

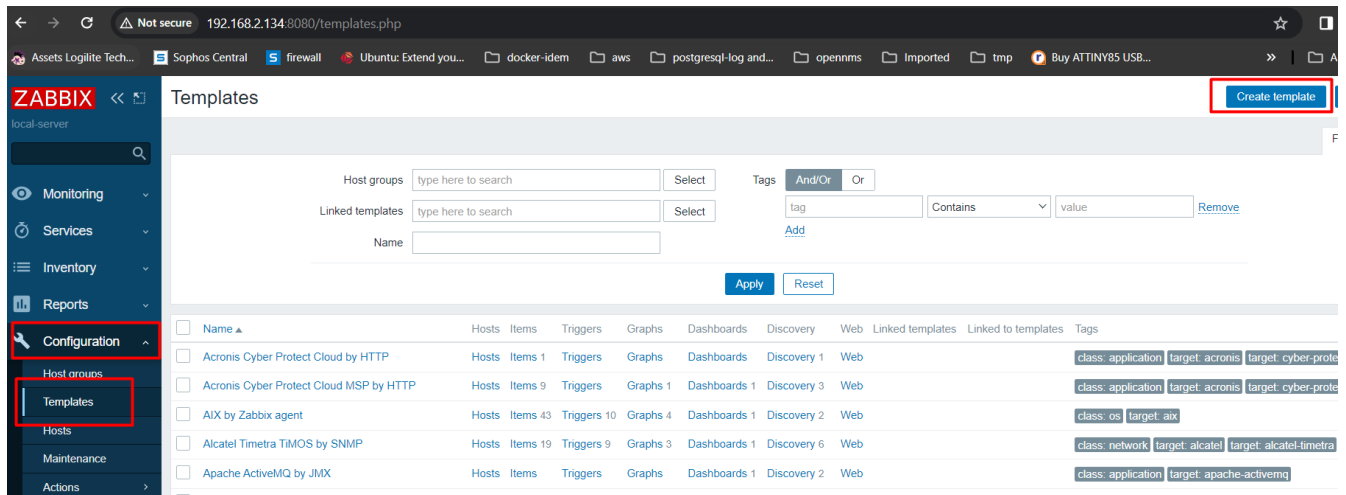
Zabbix Template Design and Alert

Table of Contents

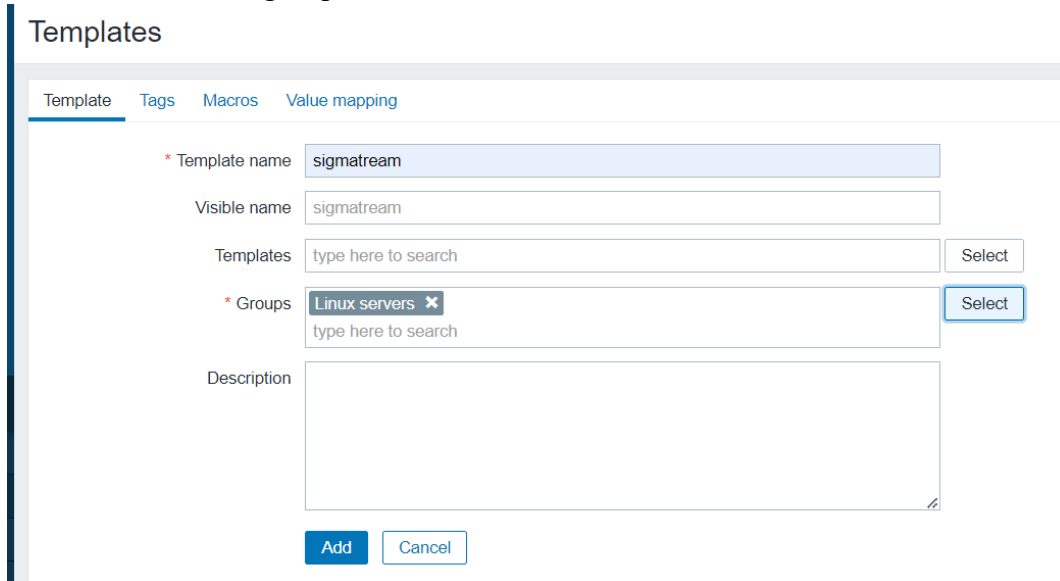
1. Create template	2
2.Custom service monitoring (by ITEM).....	4
3.Configure Trigger.....	7
4.Configure Alert on Ms team.....	12
5.Action on trigger (Run script for restart service)	18

1. Create template

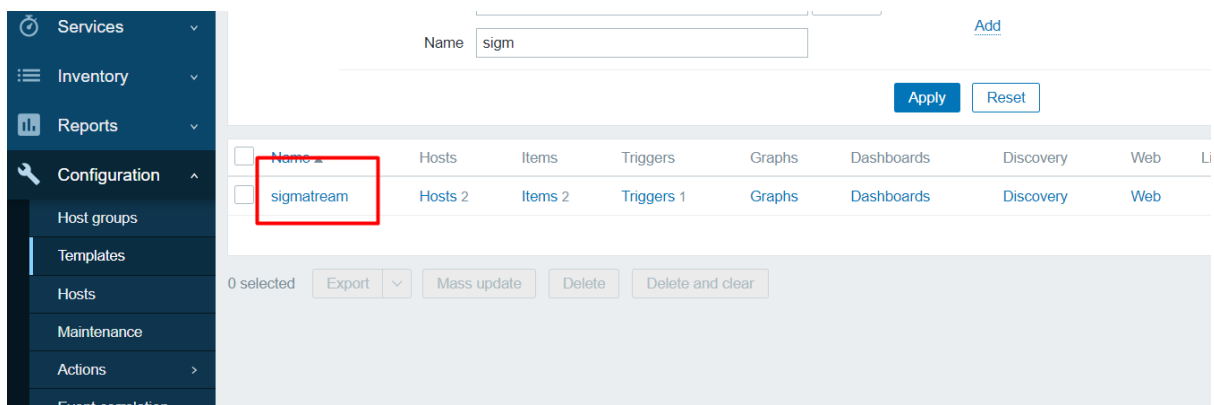
Configuration > template > create template



Add name and host group then click on Add



Search and click on template name to add ITEM and Trigger.



To add item click on Item and same for trigger.

ITEM: An individual metric that used for collecting data

Trigger: logical expressions that "evaluate" data gathered by items and represent the current system state

The screenshot displays the Zabbix web interface for configuring a template. The left sidebar shows the navigation menu with 'Configuration' expanded and 'Templates' selected. The main content area shows the configuration for the 'sigmatream' template. The 'Items' and 'Triggers' tabs are highlighted with red boxes. The 'Items' tab is active, showing the following fields:

- * Template name:
- Visible name:
- Templates:
- * Groups: (with a dropdown arrow)
- Description:

At the bottom of the form, there are several action buttons: **Update**, **Clone**, **Full clone**, **Delete**, **Delete and clear**, and **Cancel**.

2. Custom service monitoring (by ITEM).

To collect data for service we create ITEM under created template.

The screenshot shows the Zabbix web interface for configuring items. The 'Host groups' dropdown is set to 'local-server'. The 'Templates' dropdown is set to 'sigmatream'. The 'Name' field is empty. The 'Key' field is empty. The 'Value mapping' dropdown is set to 'type here to search'. The 'Type' dropdown is set to 'all'. The 'Type of information' dropdown is set to 'all'. The 'History' field is empty. The 'Trends' field is empty. The 'Update interval' field is empty. The 'Status' dropdown is set to 'all'. The 'Triggers' dropdown is set to 'all'. The 'Inherited' dropdown is set to 'all'. The 'Create item' button is highlighted in red. The 'Subfilter affects only filtered data' section shows 'TYPE OF INFORMATION' with 'Numeric (float) 1' and 'Numeric (unsigned) 1'. The 'WITH TRIGGERS' section shows 'Without triggers 1' and 'With triggers 1'. The table below shows two items: 'test' and 'YH Server - status'. The 'YH Server - status' item is highlighted in red.

Here below config create item named **YH Server – status** which check every min for service status.

Let's discuss parameter

Key: provide which kind of operation you need to perform here for service

Use: `system.run["systemctl status yhserver.service | grep -q running; echo $?"]`

Note: don't copy past above configuration it may not work you need to select key for that click on select button then click on system.run as per below.

The screenshot shows the Zabbix web interface for configuring the 'Key' field. The 'Key' field contains the command `system.run["systemctl status yhserver.service | grep -q running; echo $?"]`. The 'Select' button is highlighted in red. Below the 'Key' field, the 'Standard items' section shows a list of predefined keys. The 'system.run[command,<mode>]' key is highlighted in red.

Now remove content which present between [] and add **"systemctl status yhserver.service | grep -q running; echo \$?"**

Now you ITEM look like

* Key `system.run["systemctl status yhserver.service | grep -q running; echo $?"]`

Type of information: select as per o/p of key. After perform action as per key if we get o/p in Numeric/float then select numeric float.

Ex. For service status based on exit code use numeric(unsigned) and if we have data about cpu and memory then use Numeric(float)

Type of information	Numeric (float) ▼
Units	Numeric (unsigned) Numeric (float)
* Update interval	Character
Custom intervals	Log Text

Update intervals: Specify After how many time agent need to run key.

Now Set other parameter and tag as per need click on add

Item

Tags 1

Preprocessing

* Name

YH Server - status

Type

Zabbix agent ▼

* Key

system.run["systemctl status yhserver.service | grep -q running; echo \$?"]

Select

Type of information

Numeric (unsigned) ▼ ⓘ

Units

* Update interval

1m

Custom intervals

Type	Interval	Period	Action
Flexible	Scheduling	50s	1-7,00:00-24:00

Add

Remove

* History storage period

Do not keep history

Storage period

90d

* Trend storage period

Do not keep trends

Storage period

365d

Value mapping

type here to search

Select

Populates host inventory field

-None- ▼

Description

Enabled

☒

Add

Test

Cancel

Example of Keys for task

Key to monitor CPU

```
system.run["/etc/zabbix/resource.sh | awk 'NR>7 && NF -1 {print $9}']"
```

Key to monitor RAM

```
system.run["/etc/zabbix/resource.sh | awk 'NR>7 && NF -1 {print $10}']"
```

Create script under /etc/zabbix/ with name resource.sh

```
cd /etc/zabbix/
```

```
nano resource.sh
```

add below content into file

```
#!/bin/bash
```

```
top -n 1 -b -p $(pgrep -f yellowhammer-server)
```

```
#!/bin/bash
top -n 1 -b -p $(pgrep -f yellowhammer-server)
```

Note: Change **yellowhammer-server** with service name

Change permission

```
sudo chmod +x /etc/zabbix/resource.sh
```

Monitor via curl request

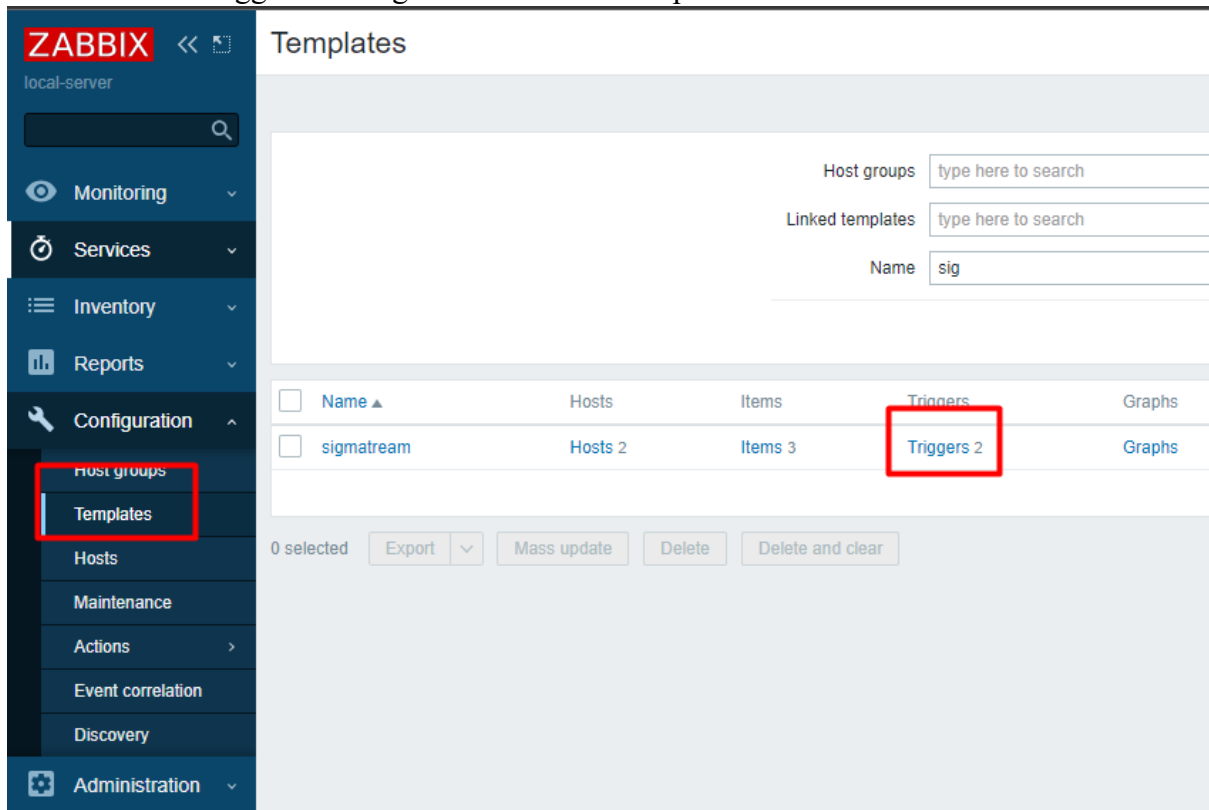
```
curl -s -k --head https://192.168.2.134:8443 | grep '200 OK' > /dev/null; echo $?
```

```
system.run["curl -s -k --head https://192.168.2.134:8443 | grep '200 OK' > /dev/null;
echo $?"]
```

Item	Tags	Preprocessing								
<div><div>* Name</div><div>CURL</div></div> <div><div>Type</div><div>Zabbix agent</div></div> <div><div>* Key</div><div>system.run["curl -s -k --head https://192.168.2.134:8443 grep '200 OK' > /dev/null; echo \$?"]</div><div>Select</div></div> <div><div>Type of information</div><div>Numeric (unsigned)</div><div>i</div></div> <div><div>Units</div><div></div></div> <div><div>* Update interval</div><div>1m</div></div> <div><div>Custom intervals</div><table><thead><tr><th>Type</th><th>Interval</th><th>Period</th><th>Action</th></tr></thead><tbody><tr><td>Flexible</td><td>Scheduling</td><td>50s</td><td>1-7,00:00-24:00</td></tr></tbody></table><div>Add</div><div>Remove</div></div> <div><div>* History storage period</div><div>Do not keep history</div><div>Storage period</div><div>90d</div></div> <div><div>* Trend storage period</div><div>Do not keep trends</div><div>Storage period</div><div>365d</div></div> <div><div>Value mapping</div><div>type here to search</div><div>Select</div></div> <div><div>Populates host inventory field</div><div>-None-</div></div> <div><div>Description</div><div></div></div>			Type	Interval	Period	Action	Flexible	Scheduling	50s	1-7,00:00-24:00
Type	Interval	Period	Action							
Flexible	Scheduling	50s	1-7,00:00-24:00							

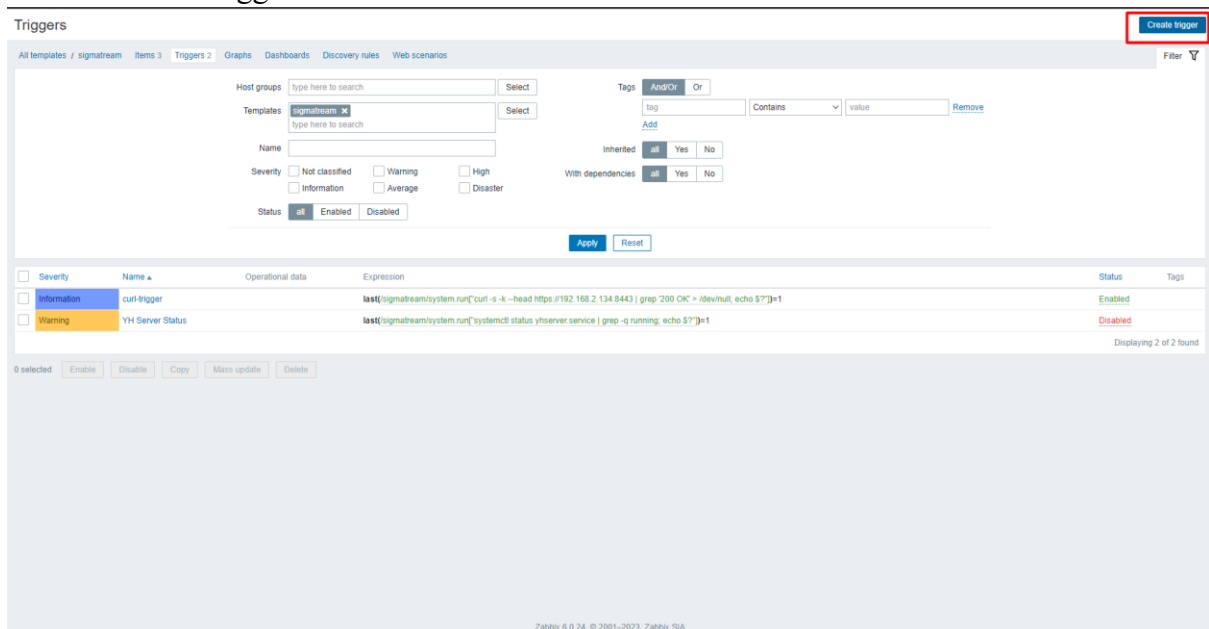
3.Configure Trigger

Now we create trigger if item give value as not accepted.



The screenshot shows the Zabbix web interface. On the left is a sidebar with navigation links: Monitoring, Services, Inventory, Reports, Configuration, Host groups, Templates, Hosts, Maintenance, Actions, Event correlation, Discovery, and Administration. The 'Configuration' section is expanded, and 'Templates' is highlighted with a red box. The main area is titled 'Templates' and contains search fields for Host groups, Linked templates, and Name (with 'sig' entered). Below these is a table with columns: Name, Hosts, Items, Triggers, and Graphs. The table lists a template named 'sigmatream' with 2 hosts, 3 items, and 2 triggers. The 'Triggers 2' link is highlighted with a red box. At the bottom of the table are buttons for '0 selected', 'Export', 'Mass update', 'Delete', and 'Delete and clear'.

Click on create trigger



The screenshot shows the Zabbix 'Triggers' page. The breadcrumb trail is 'All templates / sigmatream / Items 3 / Triggers 2'. A 'Create trigger' button is highlighted with a red box in the top right. The page contains a form for creating a new trigger with fields for Host groups, Templates, Name, Severity (Not classified, Warning, High, Information, Average, Disaster), Status (Enabled, Disabled), and Tags. There are also checkboxes for 'Inherited' and 'With dependencies'. Below the form is a table of existing triggers. The table has columns: Severity, Name, Operational data, Expression, Status, and Tags. It lists two triggers: 'curl-trigger' (Information severity, Enabled status) and 'YH Server Status' (Warning severity, Disabled status). The 'YH Server Status' trigger's expression is highlighted with a red box. At the bottom of the table are buttons for '0 selected', 'Enable', 'Disable', 'Copy', 'Mass update', and 'Delete'. The footer of the page reads 'Zabbix 6.0.24. © 2001–2023, Zabbix SIA'.

Fill out detail which required

Click on add button as show below

Trigger Tags Dependencies

* Name YH server stoped

Event name YH server stoped

Operational data

Severity Not classified Information Warning Average High Disaster

* Expression Add

[Expression constructor](#)

OK event generation Expression Recovery expression None

PROBLEM event generation mode Single Multiple

OK event closes All problems All problems if tag values match

Allow manual close ☐

URL

Description

Enabled ☒

Add Cancel

Select item

Condition

* Item Select

Function last() - Last (most recent) T value

Last of (T) Count

Time shift now-h Time

* Result = 0

Insert Cancel

Here I select host group Linux server

Under this host group I have 3 agent(agent,dockert,Windows) and one template(sigmastream) as show below not we create item on sigmastream template so click on it

Hosts

Host group Linux servers Select

Name

agent

dockert

sigmastream

Windows

Cancel

Below image show created item on sigmatream template now configure trigger for YH server -status so, click on that

Name	Key	Type	Type of information	Status
CURL	system.run["curl -s -k --head https://192.168.2.134:8443 grep '200 OK' > /dev/null; echo \$?"]	Zabbix agent	Numeric (unsigned)	Enabled
test	system.run["/etc/zabbix/cpu.sh awk 'NR>7 && NF -1 {print \$9}"]	Zabbix agent	Numeric (float)	Enabled
YH Server - status	system.run["systemctl status yhserver.service grep -q running; echo \$?"]	Zabbix agent	Numeric (unsigned)	Enabled

After it appear as below in result select = and value 1

How work:

if item run and give o/p 1 then trigger was trigger and give alert. If we need to configure for cpu or memory usage so, set ITEM and in result select > and value ex. 90 so, if CPU ITEM give value >90 then it give error

* Item: sigmatream: YH Server - status

Function: last() - Last (most recent) T value

Last of (T): Count

Time shift: now-h

* Result: = 1

Insert Cancel

Click on insert

* Name: YH server stopped

Event name: YH server stopped

Operational data:

Severity: Not classified Information Warning Average High Disaster

* Expression: last (/sigmatream/system.run["systemctl status yhserver.service | grep -q running; echo \$?"])=1

Expression constructor

OK event generation: Expression Recovery expression None

PROBLEM event generation mode: Single Multiple

OK event closes: All problems All problems if tag values match

Allow manual close: ☐

URL:

Description:

Enabled: ☒

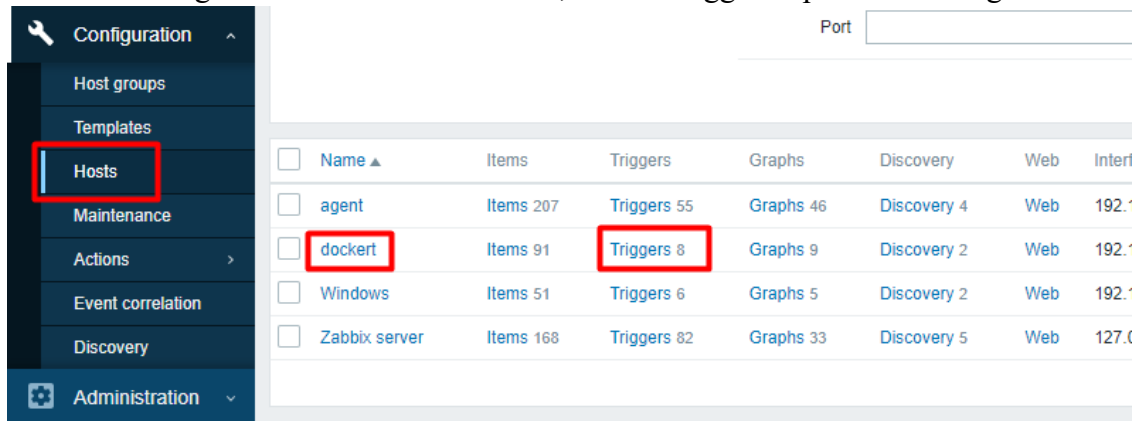
Add Cancel

Click on Add

Trigger of docker container

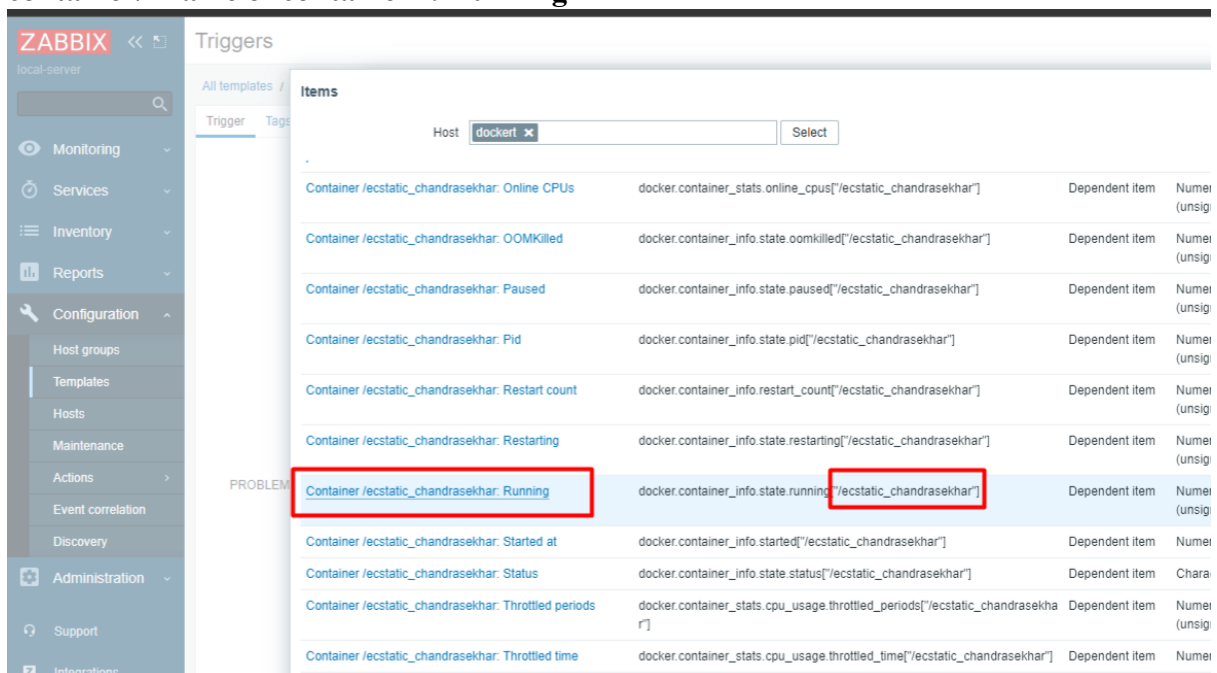
Need to follow process show as below for all docker related trigger

To set up docker based trigger go to configuration > hosts > click on trigger based on host
Here dockert agent run docker container so, I select trigger as per below image



<input type="checkbox"/>	Name ▲	Items	Triggers	Graphs	Discovery	Web	Interf
<input type="checkbox"/>	agent	Items 207	Triggers 55	Graphs 46	Discovery 4	Web	192.168.1.1
<input checked="" type="checkbox"/>	dockert	Items 91	Triggers 8	Graphs 9	Discovery 2	Web	192.168.1.1
<input type="checkbox"/>	Windows	Items 51	Triggers 6	Graphs 5	Discovery 2	Web	192.168.1.1
<input type="checkbox"/>	Zabbix server	Items 168	Triggers 82	Graphs 33	Discovery 5	Web	127.0.0.1

In trigger expression select expression named as per
container/<name of container>: Running



Items				
Host: dockert				
Container /ecstatic_chandrasekhar: Online CPUs	docker.container_stats.online_cpus["/ecstatic_chandrasekhar"]	Dependent item	Numerical (unsigned integer/float)	
Container /ecstatic_chandrasekhar: OOMKilled	docker.container_info.state.oomkilled["/ecstatic_chandrasekhar"]	Dependent item	Numerical (unsigned integer/float)	
Container /ecstatic_chandrasekhar: Paused	docker.container_info.state.paused["/ecstatic_chandrasekhar"]	Dependent item	Numerical (unsigned integer/float)	
Container /ecstatic_chandrasekhar: Pid	docker.container_info.state.pid["/ecstatic_chandrasekhar"]	Dependent item	Numerical (unsigned integer/float)	
Container /ecstatic_chandrasekhar: Restart count	docker.container_info.restart_count["/ecstatic_chandrasekhar"]	Dependent item	Numerical (unsigned integer/float)	
Container /ecstatic_chandrasekhar: Restarting	docker.container_info.state.restarting["/ecstatic_chandrasekhar"]	Dependent item	Numerical (unsigned integer/float)	
Container /ecstatic_chandrasekhar: Running	docker.container_info.state.running["/ecstatic_chandrasekhar"]	Dependent item	Numerical (unsigned integer/float)	
Container /ecstatic_chandrasekhar: Started at	docker.container_info.started["/ecstatic_chandrasekhar"]	Dependent item	Character string	
Container /ecstatic_chandrasekhar: Status	docker.container_info.state.status["/ecstatic_chandrasekhar"]	Dependent item	Character string	
Container /ecstatic_chandrasekhar: Throttled periods	docker.container_stats.cpu_usage.throttled_periods["/ecstatic_chandrasekhar"]	Dependent item	Numerical (unsigned integer/float)	
Container /ecstatic_chandrasekhar: Throttled time	docker.container_stats.cpu_usage.throttled_time["/ecstatic_chandrasekhar"]	Dependent item	Numerical (unsigned integer/float)	

Condition



* Item dockert: Container /ecstatic_chandrasekhar: Running

Select

Function last() - Last (most recent) T value



Last of (T) Count

Time shift now-h Time

* Result =



0

Insert

Cancel

Event name docker trigger

Operational data

Severity

Not classified

Information

Warning

Average

High

Disaster

* Expression

```
last (/dockert/docker.container_info.state.running["/ecstatic_chandrasekhar"]) = 0
```

[Expression constructor](#)

OK event generation

Expression

Recovery expression

None

EM event generation mode

Single

Multiple

OK event closes

All problems

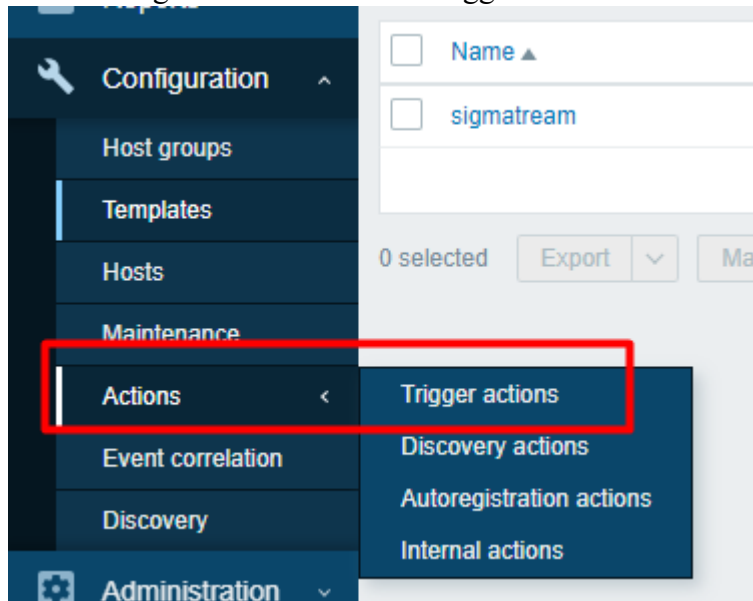
All problems if tag values match

Allow manual close

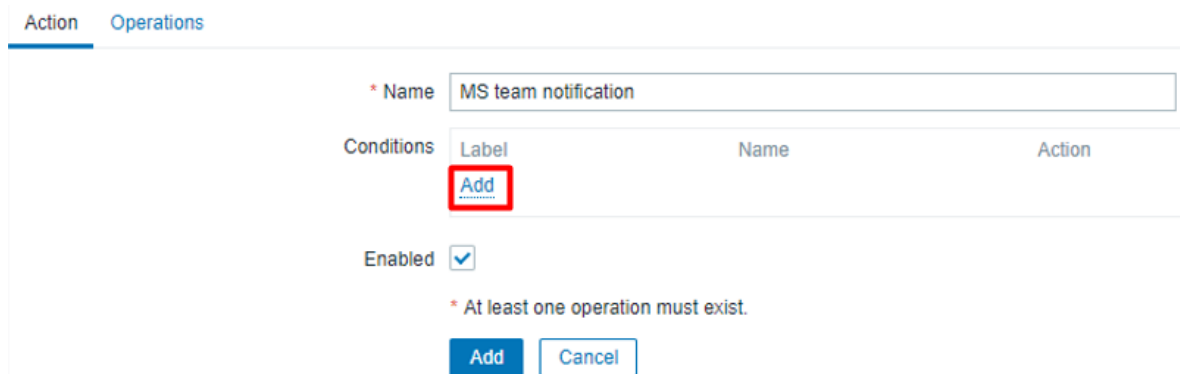
☐

4.Configure Alert on Ms team

Goto Configuration > Actions > Trigger actions



Click on Create action (present in top right corner)

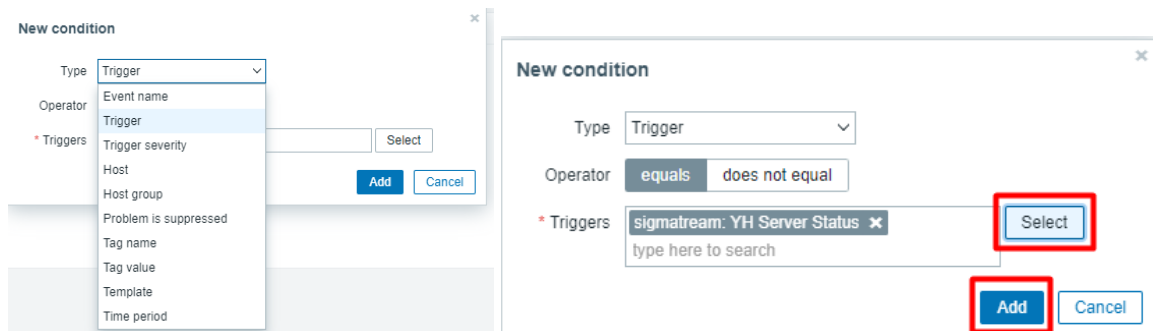


Click on add

Then as per show image below select type and value

TYPE: Trigger

Triggers: select triggers (host groups > host or template > trigger)



You can add multiple trigger as well

Actions

Action

Operations

* Name

MS team notification

Conditions

Label	Name	Action
A	Trigger equals <i>sigmatream: YH Server Status</i>	Remove
Add		

Enabled

☒

* At least one operation must exist.

Add

Cancel

Select operations

Default operation step duration: define how repeatedly notify here if problem not resolve in 1hr then it again give notification on team so, set as per requirement

Action

Operations

* Default operation step duration

1h

Operations

Steps	Details	Start in	Duration	Action
Add				

Recovery operations

Details	Action
Add	

Update operations

Details	Action
Add	

Pause operations for suppressed problems

☒

Notify about canceled escalations

☒

* At least one operation must exist.

Add

Cancel

In operation click on Add

In operation details window add info as per below

Operation: Send message

Send to Users : select any zabbix user

Send only to : MS Teams

Click on Add

The screenshot shows the Zabbix 'Operations' configuration page. The 'Operation details' dialog box is open, showing the configuration for a 'Send message' operation. The 'Operation' dropdown is set to 'Send message'. The 'Steps' are configured as 1 to 1. The 'Step duration' is set to 0. The 'Send to user groups' section is empty. The 'Send to users' section has 'Admin (Zabbix Administrator)' selected. The 'Send only to' dropdown is set to 'MS Teams'. The 'Custom message' checkbox is unchecked. The 'Conditions' section is empty. The 'Add' button at the bottom right of the dialog is highlighted with a red box.

Operation details

Operation: Send message

Steps: 1 - 1 (0 - infinitely)

Step duration: 0 (0 - use action default)

* At least one user or user group must be selected.

Send to user groups: User group Action

Send to users: User Action

Admin (Zabbix Administrator) Remove

Send only to: MS Teams

Custom message: ☐

Conditions: Label Name Action

Add Cancel

The screenshot shows the Zabbix 'Operations' configuration page after the 'Send message' operation has been added. The 'Operations' table now contains one entry: 'Send message to users: Admin (Zabbix Administrator) via MS Teams'. The 'Start in' column is set to 'Immediately' and the 'Duration' column is set to 'Default'. The 'Add' button at the bottom is highlighted with a red box.

Operations

Steps	Details	Start in	Duration	Action
1	Send message to users: Admin (Zabbix Administrator) via MS Teams	Immediately	Default	Edit Remove

Add Cancel

Click on Add

Configure Webhook URL

Goto Administrator > Media types > MS Teams

Add webhook URL in **teams_endpoint** and **Menu entry URL** value as per show in below image.

host_name {HOST.NAME} Remove

teams_endpoint https://logilitetechnologies.webh Remove

trigger_description {TRIGGER.DESRIPTION} Remove

trigger_id {TRIGGER.ID} Remove

use_default_message false Remove

zabbix_url {\$ZABBIX.URL} Remove

Add

* Script var SEVERITY_COLORS = [...]

* Timeout 30s

Process tags ☐

Include event menu entry ☒

* Menu entry name Notification

* Menu entry URL https://logilitetechnologies.webhook.office.com/webhookb2/39ec7aff-1f33-4e12-9fbi

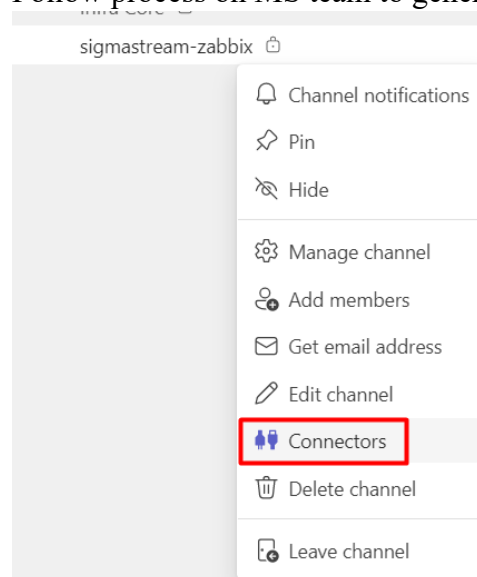
Description

Enabled ☒

Update Clone Delete Cancel

Click on Add here I already configure that's why it show update button.

Follow process on MS team to generate webhook url



Search for zabbix and click on configure

Manage

Configured

My accounts

Category

All

Analytics

CRM

Customer Support

Developer Tools

HR


Marketing


News & Social


Project Management


Others

Connectors for your team


 **Incoming Webhook** Configure
Send data from a service to your Office 365 group in real time.

 **Forms** Configure
Easily create surveys, quizzes, and polls.

 **GitHub Enterprise** Configure
Manage and collaborate on code projects hosted on a GitHub Enterprise instance.

 **Zabbix Webhook** Configure
Integrate Zabbix with Microsoft Teams by using this connector

All connectors

 **Azure DevOps** Add
Collaborate on and manage software projects online.

Copy webhook url for zabbix then click on save

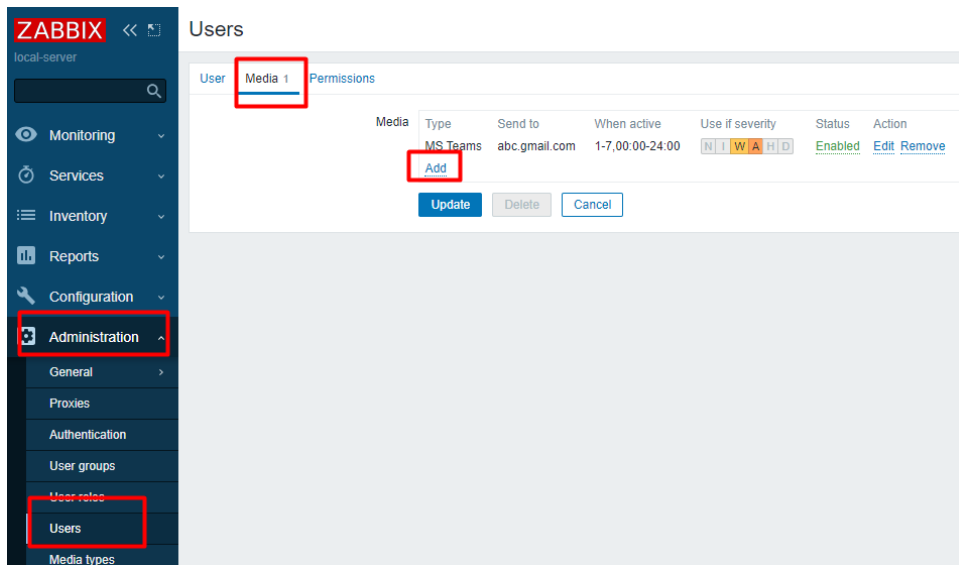
The following ways to set up Zabbix web hook connector are available:

- Import [preconfigured Microsoft teams media type XML](#) into Zabbix
- Copy the following web hook URL to Microsoft teams web hook settings in Zabbix:
`https://logilitechnologies.webhook.office.com/webhookb2/` Copy

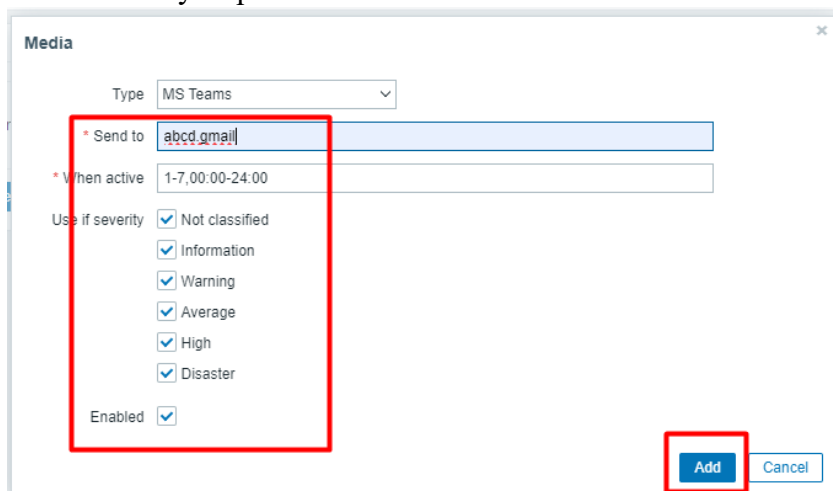
Cancel

Save

If didn't get notification then add media in user configuration as well as per shown below



Select severity as per need



5.Action on trigger (Run script for restart service)

If we need to run command on Zabbix server or agent that are connect to server via ssh then use below configuration

Run command: sudo visudo

Add below line which run systemctl related command without password and user is zabbix
: zabbix ALL=(ALL) NOPASSWD: /bin/systemctl

```
mahamadgaus ALL=(ALL:ALL) ALL
zabbix ALL=(ALL) NOPASSWD: /bin/systemctl
# Members of the admin group may gain root
%admin ALL=(ALL) ALL
```

Go to Administrator > script

Click on Create script (top right corner)

Scripts

The screenshot shows the 'Scripts' configuration page in Zabbix. The form is as follows:

- Name:** restart script
- Scope:** Action operation (selected), Manual host action, Manual event action
- Type:** Webhook, Script (selected), SSH, Telnet, IPMI
- Execute on:** Zabbix agent (selected), Zabbix server (proxy), Zabbix server
- Commands:** sudo systemctl restart yhservice.service
- Description:** (Empty text area)
- Host group:** All (dropdown menu)
- Buttons:** Update, Clone, Delete, Cancel

In above Image

Scope show when we need to run script

1.Action operation : In trigger action we add script is action in operation window so, if particular system down then it run script and restart service

2.Manual host action

In problem you can click on host and select script to run manually it help to test script execution before we automate it

The screenshot displays the Nagios XI web interface. On the left is a dark blue sidebar with navigation links: Monitoring (expanded), Dashboard, Problems, Hosts, Latest data, Maps, Discovery, Services, Inventory, Reports, Configuration, and Administration. Below these are Support, Integrations, Help, User settings, and Sign out. The main content area at the top has tabs for Show, Recent problems, Problems, and History. Below the tabs are search fields for Host groups, Hosts, and Triggers, each with a 'Select' button. A 'Problem' search field is also present. Under 'Severity', there are checkboxes for Not classified, Warning, High, Information, Average, and Disaster. The main table lists problems with columns: Time, Severity, Recovery time, Status, Info, Host, and Problem. One problem is visible: Time 17:16:07, Severity Disaster, Recovery time 17:34:07, Status RESOLVED, Host docker, and Problem docker trigger. Below the table, it says '0 selected' and 'Mass update'. A context menu is open over the 'docker' host link, showing options under 'HOST' (Inventory, Latest data, Problems, Graphs, Dashboards, Web, Configuration) and 'SCRIPTS' (Detect operating system, docker start, Ping, Traceroute). The 'docker start' option is highlighted with a red rectangle.

Time	Severity	Recovery time	Status	Info	Host	Problem
17:16:07	Disaster	17:34:07	RESOLVED		docker	docker trigger

0 selected Mass update

HOST

- Inventory
- Latest data
- Problems
- Graphs
- Dashboards
- Web
- Configuration

SCRIPTS

- Detect operating system
- docker start**
- Ping
- Traceroute

Here I configure script that start container named ecstatic_chandrasekhar

For authentication it provide two method public key and using password
We can select SSH(for remote execution) or script(for local execution)

The screenshot shows a configuration form for a new action. The 'Name' field is 'docker start'. The 'Scope' is 'Action operation'. The 'Type' is 'SSH'. The 'Authentication method' is 'Public key'. The 'Username' is 'user'. The 'Public key file', 'Private key file', and 'Key passphrase' fields are empty. The 'Port' is '22'. The 'Commands' field contains the following text:

```
#!/bin/bash
docker start ecstatic_chandrasekhar
```

. The 'Description' field is empty. The 'Host group' is 'All'. At the bottom, there are buttons for 'Update', 'Clone', 'Delete', and 'Cancel'.

To automate in trigger action add operation or recovery operation

Now add recovery operation > select script and host as per below image and click on Add

The screenshot shows the 'Actions' page with the 'Operations' tab selected. The 'Default operation step duration' is '1h'. There are three sections: 'Operations', 'Recovery operations', and 'Update operations'. The 'Operations' section has a table with one row: '1 Send message to users: Admin'. The 'Recovery operations' section has an 'Add' button. The 'Update operations' section has an 'Add' button. There are checkboxes for 'Pause operations for suppressed problems' and 'Notify about canceled escalations', both of which are checked. At the bottom, there are buttons for 'Update', 'Clone', 'Delete', and 'Cancel'. A dialog box titled 'Operation details' is open, showing the 'Operation' dropdown set to 'docker start'. The 'Target list' section has a 'Current host' checkbox, a 'Host' field with 'dockert' entered and a 'Select' button, and a 'Host group' field with 'type here to search' and a 'Select' button. At the bottom of the dialog, there are 'Add' and 'Cancel' buttons.

Save Trigger action

Example script that start service

This service run on local so no need to use ssh we use script and with updating visudo file by giving systemctl permission to zabbix user

Scripts

* Name

restart script

Scope

Action operationManual host actionManual event action

Type

WebhookScriptSSHTelnetIPMI

Execute on

Zabbix agentZabbix server (proxy)Zabbix server

* Commands

sudo systemctl restart yhservice.service

Description

Host group

All

Update

Clone

Delete

Cancel