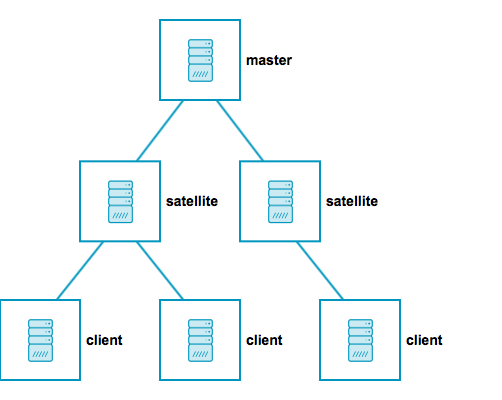
# Icinga2 - Distributed Monitoring with Master, Satellites, and Clients

Refference : https://www.icinga.com/docs/icinga2/latest/doc/02-getting-started/

## **Roles: Master, Satellites, and Clients** **[¶](https://www.icinga.com/docs/icinga2/latest/doc/06-distributed-monitoring/" \l "roles-master-satellites-and-clients)**

Icinga 2 nodes can be given names for easier understanding:

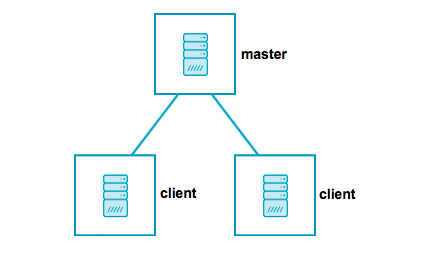
* A master node which is on top of the hierarchy.
* A satellite node which is a child of a satellite or master node.
* A client node which works as an agent connected to master and/or satellite nodes.



Rephrasing this picture into more details:

* A master node has no parent node.
* A masternode is where you usually install Icinga Web 2.
* A master node can combine executed checks from child nodes into backends and notifications.
* A satellite node has a parent and a child node.
* A satellite node may execute checks on its own or delegate check execution to child nodes.
* A satellite node can receive configuration for hosts/services, etc. from the parent node.
* A satellite node continues to run even if the master node is temporarily unavailable.
* A client node only has a parent node.
* A client node will either run its own configured checks or receive command execution events from the parent node.

In our environment we are going to configure Icinga2 with Single master and all the clients are connected to master. Service check will initiate from master to all clients.



Now let us see the steps for how to install and configure icinga2 for both master and clients servers.

Step1:- Master Installation and configuration on Icinga2 master Server.

Step2:- Client Installation and configuration on all client servers.

Step3:- Defining host, service and dependencies configurations on Icinga2 master server.

Step4:- Defining and configuring Email, Telegram, Pagerduty and SMS notifications on Icinga2 master server.

Step5:- Defining and configuring Custom commands service checks on Icinga2 master server.

Step6:- Short Rewind about Master and Client Configurations File.

**Step 1:- Icinga2 Master Installation and Configuration.**

**Icinga Master IP –** 172.22.0.140

**Icinga Master Hostname** – MUMLBMICINGAM1

**Icinga Master Server OS** - Centos7

**1.1:- Install Httpd.**

yum install httpd httpd-devel

**1.2:- Install Icing Repo and Icinga2.**

rpm --import <http://packages.icinga.org/icinga.key>

curl -o /etc/yum.repos.d/ICINGA-release.repo <http://packages.icinga.org/epel/ICINGA-release.repo>

yum makecache

yum install icinga2

systemctl enable icinga2.service

systemctl start icinga2.service

systemctl status icinga2.service

**1.3:- To List the available features in Icinga.**

The default installation will enable three features required for a basic Icinga 2 installation:

* checker for executing checks
* notification for sending notifications
* mainlog for writing the icinga2.log file

You can verify that by calling icinga2 feature list [CLI command](https://www.icinga.com/docs/icinga2/latest/doc/11-cli-commands/" \l "cli-command-feature) to see which features are enabled and disabled.

# icinga2 feature list

Disabled features: api command compatlog debuglog gelf graphite icingastatus ido-mysql ido-pgsql influxdb livestatus opentsdb perfdata statusdata syslog

Enabled features: checker mainlog notification

**1.4:- Install Nagios Plugins.**

Default plugins for service check

yum install nagios-plugins-all

**1.5:- Icinga2 mysql Connnector.**

To store all the notification details to MySQL.

yum install icinga2-ido-mysql

**1.6:- Install Mysql Server.**

Install reqiured package of MySQL server.

**1.7:- Databases and Privileges for Icinga2 and Icingaweb2.**

Once 1.6 MySQL server package is installed proceed this step.

mysql -u root -p

DB to capture service notifications for Icinga2.

CREATE DATABASE icinga;

GRANT SELECT, INSERT, UPDATE, DELETE, DROP, CREATE VIEW, INDEX, EXECUTE ON icinga.\* TO 'icinga'@'localhost' IDENTIFIED BY 'icinga';

DB to capture record for Icingaweb2 administration.

CREATE DATABASE icingaweb2;

GRANT SELECT, INSERT, UPDATE, DELETE, DROP, CREATE VIEW, INDEX, EXECUTE ON icingaweb2.\* TO 'icingaweb2'@'localhost' IDENTIFIED BY 'icingaweb2';

exit

**1.8:- Import Table and Schema for Icinga2 and Icingaweb2.**

For Icinga2

mysql -u root -p icinga < /usr/share/icinga2-ido-mysql/schema/mysql.sql

For Icingaweb2

mysql -uroot -p icingaweb2 < /usr/share/doc/icingaweb2/schema/mysql.schema.sql

**1.9:- Enable Icinga2 required features.**

Enable Mysql feature on icinga2.

icinga2 feature enable ido-mysql

Enable Service check feature on icinga2.

icinga2 feature enable command

Enable BI data feature on icinga2.

icinga2 feature enable perfdata

systemctl restart icinga2.service

**1.10:- Add apache group into icingacmd.**

usermod -a -G icingacmd apache

grep icingacmd /etc/group

**1.11:- Install required PHP package for Icingaweb2.**

yum install php-gd php-intl php-ZendFramework php-pear php-pdo php-soap php-ldap php-cli php-common php-devel php-mbstring php-mysql php-xml

vi /etc/php.ini

date.timezone = Asia/Kolkata

[ESC]:wq

**1.12:- Install Icingaweb2, Icingacli package and Configure Icingaweb2 Interface.**

yum install icingaweb2 icingacli

Generate Token for one time Icingaweb2 setup wizard login on your browser.

icingacli setup token create

Show the generated Token to copy.

icingacli setup token show #[Copy the token and

systemctl restart httpd

Go to Browser and paste the below link and do the one time setup wizard.

<http://172.22.0.140/icingaweb2/setup>

Once setup is completed. Please do the below process.

openssl passwd -1 **<admin username that u given on above icingaweb2 setup wizard>**

<$1$Ouw7Dx6y$ScuXqTzZ8BYOS6JdqIhbf1> **(example)**

mysql -uroot -p icingaweb2

INSERT INTO icingaweb\_user (name, active, password\_hash) VALUES ('**opsadmin**',1,'$1$Ouw7Dx6y$ScuXqTzZ8BYOS6JdqIhbf1');

**1.13:- Configure Icinga2 Master.**

Icinga2 master setup wizard.

#icinga2 node wizard

Welcome to the Icinga 2 Setup Wizard!

We'll guide you through all required configuration details.

Please specify if this is a satellite setup ('n' installs a master setup) [Y/n]: n

Starting the Master setup routine...

Please specifiy the common name (CN) [MUMLBMICINGAM1]: Enter

Checking for existing certificates for common name 'MUMLBMICINGAM1'...

Certificates not yet generated. Running 'api setup' now.

information/cli: Generating new CA.

information/base: Writing private key to '/var/lib/icinga2/ca/ca.key'.

information/base: Writing X509 certificate to '/var/lib/icinga2/ca/ca.crt'.

information/cli: Generating new CSR in '/etc/icinga2/pki/MUMLBMICINGAM1.csr'.

information/base: Writing private key to '/etc/icinga2/pki/MUMLBMICINGAM1.key'.

information/base: Writing certificate signing request to '/etc/icinga2/pki/MUMLBMICINGAM1.csr'.

information/cli: Signing CSR with CA and writing certificate to '/etc/icinga2/pki/MUMLBMICINGAM1.crt'.

information/pki: Writing certificate to file '/etc/icinga2/pki/MUMLBMICINGAM1.crt'.

information/cli: Copying CA certificate to '/etc/icinga2/pki/ca.crt'.

Generating master configuration for Icinga 2.

information/cli: Adding new ApiUser 'root' in '/etc/icinga2/conf.d/api-users.conf'.

information/cli: Enabling the 'api' feature.

Enabling feature api. Make sure to restart Icinga 2 for these changes to take effect.

information/cli: Dumping config items to file '/etc/icinga2/zones.conf'.

information/cli: Created backup file '/etc/icinga2/zones.conf.orig'.

Please specify the API bind host/port (optional): Enter

Bind Host []: Enter

Bind Port []: Enter

information/cli: Created backup file '/etc/icinga2/features-available/api.conf.orig'.

information/cli: Updating constants.conf.

information/cli: Created backup file '/etc/icinga2/constants.conf.orig'.

information/cli: Updating constants file '/etc/icinga2/constants.conf'.

information/cli: Updating constants file '/etc/icinga2/constants.conf'.

information/cli: Updating constants file '/etc/icinga2/constants.conf'.

Done.

Icinga2 Node Wizard does the following:

1. It enables API feature.
2. Creates the new certificate authority to sign the signing requests.
3. Generates the new certificate signing request (CSR), sign it with own CA and then copy it into /etc/icinga2/pki/.
4. Dumping configuration items (zone’s and endpoints) to /etc/icinga2/zones.conf
5. Updating the /etc/icinga2/constants.conf file with NodeName and TicketSalt.

Verify the configurations files. Make sure your master node configurations should be like below.

cat /etc/icinga2/constants.conf | egrep -i "ZoneName|TicketSalt"

const ZoneName = "MUMLBMICINGAM1"

const TicketSalt = "b58f092cd2bd27da563699938ca793ed"

**1.14:- Zone.conf file changes.**

Update the /etc/icinga2/zones.conf file with master node details.

vi /etc/icinga2/zones.conf

object Endpoint "**MUMLBMICINGAM1**" { **#[Icinga Master Hostname]**

}

object Zone "**master**" { **#[Master name to identify]**

endpoints = [ "**MUMLBMICINGAM1**" ] **#[Icinga Master Hostname]**

}

To Sync user defined command configuration file globally from master to all clients please do add the following conf in zone.conf.

vi /etc/icinga2/zones.conf **#[add at the last]**

object Zone "global-templates" {

global = true

}

**global-templates** should be the directory name created under /etc/icinga2/zones.d/ to sync command conf file from master to all cleints. We can see in more detail on step no “5”

vi /etc/icinga2/zones.conf

object Zone "global-templates" {

global = true

}

[root@MUMLBMICINGAM1 ~]# icinga2 daemon -C

[root@MUMLBMICINGAM1 ~]# systemctl restart icinga2

**Step 2:- Icinga2 Client Installation and Configuration.**

Client Host Name : MUMLBMNEWAPP1V

Client IP : 172.22.0.90

**2.1:- Install Nagios and Icinga2 Package**

Make sure you have Icinga 2 package and Nagios plugins are installed on your client machine. If you haven’t done already, please follow below steps to install the same.

RHEL/CentOS 7:

# yum install epel-release

# yum install https://packages.icinga.org/epel/7/release/noarch/icinga-rpm-release-7-1.el7.centos.noarch.rpm

RHEL/CentOS 6:

# yum install epel-release

# yum install <https://packages.icinga.org/epel/6/release/noarch/icinga-rpm-release-6-1.el6.noarch.rpm>

Now, install Icinga 2 and Nagios plugins using the following command.

yum install icinga2

yum install nagios-plugins-all

**2.2:- Generate a Ticket On Icinga Master Server.**

When you run a icinga node wizard in client machines, the wizard sends a certificate signing request to master node. To auto sign a request, master node requires a valid ticket from clients. You must generate the ticket beforehand. Run the following command to generate the ticket for the client “MUMLBMNEWAPP1V“

[root@MUMLBMICINGAM1 ~]# icinga2 pki ticket --cn 'MUMLBMNEWAPP1V'

4f11dced83f05820b09b411d40e4b643356b4a65

[root@MUMLBMICINGAM1 ~]#

**2.3:- Setup Icinga2 Client.**

Run the Icinga Node Wizard on client.

[root@MUMLBMNEWAPP1V ~]# icinga2 node wizard

Welcome to the Icinga 2 Setup Wizard!

We'll guide you through all required configuration details.

Please specify if this is a satellite setup ('n' installs a master setup) [Y/n]: Enter

Starting the Node setup routine...

Please specify the common name (CN) [MUMLBMNEWAPP1V]: Enter

Please specify the master endpoint(s) this node should connect to:

Master Common Name (CN from your master setup): MUMLBMICINGAM1

Do you want to establish a connection to the master from this node? [Y/n]: y

Please fill out the master connection information:

Master endpoint host (Your master's IP address or FQDN): 172.22.0.140

Master endpoint port [5665]: Enter

Add more master endpoints? [y/N]: n

Please specify the master connection for CSR auto-signing (defaults to master endpoint host):

Host [172.22.0.140]: Enter

Port [5665]: Enter

information/base: Writing private key to '/etc/icinga2/pki/MUMLBMNEWAPP1V.key'.

information/base: Writing X509 certificate to '/etc/icinga2/pki/MUMLBMNEWAPP1V.crt'.

information/cli: Fetching public certificate from master (172.22.0.140, 5665):

Certificate information:

Subject: CN = MUMLBMICINGAM1

Issuer: CN = Icinga CA

Valid From: Jun 12 10:01:46 2017 GMT

Valid Until: Jun 8 10:01:46 2032 GMT

Fingerprint: 02 24 6F 69 C1 D7 5C 01 9A 50 A6 E6 5A 16 58 32 C1 E1 BA 30

Is this information correct? [y/N]: y

information/cli: Received trusted master certificate.

Please specify the request ticket generated on your Icinga 2 master.

(Hint: # icinga2 pki ticket --cn 'MUMLBMNEWAPP1V'): 4f11dced83f05820b09b411d40e4b643356b4a65

information/cli: Requesting certificate with ticket '4f11dced83f05820b09b411d40e4b643356b4a65'.

warning/cli: Backup file '/etc/icinga2/pki/ca.crt.orig' already exists. Skipping backup.

warning/cli: Backup file '/etc/icinga2/pki/MUMLBMNEWAPP1V.crt.orig' already exists. Skipping backup.

information/cli: Writing signed certificate to file '/etc/icinga2/pki/MUMLBMNEWAPP1V.crt'.

information/cli: Writing CA certificate to file '/etc/icinga2/pki/ca.crt'.

Please specify the API bind host/port (optional):

Bind Host []:

Bind Port []:

Accept config from master? [y/N]: y

Accept commands from master? [y/N]: y

information/cli: Disabling the Notification feature.

critical/cli: Cannot disable feature 'notification'. Target file '/etc/icinga2/features-enabled/notification.conf' does not exist.

critical/cli: Cannot disable feature(s): notification

information/cli: Enabling the Apilistener feature.

warning/cli: Feature 'api' already enabled.

information/cli: Generating local zones.conf.

information/cli: Dumping config items to file '/etc/icinga2/zones.conf'.

information/cli: Updating constants.conf.

information/cli: Updating constants file '/etc/icinga2/constants.conf'.

Done.

Now restart your Icinga 2 daemon to finish the installation!

[root@MUMLBMNEWAPP1V ~]#

Wizard does the following,

Fetches the public certificate from the master

Client requests for a certificate from the master with the valid ticket and then it saves it into /etc/icinga2/pki/.

Updates the zones and constants.

Verify the configurations files. Make sure your client node configurations should be like below.

[root@MUMLBMNEWAPP1V ~]# vi /etc/icinga2/zones.conf

object Endpoint "MUMLBMICINGAM1" {

host = "172.22.0.140"

port = "5665"

}

object Zone "master" {

endpoints = [ "MUMLBMICINGAM1" ]

}

object Endpoint NodeName {

}

object Zone ZoneName {

endpoints = [ NodeName ]

parent = "master"

}

[root@MUMLBMNEWAPP1V ~]# vi /etc/icinga2/constants.conf

const NodeName = "MUMLBMNEWAPP1V"

/\* Our local zone name. \*/

const ZoneName = "MUMLBMNEWAPP1V"

**2.4:- Define Client Zone info on Icinga2 Master Server.**

[root@MUMLBMICINGAM1 ~]# vi /etc/icinga2/zones.d/master/global-client-zone-conf/MUMLBMNEWAPP1V.conf

object Endpoint "MUMLBMNEWAPP1V" {

host = "172.22.0.90"

}

object Zone "MUMLBMNEWAPP1V" {

endpoints = [ "MUMLBMNEWAPP1V" ]

parent = "master" //establish zone hierarchy

}

[root@MUMLBMICINGAM1 ~]# icinga2 daemon -C

[root@MUMLBMICINGAM1 ~]# systemctl restart icinga2

**2.5:- Comment un-wanted in icinga2.conf.**

Please do comment the following in all icinga2 client servers.

[root@MUMLBMNEWAPP1V ~]# vi /etc/icinga2/icinga2.conf

**//**include\_recursive "conf.d"

[root@MUMLBMNEWAPP1V ~]# icinga2 daemon -C

[root@MUMLBMNEWAPP1V ~]# systemctl restart icinga2

**3:- Defining Host, Service and Dependency configurations on Icinga2 master server.**

In Icinga2 master we need to configure the following for service checks to all Icinga2 client servers.

\* Host Configuration [ Mandatory ]

\* Service Configuration [ Mandatory ]

\* Dependency Configuration [ Optional based upon service requirement ]

Before we moving into the configurations I will let you know about variable used in Icinga2 so that it will be helpful for host, service and dependency configurations.

**3.0:- Icinga2 Variables.**

In Icinga2 we can store the values in variables for certain process. In many cases declaring variables is required on host, server and dependency configurations. Now let us see the syntax for declaring variable and fetching value to variables.

**3.0.1:- Declaring Variables.**

vars.variablename = value

vars.variablename[“arrayvalue”] = { value = [ “value1”, “value2”, … “valuen” ] }

Example:-

vars.endpoint = “MUMLBMNEWAPP1V”

vars.notification["mail"] = { groups = [ "sedbm" ]}

vars.notification[ "sms" ] = { groups = [ "level1", "level2" ] }

**3.0.2:- Fetching Values to Variables.**

We can fetch the variable declared in host, service and dependency configuration using the following syntax.

**3.0.2.1:- Fetching the values of host variable.**

Syntax:-

host.vars.variablename

Example:-

command\_endpoint = host.vars.endpoint

assign where host.vars.endpoint

host.vars.endpoint = “MUMLBMNEWAPP1V”

**3.0.2.1:- Fetching the values of service variable.**

Syntax:-

service.vars.variablename

Example:-

command\_endpoint = service.vars.pagerduty\_alert\_immediate\_appsupport

service.vars.pagerduty\_alert\_immediate\_appsupport = true

**3.1:- Host Configurations on Icinga2 Master Server.**

Hence all clients are going to establish the connection from master zone, we need to mention all the host configurations under the directory “/etc/icinga2/zones.d/” other wise master zone unable to recognize client zone.

Here I am going to create a directory named “global-client-host-conf” to identify easily by all engineers

by assuming all the global host configurations files are under this directory.

[root@MUMLBMICINGAM1 ~]# mkdir /etc/icinga2/zones.d/global-client-host-conf

Now I am going to create a host conf file with the name of “{HOSTNAME}.conf” for easy identification by engineer.

[root@MUMLBMICINGAM1 ~]# vi /etc/icinga2/zones.d/global-client-host-conf/MUMLBMNEWAPP1V.conf

object Host "MUMLBMNEWAPP1V" {

check\_command = "hostalive"

address = "172.22.0.90"

zone = "master" //optional trick: sync the required host object to the client, but enforce the "master" zone to execute the check

check\_interval = 1m

retry\_interval = 15s

max\_check\_attempts = 3

vars.endpoint = "MUMLBMNEWAPP1V"

vars.all\_servers\_application\_group = "MUMLBMNEWAPP1V"

vars.bm\_application\_servers\_group = "MUMLBMNEWAPP1V"

vars.bm\_app\_servers\_group = "MUMLBMNEWAPP1V"

vars.datadisk = "MUMLBMNEWAPP1V"

vars.hostgroup = "bm-app-cluster"

vars.enable\_pagerduty\_host\_appsupport = true

vars.notification["mail"] = { groups = [ "sedbm" ]}

vars.notification["telegram" ] = { groups = [ "icingatelegram" ]}

vars.notification[ "sms" ] = { groups = [ "level1", "level2" ] }

}

vars.endpoint = "MUMLBMNEWAPP1V" ==> for common service check like disk space,cpu,load, etc.

vars.all\_servers\_application\_group = "MUMLBMNEWAPP1V" ==> service check for http,https

vars.bm\_application\_servers\_group = "MUMLBMNEWAPP1V" ==> service check for bm node

vars.bm\_app\_servers\_group = "MUMLBMNEWAPP1V" ==> service check for app1-20 servers only

vars.datadisk = "MUMLBMNEWAPP1V" ==> service check for “diskspace /data”

vars.hostgroup = "bm-app-cluster" ==> for hostgroups “/etc/icinga2/conf.d/groups.conf”

vars.enable\_pagerduty\_host\_appsupport = true ==> for pagerduty trigger is host is down

vars.notification["mail"] = { groups = [ "sedbm" ]} ==> for mail notifications “/etc/icinga2/conf.d/users.conf & /etc/icinga2/conf.d/notifications.conf”

vars.notification["telegram" ] = { groups = [ "icingatelegram" ]} ==> for telegram notifications “/etc/icinga2/conf.d/users.conf & /etc/icinga2/conf.d/notifications.conf”

vars.notification[ "sms" ] = { groups = [ "level1", "level2" ] } ==> for sms notifications “/etc/icinga2/conf.d/users.conf & /etc/icinga2/conf.d/notifications.conf”

Whenever you are working with Icinga2 configuration files please do syntax check and restart icinga2 after your configuration changes completed.

[root@MUMLBMICINGAM1 ~]# icinga2 daemon -C

[root@MUMLBMICINGAM1 ~]# systemctl restart icinga2.service

**3.2:- Service Configurations on Icinga2 Master Server.**

Hence all clients are going to establish the connection from master zone, we need to mention all the service configurations under the directory “/etc/icinga2/zones.d/master” other wise master zone icinga2 server will not initiate service check to the host.

(Note:- A directory /etc/icinga2/zones.d/**master** is because of the master zone name the I have created in step number 1.14

[root@MUMLBMICINGAM1 ~]#cat /etc/icinga2/zones.conf

object Endpoint "**MUMLBMICINGAM1**" {

}

object Zone "**master**" { [ Here Master zone name is master ]

endpoints = [ "**MUMLBMICINGAM1**" ]

}

Here the master zone name is **master** so I suppose to create a directory named as /etc/icinga2/zones.d/**master**)

[root@MUMLBMICINGAM1 ~]# mkdir /etc/icinga2/zones.d/master

[root@MUMLBMICINGAM1 ~]# vi /etc/icinga2/zones.d/master/disk1-all\_matrimony\_servers.conf

apply Service "diskspace /" {

check\_command = "disk"

check\_interval = 5m

retry\_interval = 15s

max\_check\_attempts = 4

vars.disk\_units = "GB"

vars.disk\_partitions = [ "/" ]

vars.disk\_wfree = "10%"

vars.disk\_cfree = "1%"

vars.pagerduty\_alert\_immediate\_appsupport = true

command\_endpoint = host.vars.endpoint

assign where host.vars.endpoint

}

[root@MUMLBMICINGAM1 ~]# icinga2 daemon -C

[root@MUMLBMICINGAM1 ~]# systemctl restart icinga2.service

**3.3:- Dependency Configurations on Icinga2 Master Server.**

Here I have Inherited/created dependencies named “**check\_tcp\_bmrd1\_host**”(Parent host) and "**check\_tcp\_bmrd1\_service**"(Parent Service) for the services "**tcp\_bmrd1\_ip1**", "**tcp\_bmrd1\_ip2**" and "**tcp\_bmrd1\_ip3**".

It will suppress the service notification of "**tcp\_bmrd1\_ip1**", "**tcp\_bmrd1\_ip2**" and "**tcp\_bmrd1\_ip3**" if the **parent host(** MUMLBMBMRD1) **or parent service( tcp\_localhost\_redis-slave) is down.**

Please find the example dependency conf.

[root@MUMLBMICINGAM1 ~]# cat /etc/icinga2/zones.d/master/**dependencies**/bmservers/**disable-check\_tcp\_bmrd1.conf**

apply **Dependency** "**check\_tcp\_bmrd1\_host**" to Service {

parent\_host\_name = "MUMLBMBMRD1"

states = [ Up ]

disable\_checks = true

disable\_notifications = true

assign where service.name == "**tcp\_bmrd1\_ip1**" || service.name == "**tcp\_bmrd1\_ip2**" || service.name == "**tcp\_bmrd1\_ip3**"

}

apply **Dependency** "**check\_tcp\_bmrd1\_service**" to Service {

parent\_host\_name="**MUMLBMBMRD1**"

parent\_service\_name = "**tcp\_localhost\_redis-slave**"

states = [ OK ]

disable\_checks = true

disable\_notifications = true

assign where service.name == "**tcp\_bmrd1\_ip1**" || service.name == "**tcp\_bmrd1\_ip2**" || service.name == "**tcp\_bmrd1\_ip3**"

}

[root@MUMLBMICINGAM1 ~]# cat /etc/icinga2/zones.d/master/bm\_services/redis\_servers/tcp\_localhost\_redis-slave-service.conf

apply **Service** "**tcp\_localhost\_redis-slave**" {

check\_command = "check\_tcp"

vars.port = "6380"

vars.hostname = "localhost"

vars.respose\_time\_critical = "1.0"

command\_endpoint = host.vars.endpoint

assign where **host.vars.bm\_rd\_servers\_group**

}

[root@MUMLBMICINGAM1 ~]# cat /etc/icinga2/zones.d/global-client-host-conf/**MUMLBMBMRD1**.conf

object Host "**MUMLBMBMRD1**" {

check\_command = "**hostalive**"

address = "172.22.2.91"

zone = "master" //optional trick: sync the required host object to the client, but enforce the "master" zone to execute the check

check\_interval = 1m

retry\_interval = 15s

max\_check\_attempts = 3

vars.endpoint = "MUMLBMBMRD1"

**vars.bm\_rd\_servers\_group = "MUMLBMBMRD1"**

vars.hostgroup = "bm-redis-cluster"

vars.enable\_pagerduty\_host\_appsupport = true

vars.notification["mail"] = { groups = [ "sedbm" ]}

vars.notification["telegram" ] = { groups = [ "icingatelegram" ]}

vars.notification[ "sms" ] = { groups = [ "level1", "level2" ] }

}

[root@MUMLBMICINGAM1 ~]# icinga2 daemon -C

[root@MUMLBMICINGAM1 ~]# systemctl restart icinga2

**4:- Defining and configuring Email, Telegram, Pagerduty and SMS notifications on Icinga2 master server.**

**4.1:-** **Defining and configuring Email on Icinga2 master server.**

[root@MUMLBMICINGAM1 ~]# yum install postfix

[root@MUMLBMICINGAM1 ~]# vi /etc/icinga2/conf.d/users.conf

object User "alexbennet" {

import "generic-user"

display\_name = "Alex Bennet M"

groups = [ "sedbm","level1" ]

email = "alex.bennet@matrimony.com"

pager = "9620804034"

}

object UserGroup "sedbm" {

display\_name = "Great SEDBM Team"

}

[root@MUMLBMICINGAM1 ~]# vi /etc/icinga2/zones.d/global-client-host-conf/MUMLBMNEWAPP1V.conf

object Host "MUMLBMNEWAPP1V" {

check\_command = "hostalive"

address = "172.22.0.90"

zone = "master" //optional trick: sync the required host object to the client, but enforce the "master" zone to execute the check

check\_interval = 1m

retry\_interval = 15s

max\_check\_attempts = 3

vars.endpoint = "MUMLBMNEWAPP1V"

vars.all\_servers\_application\_group = "MUMLBMNEWAPP1V"

vars.bm\_application\_servers\_group = "MUMLBMNEWAPP1V"

vars.bm\_app\_servers\_group = "MUMLBMNEWAPP1V"

vars.datadisk = "MUMLBMNEWAPP1V"

vars.hostgroup = "bm-app-cluster"

vars.notification["mail"] = { groups = [ "sedbm" ]} #[Add this line]

}

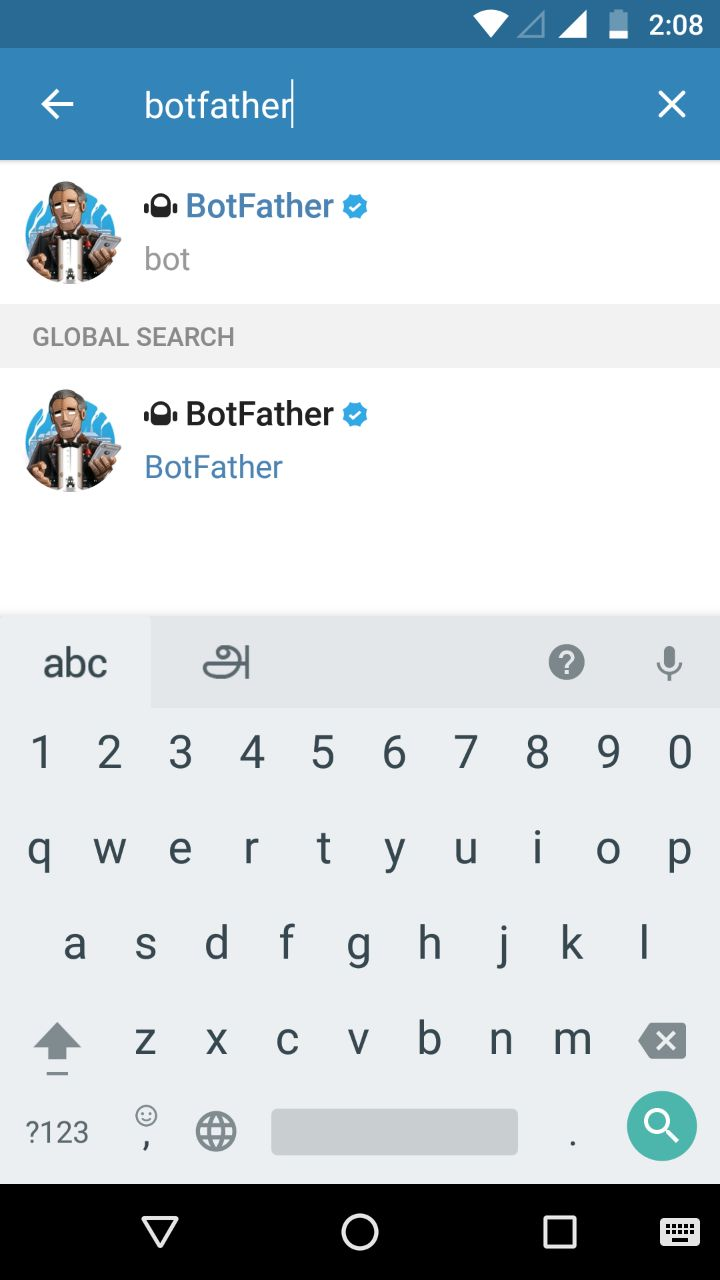
[root@MUMLBMICINGAM1 ~]# icinga2 daemon -C

[root@MUMLBMICINGAM1 ~]# systemctl restart icinga2

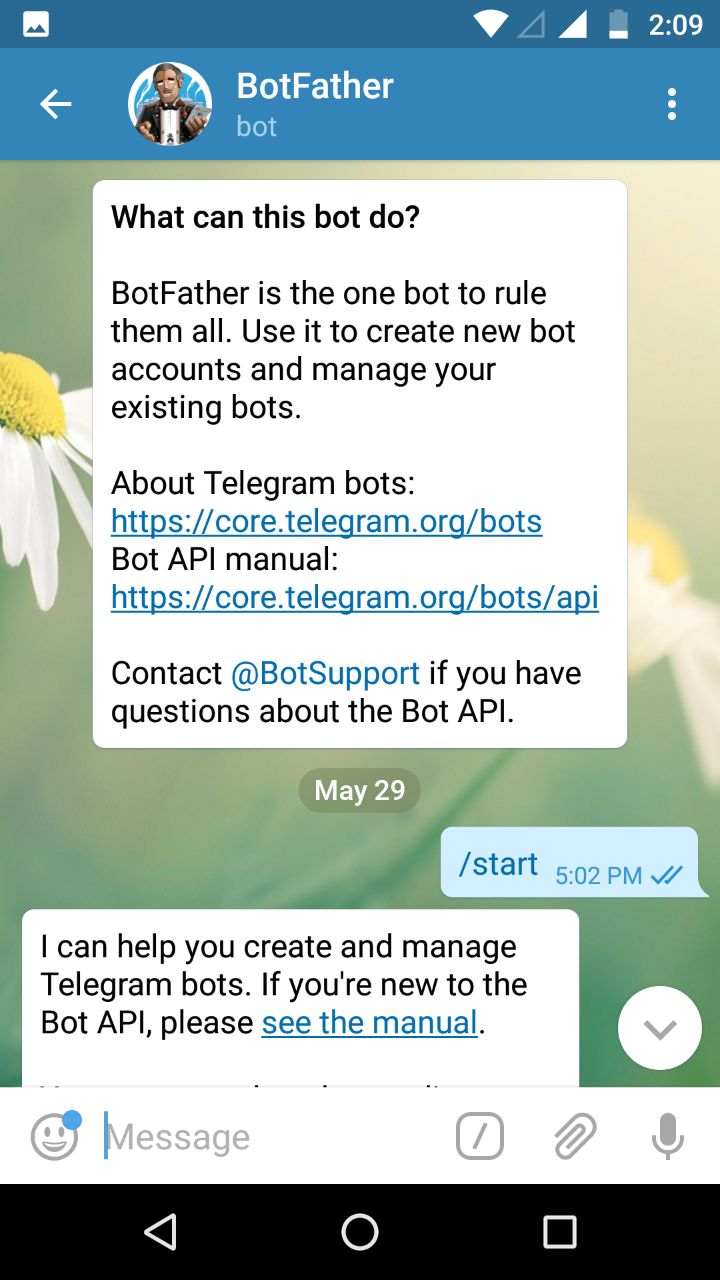
**4.2:-** **Defining and configuring Telegram Notifications on Icinga2 master server.**

**4.2.1:- Create Bot in Telegram Mobile App.**

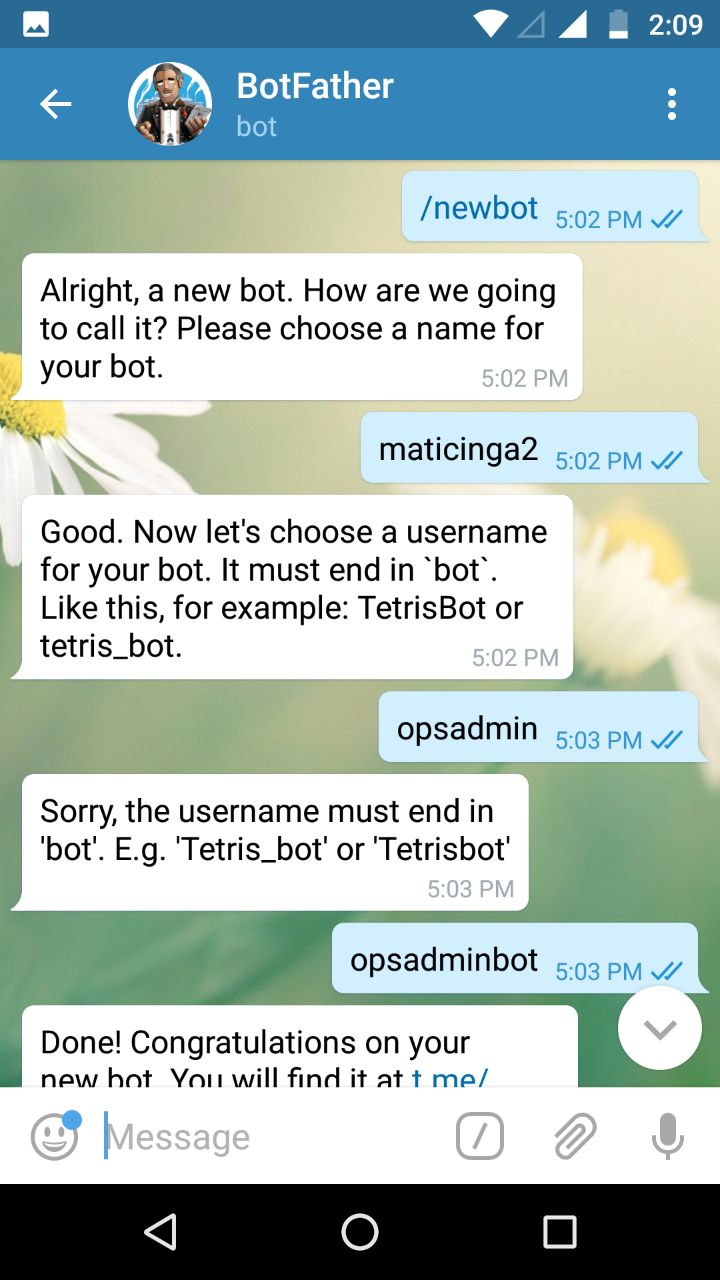
Go to Telegram app on SEDBM mobile number and on the top of the search bar type “BotFather”.

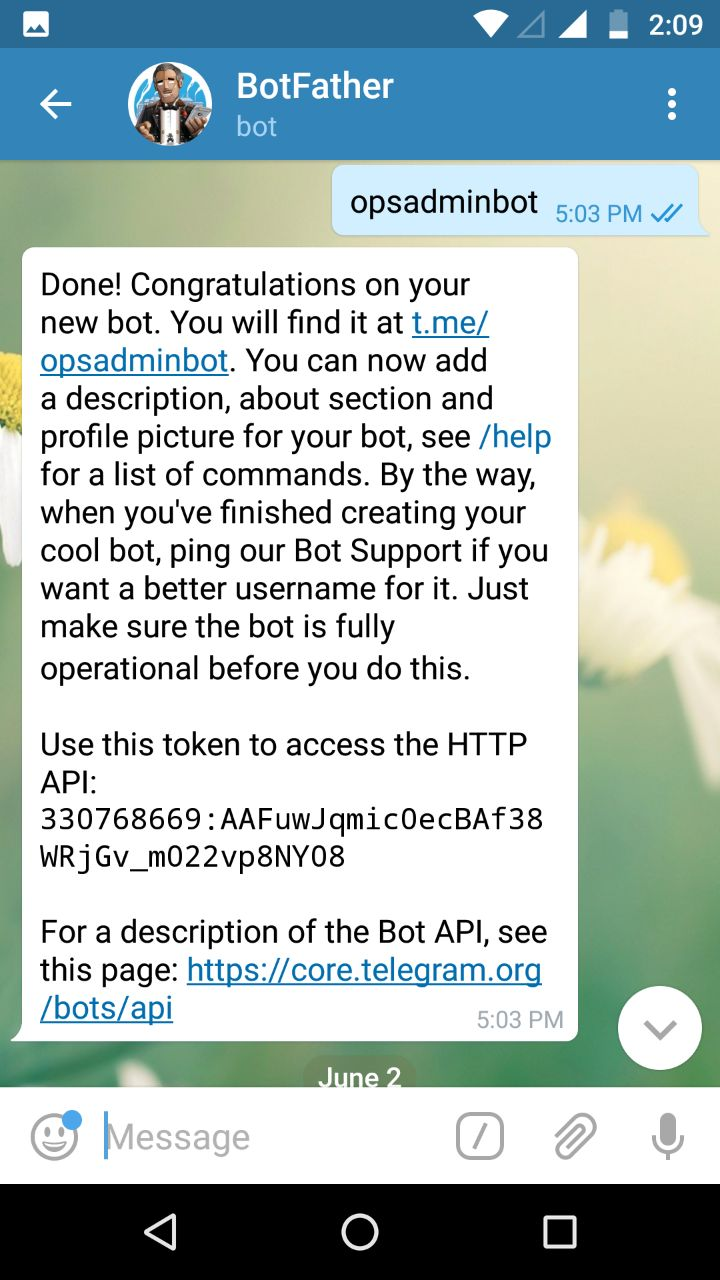


Click “BotFather”. Once you entered into BotFather type “/start”



Afterthat create new bot. Type “/newbot” continued by type bot name “maticinga2” and finally type username “opsadmin”.



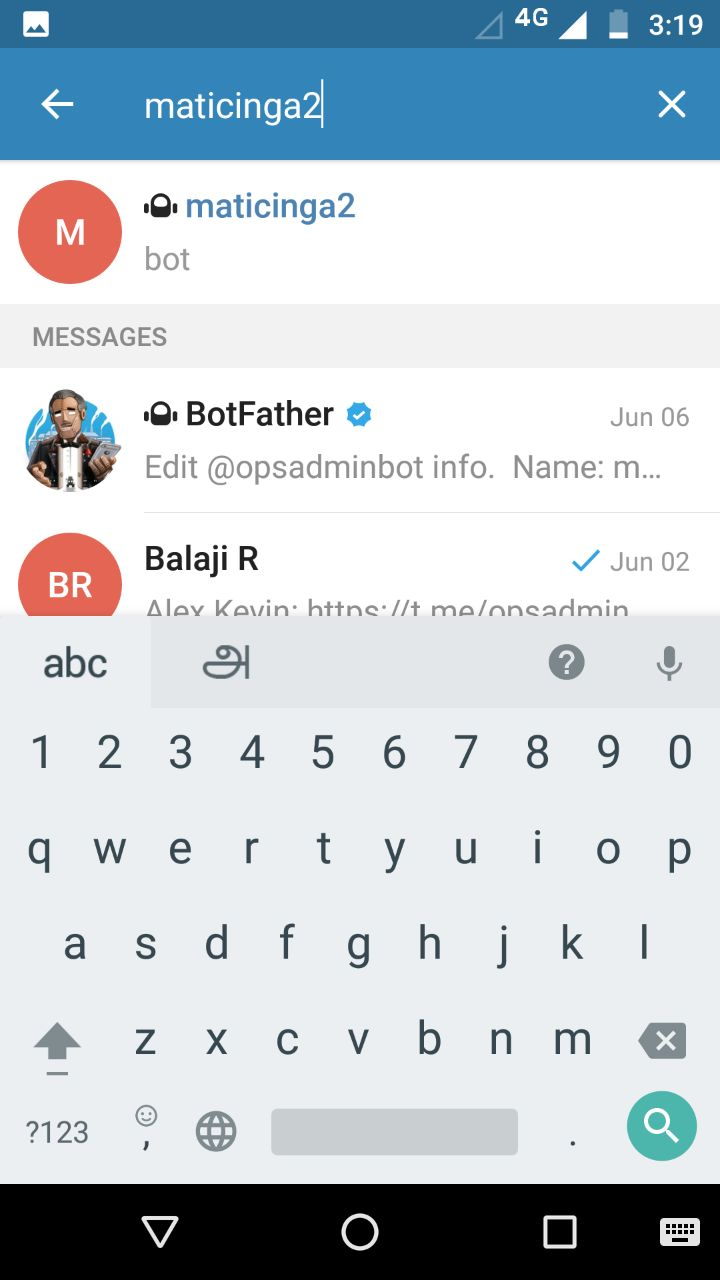
Now you have got token (**330768669:AAFuwJqmic0ecBAf38WRjGv\_m022vp8NY08**) to access the HTTP API for alert triggering.

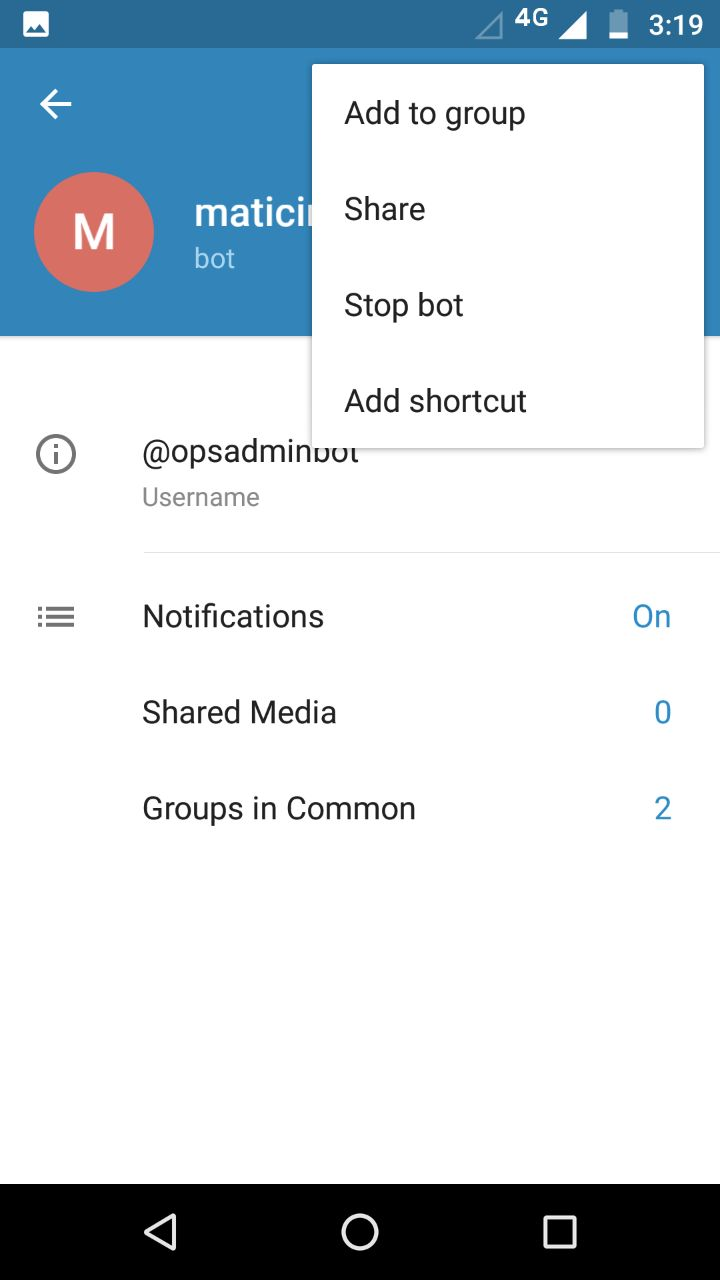
**4.2.2:- Create Group & add bot onto the group on Telegram**

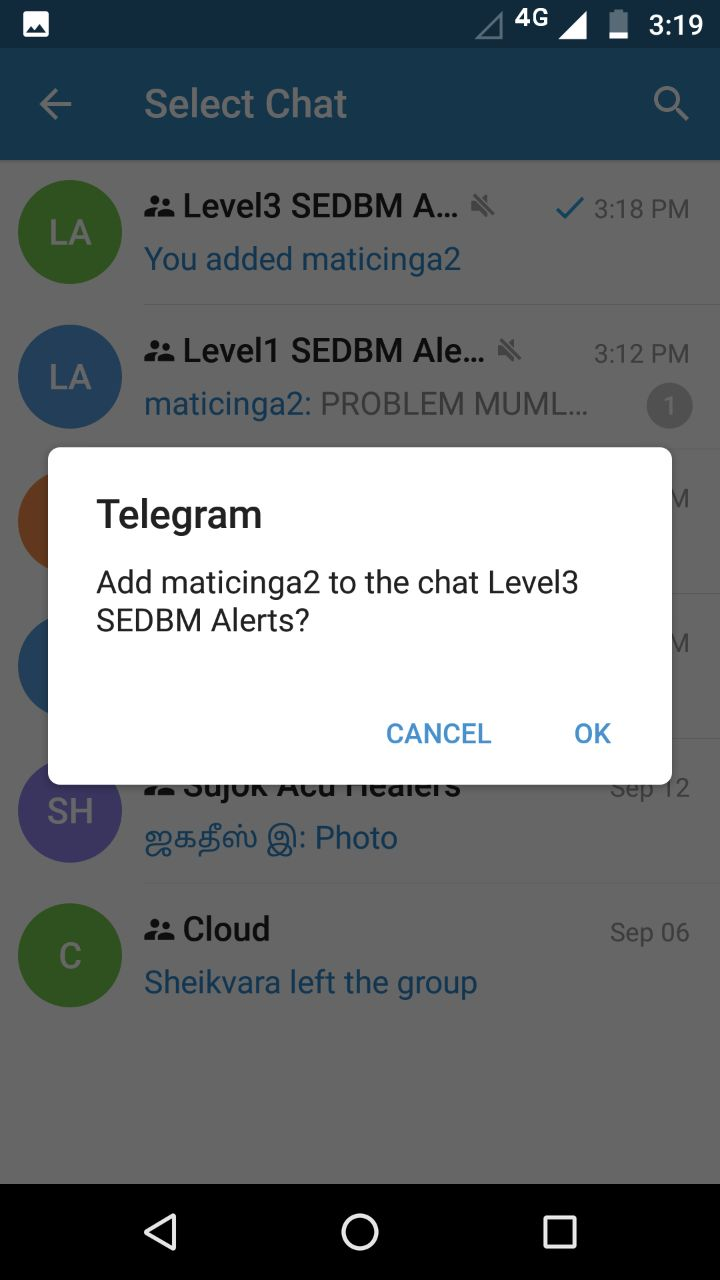
Once the Bot and username is created as shown in the above pic now got to telegram main page, on the search bar type the bot name that you have created “maticing2” , then click the bot and on the top click the bot name it gos to bot profile now click settings icon and select “Add to group” after that select the coup and touch ok.

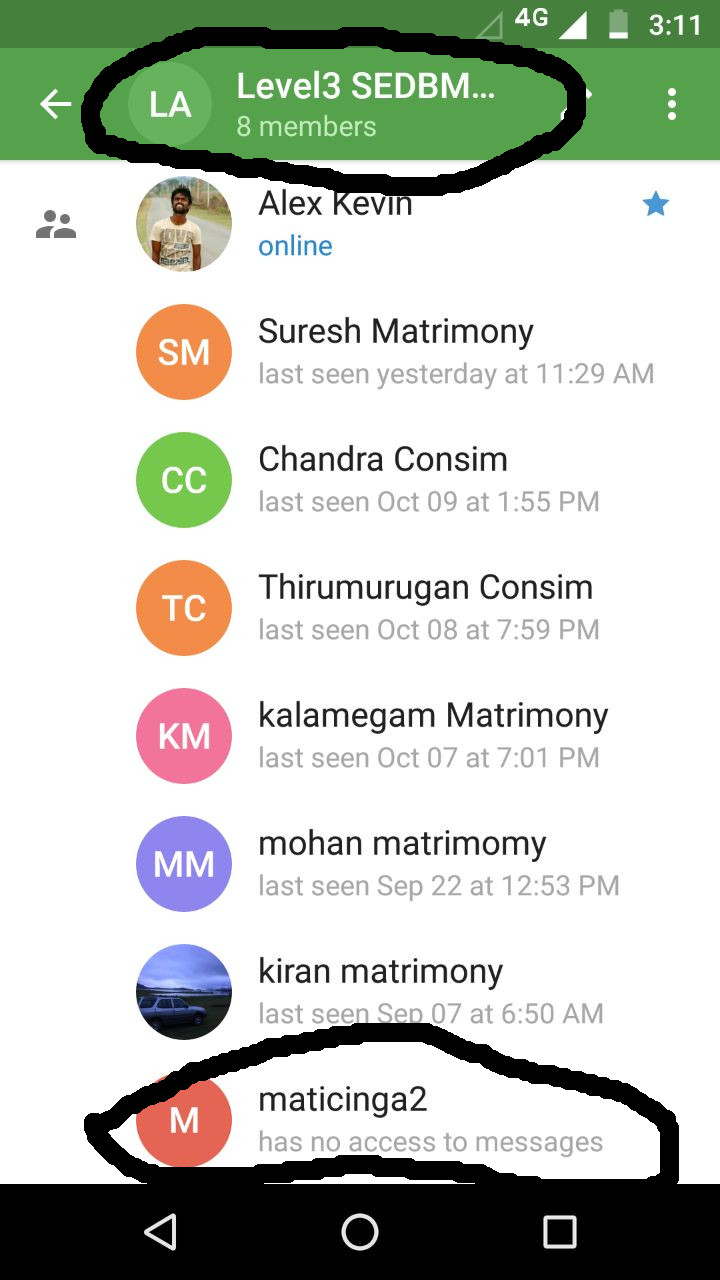
Ref Link : <http://metz.gehn.net/2016/01/monitoring-notifications-via-telegram/>

Please look into the picture below for more clarifications.









**4.2.3:- Send Test message to Group Via Bot.**

Once “maticinga2” Bot is added into a group then execute the below curl request on Icinga2 master server to find the **Chat ID** of the group that you added a “maticinga2” bot.

Note: In curl request after bot you have to give token (**330768669:AAFuwJqmic0ecBAf38WRjGv\_m022vp8NY08**).

[root@MUMLBMICINGAM1 ~]# yum install jq

[root@MUMLBMICINGAM1 ~]# curl --silent "[https://api.telegram.org/bot330768669:AAFuwJqmic0ecBAf38WRjGv\_m022vp8NY08/**getUpdates**](https://api.telegram.org/bot330768669:AAFuwJqmic0ecBAf38WRjGv_m022vp8NY08/getUpdates)" | jq

{"ok":true,"result":[{"update\_id":510983300,

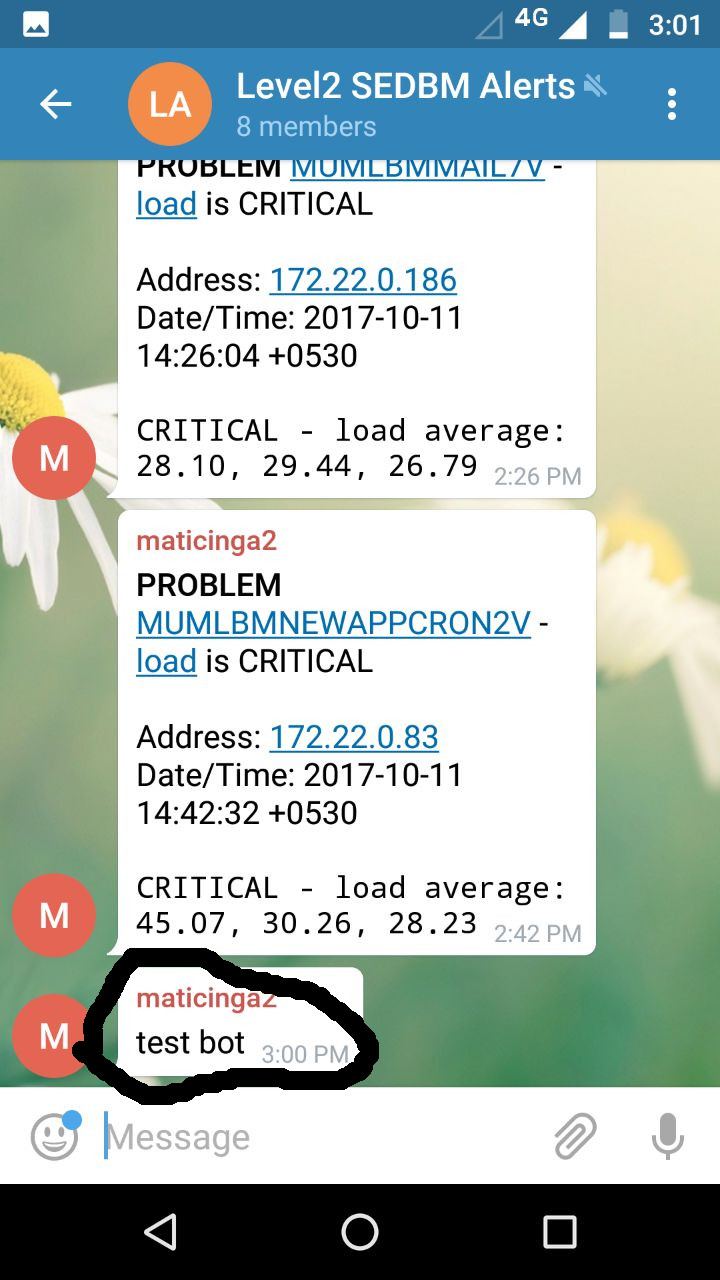
"message":{"message\_id":48460,"from":{"id":148170290,"is\_bot":false,"first\_name":"Alex","last\_name":"Kevin","username":"AlexBennet","language\_code":"en-IN"},"chat":{"id":**-229530029**,"title":"**Level2 SEDBM Alerts**","type":"group","all\_members\_are\_administrators":true},"date":1507639788,"new\_chat\_participant":{"id":477550993,"is\_bot":false,"first\_name":"Suresh","last\_name":"Reddy"},"new\_chat\_member":{"id":477550993,"is\_bot":false,"first\_name":"Suresh","last\_name":"Reddy"},"new\_chat\_members":[{"id":477550993,"is\_bot":false,"first\_name":"Suresh","last\_name":"Reddy"}]}}]}[root@MUMLBMICINGAM1 ~]#

In the above curl request output you can find the chat id **("id":-229530029)** and group name **("title":"Level2 SEDBM Alerts")** that you have added a “maticinga2” bot.

Now you can take the chat id **("id":-229530029)** and token (**330768669:AAFuwJqmic0ecBAf38WRjGv\_m022vp8NY08**) and replace it in the below curl request to send test message to the group via “maticinga2” bot.

[root@MUMLBMICINGAM1 ~]# /usr/bin/curl --silent --output /dev/null --data-urlencode "chat\_id=**-229530029**" --data-urlencode "text=**test bot**" "[https://api.telegram.org/bot**330768669:AAFuwJqmic0ecBAf38WRjGv\_m022vp8NY08**/**sendMessage**](https://api.telegram.org/bot330768669:AAFuwJqmic0ecBAf38WRjGv_m022vp8NY08/sendMessage)"

Once the above curl request is success then go to telegram and check the corresponding group.



Here we got a message on corresponding group from a “maticinga2” bot.

**4.2.4:- Telegram Notification Configurations on Icinga2 master.**

Download the following configuration file from the below link.

telegram host command configuration ==> [telegram-host-notification-command.conf](https://github.com/bodsch/icinga2-telegram-notification/blob/master/telegram-host-notification-command.conf)

telegram service command configuration ==> [telegram-service-notification-command.conf](https://github.com/bodsch/icinga2-telegram-notification/blob/master/telegram-service-notification-command.conf)

<https://github.com/bodsch/icinga2-telegram-notification>

cat the conttent [telegram-host-notification-command.conf](https://github.com/bodsch/icinga2-telegram-notification/blob/master/telegram-host-notification-command.conf) , [telegram-service-notification-command.conf](https://github.com/bodsch/icinga2-telegram-notification/blob/master/telegram-service-notification-command.conf) and append it in the below configuration file.

[root@MUMLBMICINGAM1 conf.d]# vi /etc/icinga2/conf.d/commands.conf

object NotificationCommand "telegram-host-notification" {

import "plugin-notification-command"

command = [ SysconfDir + "/icinga2/scripts/telegram-host-notification.sh" ]

env = {

NOTIFICATIONTYPE = "$notification.type$"

HOSTNAME = "$host.name$"

HOSTALIAS = "$host.display\_name$"

HOSTADDRESS = "$address$"

HOSTSTATE = "$host.state$"

LONGDATETIME = "$icinga.long\_date\_time$"

HOSTOUTPUT = "$host.output$"

NOTIFICATIONAUTHORNAME = "$notification.author$"

NOTIFICATIONCOMMENT = "$notification.comment$"

HOSTDISPLAYNAME = "$host.display\_name$"

TELEGRAM\_BOT\_TOKEN = "YOURAPIKEY"

TELEGRAM\_CHAT\_ID = "$user.pager$"

ICINGAWEB2\_URL = "http://172.22.0.140/icingaweb2"

}

}

object NotificationCommand "telegram-service-notification" {

import "plugin-notification-command"

command = [ SysconfDir + "/icinga2/scripts/telegram-service-notification.sh" ]

env = {

NOTIFICATIONTYPE = "$notification.type$"

SERVICEDESC = "$service.name$"

HOSTNAME = "$host.name$"

HOSTALIAS = "$host.display\_name$"

HOSTADDRESS = "$address$"

SERVICESTATE = "$service.state$"

LONGDATETIME = "$icinga.long\_date\_time$"

SERVICEOUTPUT = "$service.output$"

NOTIFICATIONAUTHORNAME = "$notification.author$"

NOTIFICATIONCOMMENT = "$notification.comment$"

HOSTDISPLAYNAME = "$host.display\_name$"

SERVICEDISPLAYNAME = "$service.display\_name$"

TELEGRAM\_BOT\_TOKEN = ""

TELEGRAM\_CHAT\_ID = "$user.pager$"

// optional

ICINGAWEB2\_URL = "http://172.22.0.140/icingaweb2"

}

}

[root@MUMLBMICINGAM1 conf.d]# vi /etc/icinga2/conf.d/notifications.conf

apply Notification "telegram-icingaadmin-level3" to Host {

import "mail-host-notification"

command = "telegram-host-notification"

users = [ "telegramgroupalerts-level3" ]

times = {

begin = 21m

}

assign where host.vars.notification.mail

}

apply Notification "telegram-icingaadmin-level3" to Service {

import "mail-service-notification"

command = "telegram-service-notification"

users = [ "telegramgroupalerts-level3" ]

times = {

begin = 21m

}

assign where host.vars.notification.mail

}

[root@MUMLBMICINGAM1 ~]# vi /etc/icinga2/scripts/telegram-service-notification.sh

#!/bin/sh

if [ -n "$ICINGAWEB2\_URL" ]; then

HOSTDISPLAYNAME="<a href=\"$ICINGAWEB2\_URL/host/show?host=$HOSTNAME\">$HOSTDISPLAYNAME</a>"

SERVICEDISPLAYNAME="<a href=\"$ICINGAWEB2\_URL/service/show?host=$HOSTNAME&service=$SERVICEDESC\">$SERVICEDISPLAYNAME</a>"

fi

template=$(cat <<TEMPLATE

<strong>$NOTIFICATIONTYPE</strong> $HOSTDISPLAYNAME - $SERVICEDISPLAYNAME is $SERVICESTATE

Address: $HOSTADDRESS

Date/Time: $LONGDATETIME

<pre>$SERVICEOUTPUT</pre>

TEMPLATE

)

if [ -n "$NOTIFICATIONCOMMENT" ]; then

template="$template

Comment: ($NOTIFICATIONAUTHORNAME) $NOTIFICATIONCOMMENT

"

fi

/usr/bin/curl --silent --output /dev/null \

--data-urlencode "chat\_id=${TELEGRAM\_CHAT\_ID}" \

--data-urlencode "text=${template}" \

--data-urlencode "parse\_mode=HTML" \

--data-urlencode "disable\_web\_page\_preview=true" \

"https://api.telegram.org/bot330768669:AAFuwJqmic0ecBAf38WRjGv\_m022vp8NY08/sendMessage"

[root@MUMLBMICINGAM1 ~]# vi /etc/icinga2/scripts/telegram-host-notification.sh

#!/bin/sh

if [ -n "$ICINGAWEB2\_URL" ]; then

HOSTDISPLAYNAME="<a href=\"$ICINGAWEB2\_URL/host/show?host=$HOSTNAME\">$HOSTDISPLAYNAME</a>"

fi

template=$(cat <<TEMPLATE

<strong>$NOTIFICATIONTYPE</strong> - $HOSTDISPLAYNAME is $HOSTSTATE

Host: $HOSTALIAS

Address: $HOSTADDRESS

Date/Time: $LONGDATETIME

<pre>$HOSTOUTPUT</pre>

TEMPLATE

)

if [ -n "$NOTIFICATIONCOMMENT" ]; then

template="$template

Comment: ($NOTIFICATIONAUTHORNAME) $NOTIFICATIONCOMMENT

"

fi

/usr/bin/curl --silent --output /dev/null \

--data-urlencode "chat\_id=${TELEGRAM\_CHAT\_ID}" \

--data-urlencode "text=${template}" \

--data-urlencode "parse\_mode=HTML" \

--data-urlencode "disable\_web\_page\_preview=true" \

"https://api.telegram.org/bot330768669:AAFuwJqmic0ecBAf38WRjGv\_m022vp8NY08/sendMessage"

[esc]:wq

chmod 755 /etc/icinga2/scripts/telegram-\*.sh

[root@MUMLBMICINGAM1 ~]# vi /etc/icinga2/conf.d/users.conf

object User "telegramgroupalerts-level2" {

import "generic-user"

display\_name = "telegramgroupalerts-level2"

pager = "-229530029" #[Groupname's ChatID]

vars.telegram\_bot\_token = "330768669:AAFuwJqmic0ecBAf38WRjGv\_m022vp8NY08" #[Bot Token]

}

[root@MUMLBMICINGAM1 ~]# vi /etc/icinga2/zones.d/global-client-host-conf/MUMLBMNEWAPP1V.conf

object Host "MUMLBMNEWAPP1V" {

check\_command = "hostalive"

address = "172.22.0.90"

zone = "master" //optional trick: sync the required host object to the client, but enforce the "master" zone to execute the check

check\_interval = 1m

retry\_interval = 15s

max\_check\_attempts = 3

vars.endpoint = "MUMLBMNEWAPP1V"

vars.all\_servers\_application\_group = "MUMLBMNEWAPP1V"

vars.bm\_application\_servers\_group = "MUMLBMNEWAPP1V"

vars.bm\_app\_servers\_group = "MUMLBMNEWAPP1V"

vars.datadisk = "MUMLBMNEWAPP1V"

vars.hostgroup = "bm-app-cluster"

vars.notification["mail"] = { groups = [ "sedbm" ]}

vars.notification["telegram" ] = { groups = [ "telegramgroupalerts-level2", "telegramgroupalerts-level3"]} #[add this line]

}

[root@MUMLBMICINGAM1 ~]# icinga2 daemon -C

[root@MUMLBMICINGAM1 ~]# systemctl restart icinga2

**4.3:-** **PagerDuty Integration.**

Create access key in pager duty. Once you got the key. Please do follow a below link.

<https://www.pagerduty.com/docs/guides/icinga2-integration-guide/>

[root@MUMLBMICINGAM1 ~]# yum install perl-libwww-perl perl-Crypt-SSLeay perl-Sys-Syslog

[root@MUMLBMICINGAM1 ~]# yum install perl-LWP-Protocol-https

[root@MUMLBMICINGAM1 ~]# vi /etc/icinga2/conf.d/pagerduty-icinga2.conf

object User "pagerduty\_appsupport" {

pager = "e26a8c8e631c416680417af01c12d59f"

display\_name = "PagerDuty Notification User"

states = [ OK, Warning, Critical, Unknown, Up, Down ]

types = [ Problem, Acknowledgement, Recovery ]

}

object User "pagerduty\_appsupport\_immediate" {

# pager = "5d73079a8bf0418d81cfba971ee7b0db"

pager = "9d21c1ba34c84d97a4a437a2728f6967"

display\_name = "PagerDuty Notification User"

states = [ OK, Warning, Critical, Unknown, Up, Down ]

types = [ Problem, Acknowledgement, Recovery ]

}

object User "pagerduty\_dbsupport" {

pager = "a72c729613554824b6b469d2a9ac1761"

display\_name = "PagerDuty Notification User"

states = [ OK, Warning, Critical, Unknown, Up, Down ]

types = [ Problem, Acknowledgement, Recovery ]

}

object User "pagerduty\_dbsupport\_immediate" {

pager = "b5eab49a090e4c65af9850e5c9f8abd4"

display\_name = "PagerDuty Notification User"

states = [ OK, Warning, Critical, Unknown, Up, Down ]

types = [ Problem, Acknowledgement, Recovery ]

}

object NotificationCommand "notify-service-by-pagerduty" {

import "plugin-notification-command"

command = [ SysconfDir + "/icinga2/scripts/pagerduty\_icinga.pl" ]

arguments = {

"enqueue" = {

skip\_key = true

order = 0

value = "enqueue"

}

"-f" = {

order = 1

value = "pd\_nagios\_object=service"

}

}

env = {

"ICINGA\_CONTACTPAGER" = "$user.pager$"

"ICINGA\_NOTIFICATIONTYPE" = "$notification.type$"

"ICINGA\_SERVICEDESC" = "$service.name$"

"ICINGA\_HOSTNAME" = "$host.name$"

"ICINGA\_HOSTALIAS" = "$host.display\_name$"

"ICINGA\_SERVICESTATE" = "$service.state$"

"ICINGA\_SERVICEOUTPUT" = "$service.output$"

}

}

object NotificationCommand "notify-host-by-pagerduty" {

import "plugin-notification-command"

command = [ SysconfDir + "/icinga2/scripts/pagerduty\_icinga.pl" ]

arguments = {

"enqueue" = {

skip\_key = true

order = 0

value = "enqueue"

}

"-f" = {

order = 1

value = "pd\_nagios\_object=host"

}

}

env = {

"ICINGA\_CONTACTPAGER" = "$user.pager$"

"ICINGA\_NOTIFICATIONTYPE" = "$notification.type$"

"ICINGA\_HOSTNAME" = "$host.name$"

"ICINGA\_HOSTALIAS" = "$host.display\_name$"

"ICINGA\_HOSTSTATE" = "$host.state$"

"ICINGA\_HOSTOUTPUT" = "$host.output$"

}

}

apply Notification "pagerduty-host-appsupport" to Host {

command = "notify-host-by-pagerduty"

states = [ Up, Down ]

types = [ Problem, Acknowledgement, Recovery ]

period = "24x7"

users = [ "pagerduty\_appsupport" ]

assign where **host.vars.enable\_pagerduty\_host\_appsupport** == true

}

apply Notification "pagerduty-host-dbsupport" to Host {

command = "notify-host-by-pagerduty"

states = [ Up, Down ]

types = [ Problem, Acknowledgement, Recovery ]

period = "24x7"

users = [ "pagerduty\_appsupport" ]

assign where **host.vars.enable\_pagerduty\_host\_dbsupport** == true

}

apply Notification "pagerduty-appsupport-service1" to Service {

command = "notify-service-by-pagerduty"

states = [ Critical ]

types = [ Problem, Recovery ]

period = "24x7"

users = [ "pagerduty\_appsupport\_immediate" ]

assign where **service.vars.pagerduty\_alert\_immediate\_appsupport** == true

}

apply Notification "pagerduty-appsupport-service2" to Service {

command = "notify-service-by-pagerduty"

states = [ OK, Warning, Critical, Unknown ]

types = [ Problem, Acknowledgement, Recovery ]

period = "24x7"

users = [ "pagerduty\_appsupport" ]

times = {

begin = 30m

}

assign where **service.vars.enable\_pagerduty\_service\_appsupport** == true

}

apply Notification "pagerduty-dbsupport-service1" to Service {

command = "notify-service-by-pagerduty"

states = [ Critical ]

types = [ Problem, Recovery ]

period = "24x7"

users = [ "pagerduty\_dbsupport\_immediate" ]

assign where **service.vars.pagerduty\_alert\_immediate\_dbsupport** == true

}

[root@MUMLBMICINGAM1 ~]# cd /etc/icinga2/scripts/

[root@MUMLBMICINGAM1 ~]# wget https://raw.github.com/PagerDuty/pagerduty-icinga-pl/master/pagerduty\_icinga.pl

[root@MUMLBMICINGAM1 ~]# chmod 755 /etc/icinga2/scripts/pagerduty\_icinga.pl

[root@MUMLBMICINGAM1 ~]# crontab -u icinga -e

\* \* \* \* \* /usr/local/bin/pagerduty\_icinga.pl flush

[root@MUMLBMICINGAM1 ~]# vi /etc/icinga2/zones.d/global-client-host-conf/MUMLBMNEWAPP1V.conf

object Host "MUMLBMNEWAPP1V" {

check\_command = "hostalive"

address = "172.22.0.90"

zone = "master" //optional trick: sync the required host object to the client, but enforce the "master" zone to execute the check

check\_interval = 1m

retry\_interval = 15s

max\_check\_attempts = 3

vars.endpoint = "MUMLBMNEWAPP1V"

vars.all\_servers\_application\_group = "MUMLBMNEWAPP1V"

vars.bm\_application\_servers\_group = "MUMLBMNEWAPP1V"

vars.bm\_app\_servers\_group = "MUMLBMNEWAPP1V"

vars.datadisk = "MUMLBMNEWAPP1V"

vars.hostgroup = "bm-app-cluster"

vars.notification["mail"] = { groups = [ "sedbm" ]}

vars.notification["telegram" ] = { groups = [ "telegramgroupalerts-level2", "telegramgroupalerts-level3" ]}

**vars.enable\_pagerduty\_host\_appsupport** = true #[add this line]

}

[root@MUMLBMICINGAM1 ~]# icinga2 daemon -C

[root@MUMLBMICINGAM1 ~]# vi /etc/icinga2/zones.d/master/disk1-all\_matrimony\_servers.conf

apply Service "diskspace /" {

check\_command = "disk"

check\_interval = 5m

retry\_interval = 15s

max\_check\_attempts = 4

vars.disk\_units = "GB"

vars.disk\_partitions = [ "/" ]

vars.disk\_wfree = "10%"

vars.disk\_cfree = "1%"

vars.pagerduty\_alert\_immediate\_appsupport = true #[add this line]

command\_endpoint = host.vars.endpoint

assign where host.vars.endpoint

}

[root@MUMLBMICINGAM1 ~]# icinga2 daemon -C

[root@MUMLBMICINGAM1 ~]# systemctl restart icinga2

**4.4:- SMS Integration.**

[root@MUMLBMICINGAM1 ~]# vi /etc/icinga2/conf.d/notifications.conf

apply Notification "escalation-sms-2nd-level" to Host {

import "mail-host-notification"

command = "sms-host-notification"

user\_groups = [ "level2" ]

times = {

begin = 21m

end = 30m

}

assign where host.vars.notification.mail

}

apply Notification "escalation-sms-1st-level" to Host {

import "mail-host-notification"

command = "sms-host-notification"

user\_groups = [ "level1" ]

times = {

begin = 11m

end = 20m

}

assign where host.vars.notification.mail

}

apply Notification "escalation-sms-2nd-level" to Service {

import "mail-service-notification"

command = "sms-service-notification"

user\_groups = [ "level2" ]

times = {

begin = 21m

end = 30m

}

assign where host.vars.notification.mail

}

apply Notification "escalation-sms-1st-level" to Service {

import "mail-service-notification"

command = "sms-service-notification"

user\_groups = [ "level1" ]

times = {

begin = 11m

end = 20m

}

assign where host.vars.notification.mail

}

[root@MUMLBMICINGAM1 ~]# vi /etc/icinga2/conf.d/commands.conf

object NotificationCommand "sms-host-notification" {

command = [ SysconfDir + "/icinga2/scripts/sms-host-notification.sh" ]

env = {

NOTIFICATIONTYPE = "$notification.type$"

HOSTALIAS = "$host.display\_name$"

HOSTADDRESS = "$address$"

HOSTSTATE = "$host.state$"

LONGDATETIME = "$icinga.long\_date\_time$"

HOSTOUTPUT = "$host.output$"

NOTIFICATIONAUTHORNAME = "$notification.author$"

NOTIFICATIONCOMMENT = "$notification.comment$"

HOSTDISPLAYNAME = "$host.display\_name$"

USEREMOBILE = "$user.pager$"

}

}

object NotificationCommand "sms-service-notification" {

command = [ SysconfDir + "/icinga2/scripts/sms-service-notification.sh" ]

env = {

NOTIFICATIONTYPE = "$notification.type$"

SERVICEDESC = "$service.name$"

HOSTALIAS = "$host.display\_name$"

HOSTADDRESS = "$address$"

SERVICESTATE = "$service.state$"

LONGDATETIME = "$icinga.long\_date\_time$"

SERVICEOUTPUT = "$service.output$"

NOTIFICATIONAUTHORNAME = "$notification.author$"

NOTIFICATIONCOMMENT = "$notification.comment$"

HOSTDISPLAYNAME = "$host.display\_name$"

SERVICEDISPLAYNAME = "$service.display\_name$"

USEREMOBILE = "$user.pager$"

}

}

[root@MUMLBMICINGAM1 ~]# vi /etc/icinga2/scripts/sms-host-notification.sh

#!/bin/sh

template=`cat <<TEMPLATE

\*\*\*\*\* Icinga \*\*\*\*\*

Notification Type: $NOTIFICATIONTYPE

Host: $HOSTALIAS

Address: $HOSTADDRESS

State: $HOSTSTATE

Date/Time: $LONGDATETIME

Additional Info: $HOSTOUTPUT

Comment: [$NOTIFICATIONAUTHORNAME] $NOTIFICATIONCOMMENT

TEMPLATE

`

curl --data "userId=bmcom&pass=bmcom&contenttype=1&to=$USEREMOBILE&from=FROMBM&text=(alert)$template&selfid=true&alert=1&dlrreq=true" http://push3.maccesssmspush.com/servlet/com.aclwireless.pushconnectivity.listeners.TextListener

[root@MUMLBMICINGAM1 ~]# vi /etc/icinga2/scripts/sms-service-notification.sh

#!/bin/sh

template=`cat <<TEMPLATE

\*\*\*\*\* Icinga \*\*\*\*\*

Notification Type: $NOTIFICATIONTYPE

Service: $SERVICEDESC

Host: $HOSTALIAS

Address: $HOSTADDRESS

State: $SERVICESTATE

Date/Time: $LONGDATETIME

Additional Info: $SERVICEOUTPUT

Comment: [$NOTIFICATIONAUTHORNAME] $NOTIFICATIONCOMMENT

TEMPLATE

`

curl --data "userId=bmcom&pass=bmcom&contenttype=1&to=$USEREMOBILE&from=FROMBM&text=(alert)$template&selfid=true&alert=1&dlrreq=true" <http://push3.maccesssmspush.com/servlet/com.aclwireless.pushconnectivity.listeners.TextListener>

[root@MUMLBMICINGAM1 ~]# chmod 755 /etc/icinga2/scripts/sms\*.sh

[root@MUMLBMICINGAM1 ~]# vi /etc/icinga2/conf.d/users.conf

object User "thirumurugan" {

import "generic-user"

display\_name = "Thirumurugan KK"

groups = [ "**level2**" ]

email = "thiru.murugan@matrimony.com"

pager = "9884323735"

}

object User "kalamegam" {

import "generic-user"

display\_name = "Kalamegam R"

groups = [ "**level2**" ]

email = "kalamegam.ramachandran@matrimony.com"

pager = "9980011153"

}

object UserGroup "**level1**" {

display\_name = "Level1 Esclations"

}

object UserGroup "**level2**" {

display\_name = "Level2 Esclations"

}

[root@MUMLBMICINGAM1 ~]# cat /etc/icinga2/zones.d/global-client-host-conf/MUMLBMNEWAPP1V.conf

object Host "MUMLBMNEWAPP1V" {

check\_command = "hostalive"

address = "172.22.0.90"

zone = "master" //optional trick: sync the required host object to the client, but enforce the "master" zone to execute the check

check\_interval = 1m

retry\_interval = 15s

max\_check\_attempts = 3

vars.endpoint = "MUMLBMNEWAPP1V"

vars.all\_servers\_application\_group = "MUMLBMNEWAPP1V"

vars.bm\_application\_servers\_group = "MUMLBMNEWAPP1V"

vars.bm\_app\_servers\_group = "MUMLBMNEWAPP1V"

vars.datadisk = "MUMLBMNEWAPP1V"

vars.hostgroup = "bm-app-cluster"

vars.enable\_pagerduty\_host\_appsupport = true

vars.notification["mail"] = { groups = [ "sedbm" ]}

vars.notification["telegram" ] = { groups = [ "telegramgroupalerts-level2", "telegramgroupalerts-level3" ]}

vars.notification[ "sms" ] = { groups = [ "level1", "level2" ] }

}

[root@MUMLBMICINGAM1 ~]# icinga2 daemon -C

[root@MUMLBMICINGAM1 ~]# systemctl restart icinga2

**Step5:- Defining and configuring Custom commands service checks on Icinga2 master server.**

By default the commands configurations will be in the following path while installing nagios-pluggins-all packages on all server.

[root@MUMLBMICINGAM1 ~]# ls -tlr /usr/share/icinga2/include/command-plugins.conf

-rw-r--r-- 1 root root 54828 Jul 10 13:15 /usr/share/icinga2/include/command-plugins.conf

[root@MUMLBMICINGAM1 ~]#

Now let us see how to define custom command and configure service check for that user defined commands.

By default nagios plugins will be store in the following path.

[root@MUMLBMICINGAM1 ~]# ls -ltr /usr/lib64/nagios/plugins/check\_ping

-rwxr-xr-x 1 root root 66352 Dec 17 2016 /usr/lib64/nagios/plugins/check\_ping

[root@MUMLBMICINGAM1 ~]#

We can create our custom command in the same path or else we can download some plugins available in nagios plugins repository. For Example:-

Download **cpu.sh** from the following link and put it into the nagios plugin path.

<https://exchange.nagios.org/directory/Plugins/System-Metrics/CPU->

Usage-and-Load/check\_cpu-2Esh/details

mv /tmp/check\_cpu.sh /usr/lib64/nagios/plugins/

chmod 755 /usr/lib64/nagios/plugins/**check\_cpu.sh**

[root@MUMLBMICINGAM1 ~]# vi /etc/icinga2/zones.d/global-templates/commands.conf

object CheckCommand "check\_cpu" {

command = [ PluginDir + "/**check\_cpu.sh**" ]

arguments = {

"-w" = "$global\_warning$"

"-uw" = "$user\_cpu\_warning$"

"-iw" = "$iowait\_cpu\_warning$"

"-sw" = "$system\_cpu\_warning$"

"-c" = "$global\_critical$"

"-uc" = "$user\_cpu\_critical$"

"-ic" = "$iowait\_cpu\_critical$"

"-sc" = "$system\_cpu\_warning$"

"-i" = "$interval\_seconds\_for\_iostat$"

"-n" = "$number\_report\_for\_iostat$"

}

}

Once the above file is created under a path /etc/icinga2/zones.d/ global-templates it will automatically synced to all the client once icinga2 service is restarted.

[root@MUMLBMICINGAM1 ~]# vi /etc/icinga2/zones.d/master/cpuusage-all\_matrimony\_servers.conf

apply Service "cpuusage" {

check\_command = "check\_cpu"

vars.user\_cpu\_warning = "95"

vars.user\_cpu\_critical = "98"

check\_interval = 10m

retry\_interval = 15s

max\_check\_attempts = 4

command\_endpoint = host.vars.endpoint

assign where host.vars.endpoint

ignore where match("MUMLBM\*MIMA\*", host.vars.endpoint)

}

[root@MUMLBMICINGAM1 ~]# icinga2 daemon -C

[root@MUMLBMICINGAM1 ~]# systemctl restart icinga2

Now you check the command.conf file in Icinga2 client server.

[root@MUMLBMNEWAPP1V ~]# ls -ltr /var/lib/icinga2/api/zones/global-templates/\_etc/commands.conf

-rw-r--r-- 1 icinga icinga 1281 Jul 28 11:18 /var/lib/icinga2/api/zones/global-templates/\_etc/commands.conf

[root@MUMLBMNEWAPP1V ~]# cat /var/lib/icinga2/api/zones/global-templates/\_etc/commands.conf

object CheckCommand "check\_cpu" {

command = [ PluginDir + "/check\_cpu.sh" ]

arguments = {

"-w" = "$global\_warning$"

"-uw" = "$user\_cpu\_warning$"

"-iw" = "$iowait\_cpu\_warning$"

"-sw" = "$system\_cpu\_warning$"

"-c" = "$global\_critical$"

"-uc" = "$user\_cpu\_critical$"

"-ic" = "$iowait\_cpu\_critical$"

"-sc" = "$system\_cpu\_warning$"

"-i" = "$interval\_seconds\_for\_iostat$"

"-n" = "$number\_report\_for\_iostat$"

}

}

**Step6:- Short Rewind about Master and Client Configurations File.**

**6.1:- Icinga Architecture / Technical Terms.**

Endpoints ==> All nodes are called as Endpoints.

Master zone ==> List of Endpoints in Master zone.

satellite zone ==> List of Endpoints in Satellite zone/If there is no satellite then by default all end points act as a satellite exceptmaster master.

client zone ==> List of Endpoints in Client Zone.

**6.2:- Icinga2 List of commands.**

Systemctl start/stop/restart icinga2 ==> start/stop/restart service.

systemctl enable icinga2 ==> enable service onboot.

**icinga2 daemon -C** ==> syntax check.

icinga2 object list --type service ==> to list all the services.

icinga2 object list --type zone ==> to list the zones.

icinga2 object list --type user ==> to list the user information.

icinga2 object list --type dependencies ==> to list the dependency. services.

icinga2 object list --type hostgroup ==> to list the host grouping.

icinga2 object list --type servicegroup ==> to list the service group

icinga2 object list --type host ==> to list the host informations

icinga2 object list --type PerfdataWriter ==> to list the details of BI(graph) data.

icinga2 object list --type IdoMysqlConnection ==> to list details of MySQL.

**6.3:- List Of Conf files need to check on Icinga Client.**

/etc/icinga2/zones.conf ==> Client Zone Information.

/etc/icinga2/constants.conf ==> Client Endpoint Information.

/etc/icinga2/icinga2.conf ==> have to check whether conf.d is commented or not.

**6.4:- List of Conf Files need to check on Master.**

/etc/icinga2/zones.conf ==> Master Zone Information.

/etc/icinga2/constants.conf ==> Master Endpoint Information.

/etc/icinga2/icinga2.conf ==> Icinga2 main configuration file.

/etc/icinga2/zones.d/master/global-client-zone-conf/<hostname>.conf ==> Clients zone information

/etc/icinga2/conf.d/users.conf ==> User and Group Information's for notification.

/etc/icinga2/conf.d/hostgroups.conf ==> Cluster based naming group of

icinga2 client server for easy identifications.

/etc/icinga2/conf.d/commands.conf ==> notifcation command objects

/etc/icinga2/conf.d/templates.conf ==> notifications template

/etc/icinga2/conf.d/timeperiods.conf ==> timperiods objects for notifications

/etc/icinga2/conf.d/servicegroups.conf ==> gouping service configuration

/etc/icinga2/conf.d/pagerduty-icinga2.conf ==> pagerduty notification entire configuration

/etc/icinga2/zones.d/global-templates/commands.conf ==> custom commands objects for all clients

/etc/icinga2/zones.d/global-client-host-conf/<HOSTNAME>.conf ==> clients host configurations

/etc/icinga2/zones.d/master/\*.conf ==> service configuration for all clients

/etc/icinga2/zones.d/master/bm\_services/\*.conf ==> service conf for bm nodes

/etc/icinga2/zones.d/master/bm\_services/app\_servers/\*.conf ==> service conf for bm app cluster

/etc/icinga2/zones.d/master/bm\_services/cassandra\_servers/\*.conf ==> service conf for bm cassandra cluster

/etc/icinga2/zones.d/master/bm\_services/img\_servers/\*.conf ==> service conf for bm img cluster

/etc/icinga2/zones.d/master/bm\_services/redis\_servers/\*.conf ==> service conf for bm redis cluster

/etc/icinga2/zones.d/master/bm\_services/spx\_servers/\*.conf ==> service conf for bm spx cluster

/etc/icinga2/zones.d/master/cbs\_services/ ==> service conf for cbs nodes

/etc/icinga2/zones.d/master/cbs\_services/web\_servers/ ==> service conf for cbs web cluster

/etc/icinga2/zones.d/master/cbs\_services/img\_servers/ ==> service conf for cbs img cluster

/etc/icinga2/zones.d/master/dependencies/bmservers/ ==> dependency conf for bm nodes

/etc/icinga2/zones.d/master/dependencies/cbsservers/ ==> dependency conf for cbs nodes

**6.5:- Nagios Default Plugin Path for all Endpoints.**

/usr/share/icinga2/include/command-plugins.conf

**6.6:- Global Sync Path from Master to All Client Servers.**

/var/lib/icinga2/api/zones/global-templates/\_etc/commands.conf ==> Synchronized from Iciinga2 master.