# Server Monitoring App For Splunk

Contents

[Server Monitoring App For Splunk 1](#_Toc488152895)

[Background: 2](#_Toc488152896)

[Upload App “Server Inter-dependency for Splunk” 2](#_Toc488152897)

[Pre-Requisites: 2](#_Toc488152898)

[App Details 3](#_Toc488152899)

[Server\_Interdependency\_Static\_SampleData 4](#_Toc488152900)

[Server Interdependency Dashboard 5](#_Toc488152901)

[Requirement: 5](#_Toc488152902)

[Solution: 5](#_Toc488152903)

[Prerequisite: 5](#_Toc488152904)

[Implementation: 5](#_Toc488152905)

[Step1: Nagios Setting 5](#_Toc488152906)

[a) Add servers to be monitored (also define parent child dependency) 5](#_Toc488152907)

[b) PNP4Nagios configuration 7](#_Toc488152908)

[Step2: Setup Splunk Universal forwarder: 9](#_Toc488152909)

[a) Download UF: 9](#_Toc488152910)

[b) Install UF: 9](#_Toc488152911)

[c) Set environment variables: 9](#_Toc488152912)

[d) Configure forwarder: 10](#_Toc488152913)

[e) Edit Inputs.conf: 10](#_Toc488152914)

[f) Create outputs.conf: 10](#_Toc488152915)

[g) Restart Universal forwarder: 10](#_Toc488152916)

[Step3: Splunk Server Settings 11](#_Toc488152917)

[a) Create index: 11](#_Toc488152918)

[b) Enable ports: 11](#_Toc488152919)

[c) Extract fields used dashboard: 11](#_Toc488152920)

[d) App folder Permissions 12](#_Toc488152921)

[Verify Server Monitoring HTML source code: 13](#_Toc488152922)

[D3\_visualization.js(Javascript to display the Tree diagram) 13](#_Toc488152923)

[Output: 13](#_Toc488152924)

[Appendix: 16](#_Toc488152925)

[References: 16](#_Toc488152926)

# Background:

This document can be used as a reference to upload the “Server Inter-dependency For Splunk” App and setup dashboards

# Upload App “Server Inter-dependency for Splunk”

## Pre-Requisites:

* ServerMonitoring.spl package file available with the End Users.
* Splunk server installed (Linux version - 6.6.1)
* End Users have a basic knowledge of Splunk

Follow the steps below to upload the app on Splunk Server

1. Download the ServerMonitoring.spl package on your local machine
2. Upload the App as per below screen shots

Apps -> Manage Apps -> Install app from file -> Upload file

|  |
| --- |
|  |

1. Now the app will be available to use on the Splunk server

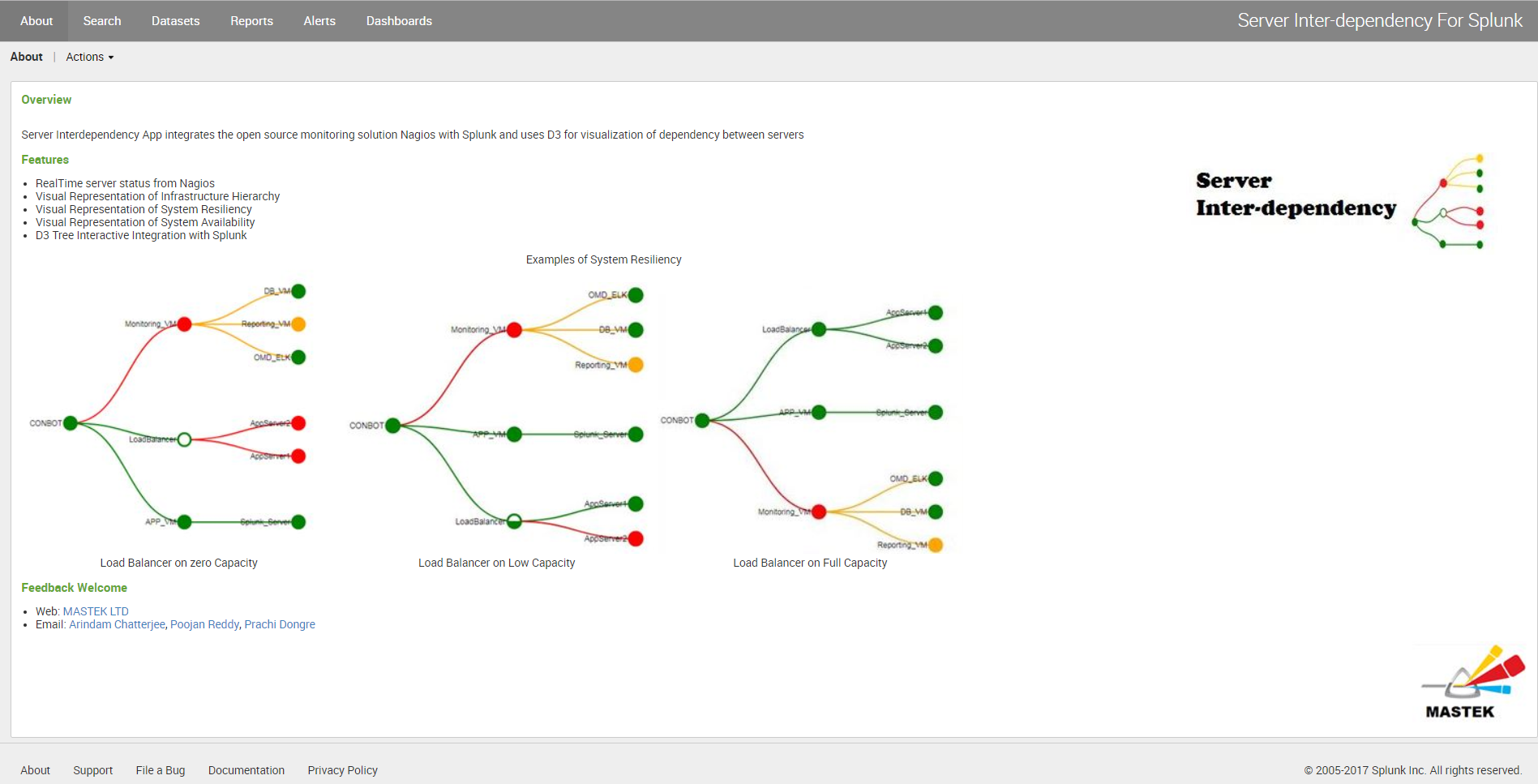
Go to Apps 🡪 Manage Apps

|  |
| --- |
|  |

1. From the dropdown arrow, select the app “Server Inter-dependency for Splunk” for further usage.

## App Details

Once the app is uploaded, under the app, the following pages have been provided



About 🡪 Gives a brief description about the app

Search 🡪 Search and Reporting tool, to create new or extract fields and search queries

Dashboards🡪 There are 2 available dashboards

* Server\_Interdependency
* Server\_Interdependency\_Sample

# Server\_Interdependency\_Sample

The sample data dashboard will be displayed immediately once app is uploaded along with the statistics table to give an indication of how the dashboard should look like to the End Users

To create the Sample –Static data dashboard the following have been used

1. Metadata.csv - for the statistics table to be displayed. A lookup has been created on this file
2. Static data in the JSON format for the D3 representation to take place.

The lookup file, Html and javascript used for the dashboard to be displayed is as follows:



Screenshot of the static Dashboard

|  |
| --- |
|  |

# Server Interdependency Dashboard

# Requirement:

Build a Splunk App to display the status of individual servers and their dependency. These servers can be of various types such as applications, load balancer etc.

# Solution:

* We have used Nagios performance logs as source
* Usage of D3 tree interactive for visualization

# Prerequisite:

* Splunk Enterprise Linux version (6.6.1) installed
* Nagios/OMD Logs for monitoring

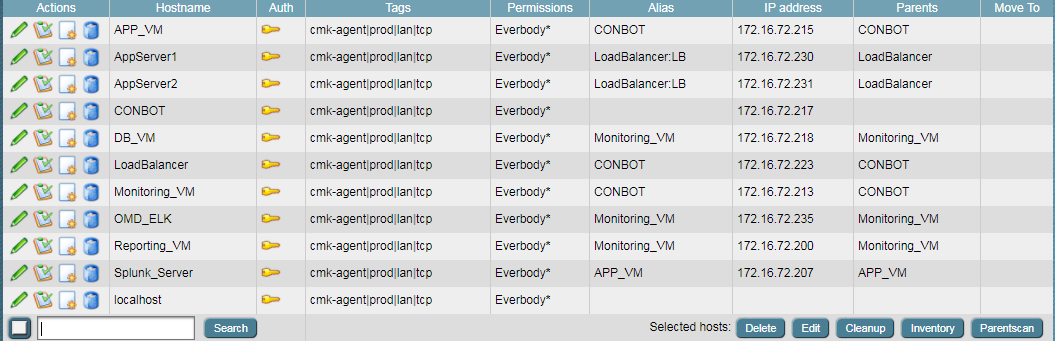
# Implementation:

### Step1: Nagios Setting

### Add servers to be monitored (also define parent child dependency)

Below is the sample list of VMs being monitored by Nagios.

<http://IP-address/monitoring/check_mk/>

  
To identify the Parent-child relationship for the hosts /servers, do the following settings via Check MK.

To setup a Parent of the Host servers, we need to set the Alias and the Parents field.

**Note:** Alias and the Parents field value should be the same.

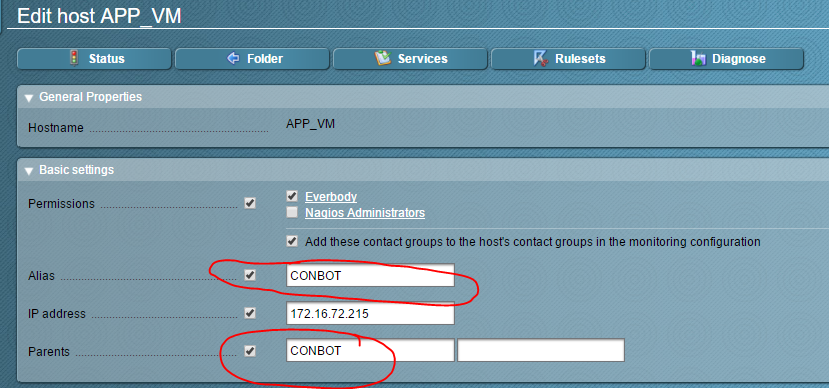
If the Host Parent is of a specific type (example: Load Balancer), set the Alias name as “<Parentname>:<Type>” example: (LoadBalancer:LB)

**Note**: The Alias field has been used instead to send the parent information through to the PNP4 Nagios logs.

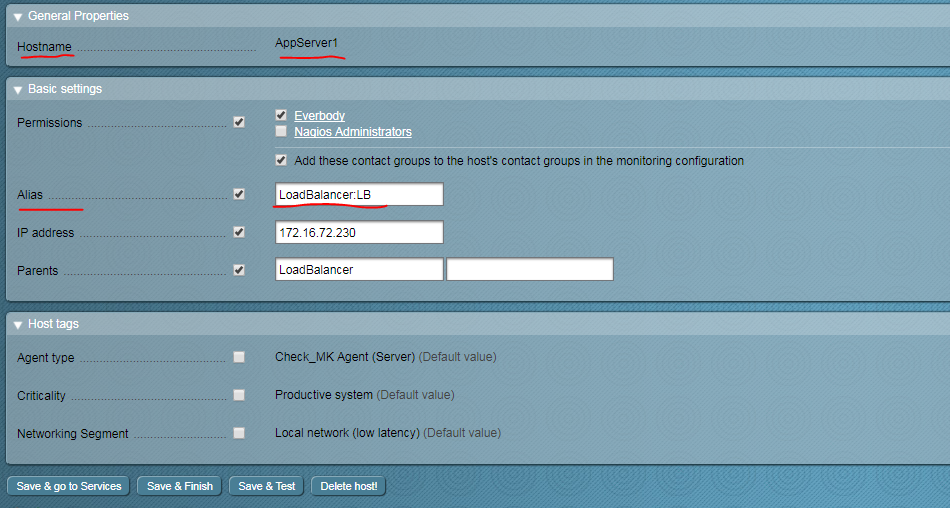
Below is an example to set up “App\_VM “host whose parent is “Conbot”

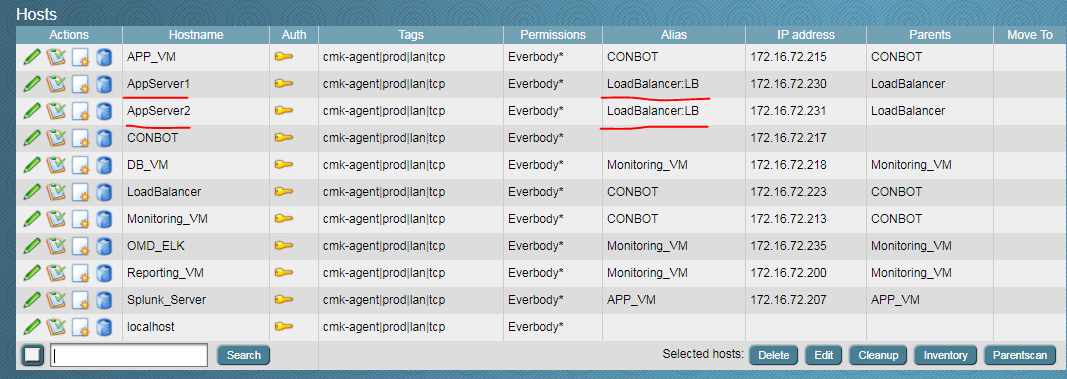
Go to Check\_MK -> Hosts -> select the child Host (APP\_VM)

Edit -> Set Alias and Parents as CONBOT (172.16.72.217) -> validate save and activate changes.



Below is a screen shot of Host “AppServer1” which is managed by a Load Balancer. To identify this, the load balancer type “LB” is mentioned in the Alias field in addition to the parent name





**Note**:

* In our host set up, the root node is “Conbot”, hence for the child nodes (App\_VM , LoadBalancer , Monitoring\_VM) , the parent node will be set as CONBOT in the Alias and Parents fields.
* Load Balancer manages hosts AppServer1 and AppServer2**.** Hence the type **“LB ”** is mentioned while adding the Alias
* The Alias information is written in the Nagios logs as HostParent information (Refer screen shot for PNP4Nagios)

Now, this information from Nagios will be captured as part of Performance logs and then the host performance logs will be sent to Splunk.

### PNP4Nagios configuration

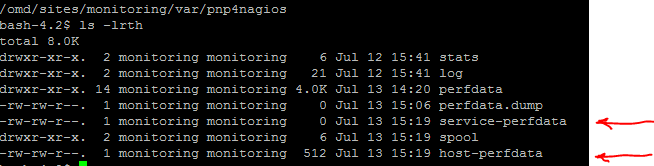
Nagios generates three logs: Nagios.log, host performance data and service performance data. We will be using host/service performance data for server monitoring.

To do this, make changes to PNP4Nagios configuration file. PNP4Nagios sends performance data to different forwarder including splunk.

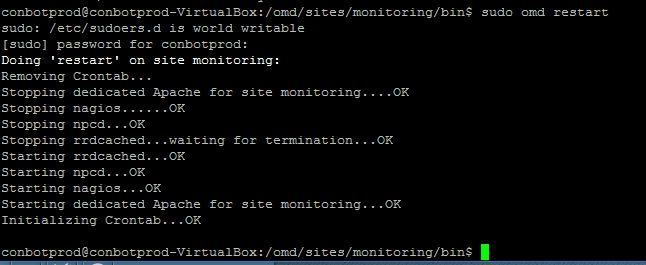
To set up the performance logs in Nagios use the following steps

|  |
| --- |
| (1) Change PNP4NAGIOS to use "NPCD with Bulk Mode" instead of NPCDMOD. This is done by redirecting the symlink for pnp4nagios.cfg:  ln -sf ~/etc/pnp4nagios/nagios\_npcd.cfg ~/etc/nagios/nagios.d/pnp4nagios.cfg  2) Go to ->  cd /omd/sites/monitoring/etc/pnp4nagios/  sudo vi Nagios\_npcd.cfg  (Added the highlighted section) Refer screen shot below    3) Update etc/nagios/conf.d/pnp4nagios.cfg (remember to replace SITENAME). |

Once the above 3 steps are done, host-perfdata and service-perfdata files will be created



1. Now Restart omd:



## Step2: Setup Splunk Universal forwarder:

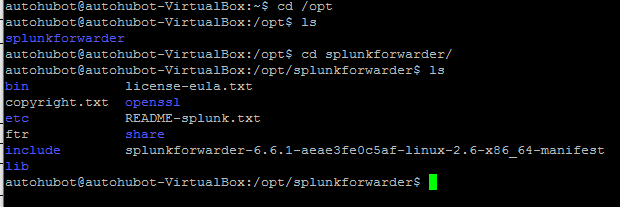
In the Linux VM from where you want to forward the logs (Nagios server in our case),lets install universal forwarder.

### Download UF:

|  |
| --- |
| wget -O splunkforwarder-6.6.1-aeae3fe0c5af-Linux-x86\_64.tgz 'https://www.splunk.com/bin/splunk/DownloadActivityServlet?architecture=x86\_64&platform=linux&version=6.6.1&product=universalforwarder&filename=splunkforwarder-6.6.1-aeae3fe0c5af-Linux-x86\_64.tgz&wget=true' |

### Install UF:

Install: tar xvzf splunkforwarder-<…>-Linux-x86\_64.tgz -C /opt



### 

### Set environment variables:

vi ~/.bashrc

export SPLUNK\_HOME="/opt/splunkforwarder"

export PATH=$PATH:$SPLUNK\_HOME/bin

export NAGIOS\_HOME=/omd/sites/monitoring

### Configure forwarder:

sudo /opt/splunkforwarder/bin/splunk start

sudo /opt/splunkforwarder/bin/splunk login –auth admin:changeme

sudo /opt/splunkforwarder/bin/splunk login –auth admin:splunk

### Edit Inputs.conf:

vi /opt/splunkforwarder/etc/system/local/inputs.conf

**Note**: index name “nagios” should remain as is

|  |
| --- |
| [monitor:///omd/sites/monitoring/var/nagios/nagios.log]  sourcetype=nagios  index=nagios  [monitor:///omd/sites/monitoring/var/pnp4nagios/host-perfdata]  sourcetype = nagioshostperf  index=nagios  [monitor:///omd/sites/monitoring/var/pnp4nagios/service-perfdata]  sourcetype = nagiosserviceperf  index=nagios |

### Create outputs.conf:

vi /opt/splunkforwarder/etc/system/local/outputs.conf

|  |
| --- |
| [tcpout]  defaultGroup=nagios  compressed=false  [tcpout:nagios]  server=172.16.72.207:9997  [tcpout-server://172.16.72.207:9997]  useACK=true |

**Note**: Here, 172.16.72.207 is our IP address of Splunk server. Replace this IP with the IP where your Splunk server is installed. 9997 is port where Splunk will listen to Nagios logs. Port ranging from 9000-9999 can be used for splunk forwarder.

### Restart Universal forwarder:

sudo /opt/splunkforwarder/bin/splunk enable boot-start

sudo /opt/splunkforwarder/bin/splunk restart

## Step3: Splunk Server Settings

1. Create index:

Create a new index with the same name as defined in universal forwarder:

Settings -> Data -> Indexes

|  |
| --- |
|  |

Click on New Index

|  |
| --- |
|  |

Put the name of the index(Nagios) and click on Save

The index will be listed as shown below

|  |
| --- |
|  |

### Enable ports:

Enable the port for receiving the data from Nagios:

Settings -> Data -> Forwarding and Receiving

|  |
| --- |
|  |

|  |
| --- |
|  |

### Extract fields used in dashboard:

There are five new fields that have been extracted from the output of the events

**Step1: Search new query in app (ServerMonitoring):**

index = nagios sourcetype="nagioshostperf"

|  |
| --- |
|  |

**Step2: Extract fields separately (as regular expression):**

Please refer Splunk documentation to extract the below new fields

[**http://docs.splunk.com/Documentation/SplunkCloud/6.6.0/Knowledge/ExtractfieldsinteractivelywithIFX**](http://docs.splunk.com/Documentation/SplunkCloud/6.6.0/Knowledge/ExtractfieldsinteractivelywithIFX)

The fields **must** be created in the below format **(Initcap)**

* HostName
* Hoststate
* Hostparent
* Timet
* Parenttype

Below is a screen shot showing setting up of field “HostName”

**Note:** the screen shot shows, how the value of the field needs to be selected to set an Extracted field with a Regular expression

|  |
| --- |
|  |

Tabular Format of the Inline Regular expressions for the Extracted fields

|  |  |
| --- | --- |
| Extracted Field | Regular Expression |
| * HostName | ^(?:[^:\n]\*:){6}(?P<HostName>[^\t]+) |
| * Hoststate | ^(?:[^\t\n]\*\t){5}\w+::(?P<Hoststate>\w+) |
| * Hostparent | ^(?:[^\t\n]\*\t){7}\w+::(?P<Hostparent>\w+) |
| * Timet | ^[^\t\n]\*\t\w+::(?P<Timet>\d+) |
| * Parenttype | ^(?:[^\t\n]\*\t){7}\w+::\w+:(?P<Parenttype>.+) |

Screenshot for Regular expressions derived for all the fields

|  |
| --- |
|  |

Now select above created fields in the visualization**.**

|  |
| --- |
|  |

### App folder Permissions

Go to Splunk server:

cd /opt/splunk/etc/apps/

Give permission to the entire app “Server Monitoring ” folder (chmod 777 <App folder Name>)

|  |
| --- |
|  |

## Verify Server Monitoring HTML source code:

Once the fields are created , the search query used will yield results, and display the statistics table and the D3 Tree Diagram should also be displayed as the D3 uses the same search querys

The final HTML used is as follows



## D3\_visualization.js(Javascript to display the Tree diagram)

Once the app is uploaded, check on the splunk server if the following files exists

* Go to folder in Splunk server where ServerMonitoring folder is available.
* Go to folder appserver/static
* Check if below three files are present in path.

/opt/splunk/etc/apps/ServerMonitoring/appserver/static

Files:

1. d3\_visualization.js
2. lowcapacity.jpg
3. nilcapacity.jpg

|  |
| --- |
|  |



**Restart Splunk server:**

cd /opt/splunk/bin

sudo ./splunk/start

## Output:

Screen shot when the App servers which are monitored by Load balancer are down . Due to this the Load Balancer capacity is determined as “ZERO”

|  |
| --- |
|  |

Screen shot when the App servers which are monitored by Load balancer are UP. Due to this the Load Balancer capacity is determined as “FULL”

|  |
| --- |
|  |

Screen shot when the App servers which are monitored by Load balancer are UP. Due to this the Load Balancer capacity is determined as “LOW”

|  |
| --- |
|  |

For the D3 tree representation, below logic has been added to depict the colour coding of nodes and links:

Nodes:

|  |  |
| --- | --- |
| **Node Colour** | **Remarks** |
| Green | when Host status is “UP” |
|  | when Host status is “Down” |
|  | when Host status is “Unreachable” |
|  | when Host status is other than above |

Links:

|  |  |  |  |
| --- | --- | --- | --- |
| **Parent Node** | **Child Node** | **Link Colour from Parent to Child Node** | **Remarks** |
| Green | Green | Green | Parent and Child Both UP, Link Available |
| Green |  |  | Parent UP, Child Down, Link Unavailable |
|  | Green |  | Parent Down, Child UP , Link Unreachable |
|  |  |  | Parent Down,Child Down , Link Unreachable |

|  |
| --- |
|  |

# Appendix:

Locations of the scripts

|  |  |
| --- | --- |
| Script Name | Location |
| metadata.csv | /opt/splunk/etc/apps/ServerMonitoring/lookups |
| Server\_Interdependency\_Static\_Data.html | /opt/splunk/etc/apps/ServerMonitoring/local/data/ui/html |
| Server\_Interdependency.html | /opt/splunk/etc/apps/ServerMonitoring/local/data/ui/html |
| d3\_static\_data\_visualization.js | /opt/splunk/etc/apps/ServerMonitoring/appserver/static |
| d3\_visualization.js | /opt/splunk/etc/apps/ServerMonitoring/appserver/static |

# References:

<http://dev.splunk.com/view/webframework-tutorials/SP-CAAAEN6>

http://docs.splunk.com/Documentation

<http://docs.splunk.com/Documentation/SplunkCloud/6.6.0/Knowledge/ExtractfieldsinteractivelywithIFX>