



Signaling Delivery Controller

Bare Metal System Installation Guide

5.1

Catalog Number: RG-016-51-28 Ver. 14

Publication Date: November 2020



Legal Information

Copyright

© 2005-2020 F5 Networks, Inc. All rights reserved.

F5 Networks, 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 Networks, OpenBloX, OpenBloX (design), Rosetta Diameter Gateway, Signaling Delivery Controller, SDC, Traffix, and Traffix [DESIGN] are trademarks or service marks of F5 Networks, 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 Networks. The information in this document may be changed at any time without notice.

About F5 Networks

F5 Networks (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 Installation Guide

Catalog Number: RG-016-51-28 Ver. 14

Publication Date: November 2020

Document Objectives

This document describes the necessary procedures to set up and install bare metal deployments of SDC and EMS sites.



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

Document History

Revision Number	Change Description	Change Location
Ver. 2 – January 2017	Upload Topology command. Added description of ports used by the SDC	<i>Uploading the Site Topology File</i> <i>Port Settings Used by the SDC</i>
Ver. 3 – February 2017	Edited description of creating a Site Topology File. Added ports used by the SDC	<i>Creating a Site Topology File</i> <i>Port Settings Used by the SDC</i>
Ver. 4 – April 2017	Added in description of data volume size	<i>Data Storage Volume</i>
Ver. 5 – May 2017	Added site topology file structure.	<i>Site Topology File Structure</i>
Ver. 6 – June 2017	Updated procedure for uploading the Site Topology File.	<i>Uploading the Site Topology File to the Installer Machines</i>
Ver. 7 – August 2017	Updated ports section	<i>Port Settings Used by the SDC</i>
Ver. 8 – September 2017	Updated ports section	<i>Port Settings Used by the SDC</i>



Revision Number	Change Description	Change Location
Ver. 9 – December 2017	Updated uploading site topology file command. Added note about the number of required OAMs.	<i>Uploading the Site Topology File; SDC Installed Components</i>
Ver. 10 – June 2018	Added nameserver (optional) to site topology file	<i>Table 25: Elements Defined in siteProperties</i>
Ver. 11 – January 2019	Added note about max length for hostname parameter	<i>Table 4: Mandatory Parameters</i>
Ver. 12 – December 2019	Added note to SDC Component description	<i>SDC Installed Components</i>
Ver. 13 – March 2020	Updated firmware and RHEL supported version numbers	<i>Supported Operating System; Installing the Operating System</i>
Ver. 14 – November 2020	Removed references to Splunk, and updated to ELK	<i>Throughout document</i>

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
<i>Normal Text Italic</i>	Links to figures, tables, and sections in the document, as well as references to other documents
<code>Script</code>	Language scripts
<code>Courier</code>	File names





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

1. Installation Overview	1
1.1 General Prerequisites.....	2
1.2 Prerequisites	2
1.2.1 Completing a Site Survey	2
1.2.2 Installing the Hardware.....	3
1.2.3 Data Storage Volume.....	3
1.2.4 Accessing the ISO Image	4
1.2.5 Supported Operating System.....	4
1.2.6 Creating a Site Topology File.....	5
2. Introduction	6
2.1 SDC Installed Components.....	6
2.1.1 Active/Active High Availability.....	7
3. Performing the Installation	8
3.1 Installing Multiple SDC Sites Managed by EMS Site	8
3.2 Setting up the Site Machines	8
3.2.1 Installing the Operating System.....	8
3.2.2 Defining Master and Minion Servers	10
3.3 Uploading the Site Topology File	13
3.3.1 Validating the Site Topology File.....	14
3.3.2 Uploading the Site Topology File to the Installer Machines.....	15
3.4 Installing the SDC Components.....	17
4. Monitoring the Installation Process.....	19
4.1 Using Logs for Verifying and Troubleshooting.....	19
4.2 Verifying the SDC Application Status.....	20
4.2.1 Authenticating the Installer REST Interface.....	20
4.2.2 Application Status per Server	21
4.2.3 Site Status.....	23
4.2.4 API Response Codes.....	25
4.2.5 Monitoring Salt Packages.....	26
5. Post Installation Procedures	27
5.1 Reinstalling Data Center/Customer Environment RPMs	27
5.2 Changing the Root Password	27
5.3 Changing the SNMP Community String	27
5.4 Add Licenses to FEP IP Addresses.....	28
Appendix A: Site Topology File Structure	29
5.5 Virtual Machines (topology:vms).....	29
5.5.1 Interfaces (topology:vms:vm:interfaces).....	30
5.5.2 Application Instances (topology:vms:vm:applicationInstance).....	33
5.5.3 Volumes (topology:vms:vm:volumes)	34
5.6 Networks (topology:networks).....	36
5.7 Applications (topology:applications)	38
5.7.1 FEP 38	
5.7.2 CPF 38	
5.7.3 Tripo	39
5.7.4 CM 41	



5.7.5 Webui.....	41
5.7.6 Nms	41
5.7.7 oamDB.....	41
5.7.8 vlnstaller.....	42
5.7.9 VIP	42
5.7.10 ELK.....	43
5.8 Site Properties (topology:siteProperties)	44
Appendix B: Using Windows for API Requests	46
B.1 Uploading the Site Topology File from Windows.....	47
Appendix C: Port Settings Used by the SDC.....	49
C.2 EMS Site Internal Ports	49
C.3 EMS Site External Ports.....	50
C.4 SDC Site Internal Ports.....	52
C.5 SDC Site External Ports	54
C.6 HP Integrated Lights-Out (iLO) Port Settings.....	56
Appendix D: ELK Components.....	58
Glossary.....	59

List of Figures

Figure 1: Installation Flow.....	1
Figure 2: GRUB Boot Loader Page.....	10

List of Tables

Table 1: Conventions.....	4
Table 2: Data Volume Storage Requirements per Server for an SDC Site	3
Table 3: Data Volume Storage Requirements per Application for an EMS Site	4
Table 4: Mandatory Parameters	11
Table 5: Optional Parameters	11
Table 6: appStatus Command Error Codes.....	21
Table 7: appStatus Return Codes.....	21
Table 8: siteStatus Command Error Codes	24
Table 9: siteStatus Return Codes	24
Table 10: API Status Output Codes.....	26
Table 11: Elements Defined in Topology	29
Table 12: Elements Defined as Part of each VM.....	30
Table 13: Elements Defined in each Interface Element.....	31
Table 14: Elements Defined in Route	32
Table 15: Elements Defined for Each Application Instance.....	33
Table 16: Elements Defined in Volumes.....	34
Table 17: Elements Defined in Partitions.....	35
Table 18: Elements Defined for Each Network Element	36



Table 19: Elements Defined for the FEP Application	38
Table 20: Elements Defined for the CPF Application.....	39
Table 21: Elements Defined for the Tripo Application	40
Table 22: Elements Defined for the oamDB Application.....	41
Table 23: Elements Defined for the vip Application	42
Table 24: Elements Defined in siteProperties.....	44
Table 25: EMS Internal Ports.....	49
Table 26: EMS External Ports	50
Table 27: SDC Internal Ports	52
Table 28: SDC External Ports.....	54
Table 29: HP iLO Ports	56
Table 31: Common Terms	59
Table 32: Abbreviations	60



1. Installation Overview

The installation process consists of two main phases:

- Setting up the site machines

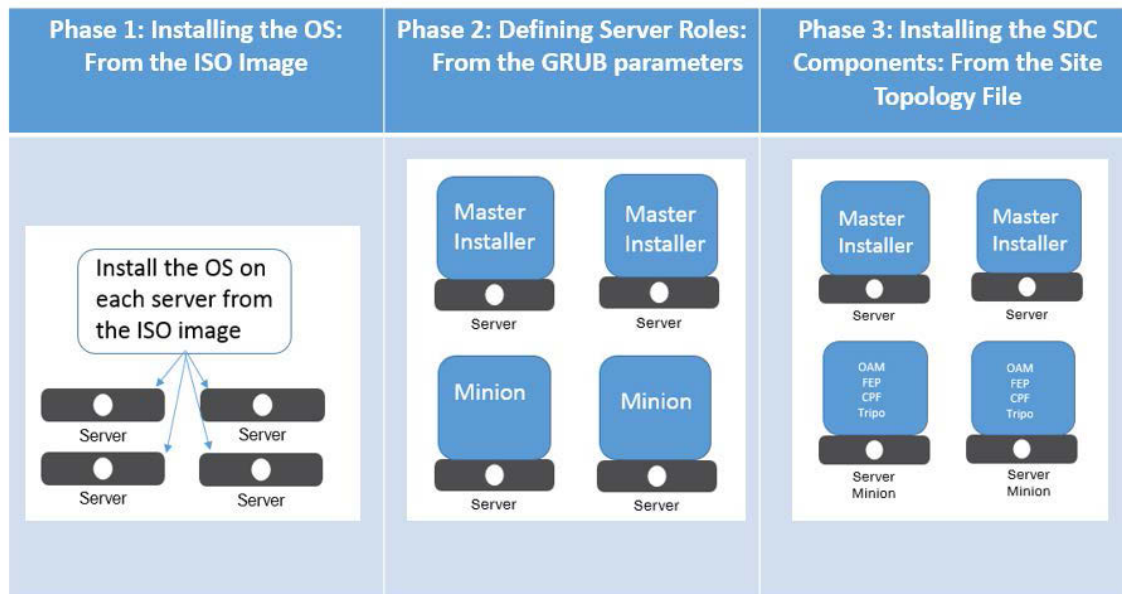
This phase includes installing the operating system on each site machine and then defining each machine's role as either a master Installer server or as a minion server, which has the role of hosting the SDC components.

The operating system is installed from the ISO image and the master-minion definition - identification process is based on GRUB boot parameters. Once the master Installer server is defined, it is configured to receive Salt API requests.

- Installing the SDC components

Using Salt API requests and based on the site Topology parameters, the master Installer server, communicates with the relevant minion servers to install the SDC components (CPF, FEP, Config Manager, NmsAgent, Web UI, Fluentd, Tripo).

Figure 1: Installation Flow





Note: The minimum number of servers is two. The installation process takes up to one hour per server.

1.1 General Prerequisites

This document assumes that you have a comprehensive understanding of:

- 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 14 and higher of the *Bare Metal Installation Guide*.

1.2 Prerequisites

1.2.1 Completing a Site Survey

To correctly assess your specific needs and ensure that the installed solution will meet them, a site survey, reviewing your anticipated traffic type and scope, is completed. Based on the site survey, a solution is built and the hardware requirements and site configuration recommendations are decided upon.

Based on the site survey, the number of needed CPFs and other components are calculated and that determines the site deployment size.



Note: This document assumes that this stage has been successfully completed.



1.2.2 Installing the Hardware

Install and verify the successful installation of the required hardware, per the recommendations in the site survey prior to performing the software installation described in this guide.



Note: This document assumes that this stage has been successfully completed.

1.2.3 Data Storage Volume

Persistent disks are used to host the OS, the data and the logs.



Note: A Warning message is generated in the logs when the volume is not configured correctly, including an invalid volume size or partition name, in the site topology file.

1.2.3.1 Data Volume Requirements for an SDC Site

For an SDC site, the total disk size must support 300 (GB) which is sub-divided into partitions to host the different server applications.

Table 2: Data Volume Storage Requirements per Server for an SDC Site

Server	Partition Name	Mount Type	Size (MB)
Per Server Note: As in a bare metal deployment, there is no minimum or maximum number of components that can be installed on a server, a server can host multiple application types.	data	/data/	174080
NMS Note: Where relevant is hosted on an OAM server.	logs	/var/log/rsyslog	40960



1.2.3.2 Data Volume Requirements for an EMS Site

For an EMS site, the total disk size must support 900 (GB) which is sub-divided into partitions to host the different server applications.

Table 3: Data Volume Storage Requirements per Application for an EMS Site

VM Server	Partition Name	Mount Type	Size (MB)
Per Server Note: As in a bare metal deployment, there is no minimum or maximum number of components that can be installed on a server, a server can host multiple application types.	data	/data/	747520
NMS Note: Where relevant is hosted on an OAM server.	logs	/var/log/rsyslog	40960

1.2.4 Accessing the ISO Image

The ISO image contains the operating system and it is packaged and provided as bootable media by F5. Verify that you have saved the ISO image in a location that you can later point to in order to load it as part of setting up the site machines.

1.2.5 Supported Operating System

SDC is certified to run on the following operating system:

- Red Hat Enterprise Linux (RHEL) 6.9 64 bit



Note: For RHEL 6.9 to be supported on a HP Smart Array P220i Controller, you must run firmware version 8.0.



1.2.6 Creating a Site Topology File

The site topology file is created jointly by the customer and F5 and is based on the customer site survey. This file is sent to and used by the master Installer servers for their configuration as master Installers. Once the site topology file has been uploaded to the master Installer servers and they are up and running, the master Installer servers then use the site topology file to install the necessary SDC components on the defined number of minion servers in the deployment.

The site topology file contains the server and network information, as well as the site properties that are needed to install the SDC components. The site topology file is an XML file and is saved in the following directory: `/srv/traffic/topology.xml`.

As part of the SDC components installation, the Site topology file is validated and uploaded to the master Installers. For more information, see *Uploading the Site Topology File*.

For more information on the structure of the site topology file, see *Appendix A: Site Topology File Structure*



2. Introduction

The installation procedure installs, configures, and enables the necessary hardware, network infrastructure, and site components needed to process F5® Traffix® Signaling Delivery Controller™ (SDC) traffic.

In this release, the installation procedure is performed using a Rest API Installer.

2.1 SDC Installed Components

An SDC site is comprised of the following components that interact with one another to provide full service and management capabilities:

- **Installer** - manages the installation and upgrade of SDC components. The Installer master machine must include the system database (Cassandra which holds the Site Topology parameters)
- **OAM** - provides the configuration, provisioning and management of FEP and CPF, and must include the configuration manager, the NMS Agent, the Web UI and the system database (Cassandra)
- **FEP** - provides the connectivity end point to the SDC for Diameter and other supported protocol peers and a Virtual IP address to the peers. The FEP load balances Diameter and other supported protocol messages to the Control Plane Functions (CPFs)
- **CPF** - provides the rules implementation of Diameter and other supported protocol traffic
- **Tripo** - maintains session information for session binding and stateful routing.



Note: In a bare metal deployment, there is no minimum or maximum number of components that can be installed on a machine.

For installations, on three or more machines, you must have three OAM components.



If you have multiple machines, you can run the Installer (Installer + Cassandra) on one machine and the OAM (CM, NMS, Web UI, and Cassandra) on a different machine.

From 5.1 CF 30, EMS deployments will use ELK components, instead of Splunk components, to manage all SDC reporting functionalities. In SDC sites, ELK components include the Fluentd Forwarder. In EMS sites, ELK components include the Fluentd EMS, Elastic Search, and Forwarder. For an overview of the ELK components, see *ELK Components*

2.1.1 Active/Active High Availability

All components are installed in active/active mode to provide high availability, where each instance of the component is installed on a separate machine.



3. Performing the Installation

The installation consists of installing the operating system from the ISO image on each site machine, defining the master and minion servers, and then installing the SDC components for each site machine.



Note: Always install the master Installer servers before installing the minion servers.

The installation process includes an OS installation. This installation removes any installed Data Center/ Customer environment specific RPMs. Copy the RPMs that you want to reinstall after the installation.

3.1 Installing Multiple SDC Sites Managed by EMS Site

When installing a deployment with multiple SDC sites that are managed by an EMS site, first install the EMS site and then the SDC sites. The installation process is the same for an EMS site and an SDC site.



Note: SDC sites can be added to an existing deployment of SDC sites managed by an EMS site. Once the SDC site is installed, it will automatically connect to the existing EMS site.

3.2 Setting up the Site Machines

You need to install the operating system on each site machine. The operating system is installed from the ISO image.



Note: It is recommended that you load the ISO image using the ILO Integrated Remote Console.

3.2.1 Installing the Operating System

In order to install the operating system, you need to load the ISO image.



Note: The ISO image must be loaded for each server in the site, always beginning with the master Installer servers. The installation process takes up to one hour per machine.

3.2.1.1 Firmware Validation

Before installing the Operating System, you must validate that Firmware version 8.0 is running to support the HP Smart Array P220i Controller.

To load the ISO image:



Note: For an EMS deployment, the following steps can be run in parallel on each EMS site.

1. Perform the required pre-installation configurations needed for the console that is being used. The following steps assume that the ILO Integrated Remote Console is being used:
 - a. Configure the ILO IP address.
 - b. Connect to the ILO Integrated Remote Console and select **Virtual Drives**.
 - c. Select **Image file CD/DVD-ROM** to set with ISO file.
 - d. Reboot the machine and press F11 when the machine starts-up.
 - e. Select **Option1 boot to CD-ROM**.
2. Mount the ISO image from where it is saved on your computer.
3. Start the installed site machine from the ISO image.

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

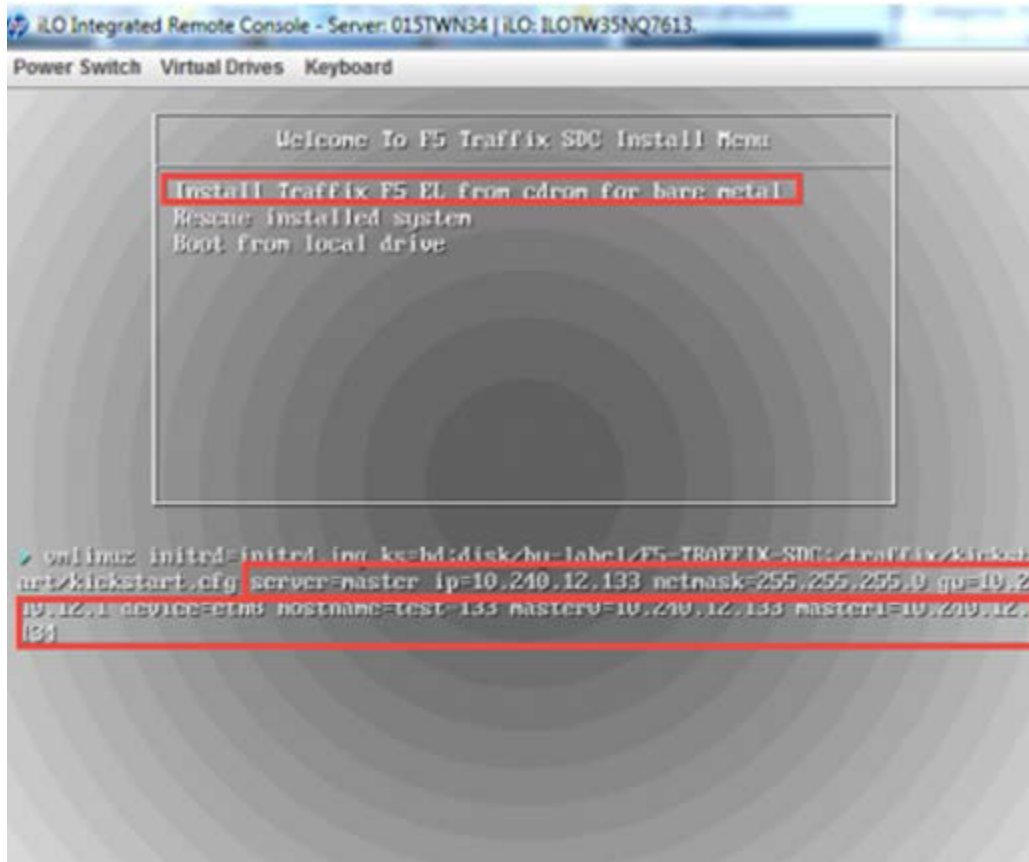
4. Under the **Welcome To F5 Traffix SDC Install Menu**, select **Install Traffix F5 EL from cdrom for bare metal**.

The GRUB boot loader page displays.



5. Press **e** (for edit) and then add the parameters

Figure 2: GRUB Boot Loader Page



6. Continue with configuring the GRUB boot parameters as in *Defining Master and Minion Servers*

3.2.2 Defining Master and Minion Servers

The GRUB boot parameters define a server's role as either a master Installer server or as a minion server that will host the SDC components. Configuring the parameters is done from the GRUB boot loader page. There are mandatory parameters and optional parameters that are only required if relevant for the deployment.



To configure the GRUB boot parameters:

1. In the prompt line, after **F5-TRAFFIX_SDC:traffix/kickstart/kickstart.cfg**, press the **TAB** key to enable editing and add the parameters, as follows:

Table 4: Mandatory Parameters

Name	Value Description
server	master/minion
hostname	the server's hostname Note: The hostname must be identical (case sensitive) to the value defined under the name attribute for the vm element in the Site topology file. The maxLength value is 64 characters.
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	The ethernet interface used by the IP address defined above

Table 5: 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 following is an example of inputted GRUB parameters:



```
server=master ip=10.240.12.133 netmask=255.255.255.0 gw=10.2
10.12.1 device=eth8 hostname=test-133 master0=10.240.12.133 master1=10.240.12.
134_
```

2. Click **ENTER** when done entering the parameters.
3. Repeat these steps for each site machine.

Each site server is now installed with an Operating System and has a defined role (master or minion). You can verify which servers in a site are defined as a master or minion with the `siteStatus` API Request. For more information, see *Site Status*.

3.2.2.1 Modifying the GRUB Boot Parameters

The GRUB boot parameters are saved in the `params` file. If you want to change any of the parameters or add a parameter, you can do so by editing the `params` file.



Note: You can only edit the parameters at this stage in the installation process, prior to uploading the Site topology file. After editing any GRUB boot parameters in `params` File, you must run an installation script.

You can reconfigure a master server as a minion server, but you cannot reconfigure a minion server as a master server.

To edit the GRUB boot parameters:

1. Go to the directory where the `params` file is saved:
cd /var/tmp/salt-install/
2. Execute the following command to edit the relevant parameters:

vi params



Note: If when entering the GRUB parameters from the GRUB boot screen, you clicked **ENTER**, but you still want to edit the parameters, you need to use the example *params* file to edit by executing the following command:

```
cp params.example params
```

3. Add/Edit a parameter according to the list of mandatory or optional parameters (see *Table 4* and *Table 5*).
4. Execute the following command to run the installation script on each of the Master Installers:

```
./salt-install.sh
```

Each installed site server is now updated with the edited parameters. You can verify which servers in a site are defined as a master or minion with the *siteStatus* API Request. For more information, see *Site Status*.



Note: If you changed the IP address of the master Installer or any of the GRUB parameters in the *params* file, then you need to run the installation script on all of the minion servers. If the master Installer IP address has not been changed, then you need to restart the Salt minion service.

3.3 Uploading the Site Topology File

The site topology file is uploaded to one of the master Installer servers using an API request. Before executing the API request to upload the site topology file, you need to validate the Site topology file and then identify it. You also need to have a valid authentication token to apply to the API request.



Note: Upload the site topology file only after the master Installer servers are up and running.



You can also upload the Site topology file from Windows. For more information, see *Appendix B: Using Windows for API Requests*

3.3.1 Validating the Site Topology File

You need to validate the site topology file before it is uploaded. This can be done on one of the Master Installer servers.

To validate the site topology file:

1. Run the following command:

```
cd /srv/traffix
```

```
python pillar/traffix_validate.py /tmp/ <topology_file_name>.xml
```

The following is an example of a successful validation:

```
validate_topology_string: succeeded to validate xml file topology=<?xml
version="1.0" ?>
[root@sdclab006-16 traffix]# python pillar/traffix_validate.py
/tmp/topology.xml
Using topology file /tmp/topology.xml
Validate topology {'siteValidations': True}
networking tp validated
siteProperties tp validated
General Site topology validation
check_pillar_network
check_pillar_applications
check_pillar_applications_vip
check_pillar_applications_cpf
check_pillar_siteProperties
siteProperties pillar validated
Topology validated 0
Topology valid!
```

The following is an example of a failed validation:



```
validate_topology_string: failed to validate xml file topology=<?xml  
version="1.0" ?>
```

3.3.2 Uploading the Site Topology File to the Installer Machines

You need to upload the site topology file to the master Installer by running an API request.

To upload the Site topology file:

1. Upload the site topology file to a master Installer server.

3.3.2.1 Authenticating the Installer REST Interface

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.

To generate an authentication token:

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

The following is the authentication API:

```
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.

3.3.2.2 Authentication Request Status Codes

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



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

3.3.2.3 UploadTopology API Request

Once the Site topology file has been validated and edited so it can be referenced and you have a valid authentication request, you can run the uploadTopology API request. The API request, references one of the master Installers < master_IP_address> to where the site topology file will be uploaded to.

To upload the Site topology file:

1. Run the following API command from where the topology file is currently located for example:

cd/tmp/

```
curl -ksi https://<master_IP_address>:8000 -H "Accept: application/x-yaml" -H "X-Auth-Token: <Token>" -d client="runner" -d fun="traffix.uploadTopology" -X POST --data-urlencode "topology=$(cat <full path to topology>topology.xml)"
```

The response indicates if the site topology file has been successfully uploaded. The following is an example of the API request with a successful response:

```
# curl -ksi https://localhost:8000 -H "Accept: application/x-yaml" -H "X-Auth-Token:fc2e8dcc1fed67dbd182b42609642b5ffcf27ed9" -d client="runner" -d fun="traffix.uploadTopology" -X POST --data-urlencode "topology=$(cat /tmp/topology.xml)"
@topology.xml
HTTP/1.1 100 Continue

HTTP/1.1 200 OK
```



```
Content-Length: 63
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, 10 Aug 2016 10:45:22 GMT
Access-Control-Allow-Origin: *
Content-Type: application/x-yaml
Set-Cookie: session_id=fc2e8dcc1fed67dbd182b42609642b5ffcf27ed9;
expires=Wed, 10 Aug 2016 20:45:22 GMT; Path=/

return:
- - 0
  - topology uploaded to the server successfully
```

The Site Topology parameters are now saved in the Cassandra database in both of the master Installers.



Note: Once the Site topology file has been uploaded successfully, the only way to modify the site configurations (in the *params* file or in the Site topology file) is to perform a new installation by reinstalling the ISO.

3.4 Installing the SDC Components

Once the master Installer servers and the other minion servers are up and running with an Operating System, they are ready to have the relevant SDC components installed on them. This is done based on the topology parameters that are configured in the Site Topology XML file.

The minion servers communicate with the master Installer servers, which then and the master Installer servers then reply to the minion servers, based on the Site Topology parameters, to know where to install the different SDC components (FEP, CPF, etc.).



Note: Immediately after installing an SDC site you may encounter multiple occurrences of the `sdcMonitProcessRestart` alarm in the Web UI. This does not impact performance.



4. Monitoring the Installation Process

You can monitor the status of the first part of the installation process (setting up the site servers/machines) by checking the different logs (see *Using Logs for Verifying and Troubleshooting*).

To monitor the second part of the installation process (installing the SDC components), you can refer to the relevant logs as described in *Using Logs for Verifying and Troubleshooting*. In addition, after the Site topology file is uploaded successfully with the Salt API request, you will see a response indicating the "topology uploaded to the server successfully."

With Rest APIs, you can verify which SDC components (applications) have been successfully installed on a specific server or on all the servers per site (see *Verifying the SDC Application Status*). These verifications can be done from a Linux (curl commands) or Windows operating systems. To use Windows, you will need to install the Google Chrome Advanced REST client plugin (see *Appendix A:*).

4.1 Using Logs for Verifying and Troubleshooting

You can refer to the following logs to verify the installation process as well as for troubleshooting the installation process:

Log	Used to...
<code>/var/log/salt-install.log</code>	verify the installation process
<code>/var/log/salt/master</code>	verify communication with minion servers, the python script, if site topology file successfully uploaded to Cassandra, (use <code>traffix API</code>)
<code>/var/log/salt/minion</code>	verify communication with the master Installer, installation, configurations, (use <code>state logs</code>)



Log	Used to...
<code>/var/log/rsyslog</code>	verify the file creation, per the OAM node, (central storage of logs)

4.2 Verifying the SDC Application Status

This verification is done by invoking the following REST APIs, `appStatus` and `siteStatus`. The master Installer checks the status of the SDC applications running on a specific server or on all the servers. These APIs are based on a standard Salt API interface and the body of the REST API message contains CLI Salt functions.

4.2.1 Authenticating the Installer REST Interface

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.

To generate an authentication token:

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

The following is the authentication API:

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

4.2.1.1 Authentication Request Status Codes

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

Return Code	Description
200	success



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

4.2.2 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.

4.2.2.1 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)
```

4.2.2.2 Command Execution Codes for appStatus API Request

Table 6: appStatus 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

4.2.2.3 Return Codes for appStatus API Request

Table 7: appStatus Return Codes

Exit Code	Description
14002	Pending Machine Start
14003	Pending SDC Installation
14004	Pending SDC Start
14006	Pending SDC Stop



Exit Code	Description
15002	Fail VM Start
15003	Fail To Install SDC
15004	Fail To Start SDC
15006	Failed To Stop SDC
13000	Suspended
12000	Successfully installed  Note: After salt highstate receives a result code of 12000, you may need to wait up to one minute until the Kibana configurations are complete and Reports are displayed as expected in the WebUI.

4.2.2.4 Status Query Answer Example

The following is an example of a status request and answer from the Installer.

```
curl -ksi https://localhost:8000 -H "Accept: application/x-yaml" -H "X-Auth-Token:fc2e8dcc1fed67dbd182b42609642b5ffcf27ed9" -d client="runner" -d fun="traffix.appStatus" -d tgt="*" -d apps=True
HTTP/1.1 200 OK
Content-Length: 386
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, 10 Aug 2016 08:07:03 GMT
Access-Control-Allow-Origin: *
Content-Type: application/x-yaml
Set-Cookie: session_id=fc2e8dcc1fed67dbd182b42609642b5ffcf27ed9; expires=Wed, 10 Aug 2016 18:07:03 GMT; Path=/
```



```
return:
- sdclab006-08:
  - Status-Code: 12000
  - Installed-Apps:
    - cm-unique
    - cpf3
    - fep-sctp-test
    - nmsagent-unique
    - oamDB-unique
    - tripol
    - vnf
    - webui-unique
sdclab006-16:
- Status-Code: 12000
- Installed-Apps:
  - cm-unique
  - cpf3
  - fep-sctp-test
  - nmsagent-unique
  - oamDB-unique
  - tripol
  - vnf
  - webui-unique
```

4.2.3 Site Status

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.

4.2.3.1 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)
```

4.2.3.2 Command Execution Codes for siteStatus API Request

Table 8: 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

4.2.3.3 Return Codes for siteStatus API Request

Table 9: siteStatus Return Codes

Exit Code	Description
14010	Installation is Running
15010	Installation Failed
12010	Installation Finished Successfully

4.2.3.4 siteStatus Answer Example

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

```
]# curl -ksi https://10.240.12.140:8000 -H "Accept: application/x-yaml" -H "X-Auth-Token:fc2e8dcc1fed67dbd182b42609642b5ffcf27ed9" -d client="runner" -d fun="traffix.siteStatus" -d apps=True
HTTP/1.1 200 OK
Content-Length: 388
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, 10 Aug 2016 08:32:12 GMT
Access-Control-Allow-Origin: *
Content-Type: application/x-yaml
Set-Cookie: session_id=fc2e8dcc1fed67dbd182b42609642b5ffcf27ed9;
expires=Wed, 10 Aug 2016 18:32:12 GMT; Path=/

return:
- - Site-Status-Code: 12010
  - Installed-Apps:
    sdclab006-08:
      - cm-unique
      - cpf3
      - fep-sctp-test
      - nmsagent-unique
      - oamDB-unique
      - tripol
      - vnf
      - webui-unique
    sdclab006-16:
      - cm-unique
      - cpf3
      - fep-sctp-test
      - nmsagent-unique
      - oamDB-unique
      - tripol
      - vnf
      - webui-unique
```

4.2.4 API Response Codes

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



Table 10: 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
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.

4.2.5 Monitoring Salt Packages

You can check which salt-srv packages are being used.

To view which Salt packages are being used:

1. Run the following command:

- yum search salt-srv



Note: This command returns the match by name and summary, as for example:
salt-srv5.1-554.noarch : Provides salt-srv.

To search for all the salt-srv packages not filtered by name or summary, use **- yum search all salt-srv**



5. Post Installation Procedures

The following procedures are performed after the installation process is successfully completed:

- *Reinstalling Data Center/Customer Environment RPMs*
- *Changing the Root Password*
- *Changing the SNMP Community String*
- *Add Licenses to FEP IP Addresses*

5.1 Reinstalling Data Center/Customer Environment RPMs

As part of the installation process, previously installed Data Center/Customer environment RPMs were removed. After performing the SDC installation, reinstall any relevant Data Center/Customer environment RPMs. The Data Center/Customer environment specific RPMs should match the OS version.

5.2 Changing the Root Password

During installation the Root password is assigned a default value. For increased security, change this value.

To change the root password:

1. Run the Unix "passwd" command.

5.3 Changing the SNMP Community String

To prevent access to the system's SNMP data, change the default value define for the community string. By default, the community string is defined as "public".



To change the SNMP community string before the site servers are started for the first time:

1. Configure the **SnmAgentSnmCommunity** parameter in the default NMS configuration file, `DEFAULT_LB_CONFIGURATION.xml`, with the desired value.

When the servers are started, all NMS Agents will be configured with the new value.

To change the SNMP community string after the site servers are started:



Note: This must be performed when the site servers are down.

1. Configure the **SnmAgentSnmCommunity** parameter in each individual NMS configuration .xml file on the site with the desired value.
2. Restart the site servers.

When the servers are started, all NMS Agents will be configured with the new value.

5.4 Add Licenses to FEP IP Addresses

Each FEP IP address must have a license. During the installation, IP addresses were added to the FEP instances. These IP addresses much each have their own license. For more information about obtaining the licenses, contact *F5 Support* and refer to the *F5 SDC User Guide* on how to add a new license key



Appendix A: Site Topology File Structure

The site topology configuration is defined under the topology element. This element contains four mandatory core elements, each defining a different aspect of the site topology. *Table 11* lists these elements.

Table 11: Elements Defined in Topology

Element (Mandatory/Optional)	Description
Vms (Mandatory)	This element defines and configures the specific virtual machines included in the site, the storage and SDC applications installed on them, and their communication paths. For more information, see Virtual Machines (topology:vms).
Networks (Mandatory)	This element defines and configures the networks used by the applications installed on the site's virtual machines for internal and external communication. For more information, see Networks (topology:networks)
Applications (Mandatory)	This element defines default values for the applications that are run on the virtual machines in the site. For more information, see Applications (topology:applications)
siteProperties (Mandatory)	This element defines and configures site-wide values. For more information, see Site Properties (topology:siteProperties)

5.5 Virtual Machines (topology:vms)

The vms element is one of the four core elements of the site topology file. This element contains virtual machine attributes and elements. The vms element defines the virtual machines that are part of the site, the installed storage on each virtual machine, the applications (SDC components) that will run on each virtual machine, and the communication paths that the applications are going to use.

The vms element contains one or more vm elements, corresponding to the number of virtual machines in the site. Each vm element defines a specific virtual machine in the site. *Table 12* lists the attributes and elements that are defined as part of each vm element.



Table 12: Elements Defined as Part of each VM

Element (Mandatory/Optional)	Description	Guidelines
Name (Mandatory)	The name of the virtual machine.	Enter a unique name.
defaultGateway (Mandatory)	The gateway used to connect the host network to the destination network.	Enter the default gateway.
VNFC (Optional)	The name of the component.	Enter the relevant name.
Interfaces (Mandatory)	The network interfaces used by the applications installed on the specific virtual machine.	The interfaces must belong to a network defined in the networks element.
applicationInstance (Mandatory)	The applications installed on the specific virtual machine.	The values defined per VM will override the default values defined in the applications element.
Volumes (Mandatory)	The file systems and persistent storage installed for the specific virtual machine.	Refer to the relevant Installation Guide, Data Volume Requirement section for more information.

5.5.1 Interfaces (topology:vms:vm:interfaces)

The interfaces element is a sub-element of a vm element and contains one or more interface element. Each interface element corresponds to a specific interface that will be used by the virtual machine.



Note: The interfaces must belong to the networks defined in the networks element. Verify that all necessary networks have been defined for the site in the networks element before defining specific interfaces for the virtual machines.



Each interface element contains attributes and sub-elements. *Table 13* lists the attributes that are defined as part of the interface element.

Table 13: Elements Defined in each Interface Element

Element (Mandatory/Optional)	Description	Guidelines
Network (Mandatory)	The name of the network associated with this interface.	The network name should match the name as it appears in the networks element.
ip4 (Optional)	The IPv4 address that is associated with this interface, corresponding to the relevant network.	Enter the relevant IPv4 address.
ip6 (Optional)	The IPv6 address that is associated with this interface, corresponding to the relevant network.	Enter the relevant IPv6 address.
Dev (Mandatory)	The name of the device on the guest OS that uses this interface.	If the interface is applied to a bond, define the element value with the bond name. For example, “bond0”.
bondDev (Optional)	The two interfaces that make up the bond defined in the “Dev” element.	Interfaces should be specified in comma separated format. E.g.: eth0, eth1
bondingOpts (Optional)	The bonding attributes on the OS level.	Enter the bonding interface configuration. For example, bondingOpts="mode=1 miimon=100 num_grat_arp=10 primary=eth10"
Name (Optional)	The name of the interface. This is used to differentiate between multiple IP addresses defined for the same interface, or when a VIP application is configured for the VM.	This value must match the “listeninterfacename” value defined for the application instance using the interface.



Element (Mandatory/Optional)	Description	Guidelines
Route (Optional)	The route is the specific route needed to be configured for a specific interface or network.	Enter the gateway for the IP



Note: The ip4 and ip6 attributes can be configured together on the same network or separately on different interfaces according to the matching network.

5.5.1.1 Route

Table 14: Elements Defined in Route

Element (Mandatory/Optional)	Description	Guidelines
Name (Mandatory)	Route name.	Enter a unique name
net4 (Optional)	The IPv4 destination network address to which to set the route.	Enter the relevant IPv4 address.
net6 (Optional)	The IPv6 destination network address to which to set the route.	Enter the relevant IPv6 address.
ip4sub (Optional)	The IPv4 destination network subnet mask.	The subnet mask can be stated in regular IPv4 format or in CIDR notation (/32, /24, etc.)
ip6sub (Optional)	The IPv6 destination network subnet mask.	The subnet mask can be stated only in CIDR notation (/64, /127, etc.)
Gateway (Mandatory)	The gateway address from which the route is configured.	This can be configured for IPv4 or IPv6.



5.5.2 Application Instances (topology:vms:vm:applicationInstance)

The applicationInstances element is a sub-element of a vm element, and contains one or more applicationInstance element. Each applicationInstance element corresponds to a specific instance of an application (SDC component) installed on the VM.

Each applicationInstance element contains attributes and sub-elements. *Table 15* lists the attributes that are defined as part of the applicationInstance element.

Each application is configured by default with the values defined – per application type – in the applications element. To override these values for this specific instance of the application, include the specific values with the correct value in the applicationInstance element.

Table 15: Elements Defined for Each Application Instance

Element (Mandatory/Optional)	Description	Guidelines
type (Mandatory)	The type of the application.	The valid values are: <ul style="list-style-type: none">▪ fep▪ vip▪ cpf▪ cpfss7▪ tripo▪ cm▪ webui▪ nms▪ oamDB▪ vInstaller▪ elk
name (Mandatory)	The name of the application instance. Each application instance should have a unique instance name to be referred to later on during the application configuration phase.	Each application instance should have a unique instance name.



IPv (Mandatory)	Specifies whether the application uses IPv4 or, IPv6	Enter either IPv4 or IPv6.
listenInterfaceName (Mandatory)	The name of the network interface (s) used by the application.	<ul style="list-style-type: none">▪ This value must match the name value defined for the interface that the application uses. <p>Note: For example, for the Tripo application instance, the listenInterfaceName field must always be defined as “ic” and the IPv field must be defined according to the IP version of the ic network.</p> <ul style="list-style-type: none">▪ All interfaces that the applications are listening to must be defined.▪ For VIP applications, all interfaces with the same routerID must be defined.

5.5.3 Volumes (topology:vms:vm:volumes)

The volumes element is a sub-element of a vm element, and defines the persistent storage that will be installed on the specific virtual machines.

The volumes element should include all persistent volumes that are expected to be attached to this specific VM, and should also include file system definition for each one of them.

Table 16: Elements Defined in Volumes

Element (Mandatory/Optional)	Description	Guidelines
cinderName (Mandatory)	The name of the storage volume.	Enter a relevant name.



Element (Mandatory/Optional)	Description	Guidelines
vmDev (Mandatory)	The device name that is expected to be seen on the guest OS level. For example: E.g. /dev/DRAOAM1	Enter the device path.
Partition (Mandatory)	A list of partitions to be defined on the guest OS level.	List all relevant partition names.

5.5.3.1 Partitions (topology:vms:vm:partition)

The partitions element is a sub-element of a volumes element, and defines a list of partitions on the guest OS level.

Note: For more information about the volume requirements, see the Data Volume Requirement section in the relevant Installation Guide.

Table 17: Elements Defined in Partitions

Element (Mandatory/Optional)	Description	Guidelines
Name (Mandatory)	Logical name for the handled partition.	Enter a relevant name, such as data, oamdb, repo, backup, logs
Size (Mandatory)	Size of the partition in GB.	Enter a relevant size per the data volume requirements.
mountPoint (Mandatory)	Mount point to which this partition will be mounted. For example: /data/Cassandra	Enter a relevant mount type per the data volume requirements.
fsType (Optional)	File system type. For example: ext4	Enter a relevant file system type.
fsParams (Optional)	Any other parameters needed for file system mount command	Enter any other relevant parameters.



5.6 Networks (topology:networks)

The networks element is one of the four core elements of the site topology file. This element contains network attributes and elements. The site topology file defines the SDC components that will run on each site server, and the communication paths – both between the SDC components and between the SDC site and external networks. The networks element defines the networks that the communication paths are going to run on.

- Management networks – this network connects between SDC components the internal SDC components for management purposes and is used to communicate between the SDC components and the Orchestrator.
- Interconnect networks – this network connects between all internal SDC components within a site.
- Signaling networks – this network connects the SDC’s FEP components with the external networks, including between geo-redundant sites.
 - Custom networks – this network is based on specific customer-requests and is defined on a case by case basis with the customer.

The networks element contains one or more network elements. Each network element defines a communication path for the site. *Table 18* lists the attributes that are defined as part of the network element.



Note: The networks element must include at least two network elements – one network element defining a management network and one network element defining an interconnect network.

Table 18: Elements Defined for Each Network Element

Element (Mandatory/Optional)	Description	Guidelines
Name (Mandatory)	The name of the network.	Enter a value that clearly indicates the nature of the



Element (Mandatory/Optional)	Description	Guidelines
		network. For example: sig-tcp-1 or sig-sctp-2.
net4 (Optional)	The IPv4 destination network address for this network.	Enter the relevant IPv4 address.
net6 (Optional)	The IPv6 destination network address for this network.	Enter the relevant IPv6 address.
ip4sub (Optional)	The network subnet – the range of IP addresses that belong to the IPv4 network.	The subnet mask can be stated in regular IPv4 format or in CIDR notation (/32, /24, etc.)
ip6sub (Optional)	The network subnet – the range of IP addresses that belong to the IPv6 network.	The subnet mask can be stated only in CIDR notation (/64, /127, etc.)
vlan (Optional)	The VLAN tag given for this specific network by the customer. The VLAN tag defines the VLAN-aware partitioning of the network.	Enter the relevant Vlan ID.
Role (Mandatory)	The type of communication that the network is going to be used for.	Define the value as follows: <ul style="list-style-type: none">▪ For a management network, define the value as “mgmt”▪ For an interconnect management network, define the value as “ic”▪ For a signaling management network, define the value as “sig”▪ For a custom network, define the value as needed.



5.7 Applications (topology:applications)

The applications element is one of the four core elements of the site topology file. This element contains sub-elements, each sub-element corresponding to one of the applications (SDC components) that can be installed in the site. These sub-elements define the default values for the general properties for each application – what version of the application is used, the communication paths that the application will use, etc.

5.7.1 FEP

The properties defined under the FEP element are listed in *Table 19*.



Note: In most cases, the FEP properties will be defined per specific instance of the FEP application, under the applicationInstances element. Any properties define per application instance will override the default values defined here.

Table 19: Elements Defined for the FEP Application

Element (Mandatory/Optional)	Description	Guidelines
fepType (Mandatory)	The protocol that the FEP application supports.	The supported FEP types are: <ul style="list-style-type: none">▪ diameter▪ http▪ radius▪ ldap The default value is “diameter”.

5.7.2 CPF

The properties defined under the CPF element are listed in *Table 20: Elements Defined for the CPF Application*.



Note: In most cases, the specific CPF application instances will use the default values defined in this element, and the only attribute that will need to be defined under the applicationInstances element is the name.



Table 20: Elements Defined for the CPF Application

Element (Mandatory/Optional)	Description	Guidelines
cpfPort (Mandatory)	The CPF port used by the Diameter FEP.	The default value is 13868.
cpfRadiusPort (Optional)	The CPF port used by the RADIUS FEP.	The default value is 11812.
cpfHttpPort (Optional)	The CPF port used by the HTTP FEP.	The default value is 18080.
cpfGtpPort (Optional)	The CPF port used by the GTP FEP.	The default value is 13386.
cpfLdapPort (Optional)	The CPF port used by the LDAP FEP.	The default value is 1389.

5.7.3 Tripo

The elements defined under the tripo element are listed in *Table 21*. These elements define the default values used by instances of the Tripo application in the site.



Note: Configuring the tripo element under the applications element is optional.

When defining the Tripo application instance, the `listenInterfaceName` must always be defined as “ic” and the `IPv` field must be defined according to the defined IP version (IPv4/IPv6) of the ic network.

For example: `<applicationInstance type="tripo" name="tripo1" IPv="IC_IPv" listenInterfaceName="ic">`



Table 21: Elements Defined for the Tripo Application

Element (Mandatory/Optional)	Description	Guidelines
secondSiteIP1 (Mandatory)	The IP address of the first Tripo instance on the second SDC server.	Enter the relevant IP address. Note: When this IP address is 1.1.1.1, replication between sites is disabled.
secondSiteIP2 (Mandatory)	The IP address of the second Tripo instance on the second SDC server.	Enter the relevant IP address. Note: When this IP address is 1.1.1.1, replication between sites is disabled.
maxSessions (Mandatory)	The maximum number of records allowed to be loaded per Tripo instance.	The value can be between 10,000 and 170,000,000.
tripoVersion (Mandatory)	The Tripo product version.	The available values are: <ul style="list-style-type: none">▪ latest▪ stated version The default is "latest".
srrInterface (Mandatory)	The name of the network that the Tripo uses to communicate with the Tripo instances on the geo-redundant site to replicate session data between sites.	The name must match the name of the network as it is defined in the networks element.
srrListenPort (Mandatory)	The port on the Tripo instance on the geo-redundant site that is used for replications.	It is assumed that both Tripo mates are using the same port.



Element (Mandatory/Optional)	Description	Guidelines
secondSiteSrrPort (Mandatory)	The port on the Tripo instance on the geo-redundant site that is used for replications.	It is assumed that both Tripo mates are using the same port.

5.7.4 CM

The cm element is currently defined as part of the applications element without any sub-elements or attributes.

5.7.5 Webui

The webui element is currently defined as part of the applications element without any sub-elements or attributes.

5.7.6 Nms

The nms element is currently defined as part of the applications element without any sub-elements or attributes.

5.7.7 oamDB

The elements defined under the oamDB element are listed in *Table 22*. These elements define the default values used by instances of the oamDB application in the site.



Note: In most cases, the oamDB properties will also be defined per specific instance of the FEP application, under the applicationInstances element. Any properties define per application instance will override the default values defined here.

Table 22: Elements Defined for the oamDB Application

Element (Mandatory/Optional)	Description	Guidelines
networkName (Mandatory)	The name of the network that the oamDB uses for internal communication and heartbeat.	The name must match the name of the network as it is



Element (Mandatory/Optional)	Description	Guidelines
		defined in the networks element. The default network is a network defined as a mgmt network.

5.7.8 vInstaller

The vInstaller element is currently defined as part of the applications element without any sub-elements or attributes.

5.7.9 VIP

The VIP application is used to enable active-standby mode for FEP applications, by associating a dedicated VIP application instance with each FEP IP address. The elements defined under the vip element are listed in *Table 23*. These elements define the default values used by instances of the vip application in the site.



Note: Verify that a dedicated VIP application is associated with each IP running on the FEP applications supported in active-standby mode. For example, to support two FEP applications using the TCP protocol, two VIP applications must be configured. To support two FEP application using the SCTP protocol, four VIP applications must be configured.



Note: Configuring the vip element under the applications element is optional.

Table 23: Elements Defined for the vip Application

Element (Mandatory/Optional)	Description	Guidelines
routerID (Optional)	A numeric value representing a group of one or more VIP	Verify that the same value is defined for all VIP applications associated with the FEP



Element (Mandatory/Optional)	Description	Guidelines
	applications that will operate in active-standby mode.	applications that will operate in active-standby mode.
transportProtocol (Optional)	Indicates whether the VIP is managing TCP or SCTP transport protocol.	Valid values are: <ul style="list-style-type: none">▪ tcp▪ sctp
applicationInstanceName (Optional)	The name of the application that the VIP is attached to. The VIP application is only valid if it is attached to an application. This attribute defines which application will be monitored for VIP failovers.	The value must be identical to the value defined for the name attribute of an application instance defined for this VM.

5.7.10 ELK



Note: Configuring the elk element under the applications element is only mandatory when the site is part of a deployment managed by an EMS site.

Adding the ELK application instance to the topology.xml, will trigger the installation and configuration of the ELK components (on the EMS site: Fluentd, elasticsearch_master, elasticsearch_data) and Kibana and on the SDC site, Fluentd).

These components are installed dynamically, and after installation, an elastic master is chosen via election between the nodes.

When defining the ELK application instance, the listenInterfaceName must always be defined as “mgmt” and the IPv field must be defined according to the defined IP version (IPv4) of the mgmt network:

```
<applicationInstance type="elk" name="elk" IPv="v4" listenInterfaceName="mgmt">
```



5.8 Site Properties (topology:siteProperties)

The siteProperties element is one of the four core elements of the site topology file. This element contains site attributes and elements that define general site properties.

Table 25 lists the attributes and elements that are defined as part of the siteProperties element.

Table 24: Elements Defined in siteProperties

Element (Mandatory/Optional)	Description	Guidelines
name (Mandatory)	The name of the EMS/SDC site.	Each site should have a unique name.
sdVersion (Mandatory)	The name of the SDC software.	The default value is “latest”.
timeZone (Mandatory)	The different time zone options by geographic areas that the system is configured to work in.	The relevant time zone geographic area.
ntpServers (Mandatory)	The server(s) used to synchronize time zones between servers.	Each ntpServer attribute within this element must contain the following two attributes: <ul style="list-style-type: none">▪ name – the name of the NTP server▪ ip – the server IP address
traffixFolder (Optional)	The folder that all site configuration files are saved to.	The default is /opt/traffix
isManager (Mandatory)	Indicates if site is an EMS site or SDC site.	The default value is False , indicating the site is an SDC site. If the site is an EMS site, set the value to True .



Element (Mandatory/Optional)	Description	Guidelines
isMultiSiteEnv (Mandatory)	Indicates if the site is an EMS site, an SDC site managed by an EMS site, or a standalone SDC site.	The default value is False , indicating the site is a standalone SDC site. If the site is an EMS site, or an SDC site managed by an EMS site, set the value to True .
emsServers (Optional)	The elk Master instance(s) on the EMS site server(s).	Each emsServer attribute within this element must contain the following two attributes: <ul style="list-style-type: none">▪ name – the name of the EMS server▪ ip – the IP address of the elk Master on the EMS server <hr/> <p>Note: If the isMultiSiteEnv element is defined with a value of False, make sure to leave the emsServers element empty.</p> <hr/>
elkAvailable (Optional)	Indicates if elk is installed.	The default value is False , indicating that elk will not be installed. If elk will be installed, set the value to True .
nameservers (Optional)	Sets the DNS IP address. For example: <nameserver index="1 " ip="192.168.16.5"/>	Can add 1-3 nameserver IP addresses.

Appendix B: Using Windows for API Requests

You have the option of using Windows to send the API requests, such as appStatus, siteStatus, and to upload the site topology file to the master Installers. To do so, you need to use a Google Chrome Advanced REST client plugin and to generate an authentication token.

To generate an authentication token:

1. Install the Google Chrome Advanced REST client plugin:

<https://chrome.google.com/webstore/detail/advanced-rest-client/hgmloofddffdnphfgcellkdfbfbjeloo>

2. Open in Google Chrome, the url with the master Installer IP address. For example: https://10.240.9.230:8000/

The following message is displayed: Your connection is not private

3. Click **Advanced**.
4. Click **Proceed to <ip address of the master Installer> (unsafe)**
5. Open the Advanced Rest Client plugin and fill in the following fields:
 - a. > https:// <ip address of the master Installer> /login
 - b. Select **POST**
 - c. Under **Headers**: Content-Type: application/x-www-form-urlencoded
 - d. Under **Payload**:
 - i. **username**: saltuser
 - ii. **password**: traffix
 - iii. **eauth**: parm

The **Response** includes the token number. The following is an example of a return response:

```
{
  "return": [
    {
      "perms": [".*", "@runner"],
      "start": 1455005380.5269041,
      "token": "e1be4b3089a802e4f51c6692dbe0fef80d25ff5c",
      "expire": 1455048580.5269051,
      "user": "saltuser",
      "eauth": "pam"
    }
  ]
}
```

B.1 Uploading the Site Topology File from Windows

You have the option of uploading the site topology file to the master Installers from Windows.

To upload the Site topology file:

1. Open the Advanced Rest Client plugin and fill in the following fields:
 - a. > https:// <ip address of the master Installer> /
 - b. Select **POST**
 - c. Under **Headers**:
 - i. Content-Type: application/x-www-form-urlencoded
 - ii. Accept: application/x-yami
 - iii. X-Auth-Token: <the generated token number>
 - d. Under **Payload**:

```
i. client=runner&fun=traffix.uploadTopology&&topology=<?xml
  version="1.0" encoding="UTF-8"?>
  <topology>
```

The Site topology file is uploaded on the screen.

The **Response** confirms if the Site topology file was successfully uploaded. The following is an example of a return response:

```
return: [
  - - 0
    - topology uploaded to the server successfully
```

Appendix C: Port Settings Used by the SDC

During an upgrade, a set of ports was enabled to ensure communication both between the different SDC components within the deployment, and between the SDC components and the necessary network elements.

This section describes the ports that have been validated for use by the SDC.

C.2 EMS Site Internal Ports

Table 25: EMS Internal Ports

Transport Protocol	Port	Network	Description
TCP	2812	IC	Monit
TCP	9200/9201 9300/9301	IC	ElasticSearch Discovery
TCP	5601	MGMT	Kibana Web Access
TCP	13868	IC	Traffic load balancing between the FEP and CPF instances
TCP	61616	IC	Communication between the configuration manager and the SDC components
TCP	61657	IC	Web UI Communication on cluster
UDP	161	IC	SNMP GET functions provided by OS snmpd service
UDP	162	IC	SNMP traps listener
UDP	1162		OS trap daemon listener

Transport Protocol	Port	Network	Description
TCP	7000	MGMT	Cassandra Database inter-site communication
TCP	7001	MGMT	Cassandra Database inter-site communication
TCP	7199	MGMT	Cassandra JMX monitoring inter-site communication
TCP	9042	MGMT	Cassandra client

C.3 EMS Site External Ports

Table 26: EMS External Ports

Transport Protocol	Port	In/Out	Network	Description
TCP	22/443	In	MGMT	SSH remote consoles
TCP	80	In	MGMT	HP Blade System web consoles
UDP	123	Out	OAM	NTP Process
UDP	514	Out	OAM	Syslog Process
UDP	1161	Out	MGMT	For External EMS Statistics Analysis
UDP	User-defined Ports (and IPs)	Out	MGMT	Trap Forwarding: For External EMS Trap listeners

Transport Protocol	Port	In/Out	Network	Description
TCP	9300 & 9320	In/Out	MGMT	Elastic search sync between two EMS nodes
TCP	5601	In/Out	MGMT	Kibana Web Access
TCP & UDP	10046	In	MGMT	Fluentd Fwd EMS (Receive TDR & Traces from site to EMS; UDP for Hearbeats)
TCP	3868	In/Out	H-TCP	Inter-site communication link for geo-redundancy
SCTP	3868	In/Out	H-SCTP-A	Primary SCTP path for domestic traffic
SCTP	3868	In/Out	H-SCTP-B	Secondary SCTP path for domestic traffic
TCP	4505/6	In/Out	MGMT	Salt Master
TCP	8000	In/Out	MGMT	Salt API
TCP	8080/8443	In	MGMT	SDC web console (Web UI)
TCP	10040	Out	MGMT	NMS Agent to NMS Manager

Transport Protocol	Port	In/Out	Network	Description
				for system status synchronization
TCP	61617	In	MGMT	Communication between the EMS and the SDC servers for new configuration propagation
TCP	7000	In/Out	MGMT	Cassandra Database inter-site communication
TCP	7001	In/Out	MGMT	Cassandra Database inter-site communication
TCP	7199	In/Out	MGMT	Cassandra JMX monitoring inter-site communication
TCP	9042	In/Out	MGMT	Cassandra client

C.4 SDC Site Internal Ports

Table 27: SDC Internal Ports

Transport Protocol	Port	Network	Description
TCP	2812	IC	Monit

Transport Protocol	Port	Network	Description
TCP	61616	IC	Communication between the configuration manager and the SDC components
TCP	13868	IC	Traffic load balancing between the FEP and the CPF instances
TCP	11812	IC	RADIUS listening port between the FEP and the CPF
TCP	18080	IC	HTTP listening port between the FEP and the CPF
TCP	13386	IC	GTP listening port between the FEP and the CPF
TCP	1389	IC	LDAP listening port between the FEP and the CPF
TCP	4444	IC	NMS to CPF communication port
TCP	23210	IC	Tripo - CPF connection to Tripo
TCP	43211	IC	Tripo – inter-site connection
TCP	23212	IC	Tripo - connection between Tripo mates within the same site
TCP	61627	IC	Default configuration REST communication
TCP	61637	IC	Default configuration REST communication
TCP	61647	IC	Default configuration REST communication NMS Agent

Transport Protocol	Port	Network	Description
TCP	61657	IC	Default configuration REST communication - UI
TCP & UDP	10046	MGMT	Fluentd Fwd Site (FWD TDR & Traces from site to EMS UDP for Hearbeats)
UDP	4545	IC	Port prefix is 4545 and the postfix is the UID of the CPF or FEP (4545 + UID)
TCP	5555	MGMT	Tripo Web statistics
TCP	7000	MGMT	Cassandra Database inter-site communication
TCP	7001	MGMT	Cassandra Database inter-site communication
TCP	7199	MGMT	Cassandra JMX monitoring inter-site communication
TCP	9042	MGMT	Cassandra client

C.5 SDC Site External Ports

Table 28: SDC External Ports

Transport Protocol	Port	In/Out	Network	Description
TCP	4505/6	In/Out	MGMT	Salt Master
TCP	8000	In/Out	MGMT	Salt API

Transport Protocol	Port	In/Out	Network	Description
TCP	8080 8443	In	MGMT	SDC web console (Web UI)
TCP	80	In	MGMT	HP Blade System web consoles
UDP	162	Out	MGMT	SNMP traps toward the EMS or third party NMS servers
TCP	3868	In/Out	H-TCP	Inter-site communication link for geo-redundancy
SCTP	3868	In/Out	H-SCTP-A	Primary SCTP path for domestic traffic
SCTP	3868	In/Out	H-SCTP-B	Secondary SCTP path for domestic traffic
TCP	61617	In	MGMT	Communication between the EMS and the SDC servers for new configuration propagation (internal and external data)
\TCP	22/80/443/623/17990/17988	In	MGMT	HP iLO4 management

Transport Protocol	Port	In/Out	Network	Description
				consoles and virtual media
TCP	10030	Out	OAM	NMS Agent
UDP	123	Out	OAM	NTP Process
UDP	514	Out	OAM	Syslog Process
TCP	7000	In/Out	MGMT	Cassandra Database inter-site communication
TCP	7001	In/Out	MGMT	Cassandra Database inter-site communication
TCP	7199	In/Out	MGMT	Cassandra JMX monitoring inter-site communication
TCP	9042	In/Out	MGMT	Cassandra client

C.6 HP Integrated Lights-Out (iLO) Port Settings

The following information is not specific to SDC, but relates to relevant ports configured on different servers.

Table 29: HP iLO Ports

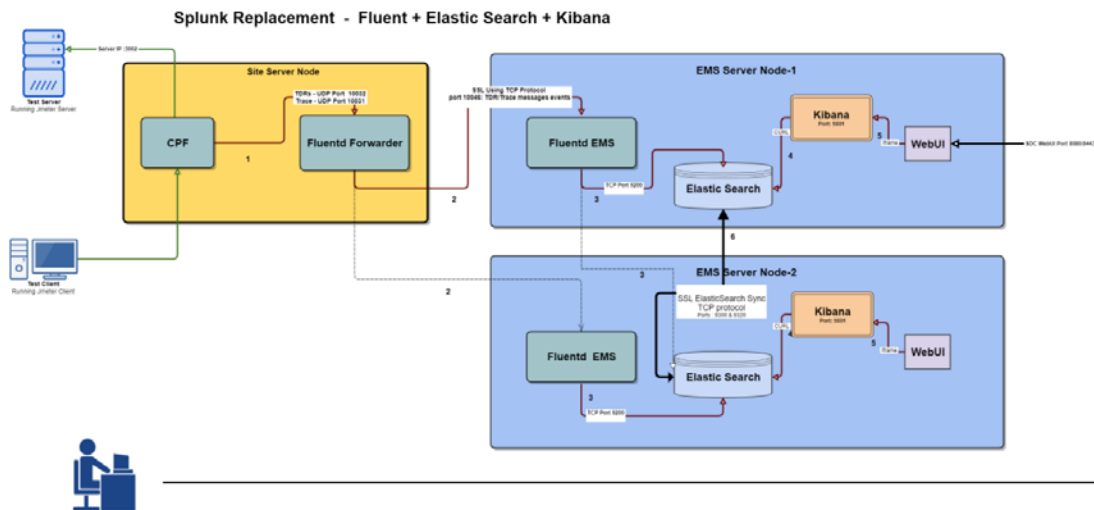
Transport Protocol	Port	iLO Function
CP	22	Secure Shell (SSH)
TCP	80	Web Server Non-SSL

Transport Protocol	Port	iLO Function
TCP	443	Web Server SSL
TCP	3389	Terminal Services
TCP	17988	Virtual Media
TCP	9300	Shared Remote Console
TCP	17990	Console Replay

Appendix D: ELK Components

As of CF 30, for EMS deployments, Splunk is replaced with ELK. There are three ELK components on the EMS (Fluentd, Elasticsearch and Kibana) and one component on the SDC (Fluentd Forwarder). These components receive and forward information to create an overview of the deployment's performance and support shared configuration across multiple sites.

The following diagram shows the full flow of how information is forwarded and collected between an SDC site and EMS sites.



All of the ELK components are managed by monit and their status (up/down) is easily viewed, as all other components, on the WebUI SDC components page.

Glossary

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

Table 30: 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 (Representational State Transfer)
Request	A message sent from one Client/Server peer to the other, followed by an answer message
RPM	RPM Package Manager

Term	Definition
Salt-API	Manages and communicates between a network of 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 31: 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

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

Term	Definition
SDC	Signaling Delivery Controller
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