Icinga Installation

About Icinga 2

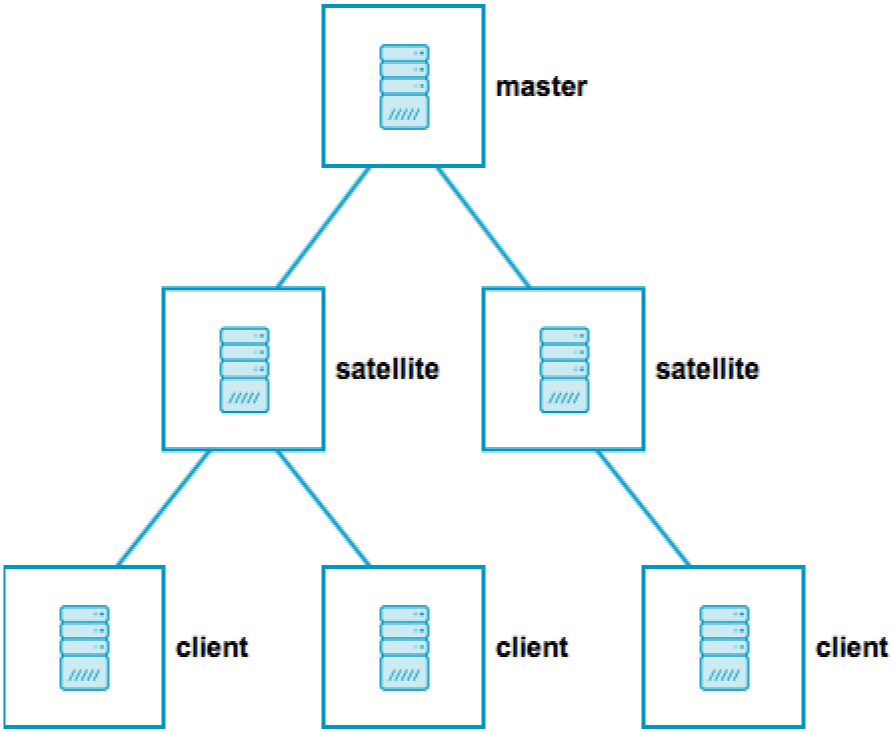
Icinga2 is an open source monitoring system which checks the availability of your network resources, notifies users of outages and generates performance data for reporting.

Scalable and extensible, Icinga 2 can monitor large, complex environments across multiple locations.

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.



Installation Steps:

Setting up Icinga 2

First off you have to install Icinga 2. The preferred way of doing this is to use the official package repositories depending on which operating system and distribution you are running.

You need to add the Icinga repository to your package management configuration.

Ubuntu:

```
# wget -O - https://packages.icinga.com/icinga.key | apt-key add -
# echo 'deb https://packages.icinga.com/ubuntu icinga-xenial main'
>/etc/apt/sources.list.d/icinga.list
# apt-get update
```

Installing Icinga 2

You can install Icinga 2 by using your distribution's package manager to install the icinga2package.

Debian/Ubuntu:

```
# apt-get install icinga2
```

Enabled Features during Installation

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 to see which features are enabled and disabled.

```
# icinga2 feature list
Disabled features: api command compatlog debuglog gelf graphite icinga status ido-mysql ido-pgsql influxdb livestatus opentsdb perfdata statusdata syslog
Enabled features: checker mainlog notification
```

Setting up Check Plugins

Without plugins Icinga 2 does not know how to check external services. The Monitoring Plugins Project provides an extensive set of plugins which can be used with Icinga 2 to check whether services are working properly.

The recommended way of installing these standard plugins is to use your distribution's package manager.

Debian/Ubuntu:

```
# apt-get install monitoring-plugins
```

Running Icinga 2

Init Script

Icinga 2's init script is installed in /etc/init.d/icinga2 (/usr/local/etc/rc.d/icinga2 on FreeBSD) by default:

```
# /etc/init.d/icinga2
Usage: /etc/init.d/icinga2 {start|stop|restart|reload|checkconfig|status}
```

Systemd Service

Some distributions (e.g. Fedora, openSUSE and RHEL/CentOS 7) use Systemd. The Icinga 2 packages automatically install the necessary Systemd unit files.

The Icinga 2 Systemd service can be (re-)started, reloaded, stopped and also queried for its current status.

```
# systemctl status icinga2
icinga2.service - Icinga host/service/network monitoring system
   Loaded: loaded (/usr/lib/systemd/system/icinga2.service; disabled)
   Active: active (running) since Mi 2014-07-23 13:39:38 CEST; 15s ago
  Process: 21692 ExecStart=/usr/sbin/icinga2 -c ${ICINGA2_CONFIG_FILE} -d -e
${ICINGA2_ERROR_LOG} -u ${ICINGA2_USER} -g ${ICINGA2_GROUP} (code=exited,
status=0/SUCCESS)
```

```
# systemctl enable icinga2
```

```
# systemctl restart icinga2
```

Configuration Syntax Highlighting using Vim (Optional)

Install the package vim-icinga2 with your distribution's package manager.

Debian/Ubuntu:

```
# apt-get install vim-icinga2 vim-addon-manager
# vim-addon-manager -w install icinga2
Info: installing removed addon 'icinga2' to /var/lib/vim/addons
```

Steps to be installed only on the master Node:

Configuring DB IDO MySQL

Installing MySQL database server

Debian/Ubuntu:

```
# apt-get install mysql-server mysql-client
# mysql_secure_installation
```

Installing the IDO modules for MySQL

The next step is to install the icinga2-ido-mysql package using your distribution's package manager.

Debian/Ubuntu:

```
# apt-get install icinga2-ido-mysql
```

Setting up the MySQL database

Set up a MySQL database for Icinga 2:

```
# mysql -u root -p

mysql>  CREATE DATABASE icinga;
mysql>  GRANT SELECT, INSERT, UPDATE, DELETE, DROP, CREATE VIEW, INDEX,
EXECUTE ON icinga.* TO 'icinga'@'localhost' IDENTIFIED BY 'icinga';
mysql> quit
```

After creating the database you can import the Icinga 2 IDO schema using the following command:

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

Enabling the IDO MySQL module

The package provides a new configuration file that is installed in /etc/icinga2/features-available/ido-mysql.conf. You will need to update the database credentials in this file.

All available attributes are explained in the IdoMysqlConnection object chapter.

You can enable the ido-mysql feature configuration file using icinga2 feature enable:

```
# icinga2 feature enable ido-mysql
Module 'ido-mysql' was enabled.
Make sure to restart Icinga 2 for these changes to take effect.
```

Webserver

Debian/Ubuntu:

```
# apt-get install apache2
```

Setting Up Icinga 2 REST API

Icinga Web 2 and other web interfaces require the REST API to send actions (reschedule check, etc.) and query object details.

You can run the CLI command icinga2 api setup to enable the api feature and set up certificates as well as a new API user root with an auto-generated password in the/etc/icinga2/conf.d/api-users.conf configuration file:

```
# icinga2 api setup
```

Edit the api-users.conf file and add a new ApiUser object. Specify the permissions attribute with minimal permissions required by Icinga Web 2.

```
# vim /etc/icinga2/conf.d/api-users.conf

object ApiUser "icingaweb2" {
  password = "Wijsn8Z9eRs5E25d"
  permissions = [ "status/query", "actions/*", "objects/modify/*", "objects/query/*" ]
}
```

Make sure to restart Icinga 2 to activate the configuration.

Installing Icinga Web 2

Installing Requirements

- Icinga 2 with the IDO database backend (MySQL or PostgreSQL)
- A web server, e.g. Apache or Nginx
- PHP >= 5.3.2 with the following modules installed: cURL, gettext, intl, mbstring, OpenSSL and xml
- Default time zone configured for PHP in the php.ini file
- LDAP PHP library when using Active Directory or LDAP for authentication
- MySQL or PostgreSQL PHP libraries

Installing Icinga Web 2 from Package

Below is a list of official package repositories for installing Icinga Web 2 for various operating systems.

Setting up Package Repositories

You need to add the Icinga repository to your package management configuration for installing Icinga Web 2. If you've already configured your OS to use the Icinga repository for installing Icinga 2, you may skip this step. Below is a list with **examples** for various distributions.

Ubuntu Xenial:

```
wget -O - http://packages.icinga.com/icinga.key | apt-key add -
add-apt-repository 'deb http://packages.icinga.com/ubuntu icinga-xenial main'
apt-get update
```

Installing Icinga Web 2

You can install Icinga Web 2 by using your distribution's package manager to install the icingaweb2 package. Below is a list with examples for various distributions. The additional package icingacli is necessary on RPM based systems for being able to follow further steps in this guide. In DEB based systems, the icingacli binary is included in the icingaweb2 package.

Debian and Ubuntu:

```
apt-get install icingaweb2 icingacli
```

Preparing Web Setup

You can set up Icinga Web 2 quickly and easily with the Icinga Web 2 setup wizard which is available the first time you visit Icinga Web 2 in your browser. When using the web setup you are required to authenticate using a token. In order to generate a token use the

```
icingacli:  icingacli setup token create
```

In case you do not remember the token you can show it using the

```
icingacli: icingacli setup token show
```

Preparing Web Setup on Debian

On Debian, you need to manually create a database and a database user prior to starting the web wizard. This is due to local security restrictions whereas the web wizard cannot create a database/user through a local unix domain socket.

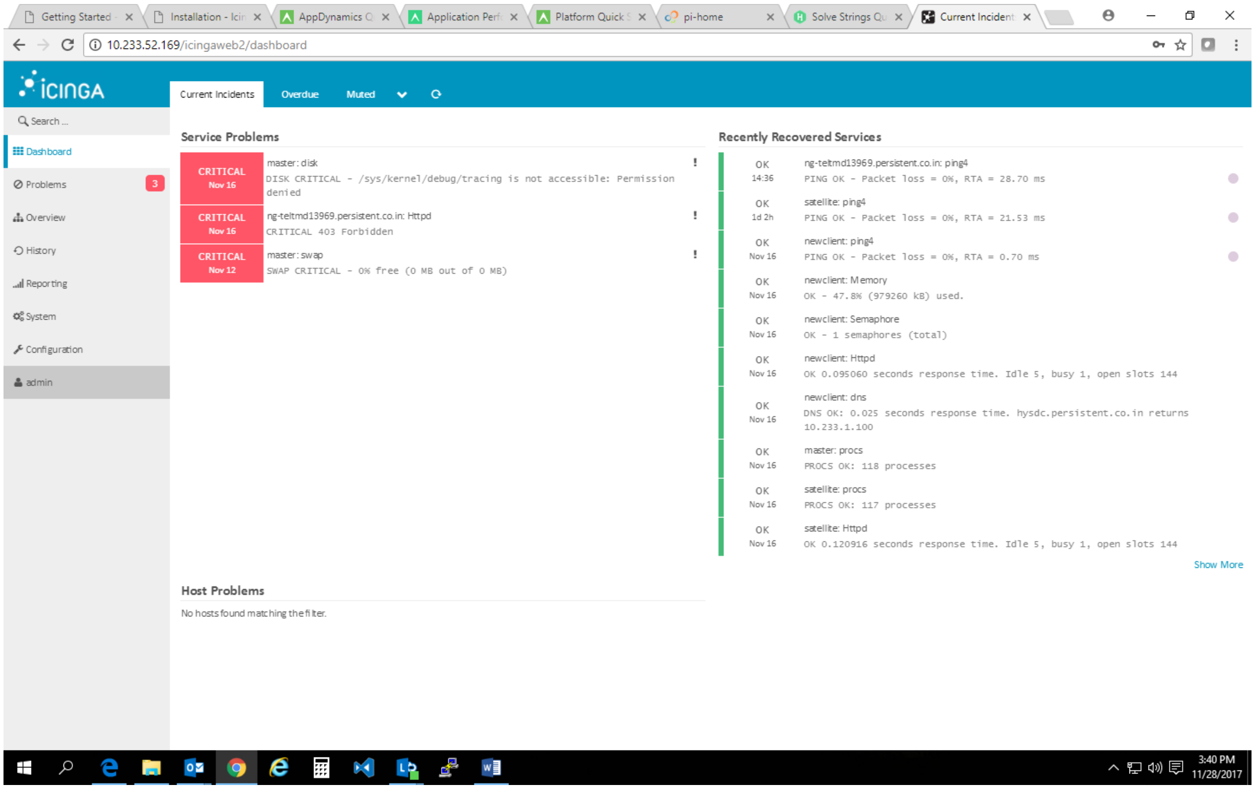
```
MariaDB [mysql]> CREATE DATABASE icingaweb2;

MariaDB [mysql]> GRANT ALL ON icingaweb2.* TO icingaweb2@localhost IDENTIFIED BY 'CHANGEME';
```

You may also create a separate administrative account with all privileges instead.

Starting Web Setup

Finally visit Icinga Web 2 in your browser to access the setup wizard and complete the installation: /icingaweb2/dashboard



Service Problems

CRITICAL Nov 16 — master: disk

```
DISK CRITICAL - /sys/kernel/debug/tracing is not accessible: Permission denied
```

CRITICAL Nov 16 — ng-teltmd13969.persistent.co.in: Httpd

```
CRITICAL 403 Forbidden
```

CRITICAL Nov 12 — master: swap

```
SWAP CRITICAL - 0% free (0 MB out of 0 MB)
```

Recently Recovered Services

OK 14:36 — ng-teltmd13969.persistent.co.in: ping4

```
PING OK - Packet loss = 0%, RTA = 28.70 ms
```

OK 1d 2h — satellite: ping4

```
PING OK - Packet loss = 0%, RTA = 21.53 ms
```

OK Nov 16 — newclient: ping4

```
PING OK - Packet loss = 0%, RTA = 0.70 ms
```

OK Nov 16 — newclient: Memory

```
OK - 47.8% (979260 kB) used.
```

OK Nov 16 — newclient: Semaphore

```
OK - 1 semaphores (total)
```

OK Nov 16 — newclient: Httpd

```
OK 0.095060 seconds response time. Idle 5, busy 1, open slots 144
```

OK Nov 16 — newclient: dns

```
DNS OK: 0.025 seconds response time. hysdc.persistent.co.in returns 10.233.1.100
```

OK Nov 16 — master: procs

```
PROCS OK: 118 processes
```

OK Nov 16 — satellite: procs

```
PROCS OK: 117 processes
```

OK Nov 16 — satellite: Httpd

```
OK 0.120916 seconds response time. Idle 5, busy 1, open slots 144
```

Show More

Host Problems

No hosts found matching the filter.

Master Setup

This section explains how to install a central single master node using the node wizard command. If you prefer to do an automated installation, please refer to the automated setup section.

Install the Icinga 2 package and setup the required plugins if you haven't done so already.

Note: Windows is not supported for a master node setup.

The next step is to run the node wizard CLI command. Prior to that ensure to collect the required information:

The setup wizard will ensure that the following steps are taken:

- Enable the api feature.
- Generate a new certificate authority (CA) in /var/lib/icinga2/ca if it doesn’t exist.
- Create a certificate for this node signed by the CA key.
- Update the zones.conf file with the new zone hierarchy.
- Update the ApiListener and constants configuration.

Here is an example of a master setup for the icinga2-master1.localdomain node on CentOS 7:

```
[root@icinga2-master1.localdomain /]# icinga2 node wizard

Welcome to the Icinga 2 Setup Wizard!

We will guide you through all required configuration details.

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

Starting the Master setup routine...

Please specify the common name (CN) [icinga2-master1.localdomain]: icinga2-master1.localdomain
Reconfiguring Icinga...
Checking for existing certificates for common name 'master1'...
Generating master configuration for Icinga 2.

Please specify the API bind host/port (optional):
Bind Host []:
Bind Port []:

Done.

Now restart your Icinga 2 daemon to finish the installation!
```

You can verify that the CA public and private keys are stored in the /var/lib/icinga2/cadirectory. Keep this path secure and include it in your backups.

In case you lose the CA private key you have to generate a new CA for signing new client certificate requests. You then have to also re-create new signed certificates for all existing nodes.

Once the master setup is complete, you can also use this node as primary CSR auto-signing master. The following section will explain how to use the CLI commands in order to fetch their signed certificate from this master node.

Configuring Zones.conf File:

```
object Endpoint "icinga2-master1.localdomain" {
  host = "192.168.56.101"
}

object Endpoint "icinga2-satellite1.localdomain" {
  host = "192.168.56.105"
}

object Zone "master" {
  endpoints = [ "icinga2-master1.localdomain" ]
}

object Zone "satellite" {
  endpoints = [ "icinga2-satellite1.localdomain" ]
  parent = "master"
}

/* sync global commands */
object Zone "global-templates" {
  global = true
}
```

Steps to be installed on the Satellite Node:

The setup wizard will ensure that the following steps are taken:

- Enable the `api` feature.
- Generate a new certificate authority (CA) in `/var/lib/icinga2/ca` if it doesn't exist.
- Create a certificate for this node signed by the CA key.
- Update the zones.conf file with the new zone hierarchy.
- Update the ApiListener and constants configuration.

Here is an example of a master setup for the `icinga2-master1.localdomain` node on CentOS 7:

```
[root@icinga2-master1.localdomain /]# icinga2 node wizard

Welcome to the Icinga 2 Setup Wizard!

We will guide you through all required configuration details.
```

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

Starting the Master setup routine...

Please specify the common name (CN) [icinga2-master1.localdomain]: icinga2-master1.localdomain
Reconfiguring Icinga...
Checking for existing certificates for common name 'master1'...
Generating master configuration for Icinga 2.

Please specify the API bind host/port (optional):
Bind Host []:
Bind Port []:

Done.

Now restart your Icinga 2 daemon to finish the installation!
```

Note: Ensure setting up a satellite node and also have a direct connection from the master node and also accept the commands from the master node and also auto sign 7the certificate from the master Node.

Get a token from the master Node by executing the following command

```
[root@icinga2-master1.localdomain /]# icinga2 pki ticket --cn icinga2-client1.localdomain
```

Configuring Zones.conf File:

```
object Endpoint "icinga2-master1.localdomain" {
  host = "192.168.56.101"
}

object Endpoint "icinga2-satellite1.localdomain" {
  host = "192.168.56.105"
}

object Zone "master" {
  endpoints = [ "icinga2-master1.localdomain" ]
}

object Zone "satellite" {
  endpoints = [ "icinga2-satellite1.localdomain" ]
  parent = "master"
}
```

```
/* sync global commands */
object Zone "global-templates" {
  global = true
}
```

Steps to be installed on the Client Node:

The setup wizard will ensure that the following steps are taken:

- Enable the api feature.
- Generate a new certificate authority (CA) in /var/lib/icinga2/ca if it doesn’t exist.
- Create a certificate for this node signed by the CA key.
- Update the zones.conf file with the new zone hierarchy.
- Update the ApiListener and constants configuration.

Here is an example of a master setup for the icinga2-master1.localdomain node on CentOS 7:

```
[root@icinga2-master1.localdomain /]# icinga2 node wizard

Welcome to the Icinga 2 Setup Wizard!

We will guide you through all required configuration details.

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

Starting the Master setup routine...

Please specify the common name (CN) [icinga2-master1.localdomain]: icinga2-master1.localdomain
Reconfiguring Icinga...
Checking for existing certificates for common name 'master1'...
Generating master configuration for Icinga 2.

Please specify the API bind host/port (optional):
Bind Host []:
Bind Port []:

Done.

Now restart your Icinga 2 daemon to finish the installation!
```

Note: Ensure setting up a satellite node and also have a direct connection from the master node and also accept the commands from the master node and also auto sign the certificate from the master Node.

Get a token from the master Node by executing the following command

```
[root@icinga2-master1.localdomain /]# icinga2 pki ticket --cn icinga2-client1.localdomain
```

Configuring Zones.conf File:

```
object Endpoint "icinga2-client1.localdomain" {
  host = "192.168.56.101"
}

object Endpoint "icinga2-satellite1.localdomain" {
  host = "192.168.56.105"
}

object Zone "client" {
  endpoints = [ "icinga2-client1.localdomain" ]
  parent = "master"
}

object Zone "satellite" {
  endpoints = [ "icinga2-satellite1.localdomain" ]
  }

/* sync global commands */
object Zone "global-templates" {
  global = true
}
```

By Default some services are installed and if we want to additional services
Add the plugin in the file /usr/lib/Nagios/plugins .
Add commands in the file /etc/icinga2/conf.d/commands.conf
Add Services in the file /etc/icinga2/conf.d/services.conf

Example:
Adding a Memory Utilization service.

1. Add a plugin check_mem in the file /usr/lib/Nagios/plugins .

Note: Download the plugins from the url
https://exchange.nagios.org/directory/Plugins

2. Add commands in the file /etc/icinga2/conf.d/commands.conf

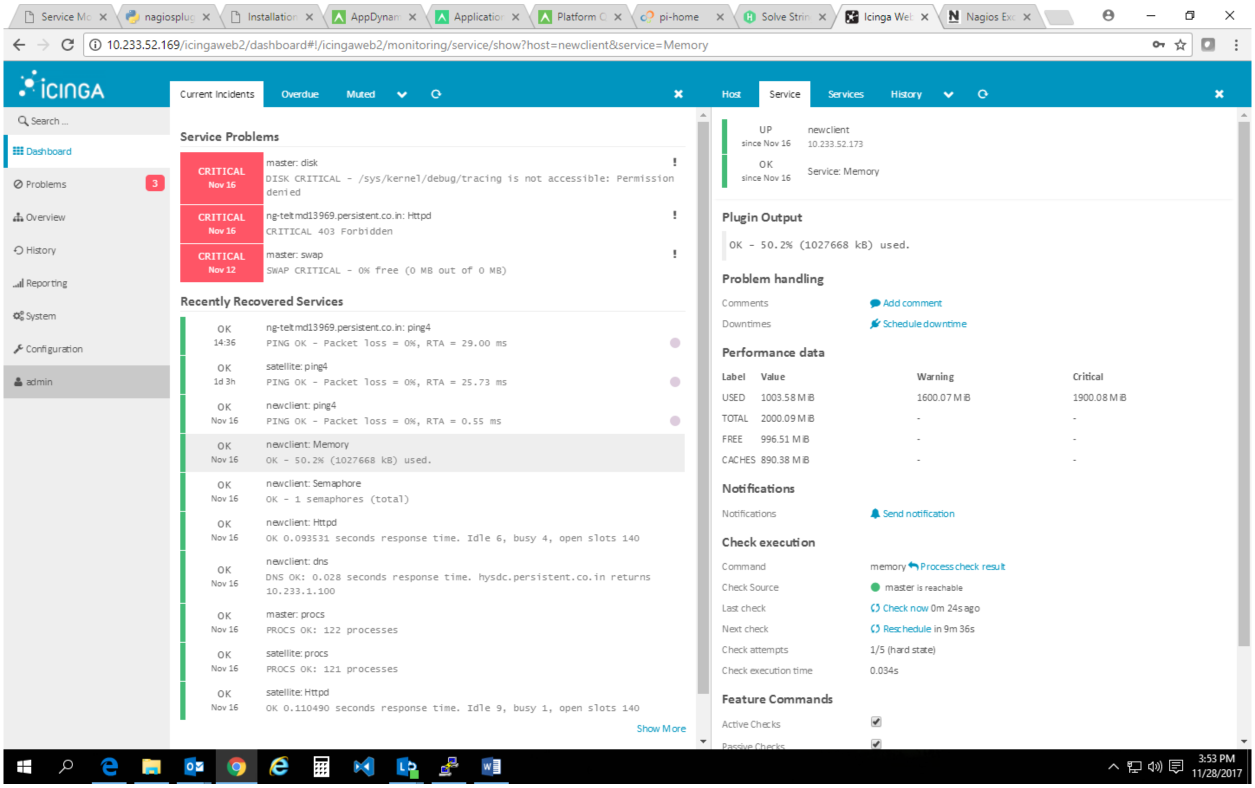
```
object CheckCommand "memory" {
    import "plugin-check-command"
    command = [ PluginDir + "/check_mem.pl" ]

    arguments = {
      "-w" = {
         value = "$mem_warning$"
      }
     "-c" = {
        value ="$mem_critical$"
      }
     "-u" = {
        set_if = "$mem_used$"
     }
     "-C" = {
        set_if = "$mem_cache$"
     }
   }

    vars.mem_warning = 80
    vars.mem_critical = 95
    vars.mem_used = true
    vars.mem_cache = true
}
```

3. Add Services in the file /etc/icinga2/conf.d/services.conf

```
apply Service "Memory" {
  import "generic-service"
  check_command = "memory"
  assign where host.address && host.zone != "satellite"
}
```



Service Problems

CRITICAL Nov 16 — master: disk
DISK CRITICAL - /sys/kernel/debug/tracing is not accessible: Permission denied

CRITICAL Nov 16 — ng-teltmd13969.persistent.co.in: Httpd
CRITICAL 403 Forbidden

CRITICAL Nov 12 — master: swap
SWAP CRITICAL - 0% free (0 MB out of 0 MB)

Recently Recovered Services

OK 14:36 — ng-teltmd13969.persistent.co.in: ping4
PING OK - Packet loss = 0%, RTA = 29.00 ms

OK 1d 3h — satellite: ping4
PING OK - Packet loss = 0%, RTA = 25.73 ms

OK Nov 16 — newclient: ping4
PING OK - Packet loss = 0%, RTA = 0.55 ms

OK Nov 16 — newclient: Memory
OK - 50.2% (1027668 kB) used.

OK Nov 16 — newclient: Semaphore
OK - 1 semaphores (total)

OK Nov 16 — newclient: Httpd
OK 0.093531 seconds response time. Idle 6, busy 4, open slots 140

OK Nov 16 — newclient: dns
DNS OK: 0.028 seconds response time. hysdc.persistent.co.in returns 10.233.1.100

OK Nov 16 — master: procs
PROCS OK: 122 processes

OK Nov 16 — satellite: procs
PROCS OK: 121 processes

OK Nov 16 — satellite: Httpd
OK 0.110490 seconds response time. Idle 9, busy 1, open slots 140

UP since Nov 16 — newclient 10.233.52.173
OK since Nov 16 — Service: Memory

Plugin Output

OK - 50.2% (1027668 kB) used.

Problem handling

Comments: Add comment
Downtimes: Schedule downtime

Performance data

Label	Value	Warning	Critical
USED	1003.58 MiB	1600.07 MiB	1900.08 MiB
TOTAL	2000.09 MiB	-	-
FREE	996.51 MiB	-	-
CACHES	890.38 MiB	-	-

Notifications

Notifications: Send notification

Check execution

Command: memory Process check result
Check Source: ● master is reachable
Last check: Check now 0m 24s ago
Next check: Reschedule in 9m 36s
Check attempts: 1/5 (hard state)
Check execution time: 0.034s

Feature Commands

Active Checks ☑
Passive Checks ☑

After adding the service restart the icinga2 services

Grafana Integration with Icinga2

Setup graphing module Grafana
Get packages

apt-get install influxdb influxdb-client adduser libfontconfig

wget https://s3-us-west-2.amazonaws.com/grafana-releases/release/grafana_4.4.3_amd64.deb
dpkg -i grafana_4.4.3_amd64.deb
rm grafana_4.4.3_amd64.deb

Enable and start Grafana service

systemctl daemon-reload
systemctl enable grafana-server
systemctl start grafana-server

Create database and user

```
influx
CREATE DATABASE icinga2;
CREATE USER icinga2 WITH PASSWORD 'your-icinga2-pwd';
```

Send performance data to influx

```
icinga2 feature enable influxdb

vim /etc/icinga2/features-enabled/influxdb.conf
-  //host = "127.0.0.1"
-  //port = 8086
-  //database = "icinga2"
-  //flush_threshold = 1024
-  //flush_interval = 10s
-  //host_template = {
-  //  measurement = "$host.check_command$"
-  //  tags = {
-  //    hostname = "$host.name$"
-  //  }
-  //}
-  //service_template = {
-  //  measurement = "$service.check_command$"
-  //  tags = {
-  //    hostname = "$host.name$"
-  //    service = "$service.name$"
-  //  }
-  //}
+  host = "127.0.0.1"
+  port = 8086
+  database = "icinga2"
+  username = "icinga2"
+  password = "your-icinga2-pwd"
+  enable_send_thresholds = true
+  enable_send_metadata = true
+  flush_threshold = 1024
+  flush_interval = 10s
+  host_template = {
+    measurement = "$host.check_command$"
+    tags = {
+      hostname = "$host.name$"
+    }
+  }
+  service_template = {
+    measurement = "$service.check_command$"
+    tags = {
+      hostname = "$host.name$"
+      service = "$service.name$"
+    }
+  }
```

systemctl restart icinga2

Navigate to Grafana web interface

Url: http://your-public-host.name:3000
Username: admin
Password: admin

Create new Grafana datasource

Add Datasource: http://your-public-host.name:3000/datasources/new?gettingstarted

Name: Influx
Type: InfluxDB
Default: Yes

Url: http://127.0.0.1:8086
Access: proxy

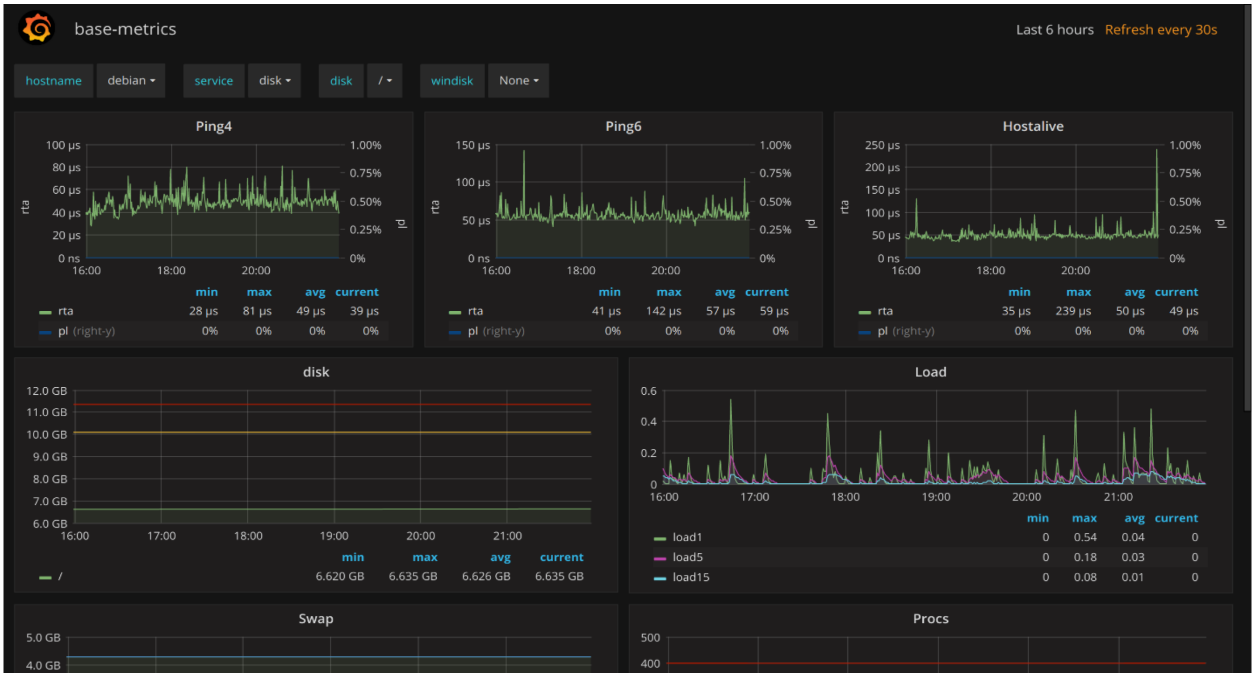
Database: icinga2
User: icinga2
Password: your-icinga2-pwd

Import Grafana dashboard

Import Dashboard:

http://your-public-host.name:3000/dashboard/new?editview=import&orgId=1

Paste JSON fromhttps://raw.githubusercontent.com/Mikesch-mp/icingaweb2-module-grafana/v1.1.8/dashboards/influxdb/base- metrics.json



Ping4

	min	max	avg	current
rta	28 µs	81 µs	49 µs	39 µs
pl (right-y)	0%	0%	0%	0%

Ping6

	min	max	avg	current
rta	41 µs	142 µs	57 µs	59 µs
pl (right-y)	0%	0%	0%	0%

Hostalive

	min	max	avg	current
rta	35 µs	239 µs	50 µs	49 µs
pl (right-y)	0%	0%	0%	0%

disk

	min	max	avg	current
/	6.620 GB	6.635 GB	6.626 GB	6.635 GB

Load

	min	max	avg	current
load1	0	0.54	0.04	0
load5	0	0.18	0.03	0
load15	0	0.08	0.01	0

Add Icinga Web Grafana module

```
cd /usr/share/icingaweb2/modules
wget -qO- https://github.com/Mikesch-mp/icingaweb2-module-grafana/archive/v1.1.8.tar.gz | tar xvz
mv icingaweb2-module-grafana-1.1.8 grafana
mkdir /etc/icingaweb2/modules/grafana
```

Configure Icinga Web Grafana module

Module configiuriation (config.ini)

```
vim /etc/icingaweb2/modules/grafana/config.ini
```

Graph configiuriation (graphs.ini)

```
vim /etc/icingaweb2/modules/grafana/graphs.ini
```

Enable anonymous access

```
vim /etc/grafana/grafana.ini
[auth.anonymous]
# enable anonymous access
-;enabled = false
+enabled = true
systemctl restart grafana-server
```

Remove scrollbars from iframe

```
vim /usr/share/icingaweb2/modules/grafana/library/Grafana/ProvidedHook/Grapher.php
-               $iframehtml = '<iframe src="%s://%s/dashboard-solo/%s/%s?var-hostname=%s&var-service=%s%s&panelId=%s&orgId=%s&theme=%s&from=now-%s&to=now" alt="%s" height="%d" frameBorder="0" style="width: 100%%;"></iframe>';

+               $iframehtml = '<script>$(".module-grafana").parent().parent().data("icingaRefresh", 30);</script><iframe src="%s://%s/dashboard-solo/%s/%s?var-hostname=%s&var-service=%s%s&panelId=%s&orgId=%s&theme=%s&from=now-%s&to=now" alt="%s" height="%d" frameBorder="0" scrolling="no" style="width: 100%%;"></iframe>';
```

Enable Icinga Web Grafana module

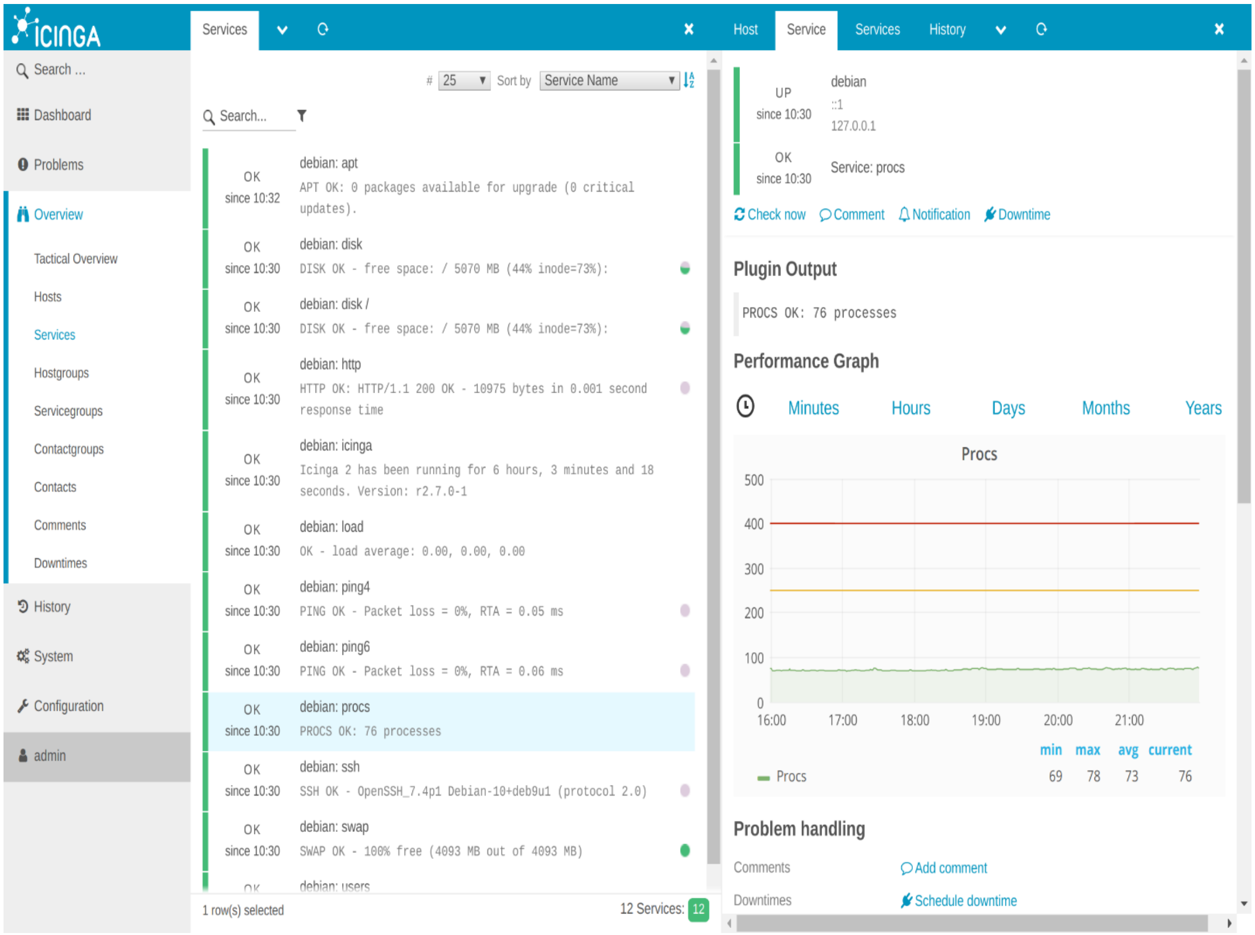
```
icingacli module enable grafana
chown -R www-data:icingaweb2 /etc/icingaweb2
```

Disable graphing for commands without graphs

Go to the service configuration and set the custom var grafana_graph_disable for all services, which have no Grafana graph: ssh, http, disk, and icinga.

```
vim /etc/icinga2/conf.d/services.conf
+  vars.grafana_graph_disable = true

systemctl restart icinga2
```



# 25 Sort by Service Name

Search...

OK since 10:32	debian: apt APT OK: 0 packages available for upgrade (0 critical updates).
OK since 10:30	debian: disk DISK OK - free space: / 5070 MB (44% inode=73%):
OK since 10:30	debian: disk / DISK OK - free space: / 5070 MB (44% inode=73%):
OK since 10:30	debian: http HTTP OK: HTTP/1.1 200 OK - 10975 bytes in 0.001 second response time
OK since 10:30	debian: icinga Icinga 2 has been running for 6 hours, 3 minutes and 18 seconds. Version: r2.7.0-1
OK since 10:30	debian: load OK - load average: 0.00, 0.00, 0.00
OK since 10:30	debian: ping4 PING OK - Packet loss = 0%, RTA = 0.05 ms
OK since 10:30	debian: ping6 PING OK - Packet loss = 0%, RTA = 0.06 ms
OK since 10:30	debian: procs PROCS OK: 76 processes
OK since 10:30	debian: ssh SSH OK - OpenSSH_7.4p1 Debian-10+deb9u1 (protocol 2.0)
OK since 10:30	debian: swap SWAP OK - 100% free (4093 MB out of 4093 MB)
OK	debian: users

1 row(s) selected 12 Services: 12

UP since 10:30 debian
::1
127.0.0.1

OK since 10:30 Service: procs

Plugin Output

```
PROCS OK: 76 processes
```

Performance Graph

	min	max	avg	current
Procs	69	78	73	76

Problem handling

Comments

Downtimes

For more Information please refer the below link:
https://www.icinga.com/docs/icinga2/latest/doc/02-getting-started/