.5.1 traffix.scaleInStatus API Request

You can send the following API request to check if the earmarked CPF has been successfully removed from the Site Topology file:

```
curl -ksi https://<master_IP_address>:8000 -H "Accept: application/x-
yaml" -H "X-Auth-Token:<Token>" -d client="runner" -d
fun="traffix.scaleInStatus" bare_metal=True
```

5.5.1.1 traffix.scaleInStatus API Request Example

The following is an example of a request to check if the earmarked CPF has been successfully removed from the Site Topology file:

```
# curl -ksi https://<master_IP_address>:8000 -H "Accept: application/x-
yaml" -H "X-Auth-Token:aa6790aa67ae5ce87715b66bf5bd58fc3ea4bdb5" -d
client="runner" -d fun="traffix.scaleInStatus" bare_metal=True

HTTP/1.1 200 OK
```


Proprietary and Confidential Information of F5 Networks

```
Content-Length: 292
Access-Control-Expose-Headers: GET, POST
Access-Control-Allow-Credentials: true
Vary: Accept-Encoding
Server: CherryPy/3.2.2
Allow: GET, HEAD, POST
Cache-Control: private
Date: Tue, 20 Oct 2015 17:46:10 GMT
Access-Control-Allow-Origin: *
Content-Type: application/x-yaml
Set-Cookie: session_id=3b3f4aec793bc67d5b562d8129609bcf7045158b;
expires=Wed, 21 Oct 2015 03:46:10 GMT; Path=/

return:
- ScaleIn request: cpf-138 successfully removed from topology
```

5.5.2 Command Execution Codes for Scale-in Host Server API Request

Table 9: Scale-in Host Server API Request Command Error Codes

Exit Code	Description
-65	"Error specific host scale in available only on bare metal"
-66	"Error could not get status of app without host name of the server"
-67	"Error specified host not in topology"
-68	"Error the specified vm is not cpf server"
-69	"Error the specified app is not configured on the specified host"

Proprietary and Confidential Information of F5 Networks

6. Migrating Traffic from a FEP

You can migrate traffic from one FEP component located on one server to another FEP component located on another server. The SDC supports a FEP failover mechanism that allows you to perform any required maintenance on an active FEP component. The failover mechanism is monitored with Keepalived that checks every six seconds the VIP-FEP connection and when one connection is down the VIP will connect to another FEP component located on another server and traffic is migrated to the newly recognized FEP component. Upon completing the maintenance activity and restarting the first FEP component, the newly recognized FEP component remains the active FEP component.

To migrate the traffic:

1. Stop the FEP on the server that you want to perform maintenance/ not have traffic running:

    d. Connect to the relevant server using SSH.

    e. Run this command: **# monit stop <FEP name>**

    After a few seconds, the Keepalived mechanism will recognize that the first FEP is down, and then the VIP will connect to another FEP component and traffic will be migrated through that FEP.

2. Upon completing the maintenance activity, restart the FEP component:

    a. Connect to the relevant server using SSH.

    b. Run this command: **# monit start <FEP name>**

    Traffic continues to be directed to the newly recognized FEP component.

Proprietary and Confidential Information of F5 Networks

7. Replacing a Server

In the case that a server is faulty, you can replace it with a new server. This can be done for faulty servers that host all the SDC components, as well as, for the server hosting one of the master Installers, as long as there is geo-redundancy and there is another server hosting the second master Installer.

The following is the process for replacing a server:

- Installing a new server
- Defining the new server's role (master/minion)
- SDC Components Automatically Uploaded to New Server

As you are shifting the SDC components from the old server to the new one, there is no need to change the Site Topology file.

Note: If the replaced server had the following data, it is not saved and cannot be retrieved to the new server:

Backed-up central syslog logs

Fluentd Forwarder buffered data that was not yet sent to the Fluentd EMS.

7.1 Installing a New Server

You need to install the operating system on the new server. The operating system is installed from the existing ISO image that is already uploaded to the site.

To load the ISO image:

1. Retrieve the ISO image from the master Installer repository: */opt/repo/iso/image.iso*
2. Start the installed site machine from the ISO image.

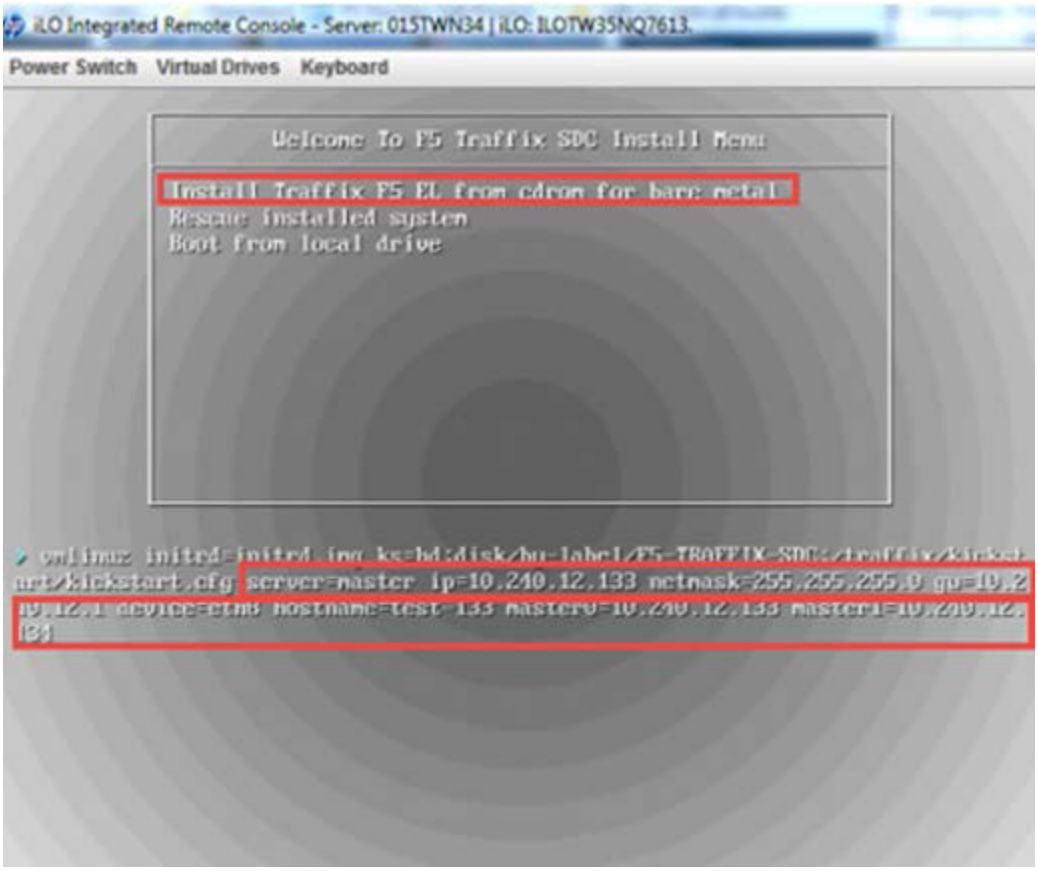
   The **Welcome To F5 Traffix SDC Install Menu** is displayed.

Proprietary and Confidential Information of F5 Networks

3. Under the **Welcome To F5 TRaffix SDC Install Menu**, select **Install Traffix F5 EL from cdrom for bare metal**.

The GRUB boot loader page displays.

Figure 3: GRUB Boot Loader Page



Note: In the case that you are replacing one of the minion servers and one of the master Installer servers is down, you need to define both master IP addresses with the same IP address so the minion server does not continue to search for a second master Installer server and only connects to the one working master Installer server. You can edit the IP address either from the GRUB boot loader page or from the params file (if you are installing from a new ISO).

Proprietary and Confidential Information of F5 Networks

Once the second master Installer is up and running, you need to re-edit the IP address of the minion server in the params file so that both master Installer IP addresses are defined. Following this, run the Salt install command (./salt-install.sh) and then proceed with *Removing the Replaced Server Master Installer Connection*.

7.1.1 Replacing a Server with Cassandra

When you replace a server that hosts either one of the master Installers or an OAM, you need to remove the existing server from the Cassandra cluster by pointing to its Host ID.

To remove the existing server from the Cassandra cluster:

1. Retrieve and verify the Host ID that you want to remove, with the following command:

   ##/opt/cassandra/bin/nodetool status

   The output displays the Host ID of the relevant server.

2. Run the following command:

   /opt/cassandra/bin/nodetool removenode <Host ID of the existing server>

   For example: /opt/cassandra/bin/nodetool removenode 33e76454-22f0-4b15-985a-e62cd8a27d4b

3. Verify that the server was removed from the Cassandra cluster:

   ##/opt/cassandra/bin/nodetool status

7.2 Defining the New Server's Role

You need to define the new server. This is done by configuring the GRUB boot parameters. There are mandatory parameters and optional parameters that are only required if relevant for the server deployment.

To configure the GRUB boot parameters:

1. In the prompt line, after **F5-TRAFFIX_SDC:traffix/kickstart/kickstart.cfg**, add the parameters, as follows:

Proprietary and Confidential Information of F5 Networks

Note: You can press the **TAB** key to enable editing.

Table 10: Mandatory Parameters

Name	Value Description
server	master/minion Note: When adding an SDC component (CPF/FEP), you can only select minion
hostname	The server's hostname Note: The hostname must be identical (case sensitive) to the value defined for the vm element in the Site Topology file
master0	The master (Installer) IP address
master1	ip address of the second master vInstaller
ip	The IP address is from the management network interface for minion and master Installer servers
netmask	netmask for the ip address defined above (only needed for IPv4, CIDR is not supported)
device	eth device the ip address defined above Note: On some HP blades eth8 is mgmt

Table 11: Optional Parameters

Name	Value Description
vlan	vlan number for interface (if vlan defined)
gw	default gateway (need to be mandatory if server = master)
debug	debug=yes enable salt log with debug
dns	DNS

The new server is now installed with an Operating System with its defined role.

Proprietary and Confidential Information of F5 Networks

7.2.1 Removing the Replaced Server Master Installer Connection

After the new server's role is identified and it is registered with the master Installers, you need to erase the replaced server's registration from both master Installers.

To remove the replaced server's registration:

1. Run the following command on all master Installer servers:

   salt-run --timeout 30 manage.down removekeys=True

Note: If you have replaced one of the master Installers, then you need to run this command on the second master Installer.

To verify that the replaced server's registration has been removed:

1. Run the following command on all master Installer servers:

   salt-run manage.status

   If the registration was successfully removed then the replaced server hostname should not appear under the **down:**prompt.

7.3 Uploading SDC Components to the New Server

Once the Operating System has been installed on the new server and the server's role has been identified, the SDC components that were previously hosted on the replaced server are now uploaded to the new server based on the GRUB parameters and the Site Topology File.

Note: If the replaced server was one of the master Installers, then you need to restart the Salt-minion service for each server with the following command: **monit restart salt-minion**

When replacing a server that hosts one of the master Installers, manually check that any changes made to the /srv/salt/ files are copied to the newly replaced master Installer, so that both /srv/salt/ files on both master Installers are identical.

Proprietary and Confidential Information of F5 Networks

7.3.1 Installing the SDC Components

After a new minion server is registered and up and running, the master Installer references it and checks to see if the relevant SDC components are installed. If they are not installed on the server, then the Master Installer initiates the installation process. While this process occurs automatically every hour, it is recommended that when replacing a server, you force this verification and installation process to occur.

To install the SDC applications on the new server:

1. Run the following command on one of the master Installers:

   salt <NEW-RESTORED-SERVER-MINION-key> state.highstate

7.4 Verifying the Replacement Process

You can check if the new server was successfully registered with the relevant SDC components with the appStatus API Request.

7.4.1 Application Status per Server

This API request checks the status of a specific server. The response includes the relevant status codes for successfully installed applications. In addition, as with all other API requests, there are related command execution codes.

7.4.1.1 appStatus API Request

```
curl -ksi https://<master_IP_address>::8000 -H "Accept: application/x-yaml" -H "X-Auth-Token:<Token>" -d client="runner" -d fun="traffix.appStatus" -d tgt="*"-d apps=True (optional for apps list)
```

7.4.1.2 Command Execution Codes for appStatus API Request

Table 12: appStatus Command Error Codes

Exit Code	Description
-50	Failed to validate site topology file - check site topology file
-51	Installation not started yet

Proprietary and Confidential Information of F5 Networks

Exit Code	Description
-52	Could not get information from DB

7.4.1.3 Return Codes for appStatus API Request

Table 13: appStatus Return Codes

Exit Code	Description
14002	Pending Machine Start
14003	Pending SDC Installation
14004	Pending SDC Start
14006	Pending SDC Stop
15002	Fail VM Start
15003	Fail To Install SDC
15004	Fail To Start SDC
15006	Failed To Stop SDC
13000	Suspended
12000	Successfully installed

Proprietary and Confidential Information of F5 Networks

8. Backing up and Restoring a Site

This backup and restore procedure provides a way for you to back up an SDC site's configuration data. In the event that an SDC site shuts down, and you install a new site to replace the old one, you can restore the backed up site configuration data to the newly replaced site.

8.1 Prerequisite

The restore procedure assumes that the deployment is geo-redundant, so that when services are stopped on one site they can continue on the geo-redundant site.

8.2 Backing up a Site's Data

The following steps explains how to back up the site data.

To back up the site data:

1. Copy the */data/backup* folder from each OAM server.
2. Copy the */srv/salt/<>* folder from each master Installer server.

The data is now backed up.

8.3 Restoring a Site's Data

When restoring a site, you need to also restore the site configuration data that was previously backed up.

Note: The Site Topology file for the restored site must have the same interconnect and signaling IP addresses as was previously defined for it to be synched with the XML schema saved in the external storage. This supports the synchronization of Tripos, CPF and FEP components between existing and restored site. It also ensures that the new site will point to the cloned external storage (cinder volumes that store the Cassandra data). Only the management IP addresses are changed for the new servers in the new site.

Proprietary and Confidential Information of F5 Networks

To restore the backed up site data to the newly restored site's OAMs:

1. Install the site from scratch.
2. Run the following command: **/opt/traffix/scripts/traffixApps.sh stop**
3. Restore the */data* folder.
4. Run the following command: **/opt/traffix/scripts/traffixApps.sh start**
5. On each master Installer server, restore the */srv/salt/* folder from the backed-up server.
6. On one of the master Installer server, run the following commands:

   /etc/init.d/salt-master restart

   salt "*" state.highstate

The site configuration data is now restored on the newly restored site.

Proprietary and Confidential Information of F5 Networks

9. Backing up and Restoring a Cassandra Database

This backup and restore procedure provides a way for you to back up an SDC site's Cassandra data. In the event that Cassandra stops working, you can restore the backed-up Cassandra data.

9.1 Backing up a Site's Cassandra Database

The following steps explains how to back up the Cassandra database by creating a snapshot.

To back up the Cassandra data:

1. Create Cassandra backup snapshots for each OAM Cassandra with a <backup name>:

   /opt/cassandra/bin/nodetool -h localhost -p 7199 snapshot -t <backup-name>

   Under /data/cassandra/data, each table in each Cassandra keyspace folder, now includes the new snapshots named with the user given name <backup-name>.

2. Go to */data/cassandra/data/<keyspace_name>/<table_name>* that has the newly created snapshots and copy the snapshots to a backup location.

Note: To ensure full backup, it is recommended to copy the newly created snapshots to a separate backup server.

9.2 Restoring the Cassandra Database

Assuming you have backed up the Cassandra data, you will be able to restore the data in the event that Cassandra shuts down unexpectedly.

To restore the backed up Cassandra data:

1. Perform the following steps on each OAM database:

   c. **monit stop Cassandra_instance_name**

Proprietary and Confidential Information of F5 Networks

d. Clear all files in the commitlog directory:

rm -rf /data/cassandra/commitlog/*

e. Delete all *.db files in:
data_directory_location/keyspace_name/table_name directory.

Note: Do not delete the */snapshots and /backups* subdirectories:

find /data/cassandra/data -name "*\.db" | egrep -v "snapshot|backup" | xargs rm –rf

---

f. Copy each table snapshot folder content into the relevant table directory:

/data/cassandra/data/<keyspace_name>/<table_name>.

Note: Check each keyspace directory, in each table, in each sub-folder, for the relevant snapshots. For example, a new snapshot, named "latest" will be saved in each table in each keyspace:

/data/cassandra/data/topology/webui-4243235252/snapshots/latest/

---

g. **monit start cassandra**

2. After all the OAM servers are up and running, run the following command on one of the OAM servers:

/opt/Cassandra/bin/nodetool repair

3. Wait for the process to finish.
4. Verify that the Cassandra data is restored:
   a. Run **monit summary** on all the OAM servers
   b. Verify that each OAM process is up and running.

Proprietary and Confidential Information of F5 Networks

10. Monitoring a Site's Status

During the maintenance procedures, you can monitor the site status with a dedicated REST API request. This API request checks the status of all site servers. The response includes the relevant status codes for the successfully installed applications on all servers within a site. In addition, as with all other API requests, there are related command execution codes.

10.1 Verifying the Authentication Token

Monitoring site status is performed with API requests. To apply the REST API, you need to have a valid authentication token that is not expired.

If the authentication token is expired, you need to request a new one (see *Generating an Authentication Token*).

10.2 siteStatus API Request

The following is the API siteStatus request:

```
curl -ksi https://<master_IP_address>:8000 -H "Accept: application/x-yaml" -H "X-Auth-Token:<Token>" -d client="runner" -d fun="traffix.siteStatus" -d apps=True (optional for apps list)
```

10.2.1 Command Execution Codes for siteStatus API Request

Table 14: siteStatus Command Error Codes

Exit Code	Description
-50	Failed to validate site topology file - check site topology file
-51	Installation not started yet
-52	Could not get information from DB

10.2.2 Return Codes for siteStatus API Request

Table 15: siteStatus Return Codes

Exit Code	Description
14010	Installation is Running

Proprietary and Confidential Information of F5 Networks

Exit Code	Description
15010	Installation Failed
12010	Installation Finished Successfully

10.2.3 siteStatus Answer Example

The following is an example of an answer from the Installer to a Site Status Query (apps =True).

```
Result:

HTTP/1.1 200 OK
Content-Length: 613
Access-Control-Expose-Headers: GET, POST
Access-Control-Allow-Credentials: true
Vary: Accept-Encoding
Server: CherryPy/3.2.2
Allow: GET, HEAD, POST
Cache-Control: private
Date: Tue, 15 Dec 2015 15:32:11 GMT
Access-Control-Allow-Origin: *
Content-Type: application/x-yaml
Set-Cookie: session_id=50479aecb5a14f333a37a0656c8b2694dc2ad4d8;
expires=Wed, 16 Dec 2015 01:32:11 GMT; Path=/

return:
- - - Site-Status-Code: 12010
    - Installed-Apps:
        sdclab005-16-cpf-1:
        - cpf1
        sdclab005-16-fep-1:
        - fep1
        sdclab005-16-master-1:
        - oamDB-unique
        - vnf
```


Proprietary and Confidential Information of F5 Networks

```
        sdclab005-16-master-2:
        - oamDB-unique
        - vnf
        sdclab005-16-oam-1:
        - CM1
        - NmsAgent1
        - Cassandra1
        - WebUI1
        sdclab005-16-oam-2:
        - CM1
        - NmsAgent1
        - Cassandra1
        - WebUI1
        sdclab005-16-tripo-1:
        - tripo1
        sdclab005-16-tripo-2:
        - tripo1

:49:56 GMT; Path=/

return:
-                                                    <machine
hostname>:
    <app>:
    - installed
```

10.3 API Response Codes

Each request has an associated API output success/error code depending on its status.

Table 16: API Status Output Codes

Status	Code	Description
Success	12000	The entire request flow was completed successfully
Failure	150xx	The request flow failed and was terminated

Proprietary and Confidential Information of F5 Networks

Status	Code	Description
Pending internal	140xx	The Installer is waiting for an internal operation to complete. For example: waiting for an application to be installed
Pending connection	130xx	The relevant application service is down.

Proprietary and Confidential Information of F5 Networks

11. Removing an SDC Site

This procedure describes how to remove an SDC site from a multi-site EMS deployment. As long as the other SDC sites in the deployment are up and running, there is no impact on traffic.

To shut down and remove the site processes:

All monit monitoring processes that are part of the SDC site that you want to remove must be shut down prior to removing the site.

1. From each site (that is to be removed) server, run the following command:

   # monit stop all

To remove an SDC site from an EMS deployment:

1. Run the following command to stop all processes except the oamDB process that is running on each EMS server: **# monit stop <process name>**
2. Verify that the relevant processes are down on each EMS server, with the following command: **# monit summary**

   The expected output from this command for each process is: “Not Monitored”.
3. On each EMS server, move the site's data directory from the Configuration Manger’s data folder to a backup location, with the following command:

   # mv /data/backup/<site name> <backup location>

Note: Make sure there is enough disk space on the backup location for target partition.

---

4. On one of the EMS servers, remove the "nms_<site name>_ems" keyspace, with the following commands:
   a. Log in to Cassandra using cql: **# /opt/cassandra/bin/cqlsh <IP address of the EMS server>**

Proprietary and Confidential Information of F5 Networks

b. Remove the keyspace, with the following command: # **drop KEYSPACE IF EXISTS nms_<site name - lower case>_ems**

Note: If the following warning message is displayed, ignore it and proceed with the next step: "Warning: schema version mismatch detected, which might be caused by DOWN nodes; if this is not the case, check the schema versions of your nodes in system.local and system.peers. OperationTimedOut: errors={}, last_host=EMS-<site name>"

---

5. Restart all the EMS site processes, with the following command: **# monit start all**
6. Verify that all processes are up and running on each EMS server, with the following command: **# monit summary**

   The expected output from this command for each process is: "Running".
7. Run a clean SDC site script on one of the EMS servers, with the following commands:

   # cd /opt/traffix/scripts/

   python clearSdcSiteFromEms.py <site name - lower case>

   The expected output from this command is the following:

   [root@EMS scripts]# python clearSdcSiteFromEms.py <SDC site name>

   Removing site data from DB

   Finished to remove site data from DB successfully

   Finished successfully

To remove the site servers that host Cassandra:

For the site that is to be removed, you need to delete each of the site servers by deleting their Host ID addresses:

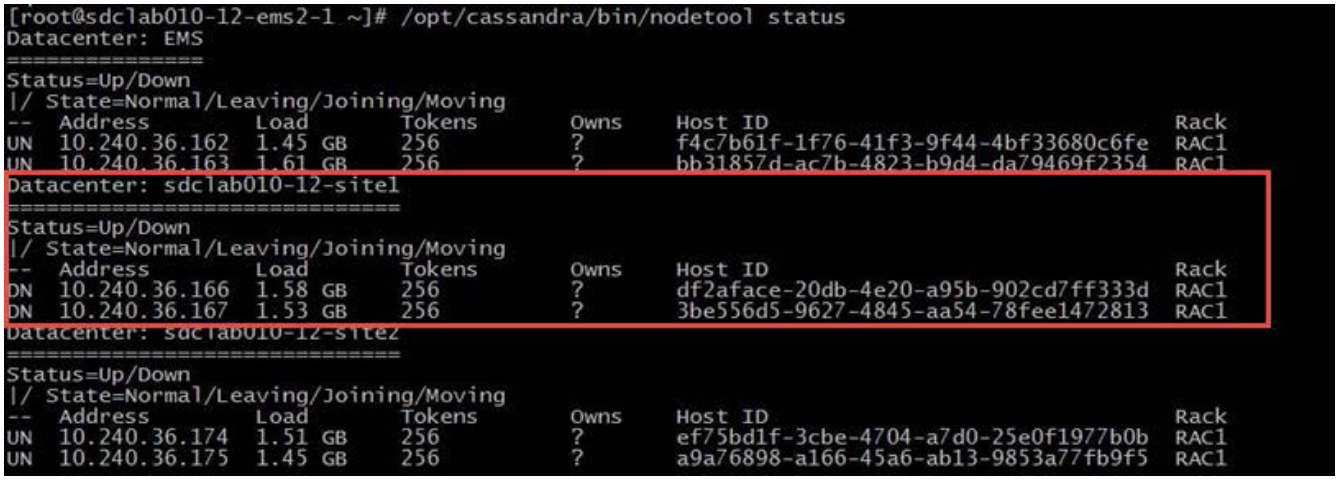
Proprietary and Confidential Information of F5 Networks

1. Retrieve and verify the server Host IDs of the relevant site with the following command:

##/opt/cassandra/bin/nodetool status

The output displays the site's server Host IDs of all the servers in Cassandra clusters.

Figure 4: Example of a Site's Servers Host IDs



2. Identify the relevant site that is to be removed and its Host IDs and run the following command for each Host ID of each server of the identified site:

/opt/cassandra/bin/nodetool removenode <Host ID of the identified site server>

For example: /opt/cassandra/bin/nodetool removenode 33e76454-22f0-4b15-985a-e62cd8a27d4b

3. Verify that each of the removed site servers were removed from the Cassandra cluster with the following command:

##/opt/cassandra/bin/nodetool status

Proprietary and Confidential Information of F5 Networks

12. Configuring SDC-Server Routes

Using API commands, you can add or remove routes when servers are added/removed with networks that were not previously recognized by the SDC. The addRoutes/deleteRoutes API can be used to add/remove route from any network.

Note: You can also add or delete routes by applying the route-api.py script. For more details, see *Adding and Deleting Routes via Script*.

Warning: Verify that any changes made with the addRoutes/deleteRoutes API will not cause the network configuration to stop running, as there is no rollback option. For example, changes to the management system, may cause the SDC to stop running. It is recommended that you take the necessary precautions to back-up your system.

12.1 Add Routes

The addRoutes API sends a request to the master Installer to add a route to the server. The addRoutes API request refers to the new XML snippet file (@addroutes.xml) and by doing so the newly created XML is saved and merged with the Site Topology file in Cassandra.

12.1.1 Prerequisites

12.1.1.1 Verifying the Authentication Token

Adding a route is performed with an API request. To apply the REST API, you need to have a valid authentication token that is not expired.

If the authentication token is expired, you need to request a new one (see *Generating an Authentication Token*).

12.1.2 Updating the Site Topology File with Added Route

The site topology file must be updated to include definitions for the added route. This is done by creating a new Topology XML file (a snippet) that is then uploaded to the Cassandra database and merged with the Site Topology file, once the addRoutes API flow request is completed.

Proprietary and Confidential Information of F5 Networks

To create a new Topology XML file:

1. Fill in the following parameters:

   - vm name
   - interface network

Note: The vm name is the name of the server that the route is being added to.

   - route name, net4, net6, ip4sub, ip6sub, gateway

Refer to the *F5 SDC Bare Metal Installation Guide: Site Topology File Structure* for more information about these parameters.

The following is an example of a Topology file for adding a route on a server:

```
hosts=<?xml version="1.0" encoding="UTF-8"?>
<topology>
        <vms>
                <vm name="sdclab005-12-fep-1">
                        <interfaces>
                                <interface network="sig-1"
ip4="10.1.81.164" ip6="" dev="eth2" name="sig-1-ip1">
                                        <route name="fep1-vip1-route1"
net4="10.1.19.0" net6="" ip4sub="255.255.255.0" gateway="10.1.81.1"/>
                                </interface>
                        </interfaces>
                </vm>
        </vms>
</topology>
```

12.1.3 addRoutes API Request

The following is the addRoutes API request:

```
# curl -ksi https://<master_IP_address>:8000 -H "Accept: application/x-
yaml" -H "X-Auth-Token:<Token>" -d client="runner" -d
fun="traffix.addRoutes" -X POST -d @addroutes.xml
```


Proprietary and Confidential Information of F5 Networks

12.2 Delete Routes

The Orchestrator sends a deleteRoutes API Request to the master vInstaller to delete the route to the server. The deleteRoutes API request refers to the new XML snippet file (@deleteroutes.xml) and by doing so the newly created XML is saved and merged with the Site Topology file in Cassandra.

12.2.1 Prerequisites

12.2.1.1 Verifying the Authentication Token

Deleting a route is performed with an API request. To apply the REST API, you need to have a valid authentication token that is not expired.

If the authentication token is expired, you need to request a new one (see *Generating an Authentication Token*).

12.2.2 Updating the Site Topology File with Deleted Route

The site topology file must be updated to remove the previously defined route network definitions for the deleted route. This is done by creating a new Topology XML file (a snippet) that is then uploaded to the Cassandra database and merged with the Site Topology file, once the deleteRoutes API flow request is completed. The API request refers to the XML file (@deleteroutes.xml.), see for more information.

To create a new Topology XML file:

2. Fill in the following parameters:

   - vm name
   - interface network

Note: The vm name is the name of the server that the route is being deleted from.

   - route name, net4, net6, ip4sub, ip6sub, gateway

Refer to the *F5 SDC Bare Metal Installation Guide: Site Topology File Structure* for more information about these parameters.

Proprietary and Confidential Information of F5 Networks

12.2.3 deleteRoutes API Request

The following is the deleteRoutes API request:

```
# curl -ksi https://<master_IP_address>:8000 -H "Accept: application/x-yaml" -H "X-Auth-Token:<Token>" -d client="runner" -d fun="traffix.deleteRoutes" -X POST -d @addroutes.xml
```

12.3 Adding and Deleting Routes via Script

The addRoute/deleteRoute script can be applied when you want to add or delete routes one by one or a group of routes.

12.3.1 Adding and Deleting a single route using the CLI

To add/delete a route:

1. Change the directory on one of the master Installer servers:

   cd /srv/traffix/pillar/

   The following script is included for adding/deleting a single route:

```
./route-api.py -r {addRoutes/deleteRoutes} -v 4/6 -i {source IP} -n {destination Net} -m {destination Netmask} -g {GATEWAY} -N {Name}
```

2. Customize the script by selecting the following:

   <addRoutes/deleteRoutes>

   a. Fill in the following parameters:

   i. < 4/6>: IP version 4 or 6

   ii. <source IP>

   iii. <destination Network>

   iv. <destination netmask>: destination network mask

   v. < GATEWAY >: source IP gateway

   vi. < route name >:-N {Name}

Proprietary and Confidential Information of F5 Networks

The following is an example of an add route script:

```
 ./route-api.py -r addRoutes -v 4 -i 10.3.118.10 -n 10.1.71.0 -m
255.255.255.0 -g 10.3.118.1 -N route1
```

3. Run the script in file process mode with the following command:

   ./route-api.py -f

   As part of this command, the script validates the inputted data. Following validation, the selected action (add/delete) is performed.

12.3.2 Adding and Deleting a single route using the /tmp/input file

Note: This method is only supported for IPv4.

To add/delete a group of routes:

1. Create a */tmp/input* file on the same master Installer server that you run the **cd /srv/traffix/pillar/** command.

   The */tmp/input* file follows the format of the./route-api.py -r script with the defined parameters:

```
./route-api.py -r {addRoutes/deleteRoutes} -v 4 -i {source IP} -n
{destination Net} -m {destination Netmask} -g {GATEWAY} -N {Name}
```

The following is an example of a group of routes to be added by the script:

```
addRoutes:4:10.3.118.11:10.1.82.0:255.255.255.0:10.3.118.1:route-1
addRoutes:4:10.3.118.10:10.1.82.0:255.255.255.0:10.3.118.1:route-2
addRoutes:4:10.3.118.12:10.1.82.0:255.255.255.0:10.3.118.1:route-3
addRoutes:4:10.3.118.13:10.1.82.0:255.255.255.0:10.3.118.1:route-4
addRoutes:4:10.3.118.14:10.1.82.0:255.255.255.0:10.3.118.1:route-5
addRoutes:4:10.3.118.3:10.1.82.0:255.255.255.0:10.3.118.1:test-route-1
addRoutes:4:10.3.118.4:10.1.82.0:255.255.255.0:10.3.118.1:test-route-2
where:
column 1 is the operation addRoutes/deleteRoutes
column 2 is the IP version 4 or 6
```


Proprietary and Confidential Information of F5 Networks

```
column 3 is the <source IP>
column 4 is the <destination network>
column 5 is the <destination network mask>
column 6 is the <source IP gateway>
column 7 is the <route name>
```

2. Run the script in file process mode with the following command:

   ./route-api.py -f

   As part of this command, the script validates the inputted data. Following validation, the selected action (add/delete) is performed.

Note: When deleting a route, the same subnet mask IP value must be used as was included in the addRoute script. For troubleshooting, refer to the /var/log/route-api.log. Show Routes

---

The showRoutes API request provides information of all the existing network routes between a server and the SDC.

12.3.3 Prerequisites

12.3.4 Verifying the Authentication Token

Showing all the routes is performed with an API request. To apply the REST API, you need to have a valid authentication token that is not expired.

If the authentication token is expired, you need to request a new one (see *Generating an Authentication Token*).

12.3.5 showRoutes API Request

The API request refers to the SDC component <hostname> in which you want to retrieve the network routes information. The request can be for more than one server. The following is the showRoutes API request:

Proprietary and Confidential Information of F5 Networks

```
# curl -ksi https://<master_IP_address>:8000 -H "Accept: application/x-yaml" -H "X-Auth-Token:<Token>" -d client="runner" -d fun="traffix.showRoutes" -d hosts='<hostname1>,<hostname2>'
```

12.3.5.1 showRoutes API Request Example

The following is an example of the showRoutes API request and answer based on showing the routes for one FEP-server connection.

```
# curl -ksi https://localhost:8000 -H "Accept: application/x-yaml" -H "X-Auth-Token:cd245f57008d3a0054d510d9b6d6237902440bd7" -d client="runner" -d fun="traffix.showRoutes" -d hosts='sdclab005-12-fep-1'

HTTP/1.1 200 OK
Content-Length: 284
Access-Control-Expose-Headers: GET, POST
Access-Control-Allow-Credentials: true
Vary: Accept-Encoding
Server: CherryPy/3.2.2
Allow: GET, HEAD, POST
Cache-Control: private
Date: Tue, 09 Feb 2016 13:32:38 GMT
Access-Control-Allow-Origin: *
Content-Type: application/x-yaml
Set-Cookie: session_id=cd245f57008d3a0054d510d9b6d6237902440bd7; expires=Tue, 09 Feb 2016 23:32:38 GMT; Path=/

return:
- sdclab005-12-fep-1:
  - - name: fep1-vip1-route12
    - interface: sig-1-ip1
    - net4: 10.1.29.0
    - ip4sub: 255.255.255.0
    - gateway: 10.1.81.1
  - - name: fep1-vip1-route2
```


Proprietary and Confidential Information of F5 Networks

```
  - interface: sig-1-ip1
  - net4: 10.1.70.0
  - ip4sub: '24'
  - gateway: 10.1.81.1
```


Proprietary and Confidential Information of F5 Networks

13. Replacing NTP Servers

Using an API command, you can replace the NTP servers that were previously configured to synchronize time zones between other servers.

The changeNtp API request to the master Installer is to replace the NTP servers. This request deletes all the existing NTP servers and replaces them with the new NTP servers that are defined in the Site Topology XML snippet file. The changeNtp API request refers to the new XML snippet file (@changentp.xml) and by doing so the newly created XML is saved and merged with the Site Topology file in Cassandra.

13.1.1 Prerequisites

13.1.1.1 Verifying the Authentication Token

Changing an NTP server is performed with an API request. To apply the REST API, you need to have a valid authentication token that is not expired.

If the authentication token is expired, you need to request a new one (see *Generating an Authentication Token*).

13.1.2 Updating the Site Topology File with New NTP Servers

The site topology file must be updated to include the definitions for any new NTP server. This is done by creating a new Topology XML file (a snippet) that is then uploaded to the Cassandra database and merged with the Site Topology file, once the changeNtp API flow request is completed.

To create a new Topology XML file:

1. Fill in the following parameters:
   - ntpServer name
   - ip

Note: The ntpServers are the new NTP servers and for each one you need to define a name (ntpServer name) and IP address. You can add multiple NTP servers, but no

Proprietary and Confidential Information of F5 Networks

matter how many are added, all of the previously configured NTP servers are removed.

Refer to the *F5 SDC Bare Metal Installation Guide: Site Topology File Structure* for more information about these parameters.

The following is an example of a Topology file for changing an NTP server:

```
ntpServers=<?xml version="1.0" encoding="UTF-8"?>
<topology>
        <siteProperties name="SDC_Site">
                <ntpServers>
                        <ntpServer name="first" ip="192.168.16.5"/>
                        <ntpServer name="second" ip="192.168.16.6"/>
                </ntpServers>
        </siteProperties>
</topology>
```

13.1.3 changeNtp API Request

The following is the changeNtp API request:

```
# curl -ksi https://<master_IP_address>:8000 -H "Accept: application/x-
yaml" -H "X-Auth-Token:<Token>" -d client="runner" -d
fun="traffix.changeNtp" -X POST -d @changentp.xml
```

A successful response will include the following response:

```
return:
- 0
- Successfully changed the NTP servers
```


Proprietary and Confidential Information of F5 Networks

14. Configuring Networks

Using API commands, you can add or remove networks to a site.

14.1 Add Networks

The addNetwork API sends a request to the master Installer to add a network to the site. The addNetwork API request refers to the new XML snippet file (traffix.addNetwork) and by doing so the newly created XML is saved and merged with the Site Topology file in Cassandra.

14.1.1 Prerequisites

14.1.1.1 Verifying the Authentication Token

Adding a network is performed with an API request. To apply the REST API, you need to have a valid authentication token that is not expired.

If the authentication token is expired, you need to request a new one (see *Generating an Authentication Token*).

14.1.2 Updating the Site Topology File with Added Network

The site topology file must be updated to include definitions for the added network. This is done by creating a new Topology XML file (a snippet) that is then uploaded to the Cassandra database and merged with the Site Topology file, once the addNetwork API flow request is completed.

To create a new Topology XML file:

1. Fill in the following parameters:
   - network name, net4, ip4sub, net6, ip6sub, role

Refer to the *F5 SDC Bare Metal Installation Guide: Site Topology File Strucutre* for more information about these parameters.

The following is an example of a Topology file for adding a network to a site:

```
networks=<?xml version="1.0" encoding="UTF-8"?>
```


Proprietary and Confidential Information of F5 Networks

```
<topology>
<topology>
          <networks>
                              <network name="mgmt" net4="10.240.9.0"
ip4sub="255.255.255.0" net6="" ip6sub="" role="mgmt"/>
                              <network name="ic" net4="10.1.79.0"
ip4sub="255.255.255.0" net6="" ip6sub="" role="ic"/>
                               <network name="sig-1" net4="10.1.81.0"
ip4sub="255.255.255.0" net6="" ip6sub="" role="sig"/>
           </networks>
</topology>
```

The following is the addNetwork API request:

```
# curl -ksi https://<master_IP_address>:8000 -H "Accept: application/x-
yaml" -H "X-Auth-Token:<Token>" -d client="runner" -d
fun="traffix.addNetwork" -X POST -d @addnetwork.xml
```

Note: Before adding a FEP, check that the relevant network interfaces are in an UP admin state.

14.2 Delete Networks

The deleteNetwork API sends a request to the master Installer to delete a network to the site. The deleteNetwork API request refers to the new XML snippet file (traffix.deleteNetwork) and by doing so the newly created XML is saved and merged with the Site Topology file in Cassandra.

14.2.1 Prerequisites

14.2.1.1 Verifying the Authentication Token

Deleting a network is performed with an API request. To apply the REST API, you need to have a valid authentication token that is not expired.

If the authentication token is expired, you need to request a new one (see *Generating an Authentication Token*).

Proprietary and Confidential Information of F5 Networks

14.2.2 Updating the Site Topology File with Deleted Network

The site topology file must be updated to remove the previously defined network. This is done by creating a new Topology XML file (a snippet) that is then uploaded to the Cassandra database and merged with the Site Topology file, once the deleteNetwork API flow request is completed.

To create a new Topology XML file:

1. Fill in the following parameter:
   - network name

Refer to the *F5 SDC Bare Metal Installation Guide: Site Topology File* for more information about this parameter.

The following is an example of a Topology file for deleting a network to a site:

```
networks=<?xml version="1.0" encoding="UTF-8"?>
<topology>
        <networks>
                        <network name="sig-3"/>
                        <network name="sig-4"/>
                        <network name="sig-5"/>
                        <network name="sig-6"/>
        </networks>
</topology>
```

14.2.3 deleteNetwork API Request

The following is the deleteNetwork API request:

```
# curl -ksi https://<master_IP_address>:8000 -H "Accept: application/x-
yaml" -H "X-Auth-Token:<Token>" -d client="runner" -d
fun="traffix.deleteNetwork" -X POST -d @deletenetwork.xml
```


Proprietary and Confidential Information of F5 Networks

15. Recovering an ELK Master Node

EMS sites are installed with aFluentd, Elasticsearch and Kibana, as well as, Fluentd on an SDC site. During installation, these components are installed dynamically. The Elasticsearch (ELK) component is installed on four nodes, two on each node, of which one is a master and the other is a data node. The Elasticsearch service chooses the master nodes via an election process.

After installation, if an assigned Elasticsearch Master is down and not running on the configured node, the Elasticsearch service assigns the other Master (located on the second node) the active Master node. If the VM/cluster is down, and, consequently, the Elasticsearch service cannot reassign a new Master, you can run a disaster recovery script to recover a Master node.

 Note: If the downtime is temporary or part of a planned maintenance, then there is no need to run the disaster recovery script.

To check if a cluster is down (not recoverable):

1. Run one of the following commands

Get the Master node:

```
curl -k -XGET "https://<username>:<password>@<IP>:<Port - Can be 9200
or 9201 according to availability>/_cat/master"
```

The following error will be generated:

```
    {"error":{"root_cause":[{
"type":"master_not_discovered_exception","reason":null}
],"type":"master_not_discovered_exception","reason":null},"status":503}

    {"error":{"root_cause":[{
"type":"security_exception","reason":"unable to authenticate user
[elastic] for REST request [/_cat/master]","header":{"WWW-
Authenticate":["Bearer realm=\"security\"","ApiKey","Basic
```


Proprietary and Confidential Information of F5 Networks

```
realm=\"security\" charset=\"UTF-8\""]}}
],"type":"security_exception","reason":"unable to authenticate user
[elastic] for REST request [/_cat/master]","header":{"WWW-
Authenticate":["Bearer realm=\"security\"","ApiKey","Basic
realm=\"security\" charset=\"UTF-8\""]}},"status":401}
```

or

Get the cluster health:

```
                                                    curl -k -XGET
"https://<username>:<password>@<IP>:<Port - Can be 9200 or 9201
according to availability>/_cluster/health?pretty"
```

The following error will be generated:

```
    {"error":{"root_cause":[{
"type":"security_exception","reason":"unable to authenticate user
[elastic] for REST request [/_cluster/health]","header":{"WWW-
Authenticate":["Bearer realm=\"security\"","ApiKey","Basic
realm=\"security\" charset=\"UTF-8\""]}}
],"type":"security_exception","reason":"unable to authenticate user
[elastic] for REST request [/_cluster/health]","header":{"WWW-
Authenticate":["Bearer realm=\"security\"","ApiKey","Basic
realm=\"security\" charset=\"UTF-8\""]}},"status":401}


    {"error":{"root_cause":[{
"type":"master_not_discovered_exception","reason":null}
],"type":"master_not_discovered_exception","reason":null},"status":503}
```

To run a disaster recovery script to recover an ELK Master:

1. Stop the Elasticsearch data node:

   monit stop sdclab010-02-ems-1_elasticsearch_data

2. Disconnect the data node from the dead cluster:

Proprietary and Confidential Information of F5 Networks

ES_PATH_CONF=/etc/elasticsearch/elasticsearch_data/
/usr/share/elasticsearch/bin/elasticsearch-node detach-cluster

3. Stop the Elasticsearch master node:

   monit stop sdclab010-02-ems-1_elasticsearch_master

4. Bootstrap the master node to form a new cluster:

   ES_PATH_CONF=/etc/elasticsearch/elasticsearch_master/
   /usr/share/elasticsearch/bin/elasticsearch-node unsafe-bootstrap

5. Start the Elasticsearch master node:

   monit start sdclab010-02-ems-1_elasticsearch_master

6. Start the Elasticsearch data node:

   monit start sdclab010-02-ems-1_elasticsearch_data

7. Once the killed VM is back online, run the following commands:

Note: Make sure both instances are stopped prior to running the following commands:

---

   a. **ES_PATH_CONF=/etc/elasticsearch/elasticsearch_master/**
      /usr/share/elasticsearch/bin/elasticsearch-node detach-cluster

   b. **ES_PATH_CONF=/etc/elasticsearch/elasticsearch_data/**
      /usr/share/elasticsearch/bin/elasticsearch-node detach-cluster

Proprietary and Confidential Information of F5 Networks

16. Replace ELK Certificate

You can replace the ELK CA, certificate, and key files. These files are stored on the Salt master nodes (EMS and SDC sites) at /srv/salt/<versiob>/elk/global/ and are distributed to

- /etc/pki/certs/
- /etc/elasticsearch/certs/

Note: If you are replacing the certificate, it is recommended to contact F5 Support.

Before you upload the new files, you need to backup the current certification files. If you need to revert to the original files, there is also a rollback option.

To replace the ELK certificate and key:

1. Login via SSH to the EMS site 1
2. Backup the files at */srv/salt/<version>/elk/global/* by doing the following commands:
   - **# cd /srv/salt/<version>/elk/global/**
   - **# mkdir orig**
   - **# cp instance.* orig**
   - **# cp ca.crt orig**
3. Copy the new files to */srv/salt/<version>/elk/global/*

Note: Use the same exact name for each file (ca.crt, instance.crt, instance.key)

---

4. Repeat **steps 1-3** on EMS site 2 and then on each Salt master on all SDC sites
5. Run the following command and then make sure that there are no errors (on EMS site 1 or any of the Salt masters):

 # salt '*' state.highstate

Proprietary and Confidential Information of F5 Networks

6. Run the following commands to verify that the new files were successfully distributed to: */etc/pki/certs/* and */etc/elasticsearch/certs/*

 # ls -ltr /etc/pki/certs/

 # ls -ltr /etc/elasticsearch/certs/

7. Restart the ELK components (Fluent, Elasticsearch and Kibana) on EMS site 1 and EMS site 2:

 # monit restart

8. Restart the ELK component (Fluent) on the CPF nodes on the SDC sites:

 # monit restart

9. Login to the EMS WebUI
10. Navigate to **Reports** and make sure that the TDR Dashboard, Transaction Data Records and Traced Messages pages are valid and available.

To roll back to the original ELK CA, certificate, and key files:

1. Login via SSH to the EMS site 1
2. Navigate to */srv/salt/<version>/elk/global/* and restore the original files with the following commands:
   - **# cd /srv/salt/<version>/elk/global/**
   - **# cp orig/instance.* ./**
   - **# cp orig/ca.crt ./**
3. Repeat **steps 1-2** on EMS site 2 and then on each Salt master on all SDC sites
4. Run the following command and then make sure that there are no errors (on EMS site 1 or any of the Salt masters):
5. **# salt '*' state.highstate**

Proprietary and Confidential Information of F5 Networks

6. Run the following commands to verify that the new files were successfully distributed to: */etc/pki/certs/* and */etc/elasticsearch/certs/*

 # ls -ltr /etc/pki/certs/

 # ls -ltr /etc/elasticsearch/certs/

7. Restart the ELK components (Fluent, Elasticsearch and Kibana) on EMS site 1 and EMS site 2:

 # monit restart

8. Restart the ELK component (Fluent) on the CPF nodes on the SDC sites:

 # monit restart

9. Login to the EMS WebUI

10. Navigate to **Reports** and make sure that the TDR Dashboard, Transaction Data Records and Traced Messages pages are valid and available.

Proprietary and Confidential Information of F5 Networks

17. Changing the Log Servers

This release supports central logging. Each SDC component sends its logging information to the OAM machine. The OAM machines have a mount point for persistent storage. You have an option to change the log server IP addresses. This configuration is done by updating the Site Topology File and using the changeLogServers API request.

Note: This logging method is in addition to the direct forwarding method in which logs are forwarded from the CPF or FEP. To configure the log servers for the direct forwarding method, refer to the *F5 SDC User Guide*.

17.1 Prerequisites

The following are the prerequisites for changing the log servers.

17.1.1 Verifying the Authentication Token

Changing the log servers is performed with an API request. To apply the REST API, you need to have a valid authentication token that is not expired.

If the authentication token is expired, you need to request a new one (see *Generating an Authentication Token*).

17.2 Updating the Site Topology File with the New Log Servers

The Site Topology file must be updated to include definitions for the specific configuration elements for the new log server IP addresses. This is done by creating a new Topology XML file (a snippet) that is then uploaded to the Cassandra database and merged with the Site Topology file, once the changeLogServers API flow request is completed. The API request refers to the XML file (@changelogservers.xml.), as shown in the *changeLogServers API Request*.

To create a new Topology XML file:

1. Fill in the following parameter for the new log servers:

   - logServer name

Proprietary and Confidential Information of F5 Networks

- Refer to the *F5 SDC Bare Metal Installation Guide: Site Topology File* for more information about this parameter.
- The following is an example of a topology file for changing the log servers:

```
logServers=<?xml version="1.0" encoding="UTF-8"?>
<topology>
        <siteProperties name="SDC_Site">
        <logServers>
            <logServer name="peervm-118-01" ip="10.240.13.74"
type="syslog" protocol="tcp" port="10514"/>
            <logServer name="peervm-118-02" ip="10.240.13.75"
type="syslog" protocol="tcp" port="10514"/>
        </logServers>
        </siteProperties>
</topology
```

17.3 Sending a changeLogServer API Request

You need to send a changeLogServers API Request to the master Installer to change the system log servers' IP addresses. The changeLogServers API request refers to the new XML snippet file (@changelogservers.xml) and by doing so the newly created XML is saved and merged with the Site Topology file in Cassandra.

17.3.1 changeLogServers API Request

The following is the changeLogServers API request:

```
curl -ksi https://<master_IP_address>:8000 -H "Accept: application/x-
yaml" -H "X-Auth-Token:<Token>" -d client="runner" -d
fun="traffix.changeLogServers" -X POST -d @changelogservers.xml
```

17.3.1.1 changeLogServers API Request Example

The following is an example of the request and answer:

```
# curl -ksi https://localhost:8000 -H "Accept: application/x-yaml" -H
"X-Auth-Token:aa6790aa67ae5ce87715b66bf5bd58fc3ea4bdb5" -d
```


Proprietary and Confidential Information of F5 Networks

```
client="runner" -d fun="traffix.changeLogServers" -X POST -d
@changelogservers.xml

HTTP/1.1 200 OK
Content-Length: 292
Access-Control-Expose-Headers: GET, POST
Access-Control-Allow-Credentials: true
Vary: Accept-Encoding
Server: CherryPy/3.2.2
Allow: GET, HEAD, POST
Cache-Control: private
Date: Tue, 20 Oct 2015 17:46:10 GMT
Access-Control-Allow-Origin: *
Content-Type: application/x-yaml
Set-Cookie: session_id=3b3f4aec793bc67d5b562d8129609bcf7045158b;
expires=Wed, 21 Oct 2015 03:46:10 GMT; Path=/

return:
- 0
- Successfully changed the remote log servers
```


Proprietary and Confidential Information of F5 Networks

18. Adding DNS Servers

You can add up to three DNS servers. To add a DNS server, you need to update the Site Topology file with the nameserver parameter and its IP address, followed by an API request.

18.1 Prerequisites

The following are the prerequisites for changing the log servers.

18.1.1 Verifying the Authentication Token

Adding DNS servers is performed with an API request. To apply the REST API, you need to have a valid authentication token that is not expired.

If the authentication token is expired, you need to request a new one (see *Generating an Authentication Token*).

18.2 Updating the Site Topology File with the New nameservers

The Site Topology file (topology:siteProperties) must be updated to include the specific configuration elements for the new DNS server. This is done by creating a new Topology XML file (a snippet) that is then uploaded to the Cassandra database, by executing the *changeNameserver* APIrequest, and merged with the Site Topology file.

To create a new Topology XML file:

1. In the nameservers XML file, add the following parameter, with an IP address:

   - nameserver

   - The following is an example of a topology file for adding DNS servers:

```
nameservers=<?xml version="1.0" encoding="UTF-8"?>
<topology>
        <siteProperties name="sdclab010-15-ems">
        <nameservers>
                <nameserver index="1" ip="192.168.16.5"/>
                <nameserver index="2" ip="192.168.16.6"/>
```


Proprietary and Confidential Information of F5 Networks

```
        </nameservers>
        </siteProperties>
</topology>
```

18.3 Sending a changeNameserver API Request

You need to send a *changeNameserver* API Request to the master Installer. The *changeNameserver* API request refers to the new XML snippet file (@nameservers.xml) and by doing so the newly created XML is saved and merged with the Site Topology file in Cassandra.

18.3.1 changeNameservers API Request

The following is the *changeNameserver* API request:

```
curl -ksi https:// master_IP_address>:8000 -H "Accept: application/x-
yaml" -H "X-Auth-Token:b0e440fd6f4deff7841716c550d2d0d4d5d1c485" -d
client="runner" -d fun="traffix.changeNameservers" -X POST -d
@nameservers.xml
```

18.4 Removing DNS Servers

You can remove DNS servers by updating the Topology XML file and sending a *changeNameserver* API request.

Note: If you want to remove only one of the DNS servers, you need to remove all the DNS servers and then add back the ones you want with a new XML snippet and *changeNameserver* API request.

To create a new Topology XML file:

1. Use the following XML (without the nameserver element):

```
nameservers=<?xml version="1.0" encoding="UTF-8"?>
<topology>
        <siteProperties name="sdclab010-15-ems">
```


Proprietary and Confidential Information of F5 Networks

```
</topology>
```

2. Send the *changeNameserver* API request

Proprietary and Confidential Information of F5 Networks

Glossary

The following tables list the common terms and abbreviations used in this document.

Table 17: Common Terms

Term	Definition
Answer	A message sent from one Client/Server Peer to the other following a request message
Client Peer	A physical or virtual addressable entity which consumes AAA services
Data Dictionary	Defines the format of a protocol's message and its validation parameters: structure, number of fields, data format, etc.
Destination Peer	The Client/Server peer to which the message is sent
Geo Redundancy	A mode of operation in which more than one geographical location is used in case one site fails
Master Session	The session for which the routing selection is performed based on the routing rules (Slave Sessions are applied with routing rules inherited from the Master Session)
Orchestrator	A workflow management solution to automate the creation, monitoring, and deployment of resources in your environment
Origin Peer	The peer from which the message is received
Pool	A group of Server Peers
QCOW2	A file format for disk image files
RADIUS	Remote Authentication Dial In User Service
REST	Representation of a resource between a client and server (**Re**presentational **S**tate **T**ransfer)
Request	A message sent from one Client/Server peer to the other, followed by an answer message
RPM	RPM Package Manager

 Proprietary and Confidential Information of F5 Networks

Term	Definition
Salt-API	Manages and communicates between an Orchestrator and network master and minion servers
SDC Site	The entire list of entities working in a single site
Server Peer	A physical or virtual addressable entity which provides AAA services
Session	An interactive information interchange between entities
Slave (Bound) Session	A session which inherits properties from a master session
Transaction	A request message followed by an answer message
Tripo	Session data repository
vCenter	Vmware Virtual Infrastructure tool for centralized management of multiple hypervisors and enabling functionalities
Virtual Server	A binding point used by SDC to communicate with the Remote Peers (Clients and Servers)

Table 18: Abbreviations

Term	Definition
AAA	Authentication, Authorization and Accounting
ACL	Access Control List
AF	Application Function
API	Application Programming Interface
AVP	Attribute Value Pair
CLI	Command Line Interface
CPF	Control Plane Function
DEA	Diameter Edge Agent
DRA	Diameter Routing Agent

 Proprietary and Confidential Information of F5 Networks

Term	Definition
EMS Site	Element Management System Site
FEP-In	In-Front End Proxy
FEP-Out	Out-Front End Proxy
HA	High Availability
HSS	Home Subscriber Server
HTTP	Hypertext Transfer Protocol
IaaS	Infrastructure as a Service
IMS	IP Multimedia Subsystem
JMS	Java Message Service
KPI	Key Performance Indicator
LDAP	Lightweight Directory Access Protocol
LTE	Long Term Evolution
MME	Mobility Management Entity
NGN	Next Generation Networking
Node	Physical or virtual addressable entity
OAM	Operation, Administration and Maintenance
OCS	Online Charging System
OVF	Open Virtualization Format
PCEF	Policy and Charging Enforcement Function
PCRF	Policy and Charging Rules Function
PLMN	Public Land Mobile Network
SCCP	Signaling Connection Control Part
SCTP	Stream Control Transmission Protocol
SDC	Signaling Delivery Controller

 Proprietary and Confidential Information of F5 Networks

Term	Definition
SDC Site	The entire list of entities working in a single site
SNMP	Simple Network Management Protocol
SS7	Signaling System No. 7
TCP	Transmission Control Protocol
TLS	Transport Layer Security
UDP	User Datagram Protocol
UE	User Equipment
URI	Universal Resource Identification.
VIP	Virtual IP
VM	Virtual Machine
VNFC	Virtualized Network Function Component
VPLMN	Visited Public Land Mobile Network
Web UI	Web User Interface
WS	Web Service

 Proprietary and Confidential Information of F5 Networks