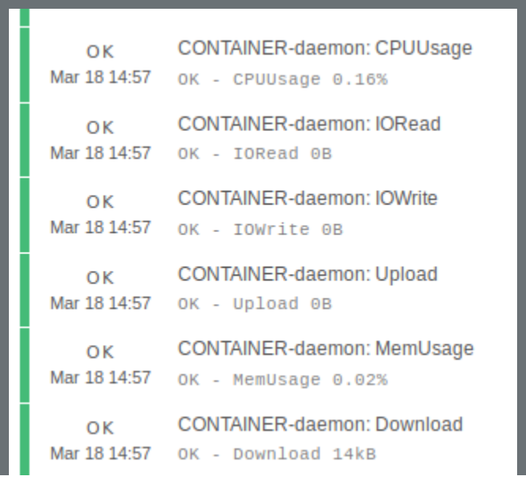
Docker Container monitoring & Icinga2

Jan Lüders Mar 21, 2019 · 3 min read

After seeking the web for a Solution to monitor Container in a good way, means with all available metricdata and not as founded with only two or three metrics(CPU/RAM || CPU/RAM/IO || CPU/RAM/NET), i found a bash script, written mainly by the Github User jryberg, which used the docker commands by them self to grep them out of the stdout.

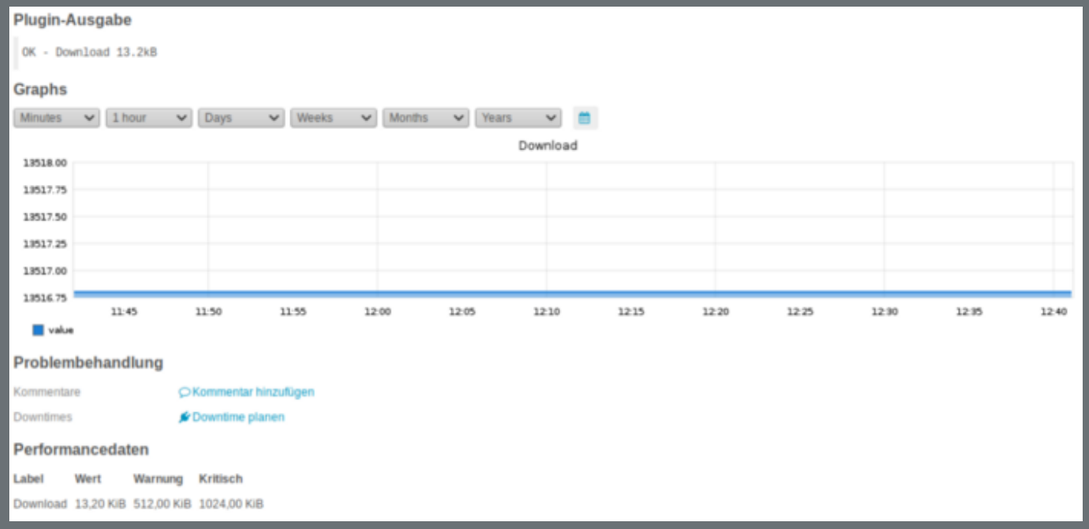
For the beginning it was an good example for me and i changed theScript as it wouzld be for me the best. You can review it here on my Github.



plete All Metrics in single Tests

As the pic above shows all available metrics would be taken, and to get these data visualized, the script print performance data out as well.

If the graphite plugin in Icingaweb2 and the performance feature in icinga2 is enabled the kown graphs would shown up like presented on the pic below



Perfomance Data and Graphsvisualisation in Icinga2

To build up this all you have to create multiple checks, in my way it were a single Hostfile with all checks inside.

object Host “CONTAINER-testdocker” {
import “generic-host”
address = “XXX.XXX.XXX”
vars.os = “Docker-Container”

check_command = “containeralive”

```
vars.notification[“mail”] = {
groups = [ “icingaadmins” ]
}
vars.containername = “testdocker”
vars.client_endpoint = “testdocker”
}

apply Service “CPUUsage” {
import “docker-service”

vars.containername = “testdocker”
vars.metric_command = “-c 50%,70%,0,100”

assign where host.name == “CONTAINER-testdocker”
}

apply Service “MemUsage” {
import “docker-service”

vars.containername = “testdocker”
vars.metric_command = “-m 50%,70%,0,100”

assign where host.name == “CONTAINER-testdocker”
}

apply Service “IORead” {
import “docker-service”
```

```
vars.containername = “testdocker”
vars.metric_command = “-ir 200KB,400KB,0,0”

assign where host.name == “CONTAINER-testdocker”
}

apply Service “IOWrite” {
import “docker-service”

vars.containername = “testdocker”
vars.metric_command = “-iw 300KB,400KB,0,0”

assign where host.name == “CONTAINER-testdocker”
}

apply Service “Upload” {
import “docker-service”

vars.containername = “testdocker”
vars.metric_command = “-nu 512KB,1024KB,0,0”

assign where host.name == “CONTAINER-testdocker”
}

apply Service “Download” {
import “docker-service”

vars.containername = “testdocker”
```

vars.metric_command = “-nd 512KB,1024KB,0,0”

assign where host.name == “CONTAINER-testdocker”
}

You need to create a template and two CheckCommands as shown below. Add the CheckCommands to the commands.conf and the template to the template.conf

template Service “docker-service” {
max_check_attempts = 5
check_interval = 15s
retry_interval = 3s

check_command = “container_metrics”
}

object CheckCommand “containeralive”{
import “by_ssh”
vars.by_ssh_command = “/usr/local/bin/docker_check -n ‘$containername$’ -s”

vars.containername = “$container_name$”
}

object CheckCommand “container_metrics” {
import “by_ssh”
vars.by_ssh_command = “/usr/local/bin/docker_check -n $containername$ $metric_command$”

vars.containername = “$container_name$”
vars.metric_command = “$metriccommand$”
}

As mentioned earlier you can get the original Script behin THIS link and the modified one here

Docker Icinga2 Monitoring Icinga Web 2 Nagios

May it helps someone else too.

Medium

About Write Help Legal