

SaaS Infrastructure metrics' thresholds

1. Application Instances

1.1 CPU thresholds

In order to measure CPU, we recommend analyzing CPU USER and CPU STEAL. Additionally, you can use the SAR tool for getting CPU information.

Note: processor_vcpus is the number of virtual processors each CPU

```
cpu_warning = 85
cpu_critical = 90

cpu_warning_stealtime = 10
cpu_critical_stealtime = 25
```

1.2 Load-average

In order to read the system Load-average, we use icinga2 plugin and monitor the value of load average for each one, five, and fifteen minutes. Also, we resize threshold values to take the numbers of CPU.

```
load_wload1 = processor_vcpus * 1.5
load_cload1 = processor_vcpus * 2.5

load_wload5 = processor_vcpus * 1.25
load_cload5 = processor_vcpus * 2.25

load_wload15 = processor_vcpus
load_cload15 = processor_vcpus * 2
```

1.3 Free RAM

In order to read freeable memory we use file `/proc/meminfo` and apply this thresholds.

```
memory_warning_threshold = 10
memory_critical_threshold = 5
```

1.4 Free space in HD

For reading freeable memory we use the icinga2 plugin and apply this threshold. The zequenze system are installed into `/opt/zequenze` so is necessary monitoring this directory

```
memory_warning_threshold = 10
memory_warning_threshold = 5
```

2. Cache Instances

2.1 CPU thresholds for Redis instances

Redis is a mono-thread service (version 6.0.6) so it only uses one CPU. That means we have to resize the CPU umbral to check the CPU behaviour correctly.

```
cpu_warning = (80 / processor_vcpus)
cpu_critical = (90 / processor_vcpus)
```

3. Application Status Monitoring

Application status can be monitored with the following http request in order to get information about ping, cache, database and locmem.

```
curl 127.0.0.1:8000/status/
```

```
{
  "ping":{
    "code":200,
    "status":"Ok",
    "time":0
  },
  "caches":{
    "default":{
      "code":200,
      "status":"Ok",
      "time":0.003637,
      "time_set":0.001012,

```

```
    "time_get":0.00225,
    "time_del":0.000375
  },
  "locmem":{
    "code":200,
    "status":"Ok",
    "time":0.000135,
    "time_set":5.5e-05,
    "time_get":7e-05,
    "time_del":1e-05
  }
},
"database":{
  "default":{
    "code":200,
    "status":"Ok",
    "time":0.003241
  },
  "replica1":{
    "code":200,
    "status":"Ok",
    "time":0.003045
  }
}
}
```

4. Database

Follows the metrics and default thresholds for each database instances

4.1 CPU Utilization

```
warning = "75"
critical = "85"
```

4.2 Freeable Memory

```
warning = "15%"
critical = "7%"
```

Revision #4

Created 2021-11-19 20:52:30 UTC by freddy@zequence.com

Updated 2024-04-08 17:53:17 UTC by freddy@zequence.com