# telegraf-os Documentation

Release 1

**Guilhem Marchand** 

Aug 29, 2021

## Contents

1	Overv	
	1.1	About
	1.2	Compatibility
		Known Issues
	1.4	Support & donate
	1.5	Download
2		oyment and configuration: 7
	2.1	Deployment & Upgrades
	2.2	Telegraf metrics ingestion
		Splunk dashboards (health views)
3		bleshoot: 31
	3.1	Troubleshoot & FAQ
4	Versi	oniong and build history: 33
	4.1	Release notes   33



The Splunk application for Operating System monitoring with Telegraf provides analytic and reporting for Linux and Windows metrics ingested in the Splunk metric store:

Service Analyzer   Notable Events Review Glass Tables Deep Dives Multi-KPI Alerts Sear	ch ▼ Configure ▼ Product Tour	IT Service Intelligence
Telegraf OS metrics		Edit Export •
entky_summany * fmilly_Hindows #0.5pust 2 mem_tstal:4006 H0	1.39 arg % cpu	48.36 % disk used
OVERVIEW CPU MEMORY DISKO NET DISK UPTIME FORECAST		
CPU utilization summary	Memory available (MB)	Disk MB/s
- max cpu load % - avg cpu load %	Available_Bytes - mem used (%) 3200 40	write_bytes read_bytes
20 200 10 mm pill 2010 20	1000 1000 1000 1000 1000 1000 22:00 12:00 1000 1000 1000 22:00 1000 1000 1000 22:00 	01 055 1200 M00 NEO 2000 2260 208
file-systems usage	Network Inbound/outbound traffic (KBps)	Context Switches (per sec)
instance      disk total (GB)      used_pct      c-     ta aa     48.352	inbound autoound	- Context_Switches_persec
C. 30.40 49.555	ради и на	1300 1300 100 1000 1

## CHAPTER 1

## Overview:

## 1.1 About

- Author: Guilhem Marchand
- First release published in October 2018
- Purposes:

The Splunk application for OS monitoring with Telegraf leverages the Influxdata Telegraf agent to provide key layer Operating System monitoring for Windows and Linux:

• Telegraf from Influxdata (https://github.com/influxdata/telegraf)

The Splunk application is backport of the ITSI module for Telegraf OS for non ITSI users.

https://da-itsi-telegraf-os.readthedocs.io

## 1.2 Compatibility

#### 1.2.1 Splunk compatibility

All the metrics are ingested into the high performance Splunk metric store, Splunk 7.0.x or higher is required.

#### 1.2.2 Telegraf compatibility

Telegraf supports various operating systems and process architectures including any version of Linux and Windows.

For more information:

• https://portal.influxdata.com/downloads

## 1.3 Known Issues

There are no known issues at the moment.

## 1.4 Support & donate

## I am supporting my applications for free, for the good of everyone and on my own private time. As you can guess, this is a huge amount of time and efforts.

If you enjoy it, and want to support and encourage me, buy me a coffee (or a Pizza) and you will make me very happy! The Splunk application for Operating System monitoring with Telegraf is community supported. To get support, use of one the following options:

#### 1.4.1 Splunk Answers

Open a question in Splunk answers for the application:

• https://answers.splunk.com/app/questions/4271.html

#### 1.4.2 Splunk community slack

Contact me on Splunk community slack, or even better, ask the community !

• https://splunk-usergroups.slack.com

#### 1.4.3 Open a issue in Git

To report an issue, request a feature change or improvement, please open an issue in Github:

• https://github.com/guilhemmarchand/telegraf-os/issues

#### 1.4.4 Email support

• guilhem.marchand@gmail.com

However, previous options are far betters, and will give you all the chances to get a quick support from the community of fellow Splunkers.

## 1.5 Download

## 1.5.1 Splunk Application for Operating System monitoring with Telegraf

#### The Splunk application can be downloaded from:

#### Splunk base

• https://splunkbase.splunk.com/app/4271

#### GitHub

• https://github.com/guilhemmarchand/telegraf-os

## CHAPTER 2

## Deployment and configuration:

## 2.1 Deployment & Upgrades

#### 2.1.1 Deployment matrix

Splunk roles	required
Search head	yes
Indexer tiers	no

If Splunk search heads are running in Search Head Cluster (SHC), the Splunk application must be deployed by the SHC deployer.

The deployment and configuration requires the creation of a dedicated metric index (by default called **telegraf**), see the implementation section.

## 2.1.2 Dependencies

The overview home page requires the deployment of the horizon chart and horseshoe third party visualisation addons on the search heads:

- https://splunkbase.splunk.com/app/3117
- https://splunkbase.splunk.com/app/3166

The metric workspace application is not required but highly recommended, if it is installed, the Metric link will be automatically available within the application bar:

• https://splunkbase.splunk.com/app/4192/

#### 2.1.3 Initial deployment

The deployment of the Splunk application for OS monitoring with Telegraf is straight forward:

- Using the application manager in Splunk Web (Settings / Manages apps)
- Extracting the content of the tgz archive in the "apps" directory of Splunk
- For SHC configurations (Search Head Cluster), extract the tgz content in the SHC deployer and publish the SHC bundle

### 2.1.4 Upgrade

Upgrading the Splunk application is pretty much the same operation, use one of the techniques that matches your conditions / requirements.

## 2.2 Telegraf metrics ingestion

Implementing Telegraf and sending its metrics in Splunk is simple, and efficient.

It is not the purpose of this documentation to expose every piece of the installation and configuration of Telegraf or Splunk.

#### 2.2.1 Telegraf installation and configuration

#### Telegraf standard installation (standalone and independant process)

The installation of Telegraf is really straightforward, consult:

• https://github.com/influxdata/telegraf

If you wish to deploy Telegraf as a Splunk TA application instead, consult the next step.

#### Telegraf deployment as Splunk application deployed by Splunk (TA)

You can publish Telegraf through a Splunk application that you push to your clients using a Splunk deployment server.

This means that you can create a custom Technology Addon (TA) that contains both the Telegraf binary and the telegraf.conf configuraton files.

#### This method has several advantages:

- If you are a Splunk customer already, you may have the Splunk Universal Forwarder deployed on your servers, you will NOT need to deploy an additional collector independently from Splunk
- · You get benefit from the Splunk centralisation and deploy massively Telegraf from Splunk
- You can maintain and upgrade Telegraf just as you do usually in Splunk, all from your Splunk Deployment Server. (DS)

Linux 64 bits example:

• https://github.com/guilhemmarchand/TA-telegraf-amd64

#### Windows 64 bits example:

• https://github.com/guilhemmarchand/TA-telegraf-windows64

#### **Telegraf minimal configuration**

A minimal configuration to monitor Operating System metrics:

· https://docs.influxdata.com/chronograf/latest/guides/using-precreated-dashboards/#system

The output configuration depends on the deployment you choose to use to ingest metrics in Splunk, consult the next sections.

Example of a minimal telegraf.conf configuration that monitors Operating System metrics for Linux:

```
# Read metrics about cpu usage
[[inputs.cpu]]
 ## Whether to report per-cpu stats or not
 percpu = true
 ## Whether to report total system cpu stats or not
 totalcpu = true
 ## If true, collect raw CPU time metrics.
 collect_cpu_time = false
 ## If true, compute and report the sum of all non-idle CPU states.
 report_active = false
# Read metrics about disk usage by mount point
[[inputs.disk]]
 ## By default stats will be gathered for all mount points.
 ## Set mount_points will restrict the stats to only the specified mount points.
 # mount_points = ["/"]
 ## Ignore mount points by filesystem type.
 ignore_fs = ["tmpfs", "devtmpfs", "devfs"]
# Read metrics about disk IO by device
[[inputs.diskio]]
 ## By default, telegraf will gather stats for all devices including
 ## disk partitions.
 ## Setting devices will restrict the stats to the specified devices.
 # devices = ["sda", "sdb", "vd*"]
 ## Uncomment the following line if you need disk serial numbers.
 # skip_serial_number = false
 ## On systems which support it, device metadata can be added in the form of
 ## tags.
 ## Currently only Linux is supported via udev properties. You can view
 ## available properties for a device by running:
 ## 'udevadm info -q property -n /dev/sda'
 # device_tags = ["ID_FS_TYPE", "ID_FS_USAGE"]
 ## Using the same metadata source as device_tags, you can also customize the
 ## name of the device via templates.
 ## The 'name_templates' parameter is a list of templates to try and apply to
 ## the device. The template may contain variables in the form of '$PROPERTY' or
 ## '${PROPERTY}'. The first template which does not contain any variables not
 ## present for the device is used as the device name tag.
 ## The typical use case is for LVM volumes, to get the VG/LV name instead of
 ## the near-meaningless DM-0 name.
 # name_templates = ["$ID_FS_LABEL", "$DM_VG_NAME/$DM_LV_NAME"]
# Get kernel statistics from /proc/stat
[[inputs.kernel]]
```

(continues on next page)

# no configuration

(continued from previous page)

```
# Read metrics about memory usage
[[inputs.mem]]
 # no configuration
# Get the number of processes and group them by status
[[inputs.processes]]
 # no configuration
# Read metrics about swap memory usage
[[inputs.swap]]
 # no configuration
# Read metrics about system load & uptime
[[inputs.system]]
 # no configuration
# # Read metrics about network interface usage
[[inputs.net]]
   ## By default, telegraf gathers stats from any up interface (excluding loopback)
#
#
   ## Setting interfaces will tell it to gather these explicit interfaces,
#
  ## regardless of status.
#
  ##
#
  # interfaces = ["eth0"]
  ##
#
  ## On linux systems telegraf also collects protocol stats.
#
  ## Setting ignore_protocol_stats to true will skip reporting of protocol metrics.
#
   ##
#
   # ignore_protocol_stats = false
#
   ##
#
# # Read TCP metrics such as established, time wait and sockets counts.
[[inputs.netstat]]
#
  # no configuration
# # Monitor process cpu and memory usage
[[inputs.procstat]]
  ## PID file to monitor process
#
  pid_file = "/var/run/nginx.pid"
#
   ## executable name (ie, pgrep <exe>)
#
#
   # exe = "nginx"
   ## pattern as argument for pgrep (ie, pgrep -f <pattern>)
#
   # pattern = "nginx"
#
   ## user as argument for pgrep (ie, pgrep -u <user>)
#
#
  user = "root"
#
  ## Systemd unit name
#
  # systemd_unit = "nginx.service"
#
  ## CGroup name or path
   # cqroup = "systemd/system.slice/nginx.service"
#
#
#
   ## override for process_name
   ## This is optional; default is sourced from /proc/<pid>/status
#
```

(continues on next page)

(continued from previous page)

```
# process_name = "bar"
#
#
#
   ## Field name prefix
#
   # prefix = ""
   ## Add PID as a tag instead of a field; useful to differentiate between
#
   ## processes whose tags are otherwise the same. Can create a large number
#
#
   ## of series, use judiciously.
   # pid_tag = false
#
#
#
   ## Method to use when finding process IDs. Can be one of 'pgrep', or
#
   ## 'native'. The pgrep finder calls the pgrep executable in the PATH while
#
   ## the native finder performs the search directly in a manor dependent on the
   ## platform. Default is 'pgrep'
#
  # pid_finder = "pgrep"
#
   pattern = ".*"
```

#### Windows additional configuration (mem inputs)

For Windows memory management, the default win\_mem inputs does not retrieve some of the metrics we need.

You need to activate the memory inputs. (which on Windows uses WMI collection):

[[inputs.mem]]
# no configuration

#### Windows Active Directory Domain Controller

#### Follow instructions for "Active Directory Domain Controller":

 https://github.com/influxdata/telegraf/tree/master/plugins/inputs/win\_perf\_counters# active-directory-domain-controller

#### Windows DNS server

#### Follow instructions for "DNS Server + Domain Controllers":

 https://github.com/influxdata/telegraf/tree/master/plugins/inputs/win\_perf\_counters#dns-server-domaincontrollers

#### Windows DFS server

#### For DFS Namespace, follow instructions for "DFS Namespace + Domain Controllers":

 https://github.com/influxdata/telegraf/tree/master/plugins/inputs/win\_perf\_counters#dfs-namespace-domaincontrollers

#### For DFS Replication, follow instructions for "DFS Replication + Domain Controllers":

 https://github.com/influxdata/telegraf/tree/master/plugins/inputs/win\_perf\_counters#dfs-replication\_domaincontrollers

#### Microsoft IIS / ASP.NET

#### For IIS and ASP.NET, follow instructions for: IIS / ASP.NET

https://github.com/influxdata/telegraf/tree/master/plugins/inputs/win\_perf\_counters#iis-aspnet

#### Linux processes monitoring (procstat)

In the linux views, the processes usage (both CPU and Memory) rely on the procstat inputs, which requires additional configuration depending on your context.

As for an example, the following configuration monitors all the processes owned by the "splunk" unix user:

```
[[inputs.procstat]]
   ## PID file to monitor process
#
  pid_file = "/var/run/nginx.pid"
#
  ## executable name (ie, pgrep <exe>)
#
#
  # exe = "nginx"
  ## pattern as argument for pgrep (ie, pgrep -f <pattern>)
#
  # pattern = "nginx"
#
  ## user as argument for pgrep (ie, pgrep -u <user>)
#
   user = "splunk"
```

#### 2.2.2 HTTP Events Collector (HEC)

Splunk deployment with HEC (available with Telegraf starting version 1.8)

Telegraf agents -> HTTP over SSL -> Splunk HEC inputs

With Telegraf starting version 1.8, you can send metrics directly from Telegraf to HTTP Events Collector using the excellent serializer leveraging the http Telegraf output.

This is extremely simple, scalable and reliable.

*Example of an HEC input definition:* 

#### Splunk inputs.conf:

```
[http://Telegraf]
disabled = 0
index = telegraf
indexes = telegraf
token = c386d4c8-8b50-4178-be76-508dca2f19e2
```

#### **Telegraf configuration:**

The Telegraf configuration is really simple and relies on defining your ouput:

Example:

```
[[outputs.http]]
  ## URL is the address to send metrics to
  url = "https://mysplunk.domain.com:8088/services/collector"
  ## Timeout for HTTP message
  # timeout = "5s"
  ## Optional TLS Config
  # tls_ca = "/etc/telegraf/ca.pem"
  # tls_cert = "/etc/telegraf/cert.pem"
```

(continues on next page)

(continued from previous page)

```
# tls_key = "/etc/telegraf/key.pem"
## Use TLS but skip chain & host verification
insecure_skip_verify = true
## Data format to output.
## Each data format has it's own unique set of configuration options, read
## more about them here:
## https://github.com/influxdata/telegraf/blob/master/docs/DATA_FORMATS_OUTPUT.md
data_format = "splunkmetric"
 ## Provides time, index, source overrides for the HEC
splunkmetric_hec_routing = true
 ## Additional HTTP headers
 [outputs.http.headers]
# Should be set manually to "application/json" for json data_format
  Content-Type = "application/json"
  Authorization = "Splunk c386d4c8-8b50-4178-be76-508dca2f19e2"
  X-Splunk-Request-Channel = "c386d4c8-8b50-4178-be76-508dca2f19e2"
```

Push this configuration to your Telegraf agents, et voila.

#### Check data availability in Splunk:

#### 2.2.3 TCP / TCP-SSL Inputs

#### Splunk deployment with TCP inputs.

This deployment requires additional indexing time parsing configuration:

• https://github.com/guilhemmarchand/TA-influxdata-telegraf

The deployment is very simple and can be described as:

Telegraf agents -> TCP or TCP over SSL -> Splunk TCP inputs

In addition and to provide resiliency, it is fairly simple to add a load balancer in front of Splunk, such that you service continues to ingest metrics depending on Splunk components availability. (HAProxy, Nginx, F5, whatsoever...)

The data output format used by Telegraf agents is the "Graphite" format with tag support enable. This is simple, beautiful, accurate and allows the management of any number of dimensions.

Example of a tcp input definition:

#### Splunk inputs.conf:

```
[tcp://2003]
connection_host = dns
index = telegraf
sourcetype = tcp:telegraf:graphite
```

#### **Telegraf configuration:**

The Telegraf configuration is really simple and relies on defining your ouput:

Example:

```
[[outputs.graphite]]
  ## TCP endpoint for your graphite instance.
  ## If multiple endpoints are configured, the output will be load balanced.
  ## Only one of the endpoints will be written to with each iteration.
  servers = ["mysplunk.domain.com:2003"]
  ## Prefix metrics name
  prefix = ""
  ## Graphite output template
  ## see https://github.com/influxdata/telegraf/blob/master/docs/DATA_FORMATS_OUTPUT.
  fmd
  # template = "host.tags.measurement.field"
  ## Enable Graphite tags support
  graphite_tag_support = true
  ## timeout in seconds for the write connection to graphite
  timeout = 2
```

Push this configuration to your Telegraf agents, et voila.

#### Check data availability in Splunk::

mcatalog values(\_dims) as dimensions values(metric\_name) as metric\_name where index=telegraf metric\_name=\*

#### 2.2.4 SPLUNK file monitoring Ingestion

#### Splunk deployment with Splunk file monitoring.

This deployment requires additional indexing time parsing configuration:

• https://github.com/guilhemmarchand/TA-influxdata-telegraf

Telegraf has a "file" output plugin that allows writing metrics to a local file on the file-system, don't say more that is much more than enough to Splunk it.

#### The Telegraf configuration is really simple and relies on defining your ouput:

Example:

```
[[outputs.file]]
## Files to write to, "stdout" is a specially handled file.
files = ["/tmp/metrics.out"]
## Data format to output.
## Each data format has its own unique set of configuration options, read
## more about them here:
## https://github.com/influxdata/telegraf/blob/master/docs/DATA_FORMATS_OUTPUT.md
data_format = "graphite"
graphite_tag_support = true
```

Notes: this is a simplistic example, in real condition do not forget to manage the file rotation using a simple logrorate configuration for Linux, and relevant solution for other OS.

#### Splunk file input configuration:

I cannot say more, this is simple, very simple. Add the following configuration to any inputs.conf configuration file of your choice:

Example:

```
[monitor:///tmp/metrics.out]
disabled = false
index = telegraf
sourcetype = file:telegraf:graphite
```

Apply this simple input.conf, if you deploy thought the Splunk deployment server ensure splunkd is configured to restart in your serverclass configuration.

Et voila, Splunk ingests the metrics continuously and metrics are forwarded to the indexing layer using your Splunk infrastructure, be on-premise, private or Splunk Cloud.

#### 2.2.5 KAFKA Ingestion

#### Splunk deployment with Kafka.

#### This deployment requires additional indexing time parsing configuration:

• https://github.com/guilhemmarchand/TA-influxdata-telegraf

If you are using Kafka, or consider using it, producing Telegraf metrics to Kafka makes a lot of sense.

First, Telegraf has a native output plugin that produces to a Kafka topic, Telegraf will send the metrics directly to one or more Kafka brokers providing scaling and resiliency.

Then, Splunk becomes one consumer of the metrics using the scalable and resilient Kafka connect infrastructure and the Splunk Kafka connect sink connector. By using Kafka as the mainstream for your metrics, you preserve the possibility of having multiple technologies consuming these data in addition with Splunk, while implementing a massively scalable and resilient environment.

On the final step that streams data to Splunk, the Kafka sink connector for Splunk sends data to Splunk HEC, which makes it resilient, scalable and eligible to all Splunk on-premise or Splunk Cloud platforms easily.

#### The deployment with Kafka can be described the following way:

*Telegraf agents -> Kafka brokers <- Kafka connect running Splunk sink connector -> Splunk HTTP Event Collector (HEC)* 

#### **Configuring Kafka connect:**

• The Kafka connect properties needs to use the "String" converter, the following example start Kafka connect with the relevant configuration:

connect-distributed.properties:

```
# These are defaults. This file just demonstrates how to override some settings.
bootstrap.servers=kafka-1:9092,kafka-2:9092,kafka-3:9092
key.converter=org.apache.kafka.connect.storage.StringConverter
value.converter=org.apache.kafka.connect.storage.StringConverter
key.converter.schemas.enable=false
value.converter.schemas.enable=false
internal.key.converter=org.apache.kafka.connect.json.JsonConverter
internal.value.converter=org.apache.kafka.connect.json.JsonConverter
internal.key.converter.schemas.enable=false
internal.value.converter.schemas.enable=false
# Flush much faster (10s) than normal, which is useful for testing/debugging
offset.flush.interval.ms=10000
plugin.path=/etc/kafka-connect/jars
group.id=kafka-connect-splunk-hec-sink
config.storage.topic=__kafka-connect-splunk-task-configs
```

(continues on next page)

(continued from previous page)

```
config.storage.replication.factor=3
offset.storage.topic=__kafka-connect-splunk-offsets
offset.storage.replication.factor=3
offset.storage.partitions=25
status.storage.topic=__kafka-connect-splunk-statuses
status.storage.replication.factor=3
status.storage.partitions=5
# These are provided to inform the user about the presence of the REST host and port.
⇔configs
# Hostname & Port for the REST API to listen on. If this is set, it will bind to the_
→interface used to listen to requests.
#rest.host.name=
rest.port=8082
# The Hostname & Port that will be given out to other workers to connect to i.e. URLs.
\rightarrow that are routable from other servers.
# rest.advertised.host.name=kafka-connect-1
```

Apply the following command against Kafka connect running the Splunk Kafka sink connector: (https://splunkbase.splunk.com/app/3862)

- replace <ip address or host> by the IP address or the host name of the Kafla connect node, if you run the command locally, simply use localhost
- replace the port if required (default is 8082)
- replace the HEC token
- replace the HEC destination
- adapt any other configuration item up to your needs

#### Achieve the following command:

```
curl localhost:8082/connectors -X POST -H "Content-Type: application/json" -d '{
"name": "kafka-connect-telegraf",
"config": {
"connector.class": "com.splunk.kafka.connect.SplunkSinkConnector",
"tasks.max": "3",
"topics":"telegraf",
"splunk.indexes": "telegraf",
"splunk.sourcetypes": "kafka:telegraf:graphite",
"splunk.hec.uri": "https://myhecinput.splunk.com:8088",
"splunk.hec.token": "fd96ffb6-fb3e-43aa-9e8b-de911356443f",
"splunk.hec.raw": "true",
"splunk.hec.ack.enabled": "true",
"splunk.hec.ack.poll.interval": "10",
"splunk.hec.ack.poll.threads": "2",
"splunk.hec.ssl.validate.certs": "false",
"splunk.hec.http.keepalive": "true",
"splunk.hec.max.http.connection.per.channel": "4",
"splunk.hec.total.channels": "8",
"splunk.hec.max.batch.size": "1000000",
"splunk.hec.threads": "2",
"splunk.hec.event.timeout": "300",
"splunk.hec.socket.timeout": "120",
"splunk.hec.track.data": "true"
} '
```

#### telegraf output configuration:

Configure your Telegraf agents to send data directly to the Kafka broker in graphite format with tag support:

```
[[outputs.kafka]]
  ## URLs of kafka brokers
brokers = ["kafka-1:19092","kafka-2:29092","kafka-3:39092"]
  ## Kafka topic for producer messages
topic = "telegraf"
  data_format = "graphite"
  graphite_tag_support = true
```

Et voila. Congratulations, you have built a massively scalable, distributable, open and resilient metric collection infrastructure.

#### Check data availability in Splunk:

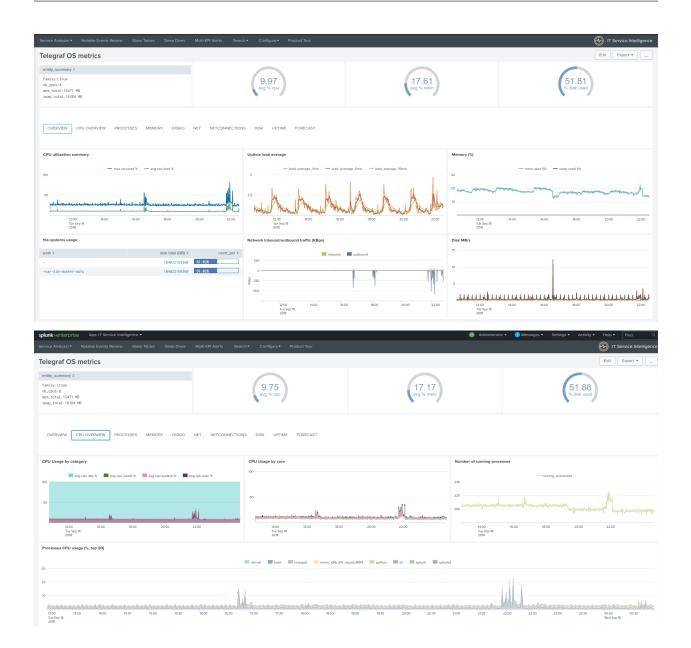
## 2.3 Splunk dashboards (health views)

#### 2.3.1 Overview (landing page)

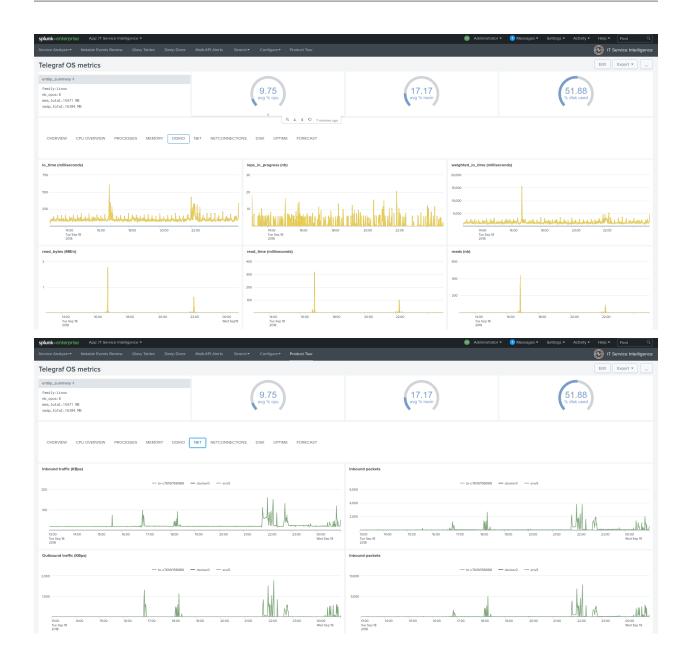
The Splunk application home page provides an overview of the Kafka infrastructure:

#### 2.3.2 View for Linux OS

The Health view for Linux OS automatically appears as "Telegraf OS (Linux)" deepdive drilldown link when entities are discovered:



splunk-enterprise Apr:17 Service Intelligence • 🛞 Messages • Settings • Activity • Help • Find Q						
Service Analyzer* Notable Events Review Glass Tables Deep Dives Multi-KPI Alerts Search* Configure* Product Tour						
Telegraf OS metrics			Edit Export •			
entity_summary 5 f maily_linox mc_cousi mem_cotal:15011 M0 swap_total:16314 M8	9.75 mg % cgu	17.17 urg 's mum	51.88 St date used			
	NET NETCONNECTIONS DISK UPTIME FORECAST					
blocked	dead	idle	paging			
0.79 03 0.25 14.00 18.00 2000 22.00	10 20 4.00 16.00 18.00 20.00 22.00	50 10 1400 1800 1800 2000 2200	50 50 1400 1600 1000 2200			
94.00 16.00 18.00 20.00 22.00 Tue Sep 18 2018	\$4.00 16.00 18.00 20.00 22.00 Tue Sep 18 2018	14:00 16:00 18:00 20:00 22:00 Tue Sep 18 2018	14.00 16.00 18.00 20.00 22.00 Tue Sep 18 2018			
running	sleeping	stopped	total			
75	240	100	240			
25 Narwallichia Un walance and have	200 in all providences and in a station with grantine of grant providences in the providence of the pr	20	200 incidence of an estimation of a production of a production of a production of a production of the			
14.00 16.00 18.00 20.00 22.00 Tue Sep 18 2018	14:00 16:00 18:00 20:00 22:00 Tue Sep 18 2018	14.00 16.00 18.00 20.00 22.00 Tue Sep 18 2018	14:00 16:00 18:00 20:00 22:00 Tue Sep 18 2018			
splunk-enterprise App: IT Service Intelligence + Service Analyzer + Notable Events Review Glass Tables Deep Dives Telegraf OS metrics	MultisKPI Alerts Search • Configure • Product Tour	🚷 Administr	ator • • • Messages • Settings • Activity • Help • Find Q. ③ IT Service Intelligence Edit _ Export •			
entity_summary \$		$\frown$				
family:Linox nb_cpos:8 men_total:15471 MB swap_total:16384 MB	9.75 avg % cpu	17.17 avg % mem	51.88 % disk used			
OVERVIEW CPU OVERVIEW PROCESSES MEMORY DISKO NET NETCONNECTIONS DISK UPTIME FORECAST						
Memory used/available (GB)	Swap used/global (GB)	Swap In / Out (MB)				
		1,500				
10	10	1,000				
	the second s	500				
14:00 16:00 18:00 20:00 Tue Sep 18 2018 buffered cached free - total use	22:00 00:00 14:00 16:00 18:00 18:00 10:00	20.00 22.00 00.00 Wed Sep19 14.00 Tile Sep 18 2018	96.00 1800 20.00 22.00			
Processes - Resident Memory (MB, top 20)	- Tree -	2018				
recessor resident methory (mb, top 20)	Processes - Resident Memory (MB, top 20)					
-don-ki i baki i mongod i mnor_jde_j-k_jduutututuk i imooparekshi i python ish i keep ii spans i guum-dyamata ispund 1500						
1,000	•					
50						
13.00 13.30 14.00 14.30 15.00 15.30 Turs Sep 18 2018	16.00 16.30 17.00 17.30 18.00 18.30	9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9	2200 2230 2300 2330 0000 0030 Wed Sep 19			



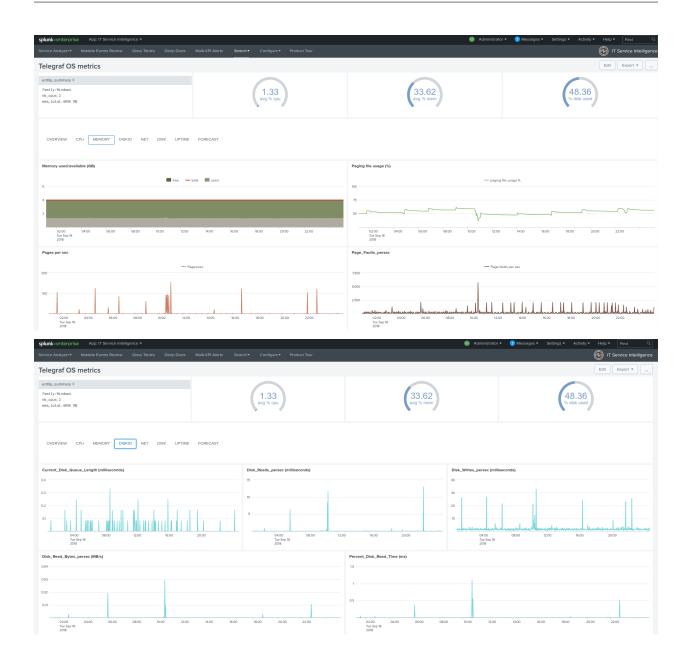
splunk>enterprise App: IT Service Intelligence * Service Analyzer* Notable Events Review Glass Tables Deep Dives	Multi-KPI Alerts Search • Configure • Product Tour	🔒 Administrati	or * 🌒 Messages * Settings * Activity * Help * Find ۹. آکی IT Service Intelligence
Telegraf OS metrics			Edit Export •
ently_summary 0 feat (y 1.1me nb_case) 1 mes, cotal:1547 10 sep_total:16344 10	9.75 org % cpu	17.17 arg 's mem	51.88 % disk used
	NET NETCONNECTIONS DISK UPTIME FORECAST		
Network Open Sockets — top_establishe	d — udp_socket	Network Sockets Created/Second — tcp_establishe	d — udp_socket
	1000 2000 2000 2000 2200 2000 Wetting 0	00 00 00 00 00 00 00 00 00 00	WWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWW
splunk-enterprise ApprIT Service Intelligence • Service Analyzer • Notable Events Review Glass Tables Deep Dives Telegraf OS metrics	Multi-KPI Akerts Search* Configure* Product Tour	Administrati	or ▼ ● Messages ▼ Settings ▼ Activity ▼ Help ▼ Find Q ④ IT Service Intelligence Edit Export ▼
entity_summary * feally_timx nb_qps:3 men_total:1591 M0 swap_total:1594 M0	9.75 arg % qu	17.17 wg % mem	51.88 N disk used
OVERVIEW CPU OVERVIEW PROCESSES MEMORY DISKO	NET NETCONNECTIONS DISK UPTIME FORECAST		
used_percent			
110		95 — -snap-care-5145 — -snap-care-5328 — -var-ib-docker-aufs	
90			
80			
60			
50			
13.00 13.30 14.00 14.30 15.00 15.30 Tur5 Np 18 2018	16.00 16.30 17.00 17.30 18.00 18.30	10:00 19:30 20:80 20:30 21:00 21:30	22:00 22:30 23:00 23:30 00:00 00:30 Wed Sep 19

splunk>enterprise App: IT Service Intelligence •		Administrato	r 🔻 🌖 Messages 💌 Settings 👻 Activity 💌 Help 💌 🛛 Find 🔍			
Service Analyzer   Notable Events Review Glass Tables Deep Dives			IT Service Intelligence			
Telegraf OS metrics			Edit Export •			
ettly_summay 2 fasliy:Linox nd_cous:8 mem_total:15471 M8 swap_total:16384 M8	9.75 mg % qu	17.17 og % men	51.88 % disk used			
OVERVIEW CPU OVERVIEW PROCESSES MEMORY DISKO	NET NETCONNECTIONS DISK UPTIME FORECAST					
	018 06:31		:30:24			
1         1						
splunk-ohtterprise App: IT Service Intelligence • Service Analyzer • Notable Events Review Glass Tables Deep Dives Telegraf OS metrics	Multi-KPI Alerts Search • Configure • Product Tour	Administrate	r ● Messages ● Settings ● Activity ● Help ● [Find ] ④ IT Service Intelligence Edit Export ●			
ently_summary 6 fastly:Lina rb_ctos:8 eme_total:1571718 swap_total:1534-18	9.75 arg % qui	17.17 eg % men	51.88 % disk und			
OVERVIEW CPU OVERVIEW PROCESSES MEMORY DISKIO	NET NETCONNECTIONS DISK UPTIME FORECAST					
Storage usage (GB) prediction for next 3 months						
700	storage_used total predict					
600						
500						
400						
300						
200						
100						
Mon Aug 20 Mon Aug 27 Mon Sep 3 Mon Sep 10 Mon Se 2018	p 17 Mon Sep 24 Mon Oct 1 Mon Oct 8 Mon Oct 15 M	on Oct 22 Mon Oct 29 Mon Nov 5 Mon Nov 12 Mon Nov 19	Mon Nov 26 Mon Dec 3 Mon Dec 10 Mon Dec 17 Mon Dec 24			

### 2.3.3 View for Windows OS

The Health view for Linux OS automatically appears as "Telegraf OS (Windows)" deepdive drilldown link when entities are discovered:

Service Analyzer   Notable Events Review Glass Tables Deep Dives Multi-KPI Alerts S	earch ▼ Configure ▼ Product Tour			🚱 r	Service Intelligence
Telegraf OS metrics				Edit	Export •
entity_summary \$	$\frown$				
family:Windows nb_cpus:2	1.39	33.5 avg % mem		48.36	
mem_total:4096 MB	avg % cpu	avg % mem		% disk used	
OVERVIEW CPU MEMORY DISKO NET DISK UPTIME FORECAST					
CPU utilization summary	Memory available (MB)		Disk MB/s		
- max cpu load % - avg cpu load %		rs — mem used (%)		- write_bytes - read_bytes	
100	3,200 2	40	015		
50	6 1600	The second secon	0.1		
	E June market	- <u>-</u>	0.05		
Local         Local <thlocal< th="">         Local         <thl< td=""><td>12:00 14:00 16:0 Tue Sep 18 2018</td><td>0 18:00 20:00 22:00</td><td>12:00 Tue Sep 18 2018</td><td>14:00 16:00 18:00 20</td><td>00 22:00</td></thl<></thlocal<>	12:00 14:00 16:0 Tue Sep 18 2018	0 18:00 20:00 22:00	12:00 Tue Sep 18 2018	14:00 16:00 18:00 20	00 22:00
		_time			
file-systems usage	Network Inbound/outbound traffic (KBps)		Context Switches (per sec		
instance ‡         disk total (GB) ‡         used_pci           C:         30.00         48,36%	t inbound	outbound	2.000	- Context_Switches_persec	
	o where the day includes allowed and the state of the second seco		1,500		
	ชื่ <sub>น</sub> เกมส์สารณ์ที่ได้ไปสารณ์ที่สารณ์ที่สารณ์ที่สารณ์ที่สารณ์ที่สารณ์ที่สารณ์ที่สารณ์ที่สารณ์ที่สารณ์ที่สารณ์ที่สารณ์		1000 Whenhand march mar		الوراييليانياتياتيات
	12.00 14.00 16 Tue Sep 18 2018	.00 18.00 20.00 22.00	12:00 Tue Sep 18 2018	14:00 16:00 18:00 20	00 22:00
splunk>enterprise App: IT Service Intelligence •			🔒 Administrat		
Service Analyzer   Notable Events Review Glass Tables Deep Dives Multi-KPI Alerts	Search   Configure   Product Tour			$\odot$	IT Service Intelligenc
Telegraf OS metrics				Ed	t Export •
entity_summary 0					
family:Windows nb_cpus:2	1.33 avg % cpu	33.62 avg % mem		48.36 % disk used	
men_total:4096 MB					
OVERVIEW CPU MEMORY DISKIO NET DISK UPTIME FORECAST					
OVERVIEW COUNTER DISKIC RELI DISK OPTIME TORECAI					
Percent_User_Time	Percent_Processor_Time		Percent_Privileged_Tim		
100	100		100		
50	50		50		
			,		
04:00 08:00 12:00 16:00 20:00 Tue Sep 18 2018	04:00 08:00 Tue Sep 18 2018	12:00 16:00 20:00	04:00 Tue Sep 18 2018	08:00 12:00 16:00	20:00
Percent_Interrupt_Time	Percent_Idle_Time		Percent_DPC_Time		
100	100		100		
	82 200 - 200 - 200 - 200 - 200 - 200 - 200 - 200 - 200 - 200 - 200 - 200 - 200 - 200 - 200 - 200 - 200 - 200 - 200		100		
	مرید مربوع می مربوع م مربوع مربوع می مربوع م	يستنقيها الألاليات ليتليقات منتقا أسليتهم	50		
	ليله فليعاود فيستحيين وسيني والمنتقي والمنصي والمنص	يستيقيل لاللك لكاكر مراكر			

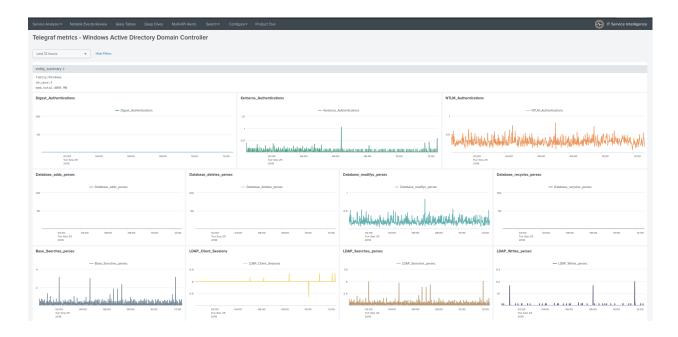


splunk>enterprise App: IT Service Intelligence *		Administrat	or 🔻 🕕 Messages 👻 Settings 👻 Activity 👻 Help 👻 🛛 Find 🔍
Service Analyzer   Notable Events Review Glass Tables Deep Dives			
Telegraf OS metrics			Edit Export •
entity_summary \$			
family:Windows	1 33	33.62	48 36
nb_cpus:2 mem_total:4096 MB	1.33 avg % cpu	avg % mem	48.36 % disk used
OVERVIEW CPU MEMORY DISKIO NET DISK UPTIME	FORECAST		
Inbound traffic (KBps)		Inbound packets	
	ionnection' 3 - isatap.eu-west-2.compute.internal	- AWS PV Network Device _0 - Local Area	Connection' 3 - isatap.eu-west-2.compute.internal
1,000		7500	
		5.000	
500		2.500	
		A.000	
02.00 04.00 06.00 08:00 10:00 12:00 Tue Sep 18 2018	14.00 16:00 18:00 20:00 22:00	02.00 04.00 06.00 08.00 10.00 12 Tue Sep 18 2018	0 14.00 16.00 18.00 20.00 22.00
Tue Sep 18 2018		Tue Sep 18 2018	
Outbound traffic (KBps)		Inbound packets	
- NE States Daries A Local Acc			
AWS PV Network Device _0 Local Area C	ionnection' 3 - isatap.eu-west-2.compute.internal	1000 - Local Area	Connection' 3 - isatap.eu-west-2.compute.internal
5		500	
new water which the second	man and the second and the second sec		
	14.00 16.00 18.00 20.00 22.00	02:00 04:00 06:00 08:00 10:00 12:	0 14.00 16.00 18.00 20.00 22.00
02:00 04:00 06:00 08:00 10:00 12:00 Tue Sep 18 2018	1600 1600 1800 2000 2200	02:00 04:00 06:00 08:00 10:00 12:1 Tue 5ep 18 2018	0 1400 1800 1800 2000 2200
splunk>enterprise App: IT Service Intelligence •		🔒 Administ	rator 🔻 🌖 Messages 👻 Settings 👻 Activity 👻 Help 👻 🛛 Find 🔍
			IT Service Intelligence
Telegraf OS metrics			Edit Export •
-			
entity_summary 0			
family:Windows nb_cpus:2	1.33 avg % cpu	33.62 avg % mem	48.36 % disk used
mem_total:4096 MB	avg % cpu	avg % mem	% disk used
	Q ± i O a minute ago		
OVERVIEW CPU MEMORY DISKIO NET DISK UPTIME			
OVERVIEW CPU MEMORY DISKIO NET DISK UPTIME	FORECAST		
used_percent			
48.375		- c.	
48.35			Γ
48.325			
10.347			
48.3			
48.275		لمر	
48,25	Α		
**** A			
48.225			
02:00 03:00 04:00 05:00 06:00	07.00 08:00 09:00 10:00 11:00 12:00	13:00 14:00 15:00 16:00 17:00 18:00	19:00 20:00 21:00 22:00 23:00 00:00 Wed Sep 19
02:00 03:00 04:00 05:00 06:00 Tut Sep 18 2018			Wed Sep 19

splunk>enterprise App: IT Service Intelligence  Service Analyzer Notable Events Review Glass Tables Deep Dives	Multi-KPI Alerts <b>Search▼</b> Configure▼ Product Tour	(B) Administrat	or • 🚯 Messages • Settings • Activity • Help • Find Q 🚱 IT Service Intelligence			
Telegraf OS metrics			Edit Export •			
entby_summary 1 faily/Nindows nh_cpusi2 mem_total:4096 MB	1.33 ang % qu	33.62 avg % mem	48.36 % disk used			
overview CPU memory diskid net disk UPTIME	FORECAST					
	018 10:30		57.0625			
1						
steps a s						
1						
Man Oct 2 Man Oct 16 Man Oct 30 Man Nev 13 Man Nev 27	Mon Dec 11 Mon Dec 25 Mon Jun 8 Mon Jun 22 Mon Feb 5 Mon Feb 19 Mon Mar 5 2018	i Mon Mar 19 Mon Apr 2 Mon Apr 16 Mon Apr 30 Mon May 14 Mon May 28 Mon Jun	11 Mon Jun 25 Mon Jul 9 Mon Jul 23 Mon Aug 6 Mon Aug 20 Mon Sep 3			
		= uptime				
splunk-enterprise App: IT Service Intelligence • Service Analyzer • Notable Events Review Glass Tables Deep Dives	Mark Mark Country Configurate Develop Tore	🔒 Administrati	or * 🌒 Messages * Settings * Activity * Help * [Find Q.] 🏟 (17 Service Intelligence			
Telegraf OS metrics	maankerzenta search - Connyche - Productiour		Edit Export •			
entity_summary \$		$\frown$				
family:Windows nb_cpus:2 mem_total:4096 MB	1.33 avg % cpu	33.62 avg % mem	48.36 % disk used			
overview CPU memory diskio net disk uptime	FORECAST					
Storage usage (GB) prediction for next 3 months	- storage_used	- total - predict				
35						
30						
25	25					
20						
5	_					
10						
3						
Mon Aug 20 Mon Aug 27 Mon Sep 3 Mon Sep 10 Mo 2018	n Sep 17 Mon Sep 24 Mon Oct 1 Mon Oct 8 Mon Oct 15	Mon Oct 22 Mon Oct 29 Mon Nov 5 Mon Nov 12 Mon	Nov 19 Mon Nov 26 Mon Dec 3 Mon Dec 10 Mon Dec 17			

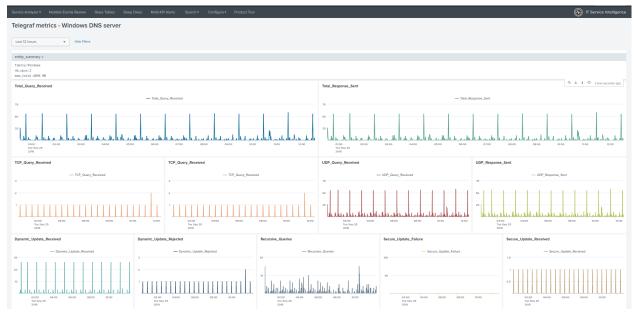
## 2.3.4 View for Windows Active Directory Domain Controller

The Health view automatically appears as "Telegraf Win AD-DC" deepdive drilldown link when entities are discovered:



## 2.3.5 View for Windows DNS

The Health view automatically appears as "Telegraf Win AD-DC" deepdive drilldown link when entities are discovered:



#### 2.3.6 View for Windows DFS

The Health view automatically appear as "Telegraf Win DFS-NS" and "Telegraf Win DFS-REP" deepdive drilldown links when entities are discovered:

## 2.3.7 View for Microsoft IIS/ASP.NET

The Health view automatically appears as "Telegraf IIS / ASP.NET" deepdive drilldown links when entities are discovered:

Service Analyzer   Notable Events Review Glass Tables Deep Dives Multi-	PI Alerts Search • Configure • Product Tour		() IT Service Intelligence
Telegraf metrics - Microsoft IIS / ASP.NET			
Last 12 hours   Hide Filters			
entity_summary \$			
family:Windows nb_cpus:2 mem_total:4096 MB			
IS Web service Bis Web Service Cache HTTP Service Request Queues ASPI	NET ASPINET Applications		
Current_Connections			
2	- Curren	t_Connections	
2			
23			
0100 0130 02:00 02:30 03:00 Tire Sep 25 2018	0330 0400 0430 0500 0530 0600 06	30 07.00 07.30 08.00 08.30 09.00 09.30	10:00 10:30 11:00 12:30 12:30
Get_Requests_persec	Post_Requests_persec	Connection_Attempts_persec	ISAPI_Extension_Requests_persec
- Get_Requests_persec	Post_Requests_persec	- Connection_Attempts_persec	
	30	03	20
2.000	20		30
02:00 04:00 05:00 05:00 10:00 12:00 Tite 5:e9:23 20:0	2200 0400 08:00 08:00 12:00 Turi 569:23 2018	82.00 04.00 06.00 88.00 10.00 12.00 Tur Feir 23 2018	02.00 94.00 05.00 06.00 10.00 12.00 Twie See 25 2019

## CHAPTER 3

## Troubleshoot:

## 3.1 Troubleshoot & FAQ

#### 3.1.1 Missing metrics for Windows memory

For Windows memory management, the default win\_mem inputs does not retrieve some of the metrics we need.

You need to activate the memory inputs. (which on Windows uses WMI collection):

```
[[inputs.mem]]
# no configuration
```

#### 3.1.2 Empy processes metrics (procstat)

In the linux views, the processes usage (both CPU and Memory) rely on the procstat inputs, which requires additional configuration depending on your context.

As for an example, the following configuration monitors all the processes owned by the "splunk" unix user:

```
[[inputs.procstat]]
   ## PID file to monitor process
#
#
   pid_file = "/var/run/nginx.pid"
#
  ## executable name (ie, pgrep <exe>)
  # exe = "nginx"
#
#
  ## pattern as argument for pgrep (ie, pgrep -f <pattern>)
#
  # pattern = "nginx"
#
  ## user as argument for pgrep (ie, pgrep -u <user>)
   user = "splunk"
```

## CHAPTER 4

## Versioniong and build history:

## 4.1 Release notes

#### 4.1.1 Version 1.0.7

• Change: JQuery simple XML dashboard update

#### 4.1.2 Version 1.0.6

- Feature: The App goes full dark mode
- Feature: Builtin out of the box alerts for basic generic system usage alerting (disk usage, cpu / memory saturation)
- Fix: Appinspect failure due to appLogo.png and appLogo\_2X.png size

#### 4.1.3 Version 1.0.5

• fix: improved nix family detection, avoid relying on kernel metrics which is Linux specific and might mot be available if not configured

#### 4.1.4 Version 1.0.4

• fix: Missing table\_bar.cssl.js

#### 4.1.5 Version 1.0.3

• fix: Missing objects and bad references

### 4.1.6 Version 1.0.2

• fix: hard coded references to savedsearches.conf

### 4.1.7 Version 1.0.1

• fix: hard coded references to index name in Overview

### 4.1.8 Version 1.0.0

• initial and first public release