Omnissiah architecture

Purpose of the Omnissiah

This platform is designed to collect information about network devices, identify them (determine type, manufacturer, model and role) and automatically manage the configuration of the Zabbix monitoring platform. The program architecture does not limit the list of possible sources of information about devices. The output of the results can also be done not only in the form of a configuration for Zabbix.

Algorithm

To obtain information about devices, scanning using the nmap program and accessing various APIs are used. The received information is stored in a relational database (Mariadb or PostgreSQL). Its processing is carried out mainly using SQL queries called from many small programs launched sequentially one after another from one batch shell script. One such launch of all programs is called a cycle. After completing one cycle, the next cycle begins automatically.

Information about the network(s) is taken from the Netbox program. But its presence or access to the Netbox API is not required. It is enough to manually fill in some tables in the database.

SQL queries that are executed by programs for data processing are stored in the database itself and can be easily changed or supplemented. Each program is responsible for one action, such as working with a specific API or processing records from a single source. All programs and database tables are divided into layers. The layer name is used as a prefix to script file or table names. Programs in the first layer, named raw, obtain data using network scanning or API access. Programs of all subsequent layers usually take records from the tables of previous layers, process them using SQL queries and write them to the tables of their layer. The last zbx layer, after preparing the tables of its layer, synchronizes them with the current Zabbix config.

Omnissiah layers

#	Layer	*Purpose*	*Source layers*	*Layer programs*	*Database tables*
1	raw	initial data		raw_activaire, raw_enplug, raw_mac, raw_map, raw_mist, raw_netbox, raw_ruckussz, raw_scan, raw_snmp	raw_*
2	info	processing of source data not about devices	raw	info_mac, info_netbox	info_*
3	ref	various reference books	info or manually	ref_netbox	ref_*
4	src	processing of initial data about devices	raw	src_activaire, src_addr, src_enplug, src_mist, src_ruckussz, src_scan, src_snmp	src_*
5	nnml	neural networks	info, src	nnml_label, nnml_predict, nnml_prepare	nnml_*
6	shot	network image in the current cycle	info, src, nnml	shot_activaire, shot_enplug, shot_host, shot_mist, shot_nnml, shot_router, shot_ruckussz, shot_wap	shot_*
7	main	network image taking into account previous cycles	shot	main_addr, main_host	main_*

8	zbx	configuration synchronization with Zabbix	main	zbx_main2zbx, zbx_omni2zbx, zbx_zbx2omni	zbx_*
9	hist	database dumps		hist_dump	hist_*
10	code	SQL queries for processing tables	-		code_*
11	cfg	configs	-	omni_config, omni_const, omni_unpwd	cfg_*
12	tmp	temporary tables	-		tmp_*
13	log	SQL query execution logs	-		log_*

Omnissiah layers table

Any program uses layer tables cfg and code (for executing SQL queries), log (for writing logs), tmp (for temporary tables) and ref (for accessing various directories). Programs of other layers convert the tables of the source layer into tables of their layer. The raw layer does not have a source layer because all the data in this layer is from sources external to the omnissiah.

Layer interaction diagram

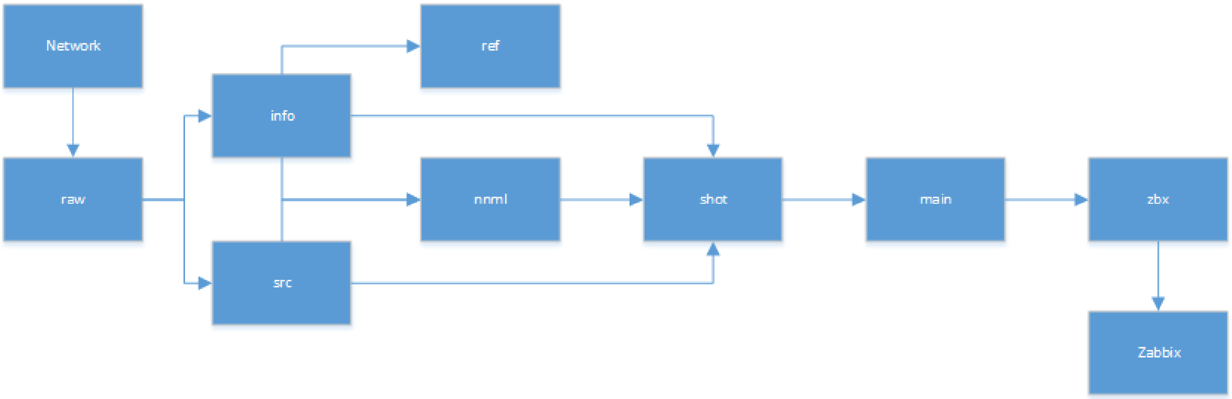

Diagram of interaction of Omnissiah layers