

EVENT-DRIVEN AUTOMATION AND CHATOPS ON IHEP COMPUTING CLUSTER

A. Kotliar^a, V. Kotliar^b

Institute for High Energy Physics named by A.A. Logunov of National Research Center "Kurchatov Institute", Nauki Square 1, Protvino, Moscow region, Russia, 142281

E-mail: ^a Anna.Kotliar@ihep.ru, ^b Viktor.Kotliar@ihep.ru

Dealing with cluster systems you have multiple ordinary situations which can be solved using automation tools. StackStorm is a quite good event-driven system which helps to manage typical problems and to communicate with cluster via ChatOps extension. Just write a rule for such event and it will be triggered and solved. In the presented work will be shown an example of a real event-driven system on IHEP computing cluster which use Nagios, Check_MK, StackStorm, Mattermost for routine work automation as a part of multicomponent cluster management system.

Keywords: Event-driven automation, ChatOps, StackStorm, cluster management system

© 2018 Anna Kotliar, Victor Kotliar

1. Introduction

To solve computation problems in the distributed environment distributed systems are build. Such systems have many elements from the distributed computing theory which allow to work smoothly and to perform their operations. Usually all such elements are gathered together in the computing clusters that have one common goal and represent one complex multicomponent system. There are many ordinary situations which should be solved in such system and automation tools play vital role here. StackStorm is quiet a good generic purpose event-driven system which is used on IHEP cluster [1] and helps to manage typical problems as automation tool. When a failure happens it troubleshoots it, fixes known problems or escalates them to system administrators if needed. It also allows to use a brand new ChatOps technology for cluster management.

2. Event-driven system overview

StackStorm consists of four main logical parts like sensors, triggers, rules and actions [2]. Sensors are python plugins for integration that receives or watches for events. When an event from external systems occurs and is processed by a sensor than an internal trigger emitted into the system. From event-driven system's view triggers are representation of external events. At IHEP it is already used timers, webhooks and integrations triggers. It is easy to define a new trigger type just by writing a sensor plugin. Rules map triggers to simple actions or to tasks as a set of actions (workflows) by applying matching criteria and by mapping trigger payload to action inputs. Actions are StackStorm outbound integrations. There are already implemented many generic actions like ssh connect, crontab, sending e-mail, REST calls and several integrations with systems like OpenStack, Docker, Puppet Actions are either python plugins, or any scripts, consumed into StackStorm system by adding a few lines of metadata. The great features of the system is that actions can be invoked directly by user via CLI or API, or used and called as a part of rules and workflows. To deploy its self-build content packs are used. They simplify the management and developing of StackStorm pluggable content by grouping integrations (triggers and actions) and automations (rules and workflows). IHEP StackStorm pack is managed by GitLab system. To audit logs of action executions, manual or automated, they are recorded and stored with full details of triggering context and execution results into a NoSQL DB. This information also captured in audit logs for integrating with external logging and analytical tools like: LogStash, Splunk, syslog. At IHEP mostly rules and actions are used for daily operations which are deployed in one pack.

2.1. Events, triggers, rules and actions

To use event-driven functionality first of all we need to have one or many systems which will generate events. At IHEP cluster Check_MK-Nagios monitoring framework is used. To bind this system with Stacktorm two steps are needed to be done:

- On Stackstorm server it needs to be installed Check_MK integration pack;
- On Check_MK server it needs to be installed StackStorm plugin.

After installation steps in Check_MK system some relevant events need to be set.

Actions	Type	Plugin	Bulk	Description	Contacts	Conditions
		stackstorm.py		Notification to stackstorm for WNs for CRITICAL&UKNOWN	• users: nagiosadmin	2 conditions

Figure 1. Check_MK event for StackStorm

On the figure 1 is shown the event (notification rule in Check_MK terms) which is triggered on CRITICAL or UNKNOWN states for the cluster computing nodes. When such events occurred the information about them is sent to the event-driven system to handle it. On the StackStorm side a special trigger installed by the integration pack is used to react to the problem. Now to really react to the event or make an action in StackStorm special rule was created for a specific problem. To bind specific problem and action a special rule parameter "Criteria" is used. Such rule-action schema is

used to bind many actions for one event depending on event parameters (one-to-many relationship see figure 2).

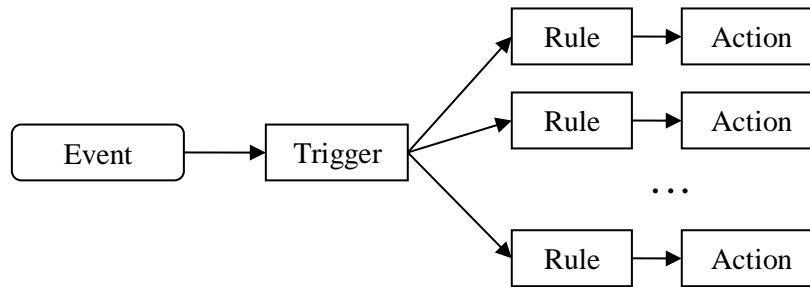


Figure 2. One-to-many relationship for event-action in StackStorm

As an example a real production trigger-rule-action set is shown on figure 3. Check_MK event handler checks against a rule NodesCheckMkCheckOrphans for three criterias: CRITICAL event, host name started with “wn0” and specific service “check orphans” which is triggered by the event. After all conditions are met an action “core.remote” is performing. In the example it just kills all orphaned processes on the computing node.

Rule

```

name: "NodesCheckMkCheckOrphans"
description: "Trigger action for check orphans check for WNs. Kill only atlas jobs"
pack: "st2ihep"
enabled: true
  
```

```

trigger:
  type: "check_mk.event_handler"
  
```

```

criteria:
  trigger.service.state:
    pattern: "CRITICAL"
    type: "equals"
  trigger.host.name:
    type: "regex"
    pattern: "^wn0"
  trigger.service.desc:
    pattern: "check orphans"
    type: "equals"
  
```

```

action:
  
```

```

  ref: "core.remote"
  parameters:
    
```

```

    hosts: puppet
    username: root
    cmd: pdsh -w {{trigger.host.name}} "/usr/share/check-mk-agent/plugins/check_unparent |
    egrep \"(at|sgmal)\" | egrep -o \"pid=[[digit:]]+\" | awk -F = '{print \\$2;}' | xargs
    kill "
  
```

```

st2 trigger get check_mk.event_handler
  
```

Trigger

Property	Value
id	5ac5dff70c4e7f15a11b9129
ref	check_mk.event_handler
pack	check_mk
name	event_handler
description	Trigger type for check_mk events
parameters_schema	
payload_schema	
tags	
uid	trigger_type:check_mk:event_handler

Figure 3. Trigger, rule, action for Check_MK event

2.2. Complex actions

Most cluster management processes consist of multiple distinct interconnected steps that need be executed in a particular order in a distributed cluster environment. A system administrator can describe such process as a set of tasks and their transitions. After that, it is possible to upload such description into the StackStorm mistral service, which will take care of state management, correct execution order, parallelism, synchronization and high availability. In mistral terminology such set of tasks and relations between them is called a workflow [3].

At IHEP computing cluster a kernel upgrade on the computing nodes must be applied periodically. Such upgrade consists of many steps which involve many different components of the cluster like batch system, monitoring system, working node itself. It was always a big headache for system administrators because such upgrade should be done as effectively as possible with minimum interruption to the service operations. To make such task more effective and simple it was developed a StackStorm mistral workflow in the distributed cluster environment. From event-driven system a workflow look like the action which consists of the set of other actions and it is possible to use such

complex action everywhere where it is possible to use simple actions. To trigger kernel upgrade action at IHEP a system console is used. One thing that should be taken in to account here is the time for workflow to be completed. That could be from few hours to several days and all that time internal StackStorm authentication token should be valid.

The procedure itself is very simple and consists of few steps:

1. A system administrator run kernel upgrade workflow through StackStorm API;
2. First step of the workflow is to put computing node offline in the batch system;
3. As soon as node will be free from computation jobs an upgrade should be applied and node need to be rebooted;
4. After node reboot it's status should be checked in the monitoring system;
5. As soon as all monitoring metrics for the node become normal the workflow put node back to the production and inform the administrator.

Looks like all these steps could be done without any special system in bunch of scripts but here is the real power of StackStorm appears. All steps in the workflow task are relatively synchronized and fully logged. It is easy to find in case of the problem where and why the workflow failed and the system administrator do not need to think how to save the states for the workflow operations.

3. ChatOps technology

ChatOps is an operation paradigm where work that is already happening in the background today is brought into a common chatroom. Such paradigm allows to fully open IT daily operations and unify the communication about what should get done with action history of work being done [4]. Operation chat is the place where any worth notification could be done, any question and conversation between sysadmins could be logged, it could speed up a new team member teaching, and it could be used as a knowledge, tips and tricks base for the operation issues. As soon as there are all software platform supported it is possible to use it from smart phones, Linux or Windows personal computers, and even from any web-browser and be in touch for operation any time from any place in a convenient way.

To implement ChatOps for IHEP IT operations Mattermost and Hubot software were chosen (Figure 4). Mattermost is a messaging workspace which has a possibility to use many different integrations. One of them is Hubot. Hubot is an open source bot written in CoffeeScript on Node.js which can do operation things that are needed. It is possible to post image to chat, to translate a text, to give a weather forecast and many others. In case of IHEP it is used to communicate with StackStorm and post answers to the common chat from the server.

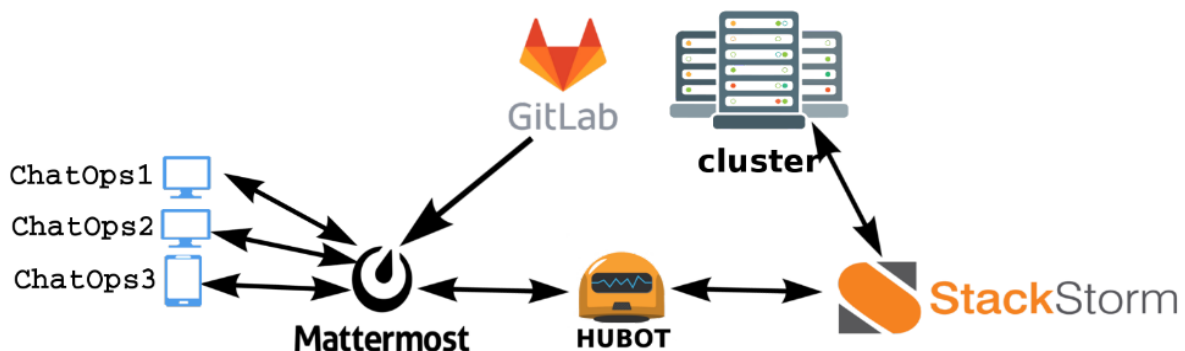


Figure 4. ChatOps system at IHEP

By placing StackStorm system for Hubot as a cluster manager software on IHEP cluster the big set of features become available from a chat room. It is possible to run remotely a command, list executions, list and install packs, re-run executions. To handle the big output for Stackstorm messages in Hubot it was modified to send a big messages to the chat room split by frames. The last but not least

piece in the system is GitLab service where cluster configuration is stored. After any modification of the configuration notification messages reflects changes at the operation chat.

4. Conclusion

A new event-driven automation system for IHEP computing cluster was presented in the described work. This system is based on open source StackStorm software. Some of the already used StackStorm features are described. StackStorm allows to automate many actions for such complex system as a cluster for distributed computing and it becomes a core part of the multicomponent cluster management system at IHEP. With growing knowledge base for solving operation issues and implementing them as actions or workflows it is possible to implement basic principles of autonomic computing for self-management systems that will allow to increase effectiveness for the computing cluster. To clue a new operation tool with system administrators team a ChatOps paradigm is used for day-to-day operations where all background operations are brought into a common chatroom.

For further developing of the event-driven system all manual actions are need to be programmed in a way that they could be used in StackStorm system. It is also planned to use this system as a part of anomaly detection system on IHEP computing cluster for self-healing. Experience in operating with event-driven systems could be used not only for IT operations but also for experimental sets for physics in the Institute.

References

- [1] Viktor Kotliar. IHEP cluster for Grid and distributed computing// CEUR Workshop Proceedings. February 2017: Vol. 1787.- pp. 312-316
- [2] StackStorm documentation [StackStorm overview]. Available at: <https://docs.stackstorm.com/overview.html>. (accessed 24.09.2018)
- [3] OpenStack documentation [Mistral overview]. Available at <https://docs.openstack.org/mistral/latest/overview.html> (accessed 24.09.2018)
- [4] StackStorm documentation [What is ChatOps]. Available at <https://docs.stackstorm.com/chatops/chatops.html> (accessed 24.09.2018)