Solace

Professional Service Group:

*Monitoring Solace Broker Statistics* The Enterprise Stats Solution

Runbook

Document Category: Runbook

Version: 1.15

Status: Release

Last Updated: Apr 17, 2020

Contents

[Contents 2](#_Toc38013542)

[1 Introduction 7](#_Toc38013543)

[1.1 Scope 7](#_Toc38013544)

[1.2 Target Audience 7](#_Toc38013545)

[1.3 Related documents 7](#_Toc38013546)

[1.4 Terminology 8](#_Toc38013547)

[2 Solution Overview 9](#_Toc38013548)

[2.1 The Solution 10](#_Toc38013549)

[2.1 Components 10](#_Toc38013550)

[2.2 The Pump 11](#_Toc38013551)

[2.3 The Receiver Framework 11](#_Toc38013552)

[2.4 Concrete Receivers 11](#_Toc38013553)

[2.5 Primary Use Case – Sequence 11](#_Toc38013554)

[2.6 Deployment Architectures 12](#_Toc38013555)

[2.6.1 Simple Self Configuration 12](#_Toc38013556)

[2.6.2 Simple Management Stats Bus Configuration 12](#_Toc38013557)

[2.6.3 Fully Redundant Enterprise Stats – in one region 13](#_Toc38013558)

[2.6.4 Global Enterprise Stats 14](#_Toc38013559)

[2.6.5 Local Plug in - Stats Without Messaging 14](#_Toc38013560)

[3 Deployment Details 16](#_Toc38013561)

[3.1 Example Setup 16](#_Toc38013562)

[4 Installation 18](#_Toc38013563)

[4.1 StatsPump 18](#_Toc38013564)

[4.1.1 Pre-requisites 18](#_Toc38013565)

[4.1.2 Installation 19](#_Toc38013566)

[4.1.3 Uninstallation 20](#_Toc38013567)

[4.2 The Stats Receiver 20](#_Toc38013568)

[4.2.1 Pre-requisites 20](#_Toc38013569)

[4.2.2 Installation 20](#_Toc38013570)

[4.2.3 Uninstallation 21](#_Toc38013571)

[4.3 The Password Generation Utility 21](#_Toc38013572)

[4.3.1 Installation 21](#_Toc38013573)

[4.3.2 Uninstallation 22](#_Toc38013574)

[4.4 Receiver Plugins 22](#_Toc38013575)

[4.4.1 Installation 22](#_Toc38013576)

[4.4.2 Uninstallation 22](#_Toc38013577)

[4.5 Local Plug ins 22](#_Toc38013578)

[4.5.1 Installation 22](#_Toc38013579)

[4.5.2 Uninstallation 23](#_Toc38013580)

[4.6 InfluxDB 23](#_Toc38013581)

[4.6.1 Pre-requisites 23](#_Toc38013582)

[4.6.2 Installation 24](#_Toc38013583)

[4.6.3 Uninstallation 24](#_Toc38013584)

[4.7 Grafana 24](#_Toc38013585)

[4.7.1 Pre-requisites 25](#_Toc38013586)

[4.7.2 Installation 27](#_Toc38013587)

[4.7.3 Uninstallation 28](#_Toc38013588)

[5 Configuration Overview 29](#_Toc38013589)

[5.1 General 29](#_Toc38013590)

[5.1.1 StatsPump publish topic formats 29](#_Toc38013591)

[5.1.2 StatsPump message formats 30](#_Toc38013592)

[5.1.3 StatsPump and High Availability 31](#_Toc38013593)

[5.2 StatsPump 31](#_Toc38013594)

[5.2.1 Pollers Configuration 31](#_Toc38013595)

[5.2.2 Poller Groups Configuration 37](#_Toc38013596)

[5.2.3 Router Configuration 38](#_Toc38013597)

[5.3 Stats Receiver 41](#_Toc38013598)

[5.3.1 Special notes with local-mgmt-msg-bus 45](#_Toc38013599)

[5.3.2 Forcing fields to numeric 45](#_Toc38013600)

[5.4 InfluxDB 46](#_Toc38013601)

[5.5 Grafana 46](#_Toc38013602)

[6 Configuration procedures 47](#_Toc38013603)

[6.1 StatsPump 47](#_Toc38013604)

[6.1.1 Poller Configuration 47](#_Toc38013605)

[6.1.2 Groups Configuration 48](#_Toc38013606)

[6.1.3 Router Configuration 48](#_Toc38013607)

[6.2 InfluxDB 49](#_Toc38013608)

[6.2.1 InfluxDB Admin Interface 49](#_Toc38013609)

[6.3 Stats Receiver 51](#_Toc38013610)

[6.4 Grafana 51](#_Toc38013611)

[7 Operational Procedures 52](#_Toc38013612)

[7.1 StatsPump 52](#_Toc38013613)

[7.1.1 Start Up and Validation 52](#_Toc38013614)

[7.1.2 Shutdown 54](#_Toc38013615)

[7.1.3 Running StatsPump From an Application Server 54](#_Toc38013616)

[7.1.4 Running a local receiver plug in inside of StatsPump 54](#_Toc38013617)

[7.1.5 Upgrading StatsPump 54](#_Toc38013618)

[7.1.6 Dealing with SolOS router upgrades 55](#_Toc38013619)

[7.2 InfluxDB 55](#_Toc38013620)

[7.2.1 Start Up and Validation 55](#_Toc38013621)

[7.2.2 Shutdown 56](#_Toc38013622)

[7.2.3 Backing up Data 56](#_Toc38013623)

[7.2.4 Restoring Data 57](#_Toc38013624)

[7.3 Grafana 57](#_Toc38013625)

[7.3.1 Start Up and Validation 57](#_Toc38013626)

[7.3.2 Shutdown 58](#_Toc38013627)

[7.3.3 Backup 58](#_Toc38013628)

[7.3.4 Restore 59](#_Toc38013629)

[7.4 Stats Receiver 60](#_Toc38013630)

[7.4.1 Start Up and Validation 60](#_Toc38013631)

[7.4.2 Shutdown 62](#_Toc38013632)

[7.4.3 Running a local receiver plug in inside of StatsPump 62](#_Toc38013633)

[7.4.4 Upgrading Stats Receiver 62](#_Toc38013634)

[8 Creating Dashboards Using Grafana 63](#_Toc38013635)

[8.1.1 Configuring InfluxDB as a datasource 63](#_Toc38013636)

[8.1.2 Creating Dashboards 65](#_Toc38013637)

[8.1.3 Adding a new graph 66](#_Toc38013638)

[8.1.4 Using interval template variables for time resolution 67](#_Toc38013639)

[8.1.5 Using query template variables to make graphs extensible 70](#_Toc38013640)

[8.1.6 Graphing Poller Stats 74](#_Toc38013641)

[8.1.7 Dashboards created as a starting point 75](#_Toc38013642)

[9 Troubleshooting common issues 76](#_Toc38013643)

[9.1 StatsPump 76](#_Toc38013644)

[9.1.1 Error connecting to the router management interface 76](#_Toc38013645)

[9.1.2 Error connecting to the message-backbone router interface 76](#_Toc38013646)

[9.1.3 Incorrect Configuration 77](#_Toc38013647)

[9.2 Stats Receiver 78](#_Toc38013648)

[9.2.1 Error connecting to the message-backbone router interface 78](#_Toc38013649)

[9.2.2 Error writing data to InfluxDB 79](#_Toc38013650)

[9.3 Debugging SEMP request times: Writing Poller Statistics 80](#_Toc38013651)

[10 Known Issues and Limitations 82](#_Toc38013652)

[History 83](#_Toc38013653)

# Introduction

## Scope

The Solace PSG Enterprise Stats Solution is a SEMP based monitoring and data republishing tool. This can be used for real-time monitoring and historical analysis of the various statistics output by the Solace routers. The solution has multiple components:

1. An application called “The StatsPump”, which polls Solace routers for statistics using XML based Solace protocol called “SEMP” on the control plane of the router, reformats the data into a more easily digestible form, and republishes these statistics as messages on a pre-defined message bus for ease of consumption.
2. A special solace subscriber called “Stats Receivers” that subscribes to these messages and persist them into a database. Typically, this would be a time series database, such as InfluxDB
3. A visualization tool such as Grafana for operational monitoring of an entire estate.

The purpose of this document is to provide the install and operational guidelines for the StatsPump deployment. Specifically, the following topics are covered:

* Solution Overview
* Solution Components
* Use Case Description
* Deployment Architectures
* Installation
* Environment preparation
* Install and post install verification
* Configuration
* Configuration overview
* Configuration procedures
* Operational Procedures
* Startup/shutdown/restart
* Operational Monitoring
* Basic Troubleshooting and Escalation

## Target Audience

This document is intended for use as a reference by infrastructure engineers and system operators responsible for the installation and subsequent operational support of the StatsPump and related components.

## Related documents

These documents contain information related to the information in this document.

| Document Number | Document Title | Version (Optional) |
| --- | --- | --- |
| [SolCLIReference] | Solace Command Line Interface Reference | 7.1.1 or above |
| [InfluxDBDownload] | https://influxdata.com/downloads/ |  |
| [InfluxDBDocs] | https://docs.influxdata.com/influxdb/v0.13/ |  |
| [InfluxDBConfig] | https://docs.influxdata.com/influxdb/v0.13/administration/config#configuration-options |  |
| [InfluxDBHardwareSizing] | https://docs.influxdata.com/influxdb/v0.13/guides/hardware\_sizing/ |  |
| [GrafanaDownload] | http://docs.grafana.org/installation/rpm/ |  |
| [GrafanaDocs] | http://docs.grafana.org/guides/basic\_concepts/ |  |
| [GrafanaConfig] | http://docs.grafana.org/installation/configuration/ |  |
| [GrafanaKey] | https://grafanarel.s3.amazonaws.com/RPM-GPG-KEY-grafana |  |

Table 1 Related Documents

## Terminology

| Term / Acronym / Abbreviation | Definition |
| --- | --- |
| VPN | (Solace) Message-VPN |
| HA | High Availability |
| SEMP | Solace Element Management Protocol |
| HTTP | Hyper Text Transport Protocol |
| VMR | (Solace) Virtual Message Router |
| XML | eXtensible Markup Language |

Table 2 Terminology

# Solution Overview

Solace exposes a rich set of statistics and metrics for effective monitoring and analysis of Solace routers. Monitoring applications can query a variety of statistics from Solace routers using the Solace Element Management Protocol (SEMP). SEMP is a request/reply protocol that uses an XML schema to interact with all the managed objects available in a Solace router. Using SEMP, monitoring applications can send an XML encoded request over HTTP to retrieve statistical information for a Solace router.   
  
Many enterprises have a large number of Solace routers. The challenge at hand is to be able to monitor all of their routers in a unified, meaningful and effective way. There are a few big challenges to do this, namely:

* SEMP is a complex XML protocol that covers Solace provisioning, administration as well as monitoring. The overhead for application developers and operational engineers to be fluent with SEMP vocabulary and effective in utilizing it for monitoring is quite high.
* Many different actors in the enterprise each may have unique needs for statistics from one or more routers. This usually results in redundant SEMP queries to the Solace router which is ineffective at least, but can put unnecessary load on the router.

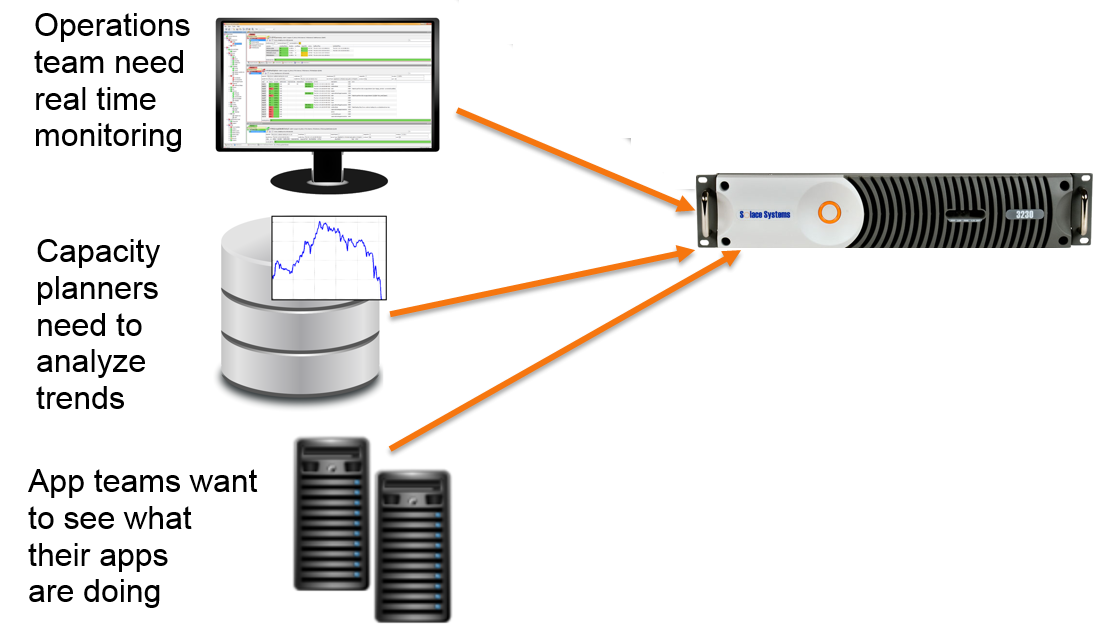


Figure 1: Monitoring using SEMP

The SEMP protocol is serviced by the control plane of the router. The control plane is reserved strictly for administrative control of the router and has much lower throughput than the message backbone of the Solace router. It has a finite capacity to process SEMP requests. If SEMP requests overload the router, the router will appear to respond poorly to other administrative clients, such as an CLI admin user. With a long list of clients requiring router statistics for a variety of purposes and potentially large number of VPNs within each router, an organization can quite easily bog down their routers with too many SEMP clients polling. Solace recommends a maximum of 10 SEMP requests per second, and a large number of uncoordinated clients requesting SEMP has the potential to reach and exceed that limit.

## The Solution

The Solace Enterprise Stats solution centralizes polling the router with SEMP and streamlines obtaining stats from multiple clients. The Stats Pump component is a highly configurable application program that polls routers in an efficient manner (bulk instead of piecemeal), reformats the data from the cumbersome SEMP XML into a more easily digestible format, and re-distributes it to a potentially long and disparate list of clients via the highly scalable Solace message bus.

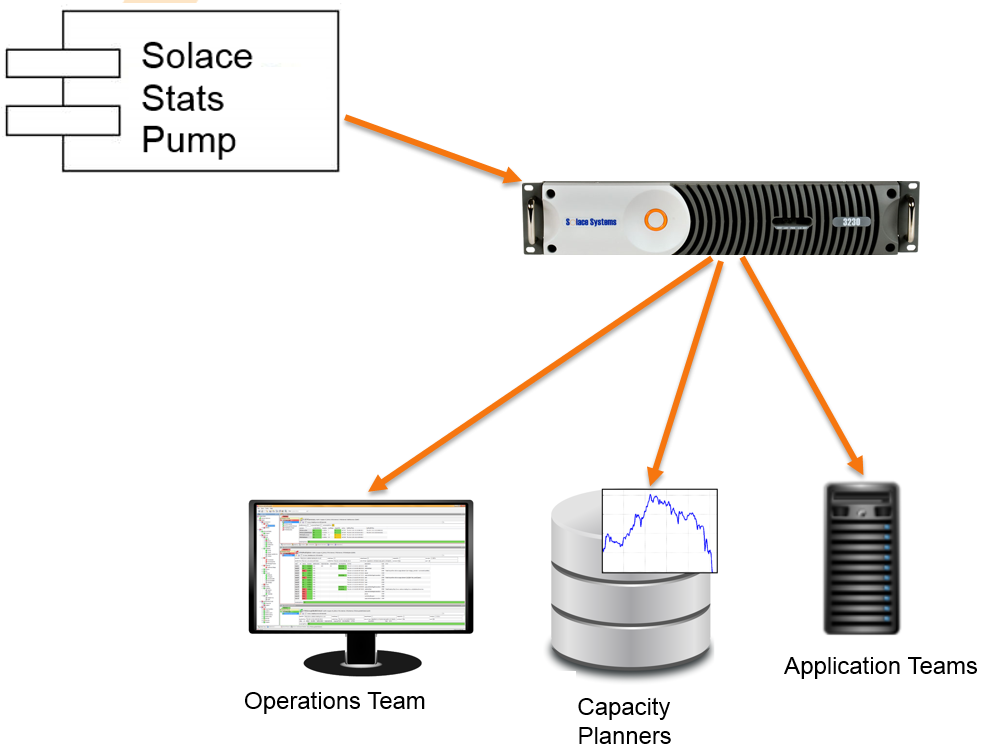


Figure 2: The StatsPump solution

A framework exists on the consumer side allowing teams to easily create stats consumers that wish to move the data into other monitoring systems. At the time of this writing, Solace has one implementation of a writer using this framework to InfluxDB. The bundled InfluxDB stats consumer receives statistics that the Pump has broadcast, and writes them into InfluxDB. From there, the data can be further processed or viewed using frontend tools such as Grafana.

## Components

There are essentially 3 main pieces of the puzzle: the pump, the receiver framework, and the concrete receivers.

## The Pump

The pump is a java application, configured using a series of XML configuration files. It connects to a set of routers, collects stats from the routers using SEMP, reformats the data into a configurable format, and re-broadcasts them as messages to a configured destination.

See the Stats Pump FS for more details.

## The Receiver Framework

A consumer plug-in component framework is available that can be used by developer to create a new stats consumer. The framework handles the configuration loading, router connections and router subscriptions, disconnect handling and re-connects, HA failover and queue management.

## Concrete Receivers

At the time of this writing, the Enterprise Stats solution is shipped with one concrete receiver, namely the InfluxDB receiver. However, based on the receiver framework as described above, additional concrete receivers can be added as required.

A concrete receiver will be given the opportunity to initialize itself from a properties file managed by the framework. From there, the Receiver framework will receive the messages and passes each stats message that arrives to the concrete receiver for further processing.

The receiver framework will also process the BROADCAST messages from the StatsPump. One of these messages is the start and end messages. Each time a poller runs, the StatsPump sends out a start message and upon its completion a corresponding end message. For example the “show client \* details” poller which sends a stats message for each client connected to the router will send a start message, followed by N client stats messages, followed by and end message. These starts and ends are handled by the receiver framework, and the concrete receiver will get simple method calls “onPollerStart()” and “onPollerEnd()”.

## Primary Use Case – Sequence

The primary use case flow for the Enterprise Stats is a sequence flow as described by the diagram below:

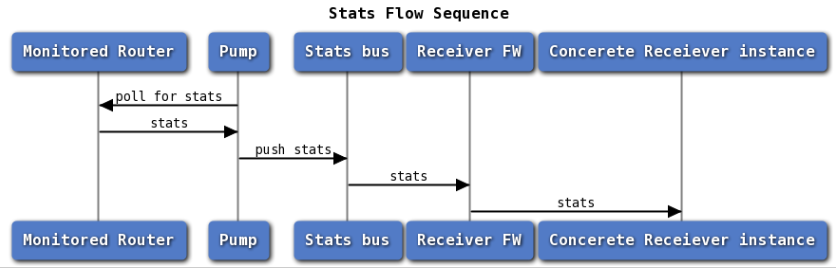


Figure 3: Use Case sequence diagram

The pump polls a set of routers to get stats, reformats them, and makes them available on the message bus through potentially a different router. One or more concrete receivers listen on these pre-defined topics for further processing.

Alternative flows surround the differences in timing of ACKs back to the router when guaranteed messaging is used. Using the JCSMP client, many messages can be in flight simultaneously. The receiver framework imposes a multi-threaded approach up the concrete receiver class, and also implements a local queueing mechanism that can keep messages in a RAM queue before being farmed out to a threaded receiver.

## Deployment Architectures

There is a wide variety of possible physical configurations for deploying and running the Enterprise Stats components. This section explores a few widely used ones, and discuss the factors that influence a particular deployment architecture. We will also look at different configurations best suited for different phases of the project.

### Simple Self Configuration

A very simple configuration of the solution is depicted in the diagram below. In this configuration, a single appliance is being monitored by a single instance of the Pump. The Pump transforms the statistics from appliance and publishes them back onto the same appliance which it is monitoring. A single receiver is listening for stats messages from the pump, and writes them to a time series database.   
  
Note that there are configurable settings in the Pumps configuration to cause the Pump to bypass stats generation for particular message VPNs.

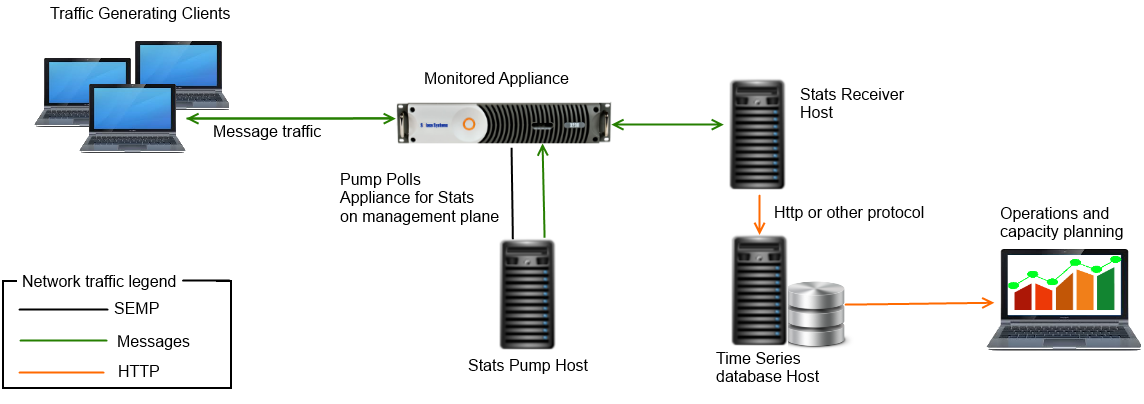


Figure 4: Simple Self Configuration

Host hardware and number of CPU requirements will vary, based on the planned capacity of the solution.

### Simple Management Stats Bus Configuration

In this configuration, instead of re-publishing the stats back into the same router being monitored, a separate Solace appliance, or possibly a Solace VMR is used as dedicated message bus for Enterprise Stats use. This separation has the advantage of stats traffic not interfering with other systems and not skewing the measured traffic with the stats messages themselves.

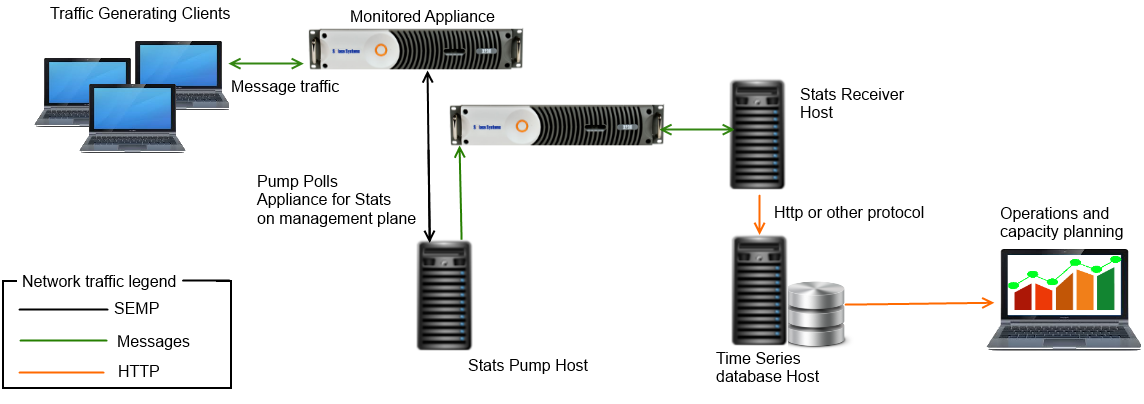


Figure 5:Simple Management Stats Bus Configuration

Host hardware and number of CPU requirements will vary, based on the planned capacity of the solution.

### Fully Redundant Enterprise Stats – in one region

In this full configuration, a redundant pair of Stats Pump hosts is configured to poll several Solace appliances. For e.g., all Solace routers in a region can be polled in a single group. A pair of redundant VMRs is used for publishing stats on the message bus. A redundant pair of Stats Receivers (possibly load balanced) process the stats messages and insert them into a time series DB, which are themselves deployed in a redundant configuration.

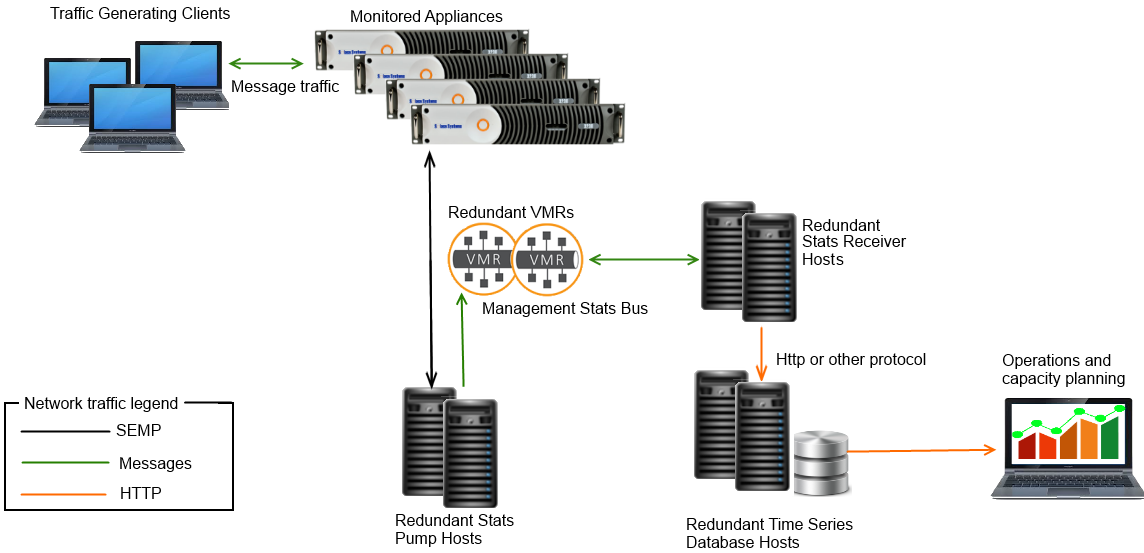


Figure 6: Fully Redundant Enterprise Stats

Host hardware and number of CPU requirements will vary, based on the planned capacity of the solution.

### Global Enterprise Stats

In this configuration, everything that is in play above in section 2.6.3 for a single region is replicated per region. In this case, options are available to either isolate each region from a statistical perspective, to roll them up into a unified global view, or both.

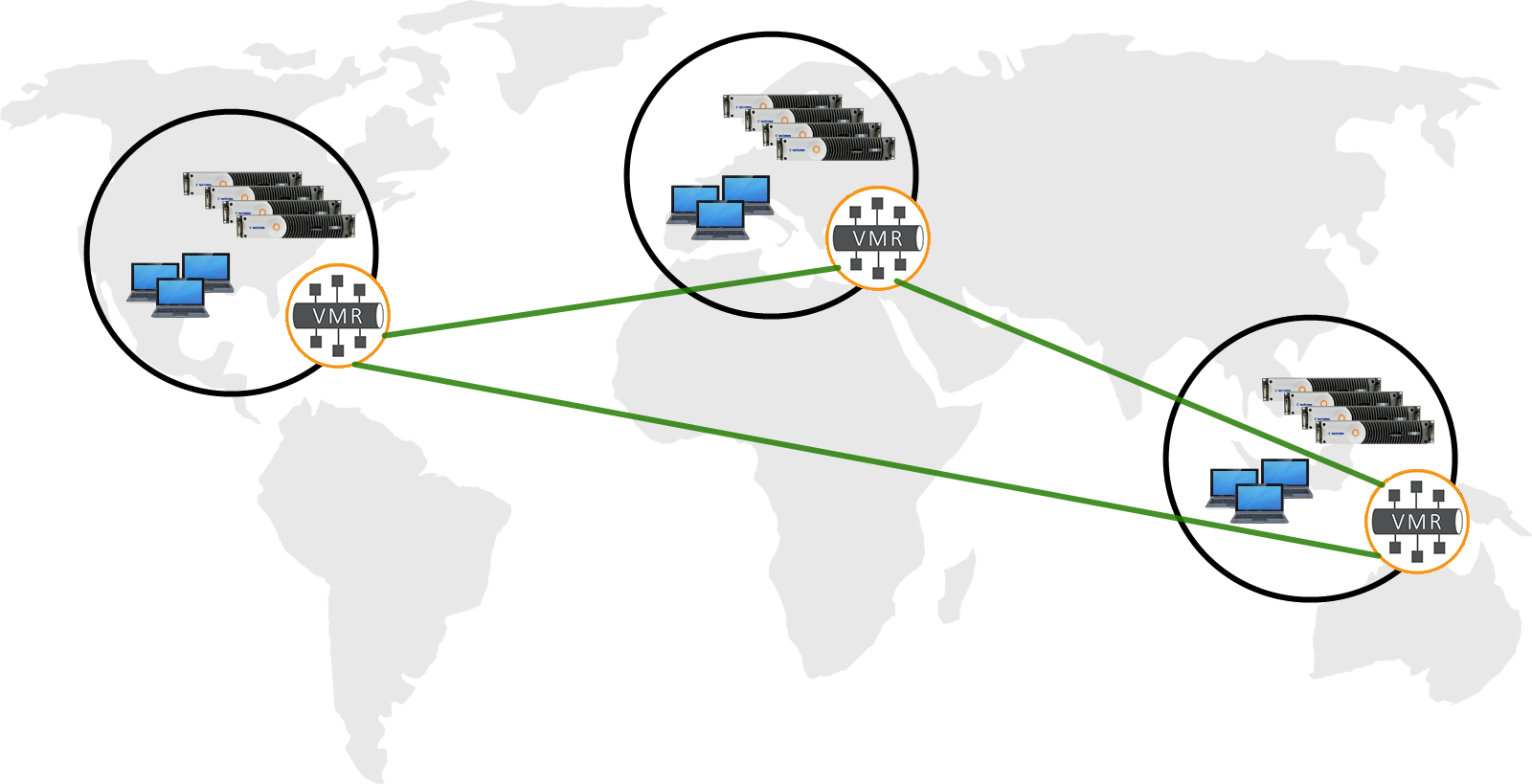


Figure 7: Global Enterprise Stats

It may be desirable for local operations teams to collect stats locally, as depicted in the single region scenario. On the other hand, both capacity planning and operations may require a global picture, and as such it may be desirable to roll all of the information up from each region. There are a couple of ways this can be done:

* Using the database technology to amalgamate local data into a global whole.
* Using Solace messaging to propagate statistical data to a global receiver. In this case, all of the local regions can be as depicted the single region scenario. Additionally, Solace features such as Multi Node Routing or VPN bridging can be used so the stats messages are also received by a global consumer.

### Local Plug in - Stats Without Messaging

The solution can be collapsed into a single process, where the StatsPump provides Stats to a receiver plug-in locally, without even sending messages on the message bus.

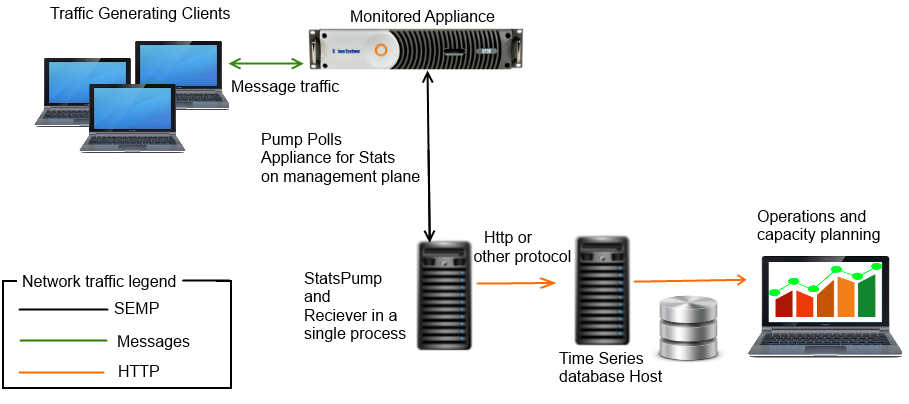


Figure 8: Local plug-in Stats

This deployment model is useful for scenarios where existing infrastructures may already exist for fan out and distribution of the Stats, and all that is desired is the data massaging functionality of the StatsPump. In this mode, the StatsPump behavior is unchanged (it polls one or more appliances using SEMP, reformats the data into the configured digestible form, and then passes this data to the Receiver plug-in for further handling). The difference is that the StatsPump simply passes the data directly to the Receiver plug-in via a Java interface, completely bypassing the message bus.

Note that it is possible to both use a local plug-in and the message bus simultaneously. In general, the StatsPump is configurable such that the stats retrieved from any appliance can be configured to push stats out to multiple destinations. The local plug-in is just another destination, and the StatsPump can both push onto the bus as well as to a local plug-in.

# Deployment Details

## Example Setup

An example deployment of the EnterpriseStats solution in a non-production environment is shown below. It follows the simple management stats bus deployment model, as described in Section 2.6.2

It is recommended to add redundancy to components for a production deployment, as indicated in Section 2.6.3

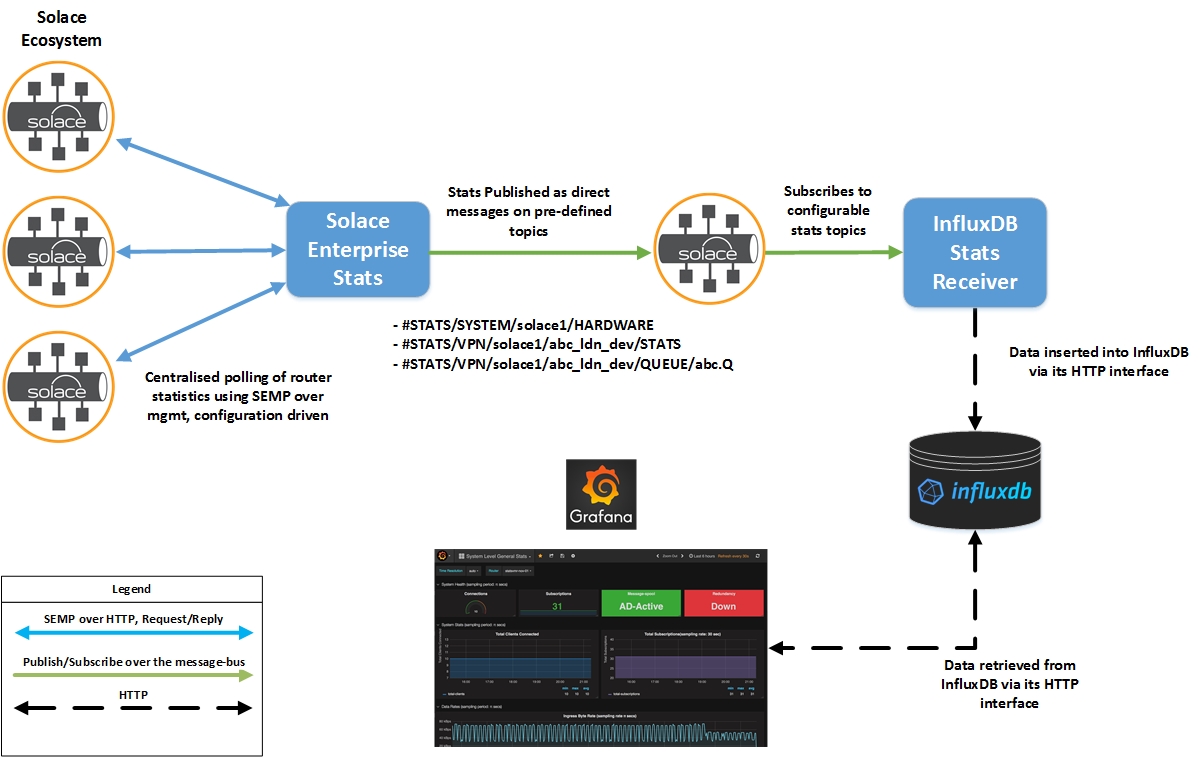


Figure 9: Sample deployment of Enterprise Stats

The Solace routers to be monitored by StatsPump are depicted at the left of the figure. Three message routers are shown in standalone mode to keep the diagram simple, but these can be extended as HA pairs. The Solace routers to be polled, SEMP requests to be performed, their corresponding polling interval, and destinations to republish this data to, can all be controlled by configuration files in the StatsPump package.

StatsPump can be configured to republish statistics from all Solace routers to a VPN on a separate Solace router or VMR - this is done to isolate the statistical data from the production environment being monitored and prevent the statistical information being published by StatsPump from consuming resources on the routers being monitored or skewing their statistics. This is the approach taken in this deployment.

At the time of writing of this document, the StatsPump only publishes statistical information as direct messages, and not guaranteed - this is done to decouple the StatsPump from downstream consumer applications. With guaranteed messaging, an offline downstream consumer may back-pressure the stats publisher, the pump. If guaranteed delivery is desired for Stats messages, a queue with subscription to required stats topics can be created. The InfluxDB receiver implementation supports receiving stats messages from a queue.

The database InfluxDB, the InfluxDB Stats Receiver, and the visualization tool Grafana, are installed on separate hosts. Both InfluxDB and Grafana are running as services. The InfluxDB Stats Receiver is running as a Java application, and configured to listen to topics on the "stats" message-vpn on the router to which statistics are being published; the receiver is deployed on the same host as InfluxDB in order to reduce the latency in writing data to the database. The InfluxDB Stats Receiver can be configured to receive the statistics message using either direct delivery by subscribing to the relevant statistics topics, or guaranteed delivery by binding to a queue configured with the appropriate topic subscriptions. Note that in the latter case, the StatsPump publishes messages as direct, and they get promoted to guaranteed messages in the queue and delivered to the InfluxDB Stats Receiver using guaranteed delivery.

The InfluxDB Stats Receiver receives the statistics messages from the Pump via the Solace router, extracts the statistics, and uses InfluxDB's HTTP API to write this information to InfluxDB through its HTTP interface (typically on port 8086).

Grafana has also been installed on the same host as InfluxDB to reduce the latency in retrieving data from the database. It has been configured through its admin interface to use InfluxDB as a data source, and it retrieves data from InfluxDB using InfluxDB's HTTP API. A Grafana administrator can use the web interface (typically on port 3000) to create dashboards, by querying statistics using the InfluxDB query language (InfluxQL), and can be customize it for display using the Grafana's rich toolset for charting and visualizing data.

Finally, an end-user uses a read-only "Viewer" account to log into Grafana's web interface to view the dashboards created, which displays the historical trend in statistical information.

For redundancy, we recommend using a warm standby model, and configure your monitoring solution to generate an alert when any of these processes goes down, so that you may start another instance.

# Installation

This section describes the procedures to install all the components in the Enterprise Stats monitoring solution. These components are independent of one another and installation a component is not dependent on the presence of other components.

## StatsPump

### Pre-requisites

* **Software:** The minimum supported JRE version for StatsPump is JRE v1.7.
* **Networking:** The host running StatsPump should be able to reach:
  + the management interface of the Solace routers it is polling
  + the message-backbone interface of the Solace routers it is publishing statistics to
* **Solace-Specific Configuration:**
  + A management CLI account should be created for StatsPump on each Solace router that the StatsPump is supposed to poll. This should have a minimum access-level of "read-only".

home

enable

configure

create username statspump\_ro password statspump

global-access-level read-only

message-vpn default-access-level read-only

end

* + One or more message-vpns should be created on the Solace router to which the StatsPump publishes statistics information depending on the destination configurations. Following CLI managed objects should be created for StatsPump:
    - message-vpn
    - client-profile - since the StatsPump publishes its statistics messages using Direct Delivery. The client profile which is used for the Pump should have the following settings:
      * Allow Guaranteed Message Send set to "No"
      * Allow Guaranteed Message Create set to "No"
      * Allow Guaranteed Message Receive set to "No"
    - [optional] an acl-profile with permissions to only connect and publish messages
    - client-username using the above client-profile and acl-profile, and optional password

The following CLI dump has a reference VPN Config for a “stats” vpn for the pump to publish stats.

<http://statsdemo.solace.com/assets/files/create_stats_vpn.txt>

### Installation

* Download the latest package for StatsPump from your designated area in the Solace SFTP site.
* Extract the contents of the package to the install location. The directory structure and details are as follows:
  + [installDir]/config: This contains the configuration files to customize the operation of StatsPump
  + [installDir]/logs: This contains the log files written by StatsPump.
  + [installDir]/doc: This contains the HTML documentation regarding StatsPump pollers
  + [installDir]/lib: This contains libraries required by the StatsPump to run.
  + [installDir]/bin: This contains the runnable scripts

The following scripts have been provided:

|  |  |
| --- | --- |
| **Script** | **Description** |
| statspump | Unix shell script to start the StatsPump process |
| statspump.bat | Window script to start the StatsPump |

Table 3: Stats Pump helper scripts

In addition, a simple service script file is provided to allow the StatsPump to be started and stopped using systemd service. This script file can be installed Unix and Linux systems that use systemd to handle starting of tasks and services during boot, and stopping them during shutdown and supervising them while the system is running.

To install the script file, run the following:

sudo cp scripts/statspump.service /lib/systemd/system/

sudo systemctl enable statspump.service

**NOTE**: when using the systemd script to start/stop the StatsPump, then the package should be placed under /opt, the configuration file should be placed in the config/ directory of the extracted package and called: appliances.xml. If named differently or placed in a different location, then update the systemd script file appropriately.

If you are using the systemd service script provided, update the log4j2.properties file to change the log file path to be under /var, such as:

property.filename = /var/log/statspump/statspump

### Uninstallation

The StatsPump package is shipped as a zip archive - in order to remove it, simply delete the directory containing the StatsPump files. This removes the StatsPump along with related configurations, logs, and other associated files.

## The Stats Receiver

Regardless of which specific database plug in you are working with, including your own implementation, there is a single runnable that you invoke. This is the Stats Receiver, aka the receiver framework.

### Pre-requisites

* **Software:** The minimum supported JRE version for StatsPump is JRE v1.7.
* **Networking:** The host running the StatsReceiver should be able to reach the message-backbone interface of the Solace router(s) that the StatsPump is publishing statistics to.
* **Solace-Specific Configuration:** StatsReceiver will connect to the message-vpn destination that the StatsPump is publishing statistics messages to.The configuration required depends on whether the Stats Receiver is using Direct or Guaranteed delivery to receive messages:
  + **For direct delivery**, a new client-username and new client-profile should be created for the StatsReceiver with an ACL-profile that only allows clients to connect and subscribe to messages. The message-vpn and client profile should deny all permissions for Guaranteed messaging. Optionally, The StatsReceiver can use the same credentials as the StatsPump to subscribe to messages, however this is not recommended in a production environment
  + **For Guaranteed delivery,** a new client-username and new client-profile should be created for the StatsReceiver with an ACL-profile that only allows clients to connect and subscribe to messages. The client profile should allow all permissions for Guaranteed messaging, and the message-spool size for the message-vpn should be set accordingly.

### Installation

* Download the latest package for StatsReceiver from your designated area in the Solace SFTP site.
* Extract the contents of the package to the install location. The directory structure and details are as follows:
  + [installDir]/config: This contains the configuration files to customize the operation of StatsReceiver
  + [installDir]/logs: This contains the log files written by StatsPump.
  + [installDir]/lib: This contains libraries required by the StatsReceiver to run.
  + [installDir]/bin: This contains the runnable scripts

The following scripts have been provided:

|  |  |
| --- | --- |
| **Script** | **Description** |
| stats-receiver.sh | Unix shell script to start the Stats Receiver process |
| stats-receiver.bat | Window script to start the Stats Receiver process |

Table 4: Stats Receiver helper scripts

In addition, a simple service script file is provided to allow the StatsReceiver to be started and stopped using systemd service. This script file can be installed Unix and Linux systems that use systemd to handle starting of tasks and services during boot, and stopping them during shutdown and supervising them while the system is running.

To install the script file, run the following:

sudo cp scripts/stats-receiver.service /lib/systemd/system/

sudo systemctl enable stats-receiver.service

**NOTE**: If using the systemd script to start/stop the Stats Receiver, the following considerations apply:

* + - * the package should be placed under /opt
      * the configuration file should be placed in the config/ directory of the extracted package and called: appliances.xml.

If named differently or placed in a different location, then update the systemd script file appropriately.

If you are using the systemd service script provided, update the log4j2.properties file to change the log file path to be under /var, such as:

property.filename = /var/log/stats-receiver/stats-receiver

### Uninstallation

The StatsReceiver package is shipped as a zip archive - in order to remove it, simply delete the directory containing the StatsPump files. This removes the StatsPump along with related configurations, logs, and other associated files.

## The Password Generation Utility

The Password generation utility is a standalone program used to encrypt authentication tokens such as Solace CLI user password before specifying them in configuration files. All the Solace configuration files used in this solution (the StatsPump and the StatsReceiver) require passwords be specified in encrypted form using this tool.

### Installation

* Download the latest package for Password Generation Utility from your designated area in the Solace SFTP site.
* Extract the contents of the package to the install location. The directory structure and details are as follows:
  + [installDir]/lib: This contains the jar files needed for the application
  + [installDir]/bin: This contains the runnable scripts

The following helper scripts have been provided:

|  |  |
| --- | --- |
| **Script** | **Description** |
| solace-pwd-util.sh | Unix shell script to call the Password Generation Utility |
| solace-pwd-util.bat | Window script to call the Password Generation Utility |

Table 5: Password Generation Utility helper scripts

### Uninstallation

The Password Generation Utility package is shipped as a zip archive - in order to remove it, simply delete the directory containing the StatsPump files. This removes the StatsPump along with related configurations, logs, and other associated files.

## Receiver Plugins

Each separate receiver plug in is simply a jar file that contains the classes needed for the plug in. This jar file is simply put into the lib folder of the Stats Receiver, where it can be located and used. At the time of this writing, the InfluxDB plug-in is the sole plug in available. In the future, more plug ins may be available. The rest of this section will detail the InfluxDB plug in.

### Installation

* Download the latest package for The InfluxDB plug in from your designated area in the Solace SFTP site.
* Extract the contents of the package to the lib folder of StatsReceiver.

### Uninstallation

The plug in is removed when you uninstall the StatsReceiver.

## Local Plug ins

In section 2.6.5, we discuss a different usage of the solution whereby the StatsPump, the Receiver framework and the specific receiver plug in all run in the same Java process. Installation is slightly different for this case. This section describes how to install the solution for this use case.

### Installation

* First, install the StatsPump normally as described above.
* Next, unpack the StatsReceiver package (the framework) to a temporary location. Copy the stats-receiver-[version].jar to the StatsPump’s lib folder.
* If using InfluxDB, download the InfluxDB plug in package as described above. This time, instead of copying the jar file to the Receiver’s lib folder, copy it to the StatsPump’s lib folder.

Modify the StatsPump’s launch script to include the two jars into the classpath.   
  
On Linux based systems, this is the bin\statspump script. On Windows based systems, modify the bin/statspump.bat file. In either file, modify the classpath line:  
  
*CLASSPATH=$APP\_HOME/lib/solace-statspump-2.1.0.jar:$APP\_HOME/lib/commons-lang-2.2.jar:$APP\_HOME/lib/commons-logging-1.1.1.jar:$APP\_HOME/lib/sol-common-7.0.0.63.jar:$APP\_HOME/lib/sol-jcsmp-7.0.0.63.jar:$APP\_HOME/lib/jsr173\_api.jar:$APP\_HOME/lib/slf4j-api-1.7.21.jar:$APP\_HOME/lib/javax.json-api-1.0.jar:$APP\_HOME/lib/commons-codec-1.6.jar:$APP\_HOME/lib/solace-pwd-utility-2.1.0.jar:$APP\_HOME/lib/log4j-api-2.7.jar:$APP\_HOME/lib/log4j-core-2.7.jar:$APP\_HOME/lib/log4j-slf4j-impl-2.7.jar:$APP\_HOME/resources*And add the references to the Stats Receiver and the plug in:

CLASSPATH=%APP\_HOME%\lib\solace-influxdb-2.1.0.jar;%APP\_HOME%\lib\stats-receiver-2.1.0.jar;   
  
In the event that you are using a later version, substitute the actual version you are using instead of the 2.1.0 versions shown here.

* Copy the sample receiver.properties file from the config folder under the temporary location you unpackaged and move it to a location accessible by StatsPump, presumably in the StatsPump’s config folder.

### Uninstallation

The local plugins are removed when you uninstall the StatsPump.

## InfluxDB

InfluxDB is an open-source time-series database, used for storing the statistics from StatsPump for historical analysis. The installation instructions for InfluxDB are correct as of InfluxDB version 1.3.1. For latest install instructions, refer to [InfluxDBDocs]

### Pre-requisites

* **System Requirements:**  Refer to the hardware sizing guidelines for InfluxDB at [InfluxDBHardwareSizing] for the system requirements. The system requirements depend on the amount of data being stored in the database; it is recommended to deploy InfluxDB on a node in accordance with the guidelines for the Low-Moderate load.
* **Network Connectivity:** InfluxDB must be installed on a host, such that the InfluxDB Stats Receiver can connect to it to write data, as well as the host running Grafana can connect to it to retrieve data.
* **Ports:** InfluxDB uses the following default ports for its operation; these can be changed in the configuration after installation if required
  + Admin Interface - Port 8083
  + HTTP API - Port 8086

Firewalls must be modified to allow incoming and outgoing access to these ports, if required.

### Installation

The following steps can be used to install InfluxDB on RPM based systems. For installation on other operating systems, refer to the documentation at [InfluxDBDocs]

**NOTE**: Root Access is required for installing InfluxDB, and to start/stop the influxdb service. However, InfluxDB runs as its own non-privileged user "influxdb"

* Download the latest RPM package, from [InfluxDBDownload]
* To Install, run:

$ sudo yum localinstall influxdb-<version\_number>.rpm

alternatively, if the installations using yum are not successful, use RPM to install:

$ sudo rpm -Uvh influxdb-<version\_number>.rpm

This command installs InfluxDB as follows:

* The InfluxDB binary is copied to/usr/bin/influxd
* The default configuration file can be found at/etc/influxdb/influxdb.conf

### Uninstallation

In order to uninstall InfluxDB, follow the below steps:

* If the InfluxDB service is running, stop it. For instructions, refer to Section 7.2.2
* If required to take the backup of the data for InfluxDB, refer to Section 0
* If you have used yum to install InfluxDB, run the below command to uninstall InfluxDB:

$ sudo yum remove influxdb

* If you have used RPM to install InfluxDB, to uninstall it, use:

$ sudo rpm -e influxdb-<version\_number>.rpm

## Grafana

Grafana is a visualization tool that connects to InfluxDB to retrieve data and can display it graphically for historical analysis. The installation instructions for Grafana are correct as of Grafana version 3.0. For latest install instructions, refer to [GrafanaDocs]

### Pre-requisites

* **Networking:** Grafana must be installed on a host, such that it has network connectivity to the HTTP port of the host running InfluxDB to retrieve data.
* **Solace-Specific Configuration:** There is no additional Solace configuration required for installing or running Grafana.
* **Ports:** Grafana uses the following default ports for its operation; these can be changed in the configuration after installation if required
  + Admin Interface - Port 3000

Firewalls must be modified to allow incoming and outgoing access to these ports, if required.

* **InitScripts, FontConfig and urw-fonts:**  The Grafana installation requires that the InitScripts, FontConfig and urw-fonts packages must be present in order to run Grafana. In order to check if these components are present, run:

[srajgopalan@athena ~]# sudo yum info initscripts

Loaded plugins: downloadonly, fs-snapshot, product-id, security, subscription-manager, versionlock

Updating certificate-based repositories.

base | 4.0 kB 00:00

errata | 1.9 kB 00:00

Installed Packages

Name : initscripts

Arch : x86\_64

Version : 9.03.27

Release : 1.el6

Size : 5.4 M

Repo : installed

From repo : anaconda-RedHatEnterpriseLinux-201111171049.x86\_64

Summary : The inittab file and the /etc/init.d scripts

URL : http://fedorahosted.org/releases/i/n/initscripts/

License : GPLv2 and GPLv2+

Description : The initscripts package contains the basic system scripts used to boot

: your Red Hat or Fedora system, change runlevels, and shut the system down

: cleanly. Initscripts also contains the scripts that activate and

: deactivate most network interfaces.

[srajgopalan@athena ~]# sudo yum info fontconfig

Loaded plugins: downloadonly, fs-snapshot, product-id, security, subscription-manager, versionlock

Updating certificate-based repositories.

Installed Packages

Name : fontconfig

Arch : x86\_64

Version : 2.8.0

Release : 3.el6

Size : 440 k

Repo : installed

From repo : anaconda-RedHatEnterpriseLinux-201111171049.x86\_64

Summary : Font configuration and customization library

URL : http://fontconfig.org

License : MIT

Description : Fontconfig is designed to locate fonts within the

: system and select them according to requirements specified by

: applications.

[srajgopalan@athena ~]$ sudo yum info urw-fonts

[sudo] password for srajgopalan:

Loaded plugins: fastestmirror, security

Loading mirror speeds from cached hostfile

\* base: mirror.vodien.com

\* extras: mirror.vodien.com

\* updates: mirror.vodien.com

Installed Packages

Name : urw-fonts

Arch : noarch

Version : 2.4

Release : 11.el6

Size : 4.2 M

Repo : installed

From repo : base

Summary : Free versions of the 35 standard PostScript fonts

URL : http://svn.ghostscript.com/ghostscript/tags/urw-fonts-1.0.7pre44/

License : GPL+ with exceptions

Description : Free, good quality versions of the 35 standard PostScript(TM) fonts,

: donated under the GPL by URW++ Design and Development GmbH.

:

: Install the urw-fonts package if you need free versions of standard

: PostScript fonts.

If the above packages are not installed, run the below command to install the packages:

$ sudo yum install initscripts fontconfig urw-fonts

### Installation

The following steps can be used to install Grafana on RPM based systems. For installation on other operating systems, refer to the documentation at [GrafanaDocs]

**NOTE**: Root Access is required for installing Grafana, and to start/stop the grafana-server service. However, Grafana runs as its own non-privileged user "grafana"

* Download the latest RPM package, from [GrafanaDownload]
* To Install, run:

$ sudo rpm -Uvh grafana-<version-number>.x86\_64.rpm

This command installs Grafana as follows:

* The Grafana binary is copied to/usr/sbin/grafana-server
* The init.d script is copied to /etc/init.d/grafana-server
* The default configuration file can be found at/etc/grafana/grafana.ini
* The logs directory can be found at: /var/log/grafana/
  + In order to start the Grafana service at system startup via systemd, run the below command:

$ sudo systemctl enable grafana-server.service

**NOTE:** The above commands install Grafana, but the service needs to be started explicitly. For instructions, refer to Section 7.3.1

If there are any errors encountered related to the RPM package not being verified, follow the below steps:

* Download the GPG key for Grafana from [GrafanaKey]
* Import the RPM Key

$ rpm --import <path-to-key>

* Verify the signature of the package. If all goes well, the following message is displayed "md5 gpg OK".

$ rpm -K <rpm-file >

* Retry the installation

### Uninstallation

In order to uninstall Grafana, follow the below steps:

* If is required to take a backup of the configuration, refer to Section 7.3.3
* If the Grafana service is running, stop it. For instructions, refer to Section 7.3.3
* Run the below command to identify the name of the Grafana package to be uninstalled:

$ rpm -qa | grep grafana

grafana-3.0.2-1463383025.x86\_64

* Run the below command to uninstall Grafana:

$ rpm -e <package-name>

# Configuration Overview

This section provides an overview of the configuration files and options present for the various components in the monitoring solution, and the meaning of each option. The first section provides general information on the StatsPump, while the procedure for configuring each of the components is detailed in the subsequent section.

## General

Before going into the details of StatsPump configuration, an overview of general concepts is provided below:

### StatsPump publish topic formats

Statistics messages published by StatsPump can be scoped as system or message-vpn level information. The topics on which the StatsPump publishes statistics differ based on the scope of the statistics, and these are defined below:

|  |  |
| --- | --- |
| **Information Scope/Type** | **Topic Structure** |
| SYSTEM | #STATS/SYSTEM/<ROUTER\_NAME>/<STAT\_TYPE>[/OPTIONAL\_OBJECT\_NAME] |
| VPN | #STATS/VPN/<ROUTER\_NAME>/<VPN\_NAME>/<STAT\_TYPE>[/OPTIONAL\_OBJECT\_NAME] |
| BROADCAST | #STATS/BROADCAST/<ROUTER\_NAME>/<BROADCAST\_TYPE> |

Table 6: StatsPump Topic formats

* SYSTEM-level messages will generally not be available for client application consumption within their VPNs. These types of state and statistics are intended for Middleware / management monitoring, and include appliance-wide and hardware-related metrics such as:
* aggregate appliance-wide message counts and rates
* total client connections
* thermal readings, fan speeds, and other environmental readings
* network configuration
* VRRP / redundancy status
* disk / hardware status
* etc.

For example, the redundancy information for a router is published on the topic: "#STATS/SYSTEM/<ROUTER\_NAME>/REDUNDANCY" and defined in its corresponding poller configuration.

* VPN-level state and statistical messages will include anything relating to, or scoped within, a VPN specifically. Several of the statistics messages will have an additional object name (e.g. queue name, client name, cache instance name) appended to the end of the topic; these allow fine-grained filtering of received messages using Solace's topic routing ability. Types of information gathered and sent as messages can include:
* VPN connection and subscription counts
* Total message and byte counts (both ingress and egress) and message rates and discard counts
* VPN message spool related information
* per-queue and per-topic endpoint information
* per-queue and per-topic endpoint message and byte rates (only SolOS R7.0 and greater)
* per-client information and statistics
* VPN bridge information
* SolCache information

For example, the details of queue "queue1" is published on the topic: "#STATS/VPN/<ROUTER\_NAME>/<VPN\_NAME>/QUEUE/queue1" and defined in its corresponding poller configuration.

* BROADCAST messages will contain various information, regarding the operation of the StatsPump, and can be used by message-bus subscriber clients to find the status of the Solace router connected to, the number of pollers running, topic format, etc. As of the time of writing, there are 4 different types of broadcast messages:
* **POLLERS**: a listing of all the various pollers running against the current appliance. Includes details about the topic list, what the actual SEMP call is, how often each poller is run, etc.
* **ANNOUNCE**: various announcements, and tips and tricks for receiving data from StatsPump
* **ROUTER**: periodic status message relating to the state of the Solace appliance, or appliance-pair if configured
* **START/END**: As each poller starts and completes, the StatsPump sends out broadcast messages indicating so. This is useful for clients that wish to build

The BROADCAST messages can be used by clients to find out status of the Solace router they are connected to, as well as all the pollers that are running and broadcasting in that VPN. Knowledge of the Pollers' topic format is very useful as it allows automated tooling to know which attributes can be indexed, and what their name should be.

### StatsPump message formats

The StatsPump has been designed to be able to republish statistics to the message bus with the payload in a number of different formats. Currently, there are several built-in "factories" for generating messages in different formats. These include:

* **SdtMapFactory:** the statistical data is formatted as a Solace SDT message, formed of nested key-value pairs, and requires the Solace API to decode
* **JsonMapFactory:** the statistical data is sent as a Text Message, and contains a JSON object representing the key-value pairs
* **FlatSdtMapFactory:** JMS does not allow for nesting within maps - it is not possible to have a map within a map; therefore, nested elements are flattened by having the keys appended together using the delimiter '/'. Array elements are indexed using |0, |1, and so on.
* **SempXmlFactory**: A fully SEMP schema compliant message that should be able to be fed directly into existing SEMP schema parsers
* **SempXmlFragmentFactory**: the SEMP XML is maintained, but all of the <rpc-reply><rpc><show>... pre/post-amble is trimmed off for brevity, leaving only the relevant statistics data in the message.

Each of these classes belong to the package: com.solace.psg.enterprisestats.statspump.containers.factories

The data contained in each format is equivalent. There are advantages to each - the SEMP XML format allows existing applications capable of parsing SEMP to change little, the SDT Map format is convenient as every Solace API has the ability to decode these, and JSON format allow the message body to pass unchanged into other downstream systems.

For examples on how a statistics message received from StatsPump looks like, refer to the StatsPump poller documentation found at <http://statsdemo.solace.com/docs>

**NOTE:** The current Solace StatsReciever framework is specifically coded to receive only **SdtMapFactory** formatted messages.

### StatsPump and High Availability

For Solace routers configured in a High-Availability Pair, each router is polled independently by the StatsPump. The StatsPump can be configured to consider a pair in either Active-Standby mode (where no clients connect using the backup message bus), or Active-Active (where both message buses are in use - N/A for Guaranteed Messaging) - more information is provided in the next section.

When a failover occurs, the <ROUTER\_NAME> portion of the topic will change to the mate name. Therefore, applications subscribing to StatsPump messages need to ensure their subscriptions will match both appliance names:

e.g. #STATS/VPN/ldnpsol1002\*/STATS, or #STATS/VPN/\*/STATS

In an Active-Standby configuration, data from only one appliance will ever be broadcast (for VPN and client info). In an Active-Active configuration, data from both appliances can be broadcast, so care must be taken to keep the information separate.

## StatsPump

There are three configuration files required for the running of StatsPump:

* Pollers Configuration
* Poller Groups Configuration
* Router Configuration

These are described in the following sections:

### Pollers Configuration

This is an XML file containing details that the StatsPump will retrieve from Solace Routers via SEMP. This file contains a number of "Pollers" - each Poller contains details of the SEMP request the poller will send to a Solace router, and how the SEMP response is to be parsed.

In addition, "Run Configurations" are defined, which are used to specify the conditions when a poller is supposed to be run. Each Poller definition is then tagged with a Run Configuration, which specifies the conditions under which the StatsPump should run the poller.

#### Run Configurations

In order to increase the efficiency of SEMP polling, the StatsPump can be configured to define the conditions under which a poller is to be run. It is not required to run all SEMP pollers on a router always; for example:

* the SEMP command to retrieve endpoint details should only be run on a router which has a message-spool state of "AD-Active"
* the SEMP command to retrieve the details of connected clients should only be run on the active router of an active-standby HA pair, but both routers in an active-active HA pair
* the SEMP commands to retrieve hardware details should always be run on a router, irrespective of its redundancy state.
* An administrator can define the conditions under which a SEMP command will be run on a Solace router, based on the model in which the Solace router is deployed in. There are 5 modes defined within the Solace Stats Publisher application for when a particular poller is supposed to run:

|  |  |
| --- | --- |
| **Mode** | **Description** |
| AD-Active | Only run this poller on a router when its message-spool state is AD-Active; For example: SEMP commands for retrieving endpoint details should be run only when the Solace router is AD-Active |
| PRIMARY\_LOCAL\_ACTIVE | Run this Poller on a router when its Primary Virtual Router is active, i.e., active for its own VRID; For example: In an active-standby HA pair, SEMP commands for retrieving connected client details should be run on the primary router under normal operating conditions |
| BACKUP\_LOCAL\_ACTIVE | Run this Poller on a router when its Backup Virtual Router is also active, i.e., active for its mate's VRID; This condition would occur only when a router in an HA-pair has failed over to its mate.For example: In an active-standby HA pair, when an HA-failover occurs to the backup routers commands for retrieving connected client details should be run on the backup router as it is for its mate's VRID |
| ALWAYS | Always run this poller, irrespective of VRID activity; For example: Chassis Stats, Environment Stats, etc. |
| NEVER | Never run this poller when the appliance is in this state |

Table 7: Run Configuration Modes

Based on the above modes, run configurations are defined on the pollers.xml file for the various modes of activity. Each run configuration defines four conditions:

* When to run the poller on the primary router in an active-standby HA pair
* When to run the poller on the backup router in an active-standby HA pair
* When to run the poller on the primary router in an active-active HA pair
* When to run the poller on the backup router in an active-active HA pair

**NOTE**: Run configuration definitions should never by modified by an administrator, and only referred when adding a new poller configuration.

An example of a run-configuration is:

<run-configuration>

<name>PBAA</name>

<run-on-primary-when-active-standby>PRIMARY\_LOCAL\_ACTIVE</run-on-primary-when-active-standby>

<run-on-backup-when-active-standby>BACKUP\_LOCAL\_ACTIVE</run-on-backup-when-active-standby>

<run-on-primary-when-active-active>ALWAYS</run-on-primary-when-active-active>

<run-on-backup-when-active-active>ALWAYS</run-on-backup-when-active-active>

</run-configuration>

* For an active-standby redundant pair this run-configuration will cause the SEMP command to be run only on the router which is currently providing service for the redundant pair
* For an active-active redundant pair, this run-configuration will cause the SEMP command to be run always on both the routers in the pair

A SEMP command which will use this run-configuration is "show stats client detail" - StatsPump will run this command only on the router providing service for an active-standby redundant pair, but both routers for an active-active redundant pair.

Another example of a run configuration is:

<run-configuration>

<name>PBPP</name>

<run-on-primary-when-active-standby>PRIMARY\_LOCAL\_ACTIVE</run-on-primary-when-active-standby>

<run-on-backup-when-active-standby>BACKUP\_LOCAL\_ACTIVE</run-on-backup-when-active-standby>

<run-on-primary-when-active-active>PRIMARY\_LOCAL\_ACTIVE</run-on-primary-when-active-active>

<run-on-backup-when-active-active>PRIMARY\_LOCAL\_ACTIVE</run-on-backup-when-active-active>

</run-configuration>

* For an active-standby redundant pair this run-configuration will cause the SEMP command to be run only on the router which is currently providing service for the HA pair
* For an active-active redundant pair, this run-configuration will cause the SEMP command both the active routers in the pair, as long as they are active for their own VRID.

A SEMP command which will use this run-configuration is "show message-vpn \* detail" - StatsPump will run this command only on the router providing service for an active-standby redundant pair, and only on the active routers in an active-active redundant pair (if one of the routers fails, it will not be polled)

#### Poller Description

Each SEMP command that the StatsPump is to poll is specified as a "poller". A poller can be of two types:

* "system-poller": These are SEMP commands for retrieving System wide statistics
* "vpn-poller": These are SEMP commands for retrieving message-vpn specific information.

Depending on the type of poller (System or Message-vpn) the StatsPump will decide which corresponding topic prefix to use when republishing the statistics data.

The various fields of a poller definition are explained in the below table:

|  |  |
| --- | --- |
| **Tag** | **Description** |
| name | CLI equivalent of the SEMP command for identification |
| description | Description of the command |
| semp-request | SEMP request to be sent |
| run-configuration | Run configuration for the SEMP command, as defined in the previous section, to control when the poller is supposed to run |
| destination | Can be one of MGMT, SELF or BOTH to indicate the destination type for publishing the statistics; It should be in accordance with destinations defined in the router configuration |
| topic-string-suffix | The topic suffix to the main StatsPump topics where the information will be published |
| base-tag | The XML path in the SEMP response from which the StatsPump should begin parsing of objects |
| vpn-name-tag | This element is only applicable to message-vpn pollers, and is used to indicate, along with the base-tag, the XML path in the SEMP response where the StatsPump can find the message-vpn name |
| object-tags | This is used to indicate one or more additional suffixes to the topic string where the StatsPump is to publish statistical information |
| min-poll-interval-sec | This allows an administrator to set the minimum configurable polling interval for this poller  **NOTE**: This feature is yet to be implemented and currently does not affect operation. |

Table 8: Tag Description for Poller Configurations

The "topic-string-suffix" is used to specify the suffix to the StatsPump topics that statistics messages are published on.

Administrators are given the flexibility of separating the statistics published backed to the message bus based on scope - Each Poller is can be configured as being of interest to just the "management" destinations, the "self" per-VPN destinations (i.e. where the particular data came from), or "both". By default, almost all of the SYSTEM-level Pollers have been assigned to just MGMT, which means that they will not get sent to every client VPN as well. These destinations are defined in the router configuration file, defined in Section 5.2.3.

There are certain pollers built into the StatsPump as these are required for the utility to run:

* show hostname
* show message-spool
* show redundancy detail
* show message-vpn \* stats detail

Each installation of StatsPump comes pre-loaded with a number of pollers in the pollers.xml file in the "configs" directory and can be used as a starting point for retrieving statistics. If an administrator wished to add a new Poller, they have only to look at the many examples within the Poller file. An example of a poller is as follows:

<system-poller>

<name>show alarm</name>

<description>Any currently triggered hardware alarms. This would include: disk down, fibre channel link down, SAN unreachable,

uncorrectable memory error, fan speed low, etc.</description>

<semp-request>

<rpc semp-version='%s'>

<show>

<alarm/>

</show>

</rpc>

</semp-request>

<run-configuration>AAAA</run-configuration>

<destination>MGMT</destination>

<topic-string-suffix>ALARMS</topic-string-suffix>

<base-tag>/rpc-reply/rpc/show/alarm/alarms</base-tag>

</system-poller>

This poller:

* Polls the Solace router for system-wide alarms using the SEMP command for "show alarm"
* Publishes the statistics on the topic "#STATS/SYSTEM/~ROUTER\_NAME~/ALARMS
* This information is only published to the MGMT destination for system wide statistics.
* The run configuration used (AAAA) indicates that this poller is to be run always, irrespective of the redundancy state on the router.

Another example is for the client-level statistics:

<vpn-poller>

<name>show client \* stats</name>

<description>Shows statistics and rates for client connections</description>

<semp-request>

<rpc semp-version='%s'>

<show>

<client>

<name>\*</name>

<vpn-name>\*</vpn-name>

<stats/>

<count/>

<num-elements>1000</num-elements>

</client>

</show>

</rpc>

</semp-request>

<run-configuration>PBPP</run-configuration>

<destination>BOTH</destination>

<topic-string-suffix>CLIENT\_STATS/~CLIENT\_NAME~</topic-string-suffix>

<base-tag>/rpc-reply/rpc/show/client/\*/client</base-tag>

<vpn-name-tag>/message-vpn</vpn-name-tag>

<object-tags>

<object-tag>

<name>CLIENT\_NAME</name>

<tag>/name</tag>

</object-tag>

</object-tags>

</vpn-poller>

This poller:

* Polls the Solace router for client-level statistics using the SEMP command for "show client \* stats"
* Publishes the statistics on the topic #STATS/VPN/~ROUTER\_NAME~/<VPN\_NAME>/CLIENT\_STATS/<CLIENT\_NAME>" - The message-vpn name is decided by the <vpn-name-tag> and the <CLIENT\_NAME> by the "<object-tag>
* This information is only published both to the MGMT and SELF destinations for system wide statistics.
* The run configuration used (PBPP) indicates that this poller is to be run on a router, only if it is actively providing service to clients.

### Poller Groups Configuration

Related pollers defined in the pollers.xml file can be classified into poller groups, for the ease of configuration and operation. Routers to be monitored can then be tagged with poller groups, rather than the individual pollers themselves. Each poller group contains the name of the poller, followed by the polling interval - an administrator can use this to control the frequency with which data is retrieved from a router. High priority statistics like the number of clients connected to a router, or queue details can be polled with a higher frequency, while lower-priority statistics, such as environment details, can be polled at lower frequencies.

A couple of examples of poller group definitions are shown:

<poller-group>

<name>Appliance</name>

<pollers>

<poller><poller-name>show hardware details</poller-name> <poll-interval-in-sec>300</poll-interval-in-sec></poller>

<poller><poller-name>show hardware post</poller-name> <poll-interval-in-sec>900</poll-interval-in-sec></poller>

<poller><poller-name>show alarm</poller-name> <poll-interval-in-sec>60</poll-interval-in-sec></poller>

...

...

</poller-group>

<poller-group>

<name>Appliance SLOW</name>

<pollers>

<poller><poller-name>show hardware details</poller-name> <poll-interval-in-sec>900</poll-interval-in-sec></poller>

<poller><poller-name>show hardware post</poller-name> <poll-interval-in-sec>1800</poll-interval-in-sec></poller>

<poller><poller-name>show alarm</poller-name> <poll-interval-in-sec>60</poll-interval-in-sec></poller>

...

...

</poller-group>

Each installation of StatsPump comes pre-loaded with standard groups configuration in the file "groups-default.xml" in the "config" directory, and can be used as a starting point for retrieving statistics.

### Router Configuration

The last piece in StatsPump configuration is the configuration of the routers to be monitored - this is present in the router configuration file. This file contains details of the various routers to be polled by StatsPump, as well as where the statistics are to be published.

A separate definition needs to be added for each appliance to be monitored, with the tag "appliance"

The various tags in the router configuration are as follows:

|  |  |
| --- | --- |
| **Tag** | **Description** |
| appliances | Set of routers to be monitored by StatsPump |
| publishTtl | Specify a TTL for statistics messages published (ms) - this is a global setting for all appliances. |
| publishDmqEligible | Specify whether the statistics messages published are to be moved to a Dead Message Queue on expiry of TTL. Can be either "true" or "false”. This is a global setting for all appliances specified in the file. |
| appliance | Configuration definition for a single router to be monitored |
| name | Unique identifier for the router |
| poller-group | Specify one or more poller-groups to be run on the router; the poller groups must be defined in the poller groups configuration file |
| primary | Connection details for the primary router in a redundant pair. Includes fields:   * "semp-host": management IP address/DNS and port of the broker * "cli-username": CLI username account for connection * "cli-password": Password for the above CLI username. The password needs to be entered in the encrypted form; In order to encrypt the password, use the genPasswd utility provided in the installation directory. * Using Secure SEMP connections is supported from version 2.6.0 onwards. To use SEMP with SSL, specify the tag <secure/> |
| backup | Connection details for the backup router in a redundant pair. For monitoring routers deployed as standalone, no backup needs to be specified. Includes the management IP address of the router, CLI Username and password, and whether to use secure connections. The "mode" attribute can be either "active" or "standby", indicating the deployment mode of the redundant pair as either "active-active" or "active-standby" |
| mgmt-msg-bus | Zero or more destinations for system-level information published, analogous to the concept of a management message-vpn in Solace. Details of fields are indicated in the subsequent paragraph |
| self-msg-bus | Zero or more destinations for message-vpn-level information published. This can either be the same message-vpn being polled for statistics (hence the name "self") or another message-vpn altogether. Details of fields are indicated in the subsequent table |
| local-mgmt-msg-bus | Zero or more destinations for local plug ins. This is effectively configuring the receiver inside of the StatsPump. This section contains:  - “local-listener-class”: Changing this property from the default is not recommended. The default setting is the Receiver framework.  - “local-listener-config”: Identifies the receiver properties file that configures the listener. See notes below about this.  -“container-factory-class”: Changing this property from the default is not recommended, unless you have a custom receiver that does not use the StdMap message format. |

Table 9: Tag description for router configurations

A destination (both MGMT or SELF) have the following attributes:

* "host": IP address of the destination router
* "vpn": Name of the message-vpn where stats are published to
* "client-username": Client Username for connecting to the above message-vpn
* "password": Password for the above client username. The password needs to be entered in the encrypted form; In order to encrypt the password, use the genPasswd utility provided in the installation directory.
* "container-factory-class": Determines the Solace message format for the stats data published. At the time of writing of the document, it is of the form "com.solacesystems.psg.enterprisestats.statspump.containers.factories.<factory-name>, where the factory name can be one of:
  + - JsonMapfactory
    - SempXmlFactory
    - SempXmlFragmentFactory
    - SdtMapFactory
    - FlatSdtMapFactory

If no message-vpn name is provided for the SELF message-bus destination, it is assumed that the message-bus provided is same as the router being polled, and the StatsPump will attempt to connect to each individual message-vpn and publish statistics back to the message-vpn it relates to.

In addition, Access Control Lists can be applied to control the publishing of statistics to destinations as shown:

|  |  |
| --- | --- |
| **Tag** | **Description** |
| vpn-connection-acl-settings | Define the ACL settings for a message-bus destination |
| default-action | Attribute for "vpn-connection-acl-settings", can be one of "ALLOW" or "DISALLOW" |
| vpn-exception | The names of message-vpns which are exceptions to the default ACL action; These can contain wildcards along with the message-vpn name, such as "RAZOR\*" |

Table 10: Tag description for additional router configurations

Each installation of StatsPump comes with a number of sample router configurations for different scenarios for the message-bus and self-destinations - these can be found at [$installDir]/config/. These can be used as a starting point for the router configuration relevant to the StatsPump deployment.

#### Special notes when using secure connections

Since EnterpriseStats version 2.6.0, Secure connections over SEMP are supported when polling the brokers for metrics. In order to enable this, the <secure/> tag must be specified in the broker configuration (as indicated above). Additionally, the public certificate of the broker must be imported into the default Java truststore on the host machine running the StatsPump.

To obtain the public certificate of the broker, run:

$ openssl s\_client -connect <serverIP:port> -servername <hostname> < /dev/null | sed -ne '/-BEGIN CERTIFICATE-/,/-END CERTIFICATE-/p' > public.crt

To import this into the default Java truststore:

sudo keytool -import -alias <hostname> -keystore <truststore> -file public.crt

where, <truststore> is the path to the default Java truststore.

<hostname> is the hostname of the Solace broker

<serverIP:port> is the management IP and secure port on the broker.

#### Special notes with local-mgmt-msg-bus

When using a local plug in receiver, you will use the exact same properties file that you would use to configure a receiver as outlined below. However, some properties are not applicable in this use case (the StatsReceiver’s connection to the Solace router is not applicable). This is detailed in the Stats receiver configuration notes below.

## Stats Receiver

The configuration for the StatsReceiver framework is driven by a configuration file, which is bundled in the StatsReceiver Installation as [$installDir]/config/influx.properties, The various options are explained below:

|  |  |
| --- | --- |
| **Configuration Option** | **Description** |
| SOLACE\_HOST | Message-backbone IP Address and port (IP:Port) of the Solace router to connect to, along with port |
| SOLACE\_VPN | Name of the message-vpn to connect to and subscribe to stats |
| SOLACE\_USERNAME | Client-username used for making the connection to the router |
| SOLACE\_PASSWORD | Password for the above client-username. The password needs to be entered in the encrypted form; In order to encrypt the password, use the genPasswd utility provided in the installation directory. |
| SOLACE\_RECONNECT\_RETRIES | Optional. Controls the number of times the receiver will try to re-connect before giving up. Use a value of -1 for “infinite retry”. Default value is 5. |
| SOLACE\_CONNECT\_RETRIES | Optional. Controls the number of times the receiver will try and initial connection at program launch before giving up. Use a value of -1 for “infinite retry”. Default value is 1. |
| SOLACE\_CONNECT\_RETRIES\_PER\_HOST | Optional. Controls the number of times the receiver will try to re-connect to a specific host in a host list before giving up. Default value is 20. |
| SOLACE\_RECONNECT\_RETRY\_WAIT\_MS | Optional. The duration (in MS) the receiver should wait in between re-connection attempts. Default value defers to the Solace JCSMP API. |
| SOLACE\_CLIENT\_NAME | Optional. The name for the client connection. |
| SOLACE\_APP\_DESCRIPTION |  |
| SOLACE\_TOPICS | One or more topic subscriptions that the InfluxDB stats receiver will subscribe to for statistics. Multiple topics should be separated by the semi-colon delimiter (";") |
| SOLACE\_QUEUE | If it is desired to use guaranteed delivery for messages receiving message, the SOLACE\_QUEUE property needs to be set to the name of a queue which has been pre-created. The Stats Receiver will fail to start unless this queue has been pre-created.  It needs to be ensured that the queue is pre-configured with the appropriate subscription topics for receiving statistics messages. These topics should be the same as entered above in the SOLACE\_TOPICS property. In addition, the queue must be configured with a topic subscription for the broadcast topic for poller info messages, without which it will not start writing messages to InfluxDB: "#STATS/BROADCAST/\*/POLLERS"  Note that the SOLACE\_TOPICS property is ignored when a SOLACE\_QUEUE is provided. |
| TAP\_PLUGIN\_CLASS | The name of the plug in class. If using Solace’s InfluxDB plug in, this value should be:  com.solace.psg.enterprisestats.receiver.influxdb.InfluxDBStatsTap |
| DB\_FIELD\_SUBSCRIPTIONS | This setting allows you to control which fields are used in the stats messages sent out by the Pump. A value of “>”, means all fields will be used for all stats messages. To restrict specific messages, use the stat message name and field. For example, the following entry selects 3 specific fields for inclusion of SYSTEM\_MEMORY stat messages:  # SYSTEM\_MEMORY/subscription-memory-usage-percent;\ # SYSTEM\_MEMORY/physical-memory-usage-percent;\ # SYSTEM\_MEMORY/slot-infos/\*/nab-buffer-load-factor;\  This next example selects all fields for the SYSTEM\_STATS\_CLIENT message.  # SYSTEM\_STATS\_CLIENT/>;\ |
| DB\_FIELD\_FORCE\_TO\_NUMERIC | This setting allows you to control which fields will be forced to numeric values. There are a number of fields that the SEMP protocol returns as strings instead of their underlying numeric values (see notes below). For example, the following entry marks a field to be converted:  SYSTEM\_MSG-SPOOL/mate-disk-partition-usage;\  The syntax for this field in the configuration file is exactly the same as above for the DB\_FIELD\_SUBSCRIPTIONS, except that wildcarding is not support here.   See notes below under section “5.3.2 Forcing fields to numeric” for more information. |
| ENABLE\_POLLER\_STATS | Optional. To receive StatsPump internal poller metrics enable the following.  Default is: false (i.e. collection of poller stats is disabled). |
| INFLUXDB\_POLLER\_STATS\_DB | Optional. The database name which InfluxDB will be writing StatsPump poller stats metrics to. If ENABLE\_POLLER\_STATS is set to true then this database  should be created prior to running the stats-receiver with the InfluxDB plugin.  Default is: statspump\_poller\_stats |

Table 11: Stats Receiver configuration options

Additionally, the following parameters can be used for performance tuning of the StatsReceiver:

|  |  |
| --- | --- |
| Configuration Option | **Description** |
| THREAD\_AWAITING\_QUEUE\_SIZE | This setting this is the maximum number of tasks that the receiver will cache in memory, awaiting to be written out to a database via the Receiver Plugin.  It is the maximum number of messages that the receiver will pull off of the queue and have cached in RAM waiting to be processed (as tasks; there is a 1-1 ratio of messages to tasks). This acts as a local shock absorber in addition to the shock absorption provided by the Solace queue, and helps to keep the receiver running at full throttle during bursts.  There are many variables that could effect this including amount of RAM, number of NICs, number of CPUs. An optimal value of 1000 is used in the samples and may be tuned depending on the environment if required. |
| THREAD\_POOL\_MINIMUM\_SIZE | This setting is the minimum number of threads in the pool for writing messages to InfluxDB. It is the minimum number of threads that are always available, regardless of how little traffic is present.  There are many variables that could effect this including amount of RAM, number of NICs, number of CPUs. An optimal value of 3 is used in the samples and may be tuned depending on the environment if required. |
| THREAD\_POOL\_MAXMUM\_SIZE | This setting is the maximum number of threads in the pool for writing messages to InfluxDB. If all existing threads are busy, new threads will be created on demand, up to a max of this number.  There are many variables that could effect this including amount of RAM, number of NICs, number of CPUs. An optimal value of 10 is used in the samples and may be tuned depending on the volume of incoming statistics messages in the environment, if required. |

Table 12: Performance tuning options

Finally, each specific plug in will have its own set of configuration parameters. At runtime, the receiver framework will give the plug in an opportunity to load it’s properties from this same file. The following parameters are required for the InfluxDB receiver plug in:

|  |  |
| --- | --- |
| Configuration Option | **Description** |
| INFLUXDB\_HOST | IP Address of the host running InfluxDB along with the port for the HTTP API (default 8086) for writing information to the database |
| INFLUXDB\_DB | Name of the Database that statistical information is written to. This database needs to be created prior to starting up the InfluxDB Stats Receiver |
| INFLUXDB\_USER | Name of the InfluxDB user for writing data to the database. This user must be given write permissions to the above database |
| INFLUXDB\_PW | Password for the above user. The password needs to be entered in the encrypted form; In order to encrypt the password, use the genPasswd utility provided in the installation directory. |
| INFLUXDB\_POLLER\_STATS\_DB | Optional. The database name which InfluxDB will be writing StatsPump poller stats metrics to. If ENABLE\_POLLER\_STATS is set to true then this database  should be created prior to running the stats-receiver with the InfluxDB plugin.  Default is: statspump\_poller\_stats |
| INFLUXDB\_HTTP\_CONNECT\_TIMEOUT | This setting controls how long the HTTP request will wait before timing out on the connection for inserts sent to InfluxDB. Default is 5000 (ms, or 5 secs) |
| INFLUXDB\_HTTP\_READ\_TIMEOUT | This setting controls how long the HTTP request will wait before timing out on inserts sent to InfluxDB. Default is 5000 (ms, or 5 secs) |
| INFLUXDB\_HTTP\_READ\_RETRIES | This setting controls the number of times the InfluxDB tap will retry upon an HTTP error before discarding the statistic. The default value is 1 - just 1 attempt. |

Table 13: InfluxDB plug in options

### Special notes with local-mgmt-msg-bus

If you are using a local plug in, this StatsReceiver confg file uses the same format, but simply ignores inapplicable settings regarding connecting to the Solace message router. The following setting do not apply when running a local receiver:

* SOLACE\_HOST
* SOLACE\_VPN
* SOLACE\_USERNAME
* SOLACE\_PASSWORD

### Forcing fields to numeric

The legacy SEMP schema has a number of fields which are returned as strings, which are actually numeric values. That is, values come back from the router in a string format, when the meaning of the field is numeric. Examples include floating point numbers which the “%” character post fixed as well as values that come back as “N/A” or “-“, instead of null.

If you have downstream processing that needs to do math on these fields, this may be problematic or at least cumbersome to have to deal with these values. To remedy this, the Receiver can be configured to force these values to numerics, by stripping off the unwanted characters and allowing them to be treated as numeric, as they should be.

Out of the box, the solution comes with many of such fields configured. You can alter the configuration of the receiver as desired to mark more or less fields for this processing. See section 5.3 for details on how to change this configuration in the receiver.

## InfluxDB

The default configuration file for InfluxDB is present at: /etc/influxdb/influxdb.conf. For more information on InfluxDB configuration options, please refer to [InfluxDBConfig]

## Grafana

The default configuration file for Grafana is present at: /etc/grafana/grafana.ini For more information on Grafana configuration options, please refer to [GrafanaConfig]

# Configuration procedures

This section contains the procedures for configuring each of the components in the StatsPump monitoring solution initially, before starting them up. Subsequent configuration changes may be performed by referring to the descriptions in the previous section.

Due to dependencies between various components, it is recommended to configure the components in the order in which they are presented.

## StatsPump

### Poller Configuration

The default poller configuration file which comes with the StatsPump installation, is a great starting point, and can be used as-is. This can be found at [$installDir]/config/pollers.xml. The default pollers file provides common monitoring statistics for a Solace environment. Additional pollers can be added for monitoring any statistics that are not included in the default poller.

The default StatsPump poller configuration file has been created to re-publish statistics to two separate destinations:

* "MGMT" for system statistics
* "SELF" for message-vpn specific statistics

In the current deployment, all the statistics retrieved by StatsPump are being published to a single destination: message-vpn "stats". Hence a single destination has been created in the router configuration for both system and client traffic, and the destination for any message-vpn specific poller has been set to "BOTH"; the reason for this being statistics retrieved by a message-vpn specific poller cannot be published to a "MGMT" destination, by the StatsPump design.

Below is an example of a system poller configuration:

<system-poller>

<name>show memory</name>

<description>Details on the various memory pools and buffers for the whole router</description>

<semp-request>

<rpc semp-version='%s'>

<show>

<memory/>

</show>

</rpc>

</semp-request>

<run-configuration>AAAA</run-configuration>

<destination>MGMT</destination>

<topic-string-suffix>MEMORY</topic-string-suffix>

<base-tag>/rpc-reply/rpc/show/memory</base-tag>

</system-poller>

Below is an example of a message-vpn poller configuration:

<vpn-poller>

<name>show client \* stats</name>

<description>Shows statistics and rates for client connections</description>

<semp-request>

<rpc semp-version='%s'>

<show>

<client>

<name>\*</name>

<vpn-name>\*</vpn-name>

<stats/>

<count/>

<num-elements>1000</num-elements>

</client>

</show>

</rpc>

</semp-request>

<run-configuration>PBPP</run-configuration>

<destination>BOTH</destination>

<topic-string-suffix>CLIENT\_STATS/~CLIENT\_NAME~</topic-string-suffix>

<base-tag>/rpc-reply/rpc/show/client/\*/client</base-tag>

<vpn-name-tag>/message-vpn</vpn-name-tag>

<object-tags>

<object-tag>

<name>CLIENT\_NAME</name>

<tag>/name</tag>

</object-tag>

</object-tags>

</vpn-poller>

### Groups Configuration

The default groups configuration file which comes with the StatsPump Installation at [$installDir]/config/groups\_default.xml, is also a great place to start, and can be used as-is. If there are any new groups to be defined, or polling intervals to be changed, these must be done at the groups configuration file and the router configuration updated accordingly.

### Router Configuration

Each StatsPump installation comes with various router configuration files, to illustrate how various destinations can be configured. These can be used as a starting point to create your configuration. You will have to create one for your usage, with pointing the Pump at your routers as the most basic step. The following information is required:

|  |  |
| --- | --- |
| **Configuration Option** | **Description** |
| Routers to be polled by StatsPump | Identity the routers to be piolled |
| Publish TTL (ms) | 30000 (for Guaranteed delivery, causes messages to be dropped on the router if the receiver is offline or can’t keep up) |
| Publish DMQ Eligible | True |
| Poller Groups to be used | Appliances, Direct and AD |
| CLI Username | The username for connecting to the administration interface of the polled router. |
| Management destination message-vpn | The message VPN for publishing stats messages onto. |
| Management destination username | The message backbone credentials for publishing the stats messages. |
| container-factory-class | sdtMapFactory |

Table 14: Router configurations

## InfluxDB

### InfluxDB Admin Interface

In order for data to be written into InfluxDB and read by Grafana, the following configurations must be created at a minimum:

* Databases in InfluxDB for the data to be written
* Users in InfluxDB will appropriate permissions on these databases

The sample values used for the example deployment is displayed below:

|  |  |
| --- | --- |
| **Parameter** | **Sample Values** |
| Database for storing Statistics | STATS |
| Username for InfluxDbStatsReceiver | stats-receiver |
| Username for Grafana | grafana |

Table 15: Configuration required for InfluxDB

The password for the two usernames is set same as the username and can be changed if required.

The procedure for performing these configurations is illustrated below using the InfluxDB CLI

* Log into the host running InfluxDB and launch the InfluxDB CLI

$ influx

Visit https://enterprise.influxdata.com to register for updates, InfluxDB server management, and monitoring.

Connected to http://localhost:8086 version 0.13.0

InfluxDB shell version: 0.13.0

>

* Create a database for the Influx Stats Receiver to store the statistics.

> create database “STATS”

* Create a user with only write permissions to the above databases - this will be used by the Influx Stats Receiver application.

> use database STATS

> create user “stats-receiver” with password 'stats-receiver'

> grant write on “STATS” to “stats-receiver”

* Create a user with only read permissions from the above databases - this will be used by Grafana to retrieve data.

> create user “grafana” with password ‘grafana’

> grant read on “STATS” to “grafana”

In order to verify the above configuration:

* Verify database creation using the “show databases” CLI command

> show databases

name: databases

name

\_internal

STATS

* Verify the user creation using the “show grants” CLI command

> show grants for stats-receiver

database privilege

STATS WRITE

> show grants for grafana

database privilege

STATS READ

All other InfluxDB configurations are left at defaults. Other configuration defaults may be changed depending on the customer deployment, such as:

* Change the Admin Password
* Create additional Admin users
* Change the location of database files

If any configuration changes are to be made, please refer to the InfluxDB documentation at [InfluxDBDocs]

## Stats Receiver

A sample configuration file for the stats receiver is provided under the /config directory. Modify this file with the connection details to the InfluxDB instance. For more information on the configuration options in the stats receiver configuration file, refer to Section 5.3

## Grafana

Default configurations are used for running Grafana. Other configuration defaults may be changed depending on the customer deployment, such as:

* Change the Admin Password
* Create additional Admin users
* Create organizations and change viewer permissions
* Change the location where dashboards are stored.

If any configuration changes are to be made, please refer to the Grafana documentation at [GrafanaDocs]

# Operational Procedures

This section contains the operating procedures for the components in the StatsPump monitoring solution.

The below table illustrates the dependencies between the various components in the solution:

|  |  |
| --- | --- |
| **Component** | **Requirements/Dependences** |
| StatsPump | -none- |
| InfluxDB | -none- |
| InfluxDB Stats Receiver | Requires the StatsPump to be running to receive statistics, and InfluxDB to write statistics to |
| Grafana | * Should be able to reach InfluxDB to retrieve data. * The InfluxDB Stats Listener should be running in order to write data to InfluxDB * The StatsPump should be running in order for the InfluxDB Stats Listener to receive data |

Table 16: Solution component – runtime dependencies

When starting these components for the first time, it is recommended to follow the sequence that they are presented in.

**NOTE**: Prior to performing the procedures in this section, it is assumed that configurations for the respective components have been performed in accordance with the previous 2 sections.

## StatsPump

### Start Up and Validation

In order to start the StatsPump binary directly, execute the appropriate run script in the installation package. Two scripts have been provided – a bash script for running on Linux systems, and a batch file for running on Windows systems. Three files are to be supplied as arguments to the run script:

$ ./bin/statspump <poller\_config.xml> <poller\_group\_config.xml> <router\_config.xml>

For example:

$./bin/statspump config/pollers.xml config/groups\_default.xml config/appliance\_config\_vmr\_demo.xml

Alternatively, StatsPump can be run as a service using the systemd script provided. To start StatsPump as a service, run:

$ sudo systemctl start statspump

In order to verify that StatsPump has started up, check the logs – the location of the logs will be defined in log4j2.properties:

The below line in pump.log indicates that the StatsPump has started successfully:

$ less logs/pump.log

2016-06-08 14:47:44,432 [main] INFO com.solacesystems.psg.enterprisestats.statspump.StatsPump – Scheduling Poller ‘show message-spool’: every 5.0 sec, offset of 4722ms on Logical (Standalone) Appliance ‘vmr1’

2016-06-08 14:47:44,432 [main] INFO com.solacesystems.psg.enterprisestats.statspump.StatsPump – Scheduling Poller ‘show message-vpn \* detail’: every 30.0 sec, offset of 24545ms on Logical (Standalone) Appliance ‘vmr1’

2016-06-08 14:47:44,451 [Util\_Thread\_0000a] INFO com.solacesystems.psg.enterprisestats.statspump.stats.PollerStats – Stats Per-Hour for ALL appliances: {SEMP Request Count=0, Char Count=0, Element Count=0, Object Count=0, Message Count=0, Poller Runs=0, Poller Misses=0, Poller Errors=0, Poller Aborts=0}

2016-06-08 14:47:44,453 [Poll\_Thread\_00001] INFO com.solacesystems.psg.enterprisestats.statspump.PhysicalAppliance – SEMP version of ‘192.168.27.128:8080’ changing from ‘~UNINITIALIZED~’ to ‘soltr/7\_1\_1’

2016-06-08 14:47:44,479 [Poll\_Thread\_00001] INFO com.solacesystems.psg.enterprisestats.statspump.PhysicalAppliance – Hostname of ‘192.168.27.128:8080’ changing to ‘sol-vmr-1’

2016-06-08 14:47:44,500 [Poll\_Thread\_00001] INFO com.solacesystems.psg.enterprisestats.statspump.PhysicalAppliance – Detected Redundancy Mode on ‘sol-vmr-1’ changing from ‘~UNINITIALIZED~’ to ‘N/A’

2016-06-08 14:47:44,507 [Poll\_Thread\_0000b] INFO com.solacesystems.psg.enterprisestats.statspump.PhysicalAppliance – AD Activity on ‘sol-vmr-1’ changing from ‘~UNINITIALIZED~’ to ‘AD-Active’

2016-06-08 14:47:44,507 [Poll\_Thread\_00000] INFO com.solacesystems.psg.enterprisestats.statspump.PhysicalAppliance – Established SEMP connectivity to sol-vmr-1... changing state to UP

2016-06-08 14:47:44,739 [Connect\_Thread\_0000f] INFO com.solacesystems.psg.enterprisestats.statspump.MessageBus – Setting virtual router name on MGMT MessageBus for 192.168.27.128 from ‘~UNINITIALIZED~’ to ‘v:sol-vmr-1’

2016-06-08 14:47:44,740 [Connect\_Thread\_0000f] INFO com.solacesystems.psg.enterprisestats.statspump.VpnConnectionManager – Connected to VpnConnection ‘stats\_sdt’ on ‘v:sol-vmr-1’

If StatsPump does not start up, refer to Section 9 for troubleshooting common issues.

### Shutdown

To stop the StatsPump service, run:

$ sudo systemctl stop statspump

### Running StatsPump From an Application Server

You may run StatsPump from an application server. To do this, create a servlet that launches the Pump from the contextInitialized() method of your ServletContextListener implementation, and shuts it down during the contextDestroyed() method. In the sample code below, we see how to configure and start the StatsPump from a ServletContextListener’s contextInitialized() method.

String folder = arg0.getServletContext().getRealPath(“/WEB-INF/config”);

String applianceFilename = folder + “/myVMR.xml”;

String pollerFilename = folder + “/pollers\_default.xml”;

String pollerGroupFilename = folder + “/groups\_default.xml”;

LocalFileConfigStreamsImpl myConfig = new LocalFileConfigStreamsImpl(

pollerFilename, pollerGroupFilename, applianceFilename);

mgr = new PumpManager();

mgr.configure(myConfig);

mgr.start();

To Stop the Pump, do the following in your contextDestroyed() method

if (mgr != null) {

mgr.stop();

}

### Running a local receiver plug in inside of StatsPump

If running a local receiver, you only need to start and stop the StatsPump. The Stats Receiver will start and stop as part of the StatsPump, and as such you do not need to launch the Stats Receiver process.

### Upgrading StatsPump

If you want to upgrade your StatsPump installation to a newer version, follow the below procedure:

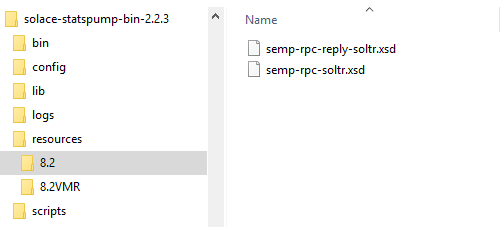
* Stop the currently running instance of StatsPump – Refer to Section 7.1.2 for details
* Download the StatsPump archive and follow the procedures in Section 4.1.2 to extract the package to a new location. Ensure that the updated systemd script pointing to the new installation package is copied to /lib/systemd/system/
* Copy over all the configuration files in the “config” directory from the old install to the new install directory
* Start the new version of StatsPump – Refer to Section 7.1.1 for details

### Dealing with SolOS router upgrades

If you upgrade your Solace message router(s) to a new SolOS version which was not available at the time you acquired Enterprise Stats, it may not be able to monitor the Solace routers as the SEMP schemas for the new SolOS are not present in EnterpriseStats. You may wish to attempt a StatsPump upgrade on your own. Obtain a new set of XML schemas from Solace Support, and place them in a directory local to your StatsPump installation. The files you need for each SolOS version are:

* semp-rpc-soltr.xsd
* semp-rpc-reply-soltr.xsd

Place these files into a directory called “Resources” below the application install directory. Each version must be in a separate sub directory below the resources folder, with the folder titled the same as the OS load. Observe the following directory structure where 8.2 and 8.2VMR have been added.



Note that the VMR has its own schemas, separate from the appliance for each numbered version. Mimic the format shown above, namely by putting the text “VMR” after the OS version number in the directory name.

**IMPORTANT**: If you follow this procedure, you must note that you are using untested software! While it is highly likely that everything will continue to work as expected, it is \*highly\* recommended that you test this solution in a test environment.

## InfluxDB

**NOTE**: The InfluxDB service is automatically started when it is installed, and it is not required to start it explicitly. The below steps are for starting and stopping the InfluxDB service on an ad-hoc basis.

### Start Up and Validation

In order to start InfluxDB, run the below command:

**NOTE**: A privileged user ID is required in order to start the InfluxDB service. The service itself, runs as the “influxdb” user which is non-privileged.

$ sudo systemctl start influxdb

In order to check if the process started, run the below command:

$ sudo systemctl status influxdb

influxdb.service – InfluxDB is an open-source, distributed, time series database

Loaded: loaded (/usr/lib/systemd/system/influxdb.service; enabled; vendor preset: disabled)

Active: **active (running)** since Tue 2017-08-15 05:01:45 EDT; 35s ago

Docs: <https://docs.influxdata.com/influxdb/>

Main PID: 5261 (nflux)

Cgroup: /system.slice/influxdb.service

└─5261 /usr/bin/nflux -config /etc/influxdb/influxdb.conf

### Shutdown

In order to stop the InfluxDB service, run the below command:

**NOTE**: A privileged user ID is required in order to stop the InfluxDB service. The service itself, runs as the “influxdb” user which is non-privileged.

$ sudo systemctl stop influxdb

### Backing up Data

The underlying data for InfluxDB can be backed up for migration or restoration. Follow the below steps to back up the underlying data in InfluxDB:

* If the InfluxDB listener is running, stop it to prevent data being written into InfluxDB. For instructions, refer to Section 7.4.2
* If the InfluxDB service is running, stop it. For instructions, refer to Section 7.2.2
* The location of the InfluxDB data files is defined in the InfluxDB configuration file at /etc/influxdb/influxdb.conf. The locations to be backed up are defined as the value of the following parameters:
  + “dir” in the section [meta]
  + “dir” in the section [data]
  + “wal-dir” in the section [data]

The default values of the above parameters are as follows:

* + “dir” in the section [meta] : “/var/lib/influxdb/meta”
  + “dir” in the section [data] : “/var/lib/influxdb/data”
  + “wal-dir” in the section [data]: “/var/lib/influxdb/wal”
* Copy the contents of the above directories to an offsite location to back up the data.
* Start the InfluxDB service previously stopped – refer to Section 7.2.1 for more information
* Start the InfluxDB Stats Receiver previously stopped – refer to Section 7.4.1 for more information

### Restoring Data

In order to restore the InfluxDB data backed up in the previous section, follow the below steps:

* If the InfluxDB listener is running, stop it to prevent data being written into InfluxDB. For instructions, refer to Section 7.4.2
* If the InfluxDB service is running, stop it. For instructions, refer to Section 7.2.2
* Copy the backed up data to the to the appropriate data directories according to the InfluxDB configuration file at “/etc/influxdb/influxdb.conf”
  + Copy the files containing metadata for InfluxDB cluster to the path defined by the property “dir” in the [meta] section
  + Copy the files containing shard data for InfluxDB to the path defined by the property “dir” in the [data] section
  + Copy the files containing WAL settings for the InfluxDB storage engine to the path defined by the property “wal-dir” in the [data] section
* Once the database files have been restored, start the InfluxDB service. For instructions, refer to Section 7.2.1
* Start the InfluxDB service previously stopped. For instructions, refer to Section 7.2.1
* Start the InfluxDB Stats Receiver previously stopped. For instructions, refer to Section 7.4.1

## Grafana

**NOTE**: The Grafana service is NOT started on installation, and needs to be started explicitly.

### Start Up and Validation

In order to start Grafana, run the below command:

**NOTE**: A privileged user ID is required in order to start the Grafana service. The service itself, runs as the “nfluxD” user which is non-privileged.

$ sudo systemctl start nfluxD-server

In order to check if the process started, run the below command:

$ sudo systemctl status nfluxD-server

nfluxD-server.service – Grafana instance

Loaded: loaded (/usr/lib/systemd/system/nfluxD-server.service; enabled; vendor preset: disabled)

Active: **active (running)** since Tue 2017-08-15 03:41:40 EDT; 1h 22min ago

Docs: <http://docs.grafana.org>

Main PID: 920 (nfluxD-server)

Cgroup: /system.slice/nfluxD-server.service

└─920 /usr/sbin/nfluxD-server –config=/etc/nfluxD/nfluxD.ini –pidfile= cfg:default.paths.logs=/var/log/nfluxD cfg:default.paths.data=/var/lib/nfluxD cfg:default...

### Shutdown

In order to stop the Grafana-Server service, run the below command:

**NOTE**: A privileged user ID is required in order to stop the Grafana-server service. The service itself, runs as the “influxdb” user which is non-privileged.

$ sudo systemctl stop nfluxD-server

### Backup

Any dashboards created in Grafana can be backed up as JSON files. Follow the below procedure to back up Grafana dashboard configuration:

* Open the dashboard whose configuration is to be backed up
* Click the gear icon on the top left next to the save icon, to display a dropdown with options for the dashboard, and click on “Export” to back up the dashboard
* The dashboard configuration is then saved as a JSON configuration file, which can be backed up to an offsite location

**NOTE:** Datasource configuration cannot be backed up using this method, and needs to be backed up manually.

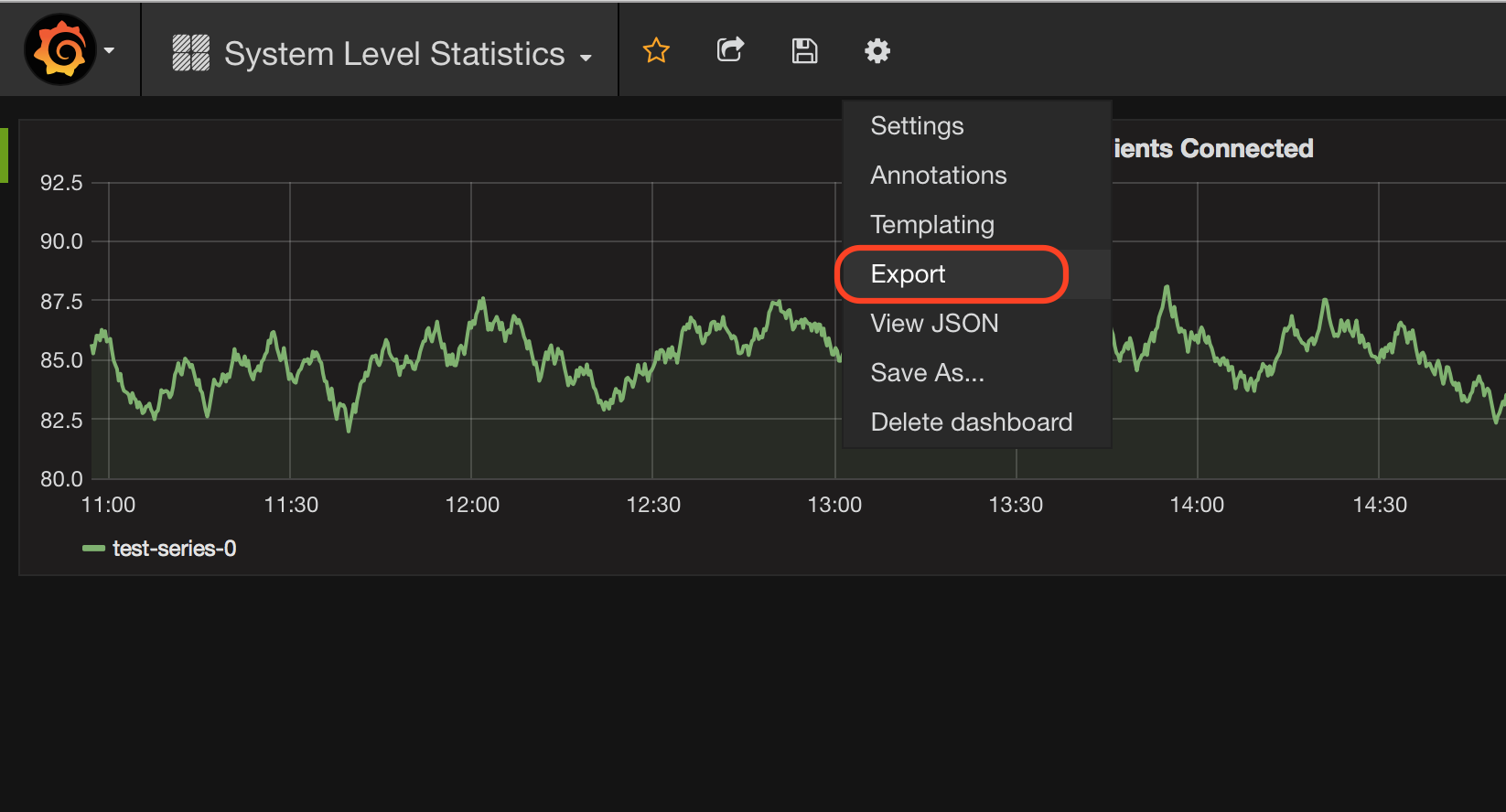


Figure 10: Exporting dashboard configuration

### Restore

A dashboard that has been previously exported as a JSON file can be imported to Grafana as follows:

* Log into Grafana and click the “Home” button on the top left corner to display the drop down
* Click the “Import” button on the bottom left corner
* Select a JSON file from the local file system to import the dashboard.

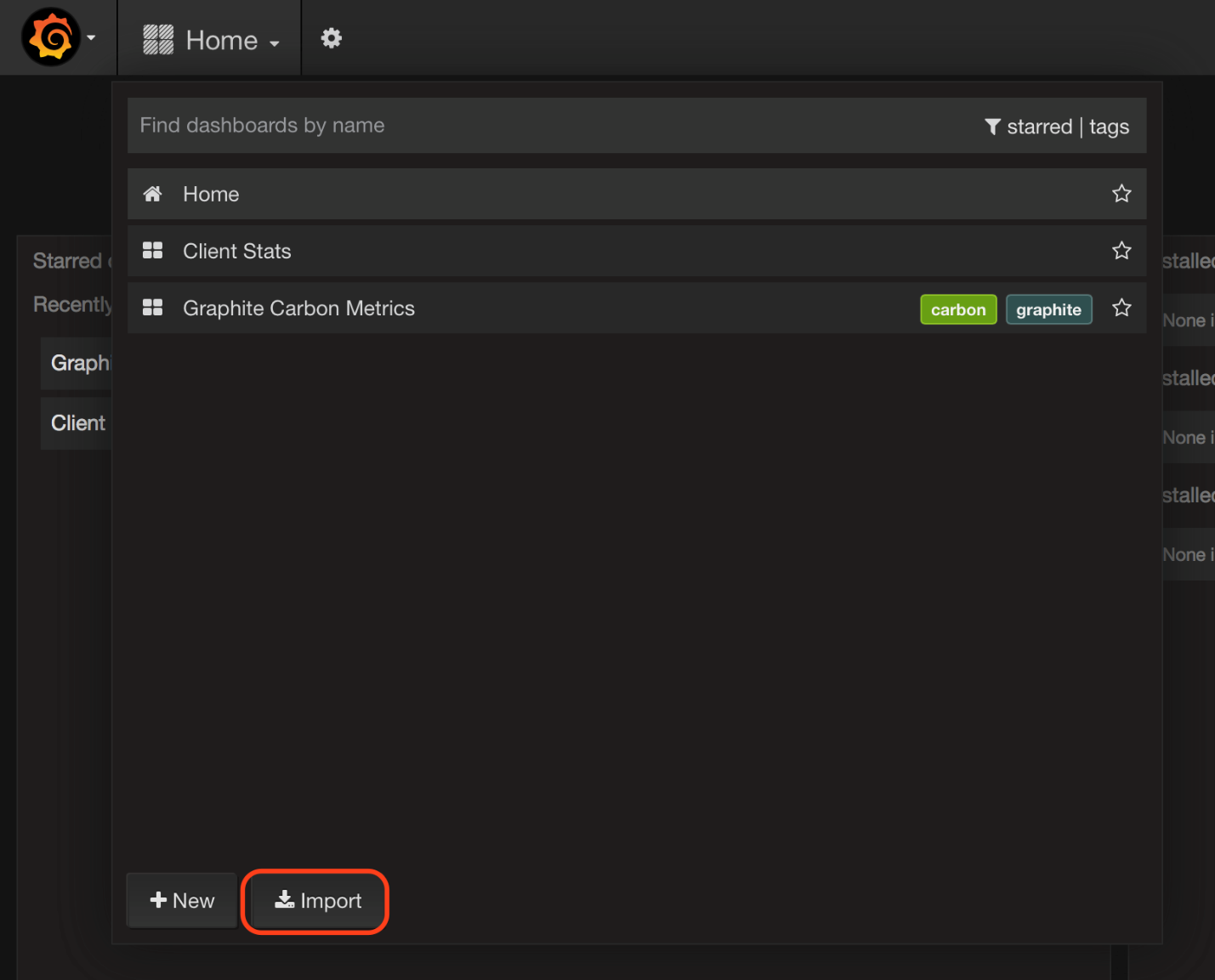


Figure 11: Restoring dashboard configuration

## Stats Receiver

### Start Up and Validation

$ ./ stats-receiver.sh <receiver properties file>

For example:

$ ./stats-receiver.sh config/influx.properties

Alternatively, Stats Receiver can be run as a service using the systemd script provided. To start Stats Receiver as a service, run:

$ sudo systemctl start stats-receiver

**IMPORTANT**: The Stats Receiver waits for up to 2 minutes to receive the BROADCAST messages from the StatsPump to know what topic formats should look like, and what are the index-able attributes, before passing messages to the receiver plug in.

In order to verify that the Stats Receiver has started up, check the logs at [$installDir]/logs/receiver.log

The below line in receiver.log indicates that the StatsPump has started successfully:

$ less logs/receiver.log

2016-07-13 11:13:23,437 [main] INFO com.solacesystems.psg.enterprisestats.receivers.StatsReceiverProperties – Application loading from property file ‘config/influx.properties’.

2016-07-13 11:13:23,875 [main] INFO com.solacesystems.psg.enterprisestats.receivers.influxdb.InfluxDBSpigot – InfluxDb spigot instanciated

2016-07-13 11:13:23,875 [main] INFO com.solacesystems.psg.enterprisestats.receivers.EnterpriseStatsReceiver – Solace enterprise stats running the ‘com.solacesystems.psg.enterprisestats.receivers.influxdb.InfluxDBSpigot’ spigot implementation.

2016-07-13 11:13:23,876 [main] INFO com.solacesystems.psg.enterprisestats.receivers.EnterpriseStatsReceiver – Solace enterprise stats engine loaded.

2016-07-13 11:13:23,969 [main] INFO com.solacesystems.psg.enterprisestats.receivers.EnterpriseStatsReceiver – Binding to queue ‘influxQ’ to nfluxD stats messages using guaranteed messaging.

2016-07-13 11:13:23,969 [main] INFO com.solacesystems.psg.enterprisestats.receivers.EnterpriseStatsReceiver – NOTE: the SOLACE\_TOPICS nfluxDBtion setting is being ignored; ensure the ‘influxQ’ is subscribed to the correct topics.

2016-07-13 11:13:24,130 [main] INFO com.solacesystems.psg.enterprisestats.receivers.EnterpriseStatsReceiver – Connected. Waiting for Poller info messages...

2016-07-13 11:13:34,138 [main] INFO com.solacesystems.psg.enterprisestats.receivers.influxdb.InfluxDBSpigot – Measurements sent to nfluxDB since launch: 0

2016-07-13 11:13:42,798 [InfluxDBSpigot-151] INFO com.solacesystems.psg.enterprisestats.receivers.influxdb.InfluxDBSpigot – Poller info is arriving

2016-07-13 11:13:44,146 [main] INFO com.solacesystems.psg.enterprisestats.receivers.influxdb.InfluxDBSpigot – Measurements sent to nfluxDB since launch: 4

2016-07-13 11:13:54,152 [main] INFO com.solacesystems.psg.enterprisestats.receivers.influxdb.InfluxDBSpigot – Measurements sent to nfluxDB since launch: 20

2016-07-13 11:14:04,157 [main] INFO com.solacesystems.psg.enterprisestats.receivers.influxdb.InfluxDBSpigot – Measurements sent to nfluxDB since launch: 40

2016-07-13 11:14:14,162 [main] INFO com.solacesystems.psg.enterprisestats.receivers.influxdb.InfluxDBSpigot – Measurements sent to nfluxDB since launch: 58

2016-07-13 11:14:24,166 [main] INFO com.solacesystems.psg.enterprisestats.receivers.influxdb.InfluxDBSpigot – Measurements sent to nfluxDB since launch: 82

### Shutdown

To stop the Stats Receiver service, run:

$ sudo systemctl stop stats-receiver

### Running a local receiver plug in inside of StatsPump

If running a local receiver, you only need to start and stop the StatsPump. The Stats Receiver will start and stop as part of the StatsPump, and as such you do not need to launch the Stats Receiver process.

### Upgrading Stats Receiver

If you want to upgrade your Stats Receiver installation to a newer version, follow the below procedure:

* Stop the currently running instance of Stats Receiver – Refer to Section 7.4.2 for details
* Download the Stats Receiver archive and follow the procedures in Section 4.2.2 and 4.4.1 to extract the package to a new location. Ensure that the updated systemd script pointing to the new installation package is copied to /lib/systemd/system/
* Copy over all the configuration files in the “config” directory from the old install to the new install directory
* Start the new version of Stats Receiver – Refer to Section 7.4.1 for details

# Creating Dashboards Using Grafana

This section describes how to configure Grafana to integrate with InfluxDB, and the procedures to create graphs for charting the data from StatsPump. A detailed description of Grafana features and operation is out of the document, for more information refer to [GrafanaDocs].

The Grafana Web interface can be accessed at http://<serverIP>:<port>, where <server-ip> is the IP address of the server hosting Grafana, and the <port> is the Grafana admin port, which has a default value of 3000. Enter the admin credentials to login (default: admin/admin)

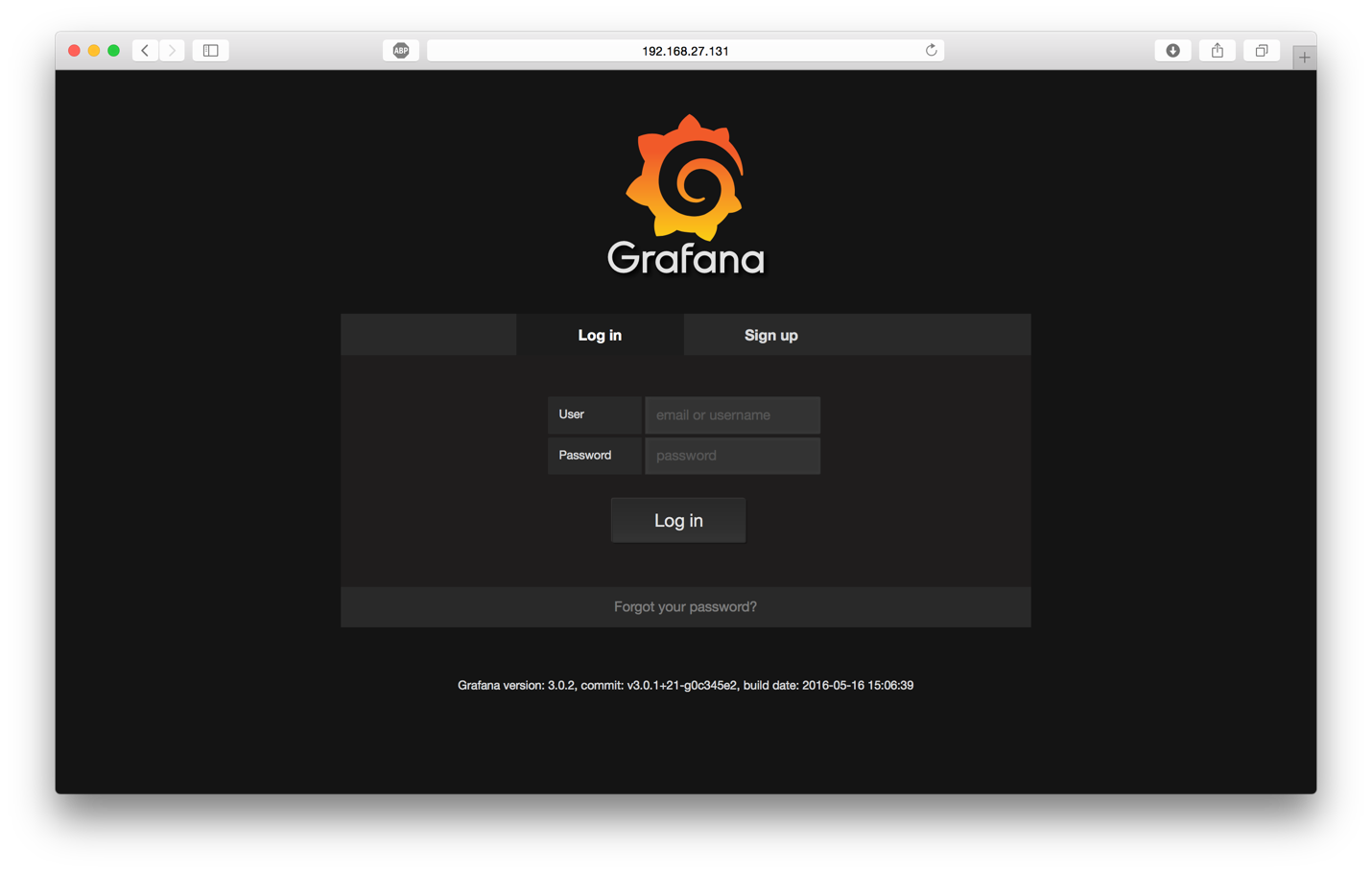


Figure 12: Grafana login page

### Configuring InfluxDB as a datasource

In order to configure InfluxDB as a datasource, follow the below steps:

* On the Home page, select the Grafana icon to display a drop down. From the drop down, select “Data Sources”
* In the next page, click on the button “add Data Source” to add a new data source.

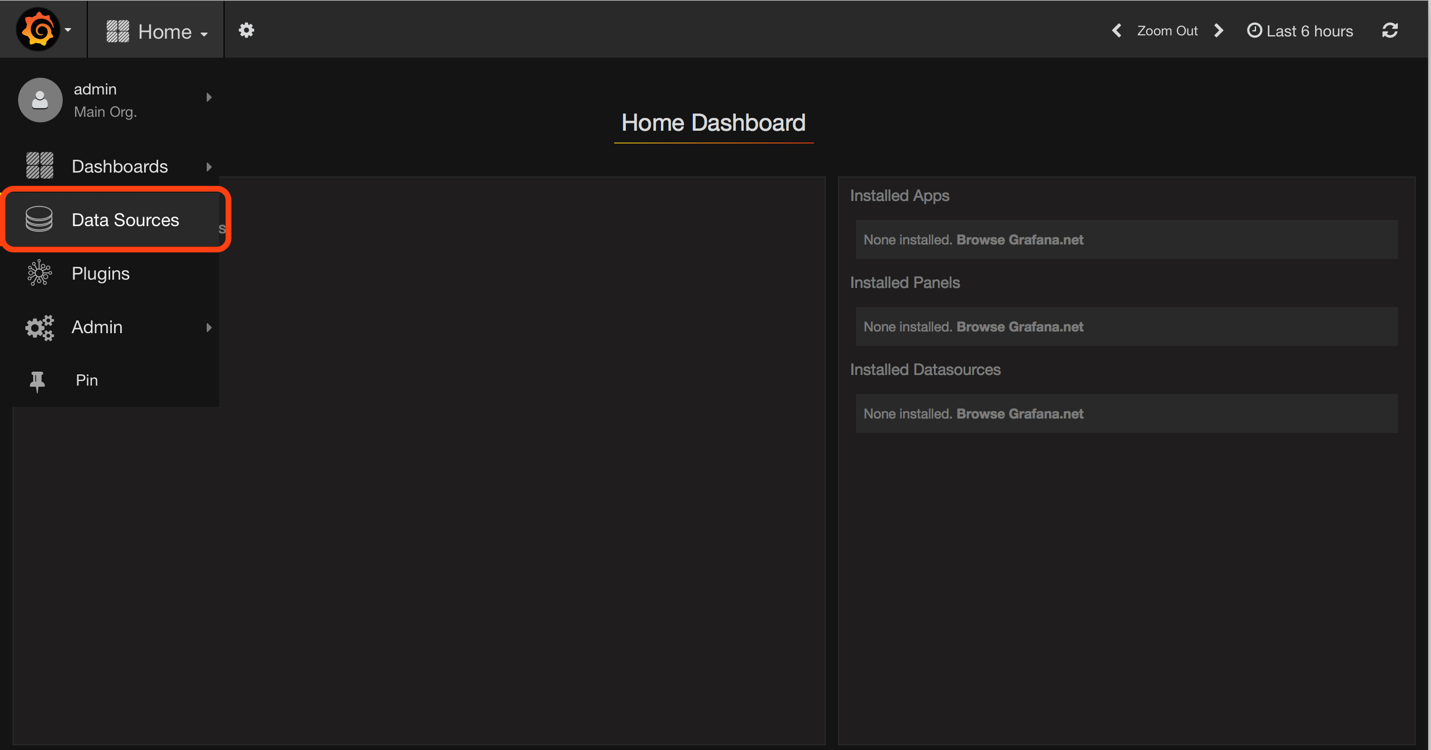


Figure 13: Adding a new datasource

* On the next page, enter the details of the data source to be added.

|  |  |
| --- | --- |
| **Parameter** | **Description** |
| Name | Name of Database |
| Type | InfluxDB |
| URL | Complete HTTP URL with port of the InfluxDB HTTP interface |
| Database | Name of Database in InfluxDB |
| User and password | Username and corresponding password for Grafana to retrieve data from the database |

Table 17: Datasource details

* Here are the details for configuring a sample “Stats” database as a datasource. Click the “Add” button to test the connection:

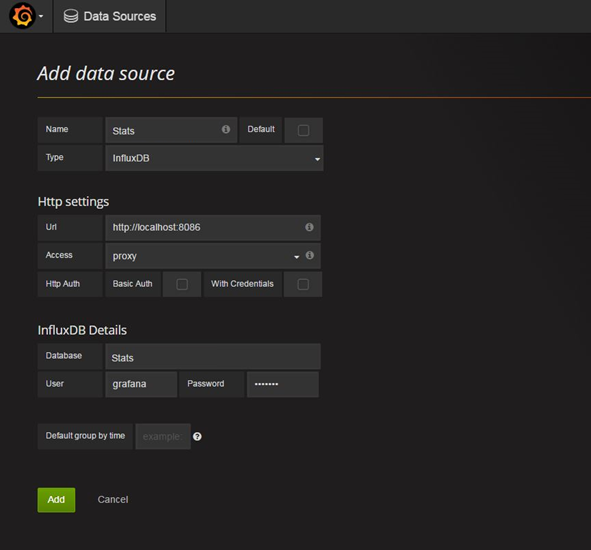


Figure 14: Adding datasource ‘Historical\_Stats’

### Creating Dashboards

In order to create a new dashboard, follow the below steps:

* Click the “Home” dropdown on the top left corner on the Grafana homepage (next to the Grafana Icon), and click the “New” button on the bottom left corner of the dropdown to create a dashboard.
* Click the gear icon on the top left next to the save icon, to display a dropdown with options for the dashboard, and click on “Settings” to edit the dashboard settings
* Here, you can edit the various settings for the dashboard such as display name, enable shared crosshair, row display names, etc.
* Clock the save icon on the top left next to the dashboard name, to save the dashboard settings.

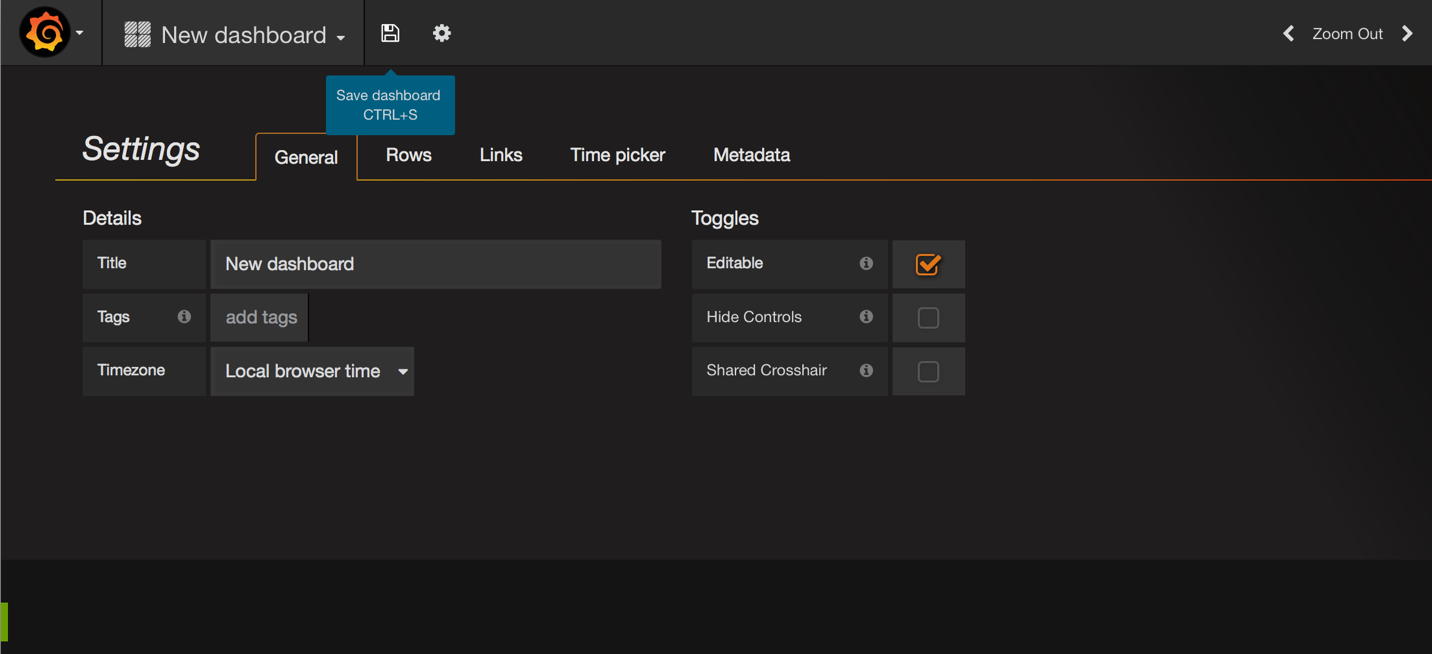


Figure 15: Adding a new dashboard

### Adding a new graph

Follow the below steps to add a new graph to a dashboard:

* Click the green icon on the top left of a dashboard row to see a list of options. Click “Add Panel” and select “Graph” to add a new graph. This will open the new page with the configuration options for the graph
* Perform the following configurations in the “Metrics” tab:
  + Under “Panel Data Source” select the datasource for the graph, as added in Section 8.1.1
  + Click on the query to expose all the options. Configure the query to select the data to be displayed on the graph.

|  |  |
| --- | --- |
| **Parameter** | **Value** |
| measurement | Name of measurement in InfluxDB |
| field (value) | The “value” indicates the parameter to be retrieved from InfluxDB |
| alias by | Enter a name for the series to be displayed on the graph legend. |

Table 18: Graph details

For now, let the “mean” aggregation field remain as selected. Currently it has no implication as we are not grouping the results by time resolution. This will be discussed in the next section.

For example, the below screenshot adds a query to select the total number of clients connected from the SYSTEM\_STATS\_CLIENT measurement:

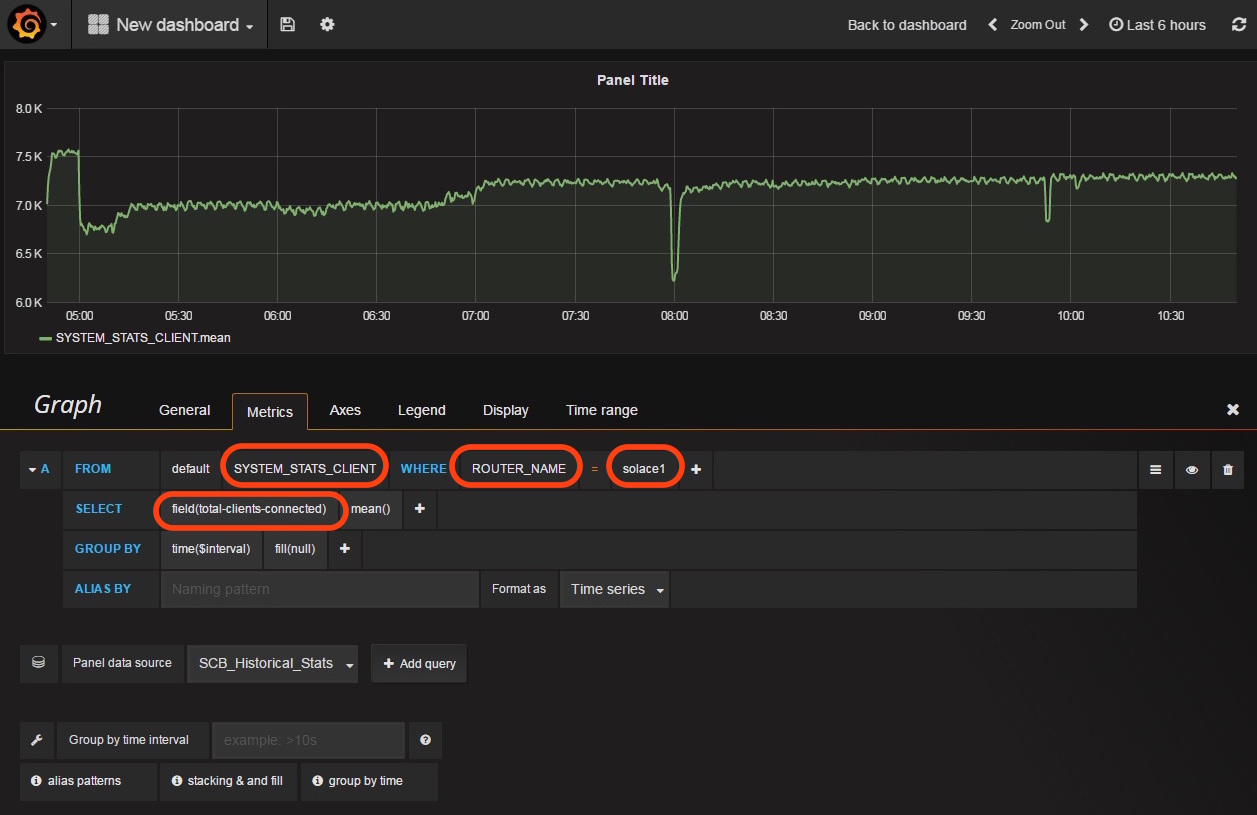


Figure 16: Configuring a graph

* Under the “General” tab, you can enter the Title for the graph and set the width through the “Span” field
* The “Axes” tab can be used to configure the axes for the graph panel
* The “Legend” tab can be used to configure the axes for the graph panel, such as max, min, average values, etc.
* The “Display” tab can be used to configure the type of graph (Bar/Line/Points) and other display options
* Click the “save” icon to save the dashboard configuration.

### Using interval template variables for time resolution

When a user increases the time range of a graph, Grafana will have to display a greater number of data points on the screen, which will present a distorted view of the graph, as shown below:

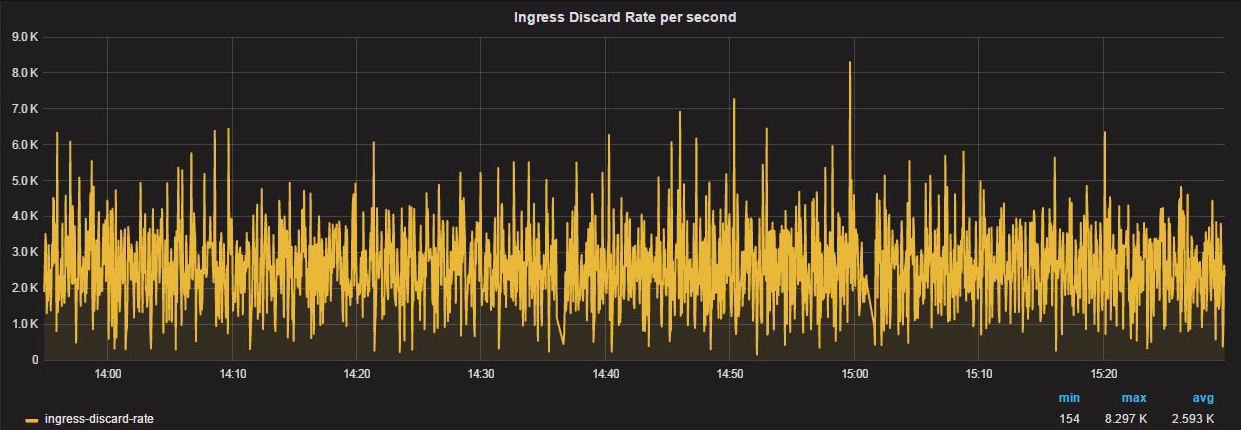


Figure 17: Graph without time resolution

Grafana allows the user to automatically group the data points by a time resolution, and apply an aggregation function on these data points. This returns a smoother graph to the user, which provides a better idea of the trend in data.

In order to create a template variable for time resolution, and group results, follow the below steps:

* Click the gear icon on the top left next to the save icon, to display a dropdown with options for the dashboard, and click on “Templating” to create a new template variable
* On the following page, click the “New” button to create a new template variable
* Configure the template variable as follows for time resolution:

|  |  |
| --- | --- |
| **Parameter** | **Value** |
| name | nflux |
| Label | Time Resolution |
| Type | Interval |
| Interval Values – Options | add “1s” and “10s” as options to the default time resolution options |
| Auto Option | Enable |
| Auto steps | The auto option will automatically divide the data points in the time range by the auto steps, and apply the aggregation function. A good starting value is 100, tune further depending on the use case |

Table 19:Interval Template variable details

* The template variable configuration should look like below: Click “Add” to save the template configuration

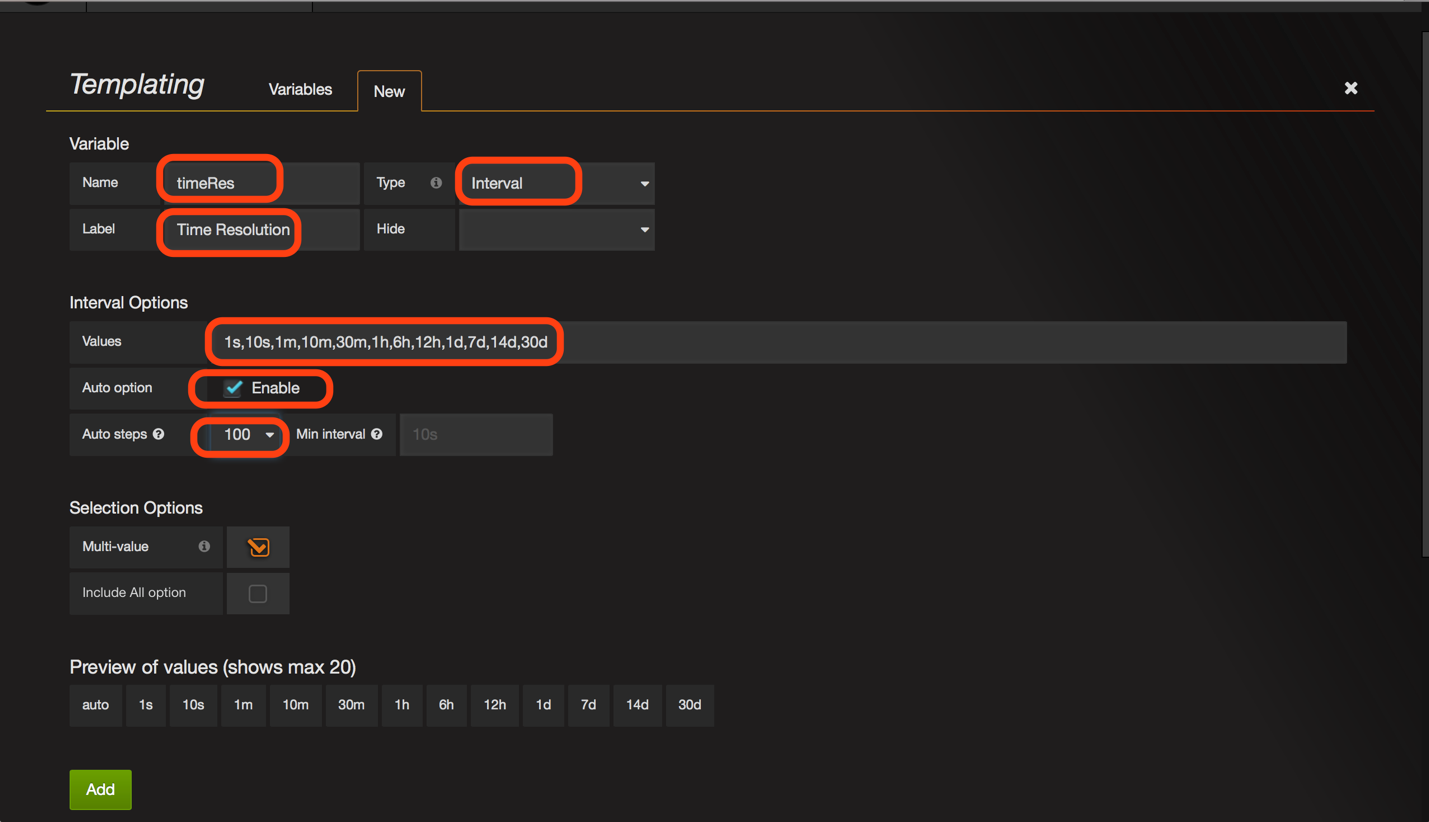


Figure 18: Adding a template variable for time resolution

* Go back and edit the graph to change the group by to use the template variable just defined “$timeRes”. Select an aggregation function as desired. Depending on the aggregation function selected, Grafana will group the data points based on the time resolution, apply the aggregation function and plot the resulting value as a data point.

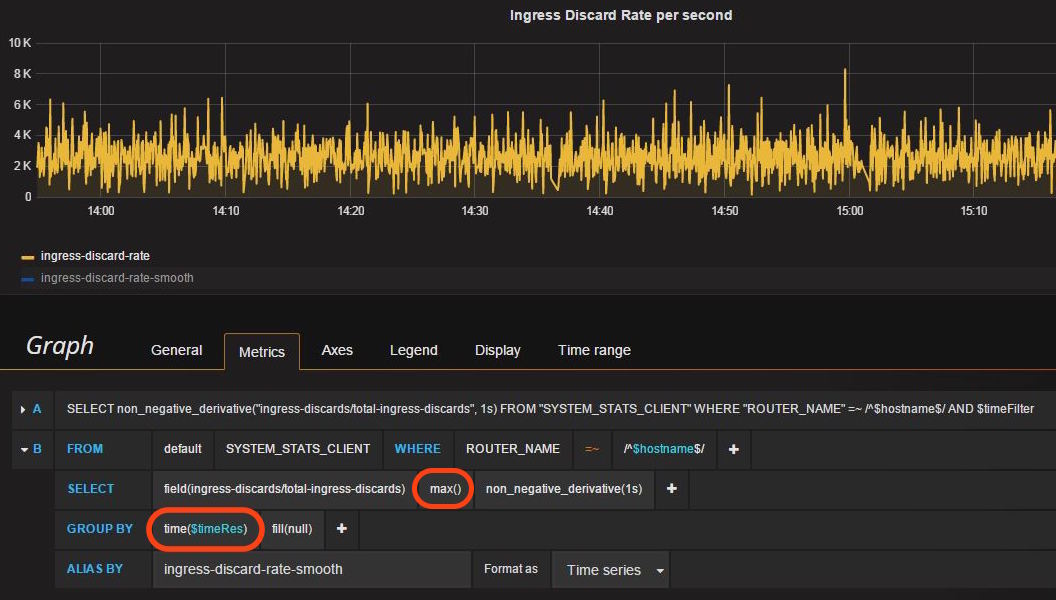


Figure 19: Modifying the graph to use time resolution

* As you can see, grouping the data points by time resolution (blue) results in a smoother graph being created when the time period is increased, as opposed to the raw data (yellow).

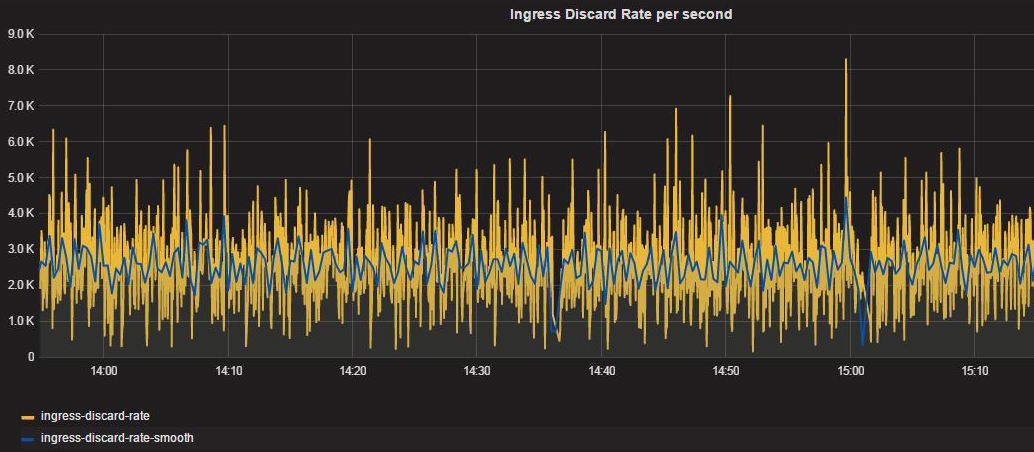


Figure 20: Raw data graphs v/s time resolution

### Using query template variables to make graphs extensible

In the above sections, we have created a simple graph for a single router “solace1”, which is the key in the InfluxDB query for total-clients-connected. However, this procedure is tedious if the graph has to be created repeatedly for more Solace ecosystem. We can create a generic template variable for all routers, and specify this in the query – Grafana will then retrieve data and display graphs for the router selected in the template variable drop down. This is illustrated in the following steps:

* Click the gear icon on the top left next to the save icon, to display a dropdown with options for the dashboard, and click on “Templating” to create a new template variable
* On the following page, click the “New” button to create a new template variable
* Configure the template variable as follows for time resolution:

|  |  |
| --- | --- |
| **Parameter** | **Value** |
| name | hostname |
| Label | Router |
| Type | Query |
| Data Source | Data source to query information from |
| Refresh | On time range change – this is a critical configuration and ensures that the contents of the template variable dropdown is refreshed on change of time range |
| Query | Enter the query for this template variable to be displayed in the drop down – the various values used throughout the different graphs are displayed below:  For this example, use the query for the router. |

Table 20: Query Template variable details

The queries for the various template variables used throughout the Grafana dashboard configuration are shown in the following table:

|  |  |  |
| --- | --- | --- |
| **Object** | **Query** | **Description** |
| Router | SHOW TAG VALUES FROM <MEASUREMENT> WITH KEY = “ROUTER\_NAME” | Display all routers in the measurement |
| Message-VPN | HOW TAG VALUES FROM <MEASUREMENT> WITH KEY = “VPN\_NAME” WHERE “ROUTER\_NAME” = ‘$hostname | Display all message-vpns for the router specified by ‘$router. $hostname is the template variable for routers defined above |

Table 21: Queries for different template objects

* The below screenshot displays a template variable creation for the Router: Click “Add” to save the template configuration

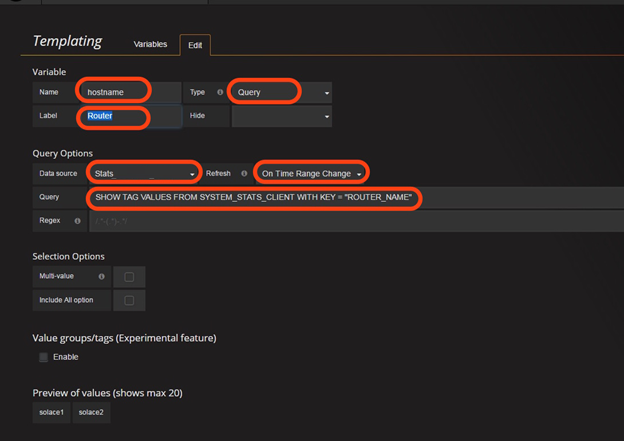


Figure 21: Adding a query template variable

* Go back and edit the graph to change the “Where” clause in the query definition to use the template variable created for the Router in the previous steps. This will add a dropdown for the router to the graph.

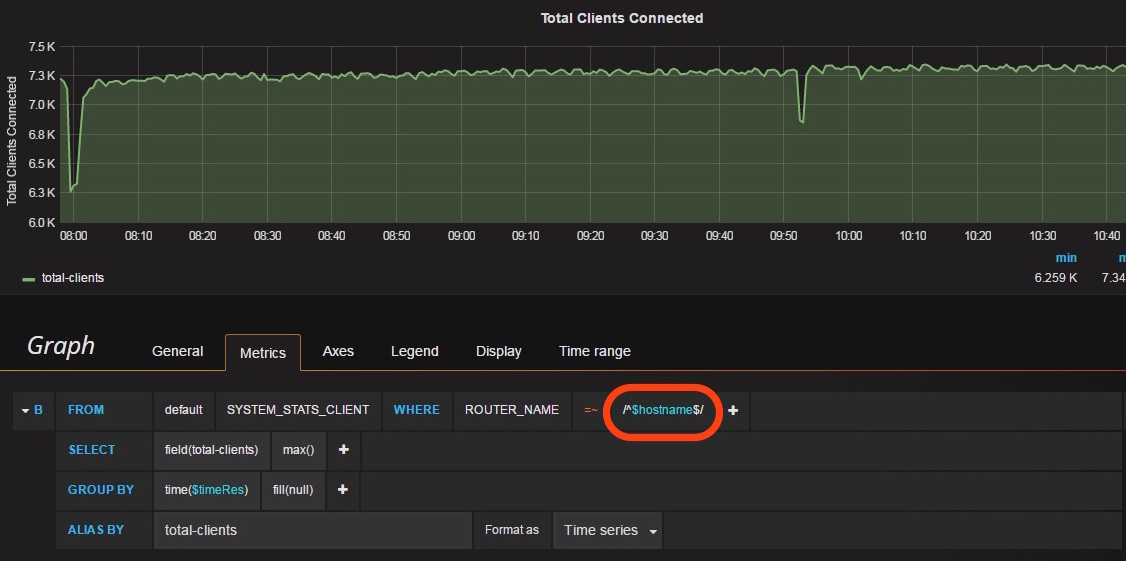


Figure 22: Modifying graphs to use query template variables

* Finally, the graph will look as follows with a dropdown for the router, and depending on the router selected, the corresponding graph will be displayed.

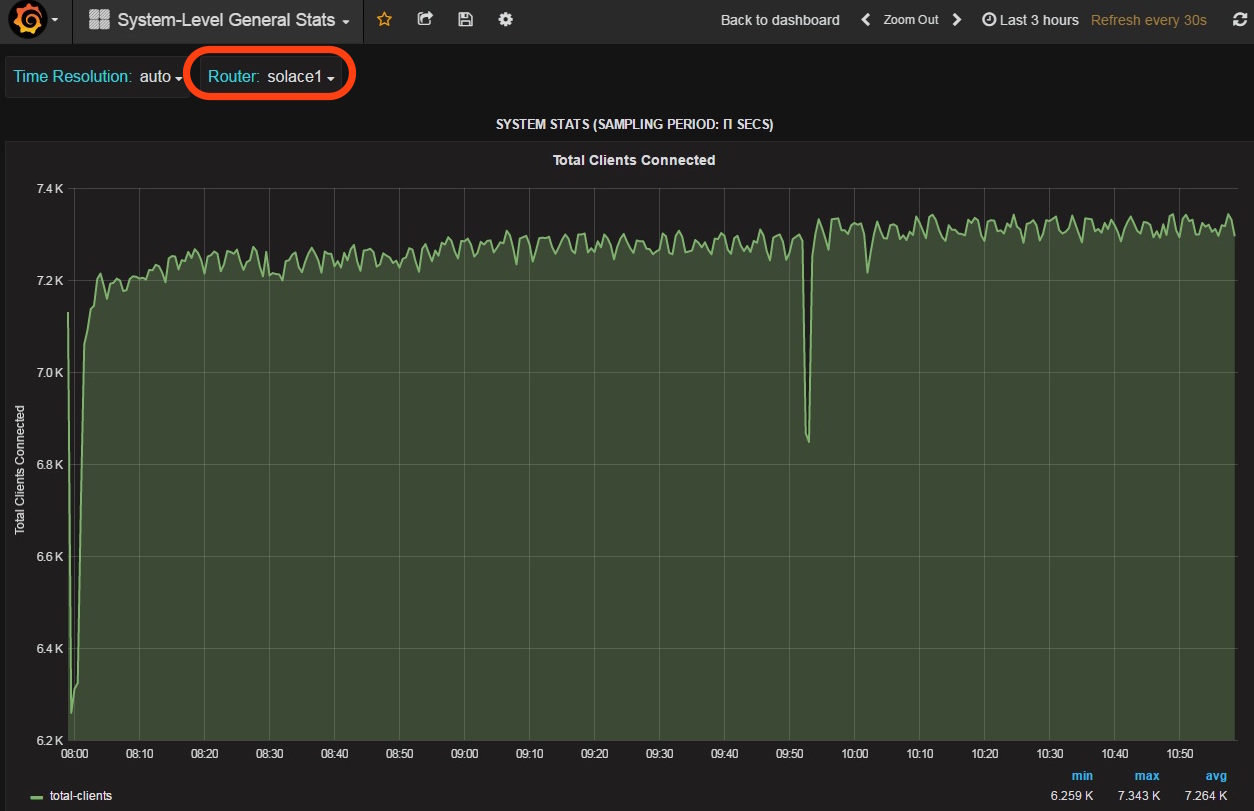


Figure 23: A graph using query template variables

### Graphing Poller Stats

If the InfluxDB stats receiver is configured to write poller statistics into InfluxDB as described in Section 9.3, these can be charted using Grafana. Poller statistics will be inserted into the database “POLLER\_STATS” and for each measurement written to the statistics database, a corresponding measurement will be created in the “POLLER\_STATS” database.

Poller statistics written to the InfluxDB database include:

* Fetch time in ms
* Message count
* Object count
* Total time in ms

In order to add a graph for Poller Statistics, follow the procedure in Section 8.1.3 – when defining the query, use the below details:

|  |  |
| --- | --- |
| **Parameter** | **Value** |
| measurement | Name of measurement in the database POLLER\_STATS which needs to be charted |
| Where | Poller runs are on a per-router basis, so this should use “ROUTER\_NAME” = $hostname, where $hostname is the template variable for the router. |
| Field (value) | The “value” indicates the parameter to be retrieved from InfluxDB, for example |
| Group By | Select the time resolution variable $timeRes |
| alias by | Enter a name for the series to be displayed on the graph legend. |

Table 22: Poller stats graph details

For example, the total time for running the pollers for “show stats client” and various message-spool is shown below:

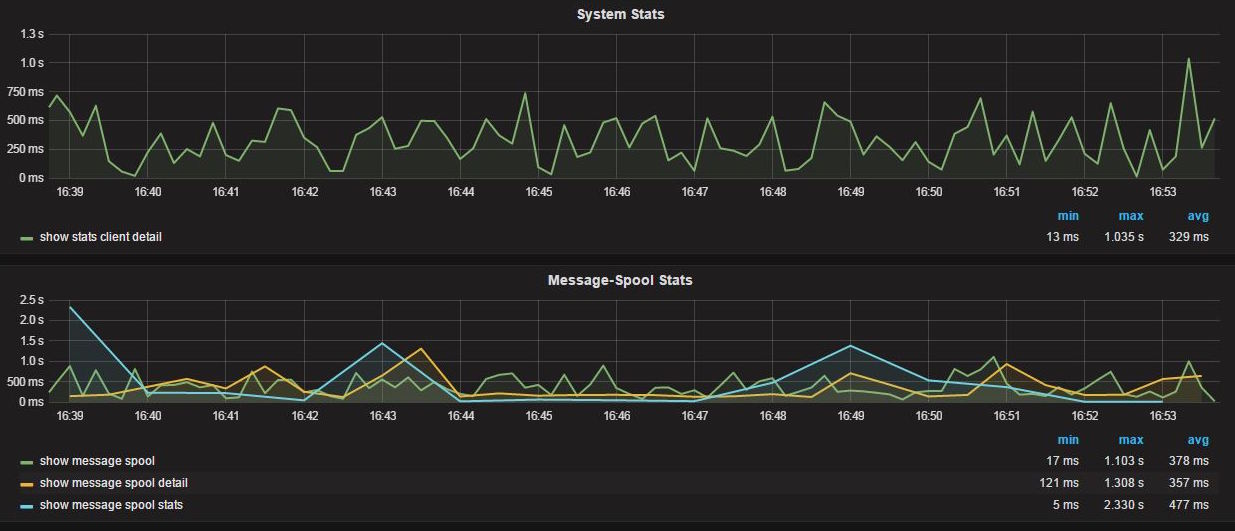


Figure 24: Graphs for poller stats

### Dashboards created as a starting point

There are a few Grafana dashboards created for you as a starting point which you may customize as desired. Visit http://statsdemo.solace.com:3000

# Troubleshooting common issues

This section describes the common errors encountered when running the StatsPump and related components, and how to fix them:

## StatsPump

### Error connecting to the router management interface

* **Detection**: If the StatsPump is unable to connect to the management interface of Solace routers, this will be indicated by the message:

2016-06-08 16:00:30,131 [Poll\_Thread\_00001] INFO com.solacesystems.psg.enterprisestats.statspump.PollerRunnable – Caught on Poller ‘show hostname’ on 192.168.27.128:8080 in non-streaming-mode while fetching due to: java.io.IOException: Server returned HTTP response code: 401 for URL: <http://192.168.27.128:8080/SEMP>

* **Resolution**: Check if the router configuration file passed to StatsPump has the correct management IP address, port, and CLI username and password for connecting to the management interface of the routers. The StatsPump process does not stop if it cannot connect the management interface of a single router; it keeps retrying the connection based on a back-off interval.If any configurations are to be changed, the StatsPump process needs to be stopped, changes to configuration made, and the StatsPump process restarted.

### Error connecting to the message-backbone router interface

* **Detection**: If the StatsPump is unable to connect to the message-backbone interface of one of the Solace routers, this will be indicated by the message:

2016-06-08 16:11:51,314 [Connect\_Thread\_0000e] INFO com.solacesystems.psg.enterprisestats.statspump.VpnConnectionManager – Couldn’t connect to VpnConnection ‘stats\_sdt’ on ‘192.168.27.131’ (attempt 1) due to ‘(Client name: Stats\_Publisher\_\_#NjU3Nzky Local port: -1 Remote addr: 192.168.27.131:55555) – Error communicating with the router.’. Blocking connect attempts for 15 sec

* **Resolution**: Check if the router configuration file passed to StatsPump has the correct settings for the destinations configured: the message-backbone IP address, port, username and password for connecting to the message-backbone interface of the routers. The StatsPump process does not stop if it cannot connect to a single VPN connection; it keeps retrying the connection based on a back-off interval; If any configurations are to be changed, the StatsPump process needs to be stopped, changes to configuration made, and the StatsPump process restarted.

### Incorrect Configuration

* **Detection:** If the StatsPump is unable to load or parse the supplied configuration, it shuts down and an error is written to its log files. For example, one of the poller configuration files were modified to corrupt the SEMP request for “show hardware”. The corresponding error in pump.log is displayed below:

2016-06-08 17:02:41,473 [main] ERROR com.solacesystems.psg.enterprisestats.statspump.StatsPump – There was an issue loading the configuration files

com.solacesystems.psg.enterprisestats.statspump.config.ConfigLoaderException: asdlkfj 6

at com.solacesystems.psg.enterprisestats.statspump.config.ConfigLoader.loadPollerConfig(ConfigLoader.java:189)

at com.solacesystems.psg.enterprisestats.statspump.config.ConfigLoader.loadConfig(ConfigLoader.java:87)

at com.solacesystems.psg.enterprisestats.statspump.StatsPump.load(StatsPump.java:70)

at com.solacesystems.psg.enterprisestats.statspump.StatsPump.main(StatsPump.java:266)

Caused by: javax.xml.bind.UnmarshalException

- with linked exception:

[org.xml.sax.SAXParseException; systemId: file:/opt/solace/nfluxDB/statspump/config/pollers.xml; lineNumber: 151; columnNumber: 15; The element type “hardware” must be terminated by the matching end-tag “</hardware>”.]

at javax.xml.bind.helpers.AbstractUnmarshallerImpl.createUnmarshalException(AbstractUnmarshallerImpl.java:335)

at com.sun.xml.internal.bind.v2.runtime.unmarshaller.UnmarshallerImpl.createUnmarshalException(UnmarshallerImpl.java:563)

at

...

...

...

...

2016-06-08 17:02:41,477 [main] ERROR com.solacesystems.psg.enterprisestats.statspump.StatsPump – The system will now exit

* **Resolution:** Fix the error in the configuration file as indicated in the logs, and restart the StatsPump process.

## Stats Receiver

### Error connecting to the message-backbone router interface

* **Detection**: If the Stats Receiver is unable to connect to the message-backbone interface of the router to subscribe statistics from, and error will be logged in the receiver.log:

2016-07-13 11:24:22,744 [main] INFO com.solacesystems.psg.enterprisestats.receivers.StatsReceiverProperties – Application loading from property file ‘config/influx.properties’.

2016-07-13 11:24:23,189 [main] INFO com.solacesystems.psg.enterprisestats.receivers.influxdb.InfluxDBSpigot – InfluxDb spigot instanciated

2016-07-13 11:24:23,189 [main] INFO com.solacesystems.psg.enterprisestats.receivers.EnterpriseStatsReceiver – Solace enterprise stats running the ‘com.solacesystems.psg.enterprisestats.receivers.influxdb.InfluxDBSpigot’ spigot implementation.

2016-07-13 11:24:23,189 [main] INFO com.solacesystems.psg.enterprisestats.receivers.EnterpriseStatsReceiver – Solace enterprise stats engine loaded.

2016-07-13 11:24:23,283 [main] INFO com.solacesystems.psg.enterprisestats.receivers.EnterpriseStatsReceiver – Binding to queue ‘influxQ’ to nfluxD stats messages using guaranteed messaging.

2016-07-13 11:24:23,283 [main] INFO com.solacesystems.psg.enterprisestats.receivers.EnterpriseStatsReceiver – NOTE: the SOLACE\_TOPICS nfluxDBtion setting is being ignored; ensure the ‘influxQ’ is subscribed to the correct topics.

2016-07-13 11:24:56,389 [main] FATAL com.solacesystems.psg.enterprisestats.receivers.EnterpriseStatsReceiver – Fatal error in stats receiver

com.solacesystems.psg.enterprisestats.receivers.ReceiverException: com.solacesystems.jcsmp.JCSMPTransportException: (Client name: InfluxDBSpigot-1468380263213 Local port: -1 Remote addr: 192.168.40.31:55555) – Timeout happened when reading response from the router.

At com.solacesystems.psg.enterprisestats.receivers.EnterpriseStatsReceiver.bindToQueue(EnterpriseStatsReceiver.java:367)

at com.solacesystems.psg.enterprisestats.receivers.EnterpriseStatsReceiver.start(EnterpriseStatsReceiver.java:228)

at com.solacesystems.psg.enterprisestats.receivers.EnterpriseStatsReceiver.main(EnterpriseStatsReceiver.java:454)

Caused by: com.solacesystems.jcsmp.JCSMPTransportException: (Client name: InfluxDBSpigot-1468380263213 Local port: -1 Remote addr: 192.168.40.31:55555) – Timeout happened when reading response from the router.

At ...

...

...

* **Resolution**: Check the configuration file for the InfluxDB Stats receiver process and ensure that the properties for connecting to the Router are correct. If not, modify the configuration file, and restart the InfluxDB Stats Receiver process.

### Error writing data to InfluxDB

* **Detection**: If the InfluxDB plug in for the Stats Receiver is unable to connect the HTTP Interface of InfluxDB, due to incorrect credentials or a network issue, to write data, an error will be displayed in the receiver.log as follows. . A WARN log will be displayed if the receiver is configured to retry on error, and an ERROR log will be displayed if the receiver has reached the maximum number of retries and is going to discard the statistic.

**WARN Log**:

2018-04-23T04:28:03,465 [InfluxDBStatsTap-65620095] WARN c.s.p.e.r.i.InfluxDBStatsTap - FAILED to write update for measurement: VPN\_CLIENT\_STATS: {ROUTER\_NAME=solace1, VPN\_NAME=ABC\_GLOBAL\_DEV\_01, CLIENT\_NAME=gu.efxqa12.b362745b-dfce-4e83-b1ed-cafcabc73d9f}. 'Server returned HTTP response code: 400 for URL: http://192.168.1.2:8086/write?db=STATS&precision=ms

HTTP/1.1 400 Bad Request

X-Influxdb-Error = partial write: max-series-per-database limit exceeded: (1000000) dropped=1'.

The HTTP request will be retried.

**ERROR log:**

2018-04-23T04:28:03,464 [InfluxDBStatsTap-65620089] ERROR c.s.p.e.r.TapWorkerThreadedTask - Tap failure while processing Stats

com.solace.psg.enterprisestats.receiver.ReceiverException: FAILED to write update for measurement: VPN\_CLIENT\_STATS: {ROUTER\_NAME=solace2, VPN\_NAME=ZYX\_GLOBAL\_DEV\_01, CLIENT\_NAME=gu.efxqa12.a9640feb-f23d-4142-af0d-a5d5c0341f98}. 'Server returned HTTP response code: 400 for URL: http://192.168.1.2:8086/write?db=STATS&precision=ms

HTTP/1.1 400 Bad Request

X-Influxdb-Error = partial write: max-series-per-database limit exceeded: (1000000) dropped=1'.

This statistic will be discarded.

at com.solace.psg.enterprisestats.receiver.influxdb.InfluxDBStatsTap.postToInfluxDb(InfluxDBStatsTap.java:157) ~[solace-influxdb-2.3.6.jar:2.3.6]

at com.solace.psg.enterprisestats.receiver.influxdb.InfluxDBStatsTap.onRouterStats(InfluxDBStatsTap.java:235) ~[solace-influxdb-2.3.6.jar:2.3.6]

at com.solace.psg.enterprisestats.receiver.TapWorkerThreadedTask.processRegularStatsMessage(TapWorkerThreadedTask.java:196) ~[stats-receiver-2.3.6.jar:2.3.6]

at com.solace.psg.enterprisestats.receiver.TapWorkerThreadedTask.processStats(TapWorkerThreadedTask.java:218) ~[stats-receiver-2.3.6.jar:2.3.6]

at com.solace.psg.enterprisestats.receiver.TapWorkerThreadedTask.call(TapWorkerThreadedTask.java:83) [stats-receiver-2.3.6.jar:2.3.6]

at com.solace.psg.enterprisestats.receiver.TapWorkerThreadedTask.call(TapWorkerThreadedTask.java:41) [stats-receiver-2.3.6.jar:2.3.6]

at java.util.concurrent.FutureTask.run(FutureTask.java:266) [?:1.8.0\_161]

at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1149) [?:1.8.0\_161]

at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:624) [?:1.8.0\_161]

at java.lang.Thread.run(Thread.java:748) [?:1.8.0\_161]

* **Resolution**: The HTTP response error code in the logs indicates why the Stats Receiver is unable to reach InfluxDB – verify that the InfluxDB stats receiver can reach the HTTP interface of InfluxDB using those properties, and the InfluxDB service is running. Check the configuration file for the InfluxDB Stats receiver process and ensure that the properties for connecting to InfluxDB’s HTTP interface are correct. If the file needs updating, modify the configuration file, and restart the InfluxDB Stats Receiver process. If the InfluxDB HTTP interface is reachable and the service is running, the Stats Receiver may be unable to write data due to a variety of reasons such as resource starvation on the host running InfluxDB. To troubleshoot, check the InfluxDB logs.

## Debugging SEMP request times: Writing Poller Statistics

The InfluxDB Stats Receiver can be configured to write statistics about the various pollers that the StatsPump is running – such as the amount of time per SEMP request, number of elements returned, etc. This information in InfluxDB can them be charted using Grafana, and can be used to change the polling interval of various statistics on the StatsPump accordingly. Currently, this feature can only be turned on or off. Prior to turning this feature on, a database named “POLLER\_STATS” must be created in InfluxDB, and the same InfluxDB user writing data into the database for statistics must be granted write-access to this database.

For instructions on creating the InfluxDB database, creating a user and giving it read-write access, refer to Section 6.2.1

To turn on this feature:

* Stop the Stats Receiver. For more information, refer to Section 7.4.2
* Modify the InfluxDB Stats configuration file at [$installDir]/config to modify the below two options:
  + Set “ENABLE\_POLLER\_STATS” to true
  + Set “INFLUXDB\_POLLER\_STATS\_DB” to the name of the database for poller stats
* Start the Stats Receiver. For more information, refer to Section 7.4.1

This will now insert into the Database “POLLER\_STATS” a measurement with the statistics of each Poller run by StatsPump, and can be charted using Grafana. If this database does not exist, this data will not be inserted.

If a particular SEMP poller takes more time that its polling interval as defined in the poller groups configuration (groups\_default.xml) , it will generate log messages in the statspump logs such as:

2018-03-12T09:28:32,404 [Poll\_Thread\_0005c] INFO c.s.p.e.s.ScheduledPollerHandle - Poller 'show hostname' on solace1 did not complete after 399.1 seconds (supposed to have a period of 120.0 seconds). Terminating the Poller.

And:

2018-03-12T09:34:54,056 [Poll\_Thread\_000e1] INFO c.s.p.e.s.ScheduledPollerHandle - \*Warning\* Poller 'show bridge \* subscriptions' on solace1 took 903.6 sec to run, but is supposed to run every 300.0 sec

It is recommended that you use the Poller Stats feature to review and determine the average time that a poller takes to return for each router, and use these to tune your groups configuration on the StatsPump accordingly.

# Known Issues and Limitations

The below table lists any known issues and limitations with the StatsPump or InfluxDB Stats Receiver, and provides workarounds for correcting them, where available:

|  |  |  |
| --- | --- | --- |
| **Reference Number** | **Issue** | **Workaround (where available)** |
| 1 | If “management” and “self” destinations are created in the router configuration file for the StatsPump with both pointing to the same message-vpn, it will result in double publishing of poller statistics configured with the destination of “SELF” in the pollers file, i.e. two statistics messages are published instead of one. | If it is required that the message-vpn specific statistics be published to the same destination as the system level statistics, the destination for the relevant message-vpn specific pollers should be changed from “SELF” to “BOTH” to prevent this issue. |
| 2 | If any CLI objects contain any special characters in their name, such as a double quote (“”) the statistics message for this object cannot be written to InfluxDB and an error is seen in the logs. Subsequent statistics messages received are written to InfluxDB normally. | Ensure that CLI objects do not contain any special characters in their names |
| 3 | StatsPump and Stats Receiver do not support SSL or Client Certificate authentication | The Internal, LDAP, or no authentication scheme can be used for authenticating StatsPump and InfluxDB Stats Receiver connections. |
| 4 | The StatsPump does not support publishing messages with compression |  |

History

| Version | Status | Date | Author | Reason for changes |
| --- | --- | --- | --- | --- |
| 1.0 | Release | July 13th, 2016 | Shrikanth Rajgopalan | Updated runbook based on the improved version of StatsPump |
| 1.1 | Release | Nov 14, 2016 | Mike O’Brien | Added architectural overview |
| 1.2 | Release | Jan 17, 2017 | Mike O’Brien | Updated with new packaging and procedures to control execution from a J2EE server. |
| 1.3 | Release | Feb 3/2017 | Mike O’Brien | Updated with new “local plug in” configuration and procedures |
| 1.4 | Release | Apr 27/2017 | Steve Chan | Cosmetic changes |
| 1.5 | Release | Aug 14/2017 | Mike O’Brien | Added notes about upgrading the Pump with new versions of SolOS. |
| 1.6 | Release | Aug 15/2017 | Shrikanth Rajgopalan | Updated runbook with details of Systemd Linux service script, moved poller stats to a section 9.3 on debugging SEMP requests, updated deployment diagrams, reviewed entire document |
| 1.7 | Release | Oct 27/2017 | Shrikanth Rajgopalan | Added Sections 7.1.5 and 7.4.4 on upgrading the StatsPump and StatsReceiver |
| 1.8 | Release | April 1/2018 | Ramesh Natarajan | Additional notes and edits. |
| 1.9 | Release | April 4/2018 | Mike O’Brien | Added notes regarding reconnect arguments for receiver config file. |
| 1.10 | Release | April 17/2018 | Mike O’Brien | Added notes regarding HTTP connect retries for receiver config file. |
| 1.11 | Release | April 24/2018 | Shrikanth Rajgopalan | Updates to InfluxDB configuration commands in Sec 6.2.1. |
| 1.12 | Release | April 26/2018 | Mike O’Brien | Added notes regarding new poller start/end broadcasts |
| 1.13 | Release | May 02/2018 | Shrikanth Rajgopalan | Updates to Section 9.2 and 9.3 on updated error messages from the StatsReceiver and Pump |
| 1.14 | Release | Feb 11/2019 | Mike O’Brien | Add notes re: force fields to numeric feature in receiver |
| 1.15 | Release | Apr 17/2020 | Shrikanth Rajgopalan | Add notes on the secure SEMP feature introduced in 2.6.0 |