# KPI Pipeline to Telegraf/Influx DB/Grafana

## **Table of Contents**

- 1. Overview
- 2. Setup/Configure Influx DB
- 3. Setup/Configure Telegraf
  - Service Input Plugin
  - Output Plugin
- 4. Configure and Run Tests on TAS (Test Administration Server)
  - HTTP Post (JSON format) "Favorites" measurements to URL
- 5. Exploring "Favorites" measurements on Influx DB
- 6. Set up Grafana
  - Add Influx DB as a data source on Grafana
  - Add Flux queries in Grafana query explorer & Build a Grafana dashboard

## 1. Overview

TAS provides a mechanism to HTTP POST a JSON format of the Favorite Measurements to a URL.

This documentation explains on how to publish "Favorites" KPIs into Grafana via Telegraf -> Influx DB -> Grafana Pipeline.



# 2. <u>Setup/Configure Influx DB</u>

Influx DB is a time series database designed to handle high write and query loads.

Once the Influx DB is running, navigate to http://localhost:8086 and user should be greeted with this page:



Clicking "Get Started" will take you to this page:

	Velcome — o Initial User Setup — o Complete	
	Setup Initial User	
	Username	
	admini	
	Password Confirm Password	
	An organization is a workspace for a group of users.	
	Initial Bucket Name 🖗	
R		
$\checkmark$		
$\sum D$		
X		
M		

Organizations are an umbrella under which your data and queries are organized in 2.0. A bucket is the equivalent of a database. You can choose whatever organization, bucket name, username, and password you like. Clicking "Continue" will take you to this page:

	Welcome ——— 🗸 Initial User Setup ——— o Compl	lete
	YOU ARE REACY TO GO! Your InfluxDB 2.0 has 1 organization, 1 user, and 1 bucket.	
	×	
	Let's start collecting data!	
Quick Start	Advanced	Configure Later
Timing is everything!	Whoa looks like you're an expert!	I've got this
This will set up local metric collection and allow you to explore the features of influxOB 2.0 quickly.		Jump into InfluxDB 2.0 and set up data collection when you're ready.

Influx DB tokens for authentication can be found here:

Ø	<b>influx</b> db	Load Data					
	admin -						
8	Load Data	Sources Buckets	Telegraf	Scrapers	Tokens		
	Sources Buckets Telegraf Scrapers	<b>Q</b> Filter Tokens			Sort by Descripti		
~	Tokens Data Explorer	admin's Token Created at: 2020-10-26T22:20:03					

# 3. Setup/Configure Telegraf

Telegraf is an agent written in Go for collecting, processing, aggregating, and writing metrics.

Telegraf is plugin-driven and has the concept of 4 distinct plugins. Out of which we are using the below 2 plugins for our use case:

- Input Plugins collect metrics from the system, services, or 3rd party APIs.
- Output Plugins write metrics to various destinations (Influx DB, MySQL etc.)

### <u>Service Input Plugin</u>

In our use case, TAS is providing a mechanism to HTTP POST a JSON format of "Favorites" measurements to a URL. So, Telegraf is configured to use http\_listener\_v2 service input plugin.

The configuration changes can be set in the Telegraf configuration file as below:

```
# Generic HTTP write listener
[[inputs.http_listener_v2]]
  ## Address and port to host HTTP listener on
  service_address = ":8080"
  ## Path to listen to.
  path = "/telegraf"
  ## HTTP methods to accept.
  methods = ["POST", "PUT"]
```

Data format to consume is JSON format in our use case scenario. 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\_FORM ATS\_INPUT.md



Enable HTTP Basic Authentication by adding the below configuration in the configuration file. Also, the user can enable HTTPS by adding Service certificate and Key.

```
## Set one or more allowed client CA certificate file names to
## enable mutually authenticated TLS connections
# tls_allowed_cacerts = ["/etc/telegraf/clientca.pem"]
## Add service certificate and key
#tls_cert = "/usr/local/etc/telegraf/localhost.cer.pem"
#tls_key = "/usr/local/etc/telegraf/localhost.key.pem"
## Optional username and password to accept for HTTP basic authentication.
## You probably want to make sure you have TLS configured above for this.
basic_username = "admin"
basic_password = "password"
```

#### Output Plugin

To start pushing Telegraf data to the Metrics platform, user just need to add an Influx output plugin for writing metrics to a specific bucket in Influx DB as described below:

### 4. <u>Configure and Run Tests on TAS (Test Administration</u> <u>Server)</u>

TAS provides a mechanism to HTTP Post (JSON format) "Favorites" measurements to URL (Telegraf/Influx DB/Grafana)

By clicking "HTTP POST" on Test Session, User can configure below HTTP POST Output parameters which sends metrics on each interval to Influx DB via Telegraf:

 HTTP POST URL - The service address, port and path to host HTTP listener, which is the same configuration in Telegraf.

Also, it allows multiple users to post different URLs based on Telegraf configuration as below example:

#User 1 - <u>http://10.39.132.97:8080/telegraf</u> #User 2 - <u>http://10.39.132.97:8090/telegraf</u>

 JSON\_target\_ - This is a measurement name user can input which is conceptually similar to a table in Influx DB. Each user can have their own bucket and separate table sharing across the same Influx DB instance

Note- Target must be 1 to 32 characters only letters, numbers, \_, -, or ...

 BASIC AUTH Username – HTTP Basic Authentication Username (configured in Telegraf)

Note- Username must be 0 to 32 characters only letters, numbers, \_, -, or ..

 BASIC AUTH Password – HTTP Basic Authentication Password (configured in Telegraf)

Note- If username is filled out: Password must be 1 to 32 characters ASCII.

🔎 Test Session - 2: sms/Simple Data Test 🛛 🖉									
<u>File Settings Logs R</u> eports									
	Test Complete.								
Overall State: COMPLETE									
Session Builder Automation Control Pass/Fail Port Capture Reports Favorites Info Logs									
B Co HTTP POST									
Measurement	892	893	894	895	Current				
Elapsed Time	3 Hr(s) 43 Min								
Actual Time	11/30 20:40:36	11/30 20:40:51	11/30 20:41:06	11/30 20:41:23	11/30 20:41:22				
L5-7 Client Advanced::RTD Command Message Count	401250	401700	402150	402270	402270				
L5-7 Client Advanced::Command Messages Sent	535000	535600	536200	536360	536360				
L5-7 Client Advanced::Command Messages Received	601875	602550	603225	603405	603405				
L3 Client:Total Packets Sent	735625	736450	737275	737495	737495				
L3 Client::Total Packets Received	735625	736450	737275	737495	737495				
L3 Client::Total Packets Sent         735025         736450         737275         737495         737495           L3 Client::Total Packets Received         735625         736450         737275         737495         737495           Test Summary::Total Data Sent + Re									
Moort Stop Run				Sav	Close				

After configuring HTTP POST Output, click "Run" to run the test and send measurements to Telegraf.

The measurements get converted to JSON format in the background as below:

📈 Test Session - 3: sms/Simple Data Test					۰ď 🛛	http://10.39.132.97:8080/telegrafsending>
<u>File Settings Logs Reports</u>						{
Running indef	initely, stop tes	t manually				"_target_": "table1",
Overall State: RUINNING Session Builder Automation Control Pass/Fail Port Capture Reports Favorites Info Logs						"_test_": "Simple Data Test", "_run_": "2020\/11\/10 13:17:00_RID-3", "_interval_": 12,
	al 7 of 7 🐌	Go to i	nterval	Go HTTP	POST	"_elapsedTime_": 180, "_actualTime_": 1605036006363,
Measurement	4	5	6	7	Current	"_iteration_": 1, "IS-7 Client Makaneed: PTD Command Message Count": 5250
Elapsed Time	1 Min(s) 0 S	1 Min(s) 15	1 Min(s) 30	1 Min(s) 45	1 Min(s) 55	US-7 Citeri LavancedKib Command Message Count . 329
Actual Time	11/10 14:18:	11/10 14:18:	11/10 14:18	. 11/10 14:18:	11/10 14:19:	LS-7 CLEENLAWARDEN, COMMAND MESSages Bent, 7000,
L5-7 Client/Advanced:RTD Command Message Count	1650	2100	2550	3000	3000	L3-7 Citeri Advanced: Command Hessages Received: 7875,
L5-7 ClientlAdvanced Command Messages Sent	2200	2800	3400	4000	4000	"L3 Client::Total Packets Sent": 9625,
L5-7 ClientlAdvanced: Command Messages Received	2475	3150	3825	4500	4500	"L3 Client::Total Packets Received": 9625,
L3 Client: Total Packets Sent	3025	3850	4675	5500	5500	"Test Summary::Total Data Sent + Received Packets\/Sec (P-I) ": 219.99266715552295
L3 Client: Total Packets Received	3025	3850	4675	5500	5500	"L4 Client::Socket Payload Bytes Sent": 333375,
Test Summary: Total Data Sent + Received Packets/Sec (P-I)	220	220 007	220	220	0	"L4 Client::Socket Payload Bytes Received": 98000,
L4 Client: Socket Payload Bytes Sent	104775	133350	161925	190500	190500	"L4 Client::Socket Connect Count": 875
L4 Client: Socket Payload Bytes Received	30800	39200	47600	56000	56000	}
L4 Client: Socket Connect Count	275	350	425	500	500	START-RUN-SAFE
		1	140			SENDING
						RESPONSE: 204:No Content
C Abort Stop Run				Sav	e Close	Sent, response: END-RUN-SAFE

Start Telegraf to verify the live measurements being written to Influx DB in line protocol format.



## 5. Exploring "Favorites" measurements on Influx DB

As the user can see, the "Favorites" measurements are being written to Influx DB via Telegraf.

In the Influx DB Data explorer, the first column is Bucket which is equivalent to database. The second column is measurement conceptually similar to table. The third column shows fields where metrics are stored. And the other columns are tag values where strings are stored.

∿	Data Explorer		Create a guery. Have fun!							
⊞	Dashboards									
<b>.</b>	Tasks									
Δ	Alerts	Query 1 +			View Raw Data 💽 👃 C.	SV II 🔻 💭 Pastóh	<ul> <li>Script Editor</li> <li>Submit</li> </ul>			
-	Alert History		Filter 👻	Filter 💌 🗙	Filter 🔹 🗙	Filter 👻	WINDOW PERIOD			
a			_measurement - 1	_field 🔻 6		_test_ 🔹	auto (1m)			
	Settings									
	Variables		table1	L3 Client::Total Packets	2020/11/30 21:27:18RID-3	Simple Data Test	AGGREGATE FUNCTION			
	Templates bucket1		user1	L3 Client::Total Packets _			Custom Auto			
	Labels	user1		L5-7 Client Advanced::Com			mean			
			L5-7 Clien							
		+ Create Bucket		L5-7 Client Advanced::RTD						
				Test Summary::Total Data						

# 6. Setup Grafana

**Grafana** is a multi-platform open source analytics and interactive visualization web application. Each user can have their own pre-configured Grafana dashboard which periodically refreshes (configured to 10 secs) showing live measurements through visualizations.

### Add Influx DB as a data source on Grafana

In the left menu, click on the Configuration > Data sources section.



In the next window, click on "Add datasource".



In the datasource selection panel, choose Influx DB as a datasource.



Here is the configuration you have to match to configure Influx DB on Grafana.

Data So         Type: InfluxD         III         Settings	urces / InfluxDB	3				
Name :	InfluxDB	63	Default			
Query Language		~				
Support for flux in G Please report any issue <u>https://github.com/gra</u>	Support for flux in Grafana is currently in beta Please report any issues to: <u>https://github.com/grafana/grafana/issues</u>					
Connection						
URL :	http://localhost:8086					
Organization	Organization landslide					
Token	configured		Reset			
Default Bucket	user1					
Min time interval 🔅	15s					

Click on "Save and Test", and make sure that you are not getting any errors.



### Add Flux queries in Grafana query explorer & Build a Grafana dashboard

Click on "Create dashboard" and click on "Add new Panel"



Select Influx DB in the Query explorer and write Flux QL to create a Grafana dashboard

	Fill F	t Exact	<ul> <li>Last 15 minutes</li> </ul>	~ Q	යි <mark>10s</mark> ∽			
liert								
50 K								
40 K								
30 K 20 K								
10K								
22:27 22:28 22:29 22:30 22:31 22:32 22:33 22:34 22:35	22:36	22:37	22:38 22:39	22:40	22:41			
E3 Client: Total Packets Received E3 Client: Total Packets Sent E L5-7 Client(Advanced::Command Messages Received E L5-7 Client(Advanced::RTD Command Message Count E Test Summary::Total Data Sent + Received Packets/Sec (P-I)	Client Advanced	Command Me	ssages Sent					
E Query 1 Alert 0								
③ InfluxDB ✓ ③ → Query options MD = 5000 Interval = 15s				Query ir	nspector			
				00				
<pre>1 from(bucket: "user1") 2  &gt; range(start: v.timeRangeStart, stop: v.timeRangeStop) 3  &gt; filter(fn: (r) =&gt; r["_measurement"] == "table1" 4 and rfield != "_elapsedTime_" and rfield != "_interval_" and rfield != "_iteration_") 5  &gt; yield(name: "mean") 6</pre>								

And configure the time period (say Last 1 hour) and interval (say 10 seconds) on the top right corner to refresh periodically and display live measurements every 10 seconds on the Grafana dashboard.



Live "Favorites" measurements on Grafana dashboard:

