PGConf EU | Warsaw | 26.10.2017

Monitoring Open Source Databases with Icinga

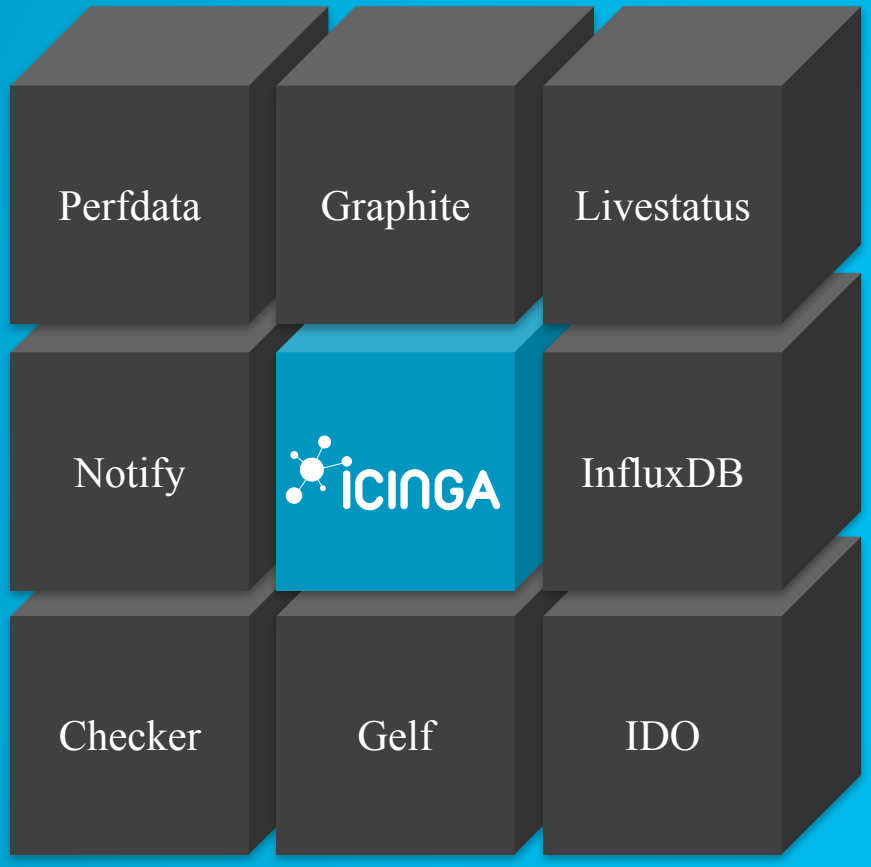
- Blerim Sheqa
  - Product Manager
  - Working @netways
  - @bobapple

Introduction to Icinga2

Quick Poll

Icinga is a scalable and extensible monitoring system which checks the availability of your resources, notifies users of outages and provides extensive metrics.

ICINGA



- Multithreaded
- Modular Features
- Zone support
- Secure Agent
- No Nagios®

ICINGA

- Availability and scaling zones
- Automatic redistribution of checks
- Zones for multitenancy environments

Monitoring Databases

- MySQLish
- PostgreSQL
- MongoDB
- Firebird
- SQLite

- check_mongodb_py

  - connect
  - connections
  - replicaton_lag
  - memory
  - memory_mapped
  - lock
  - flushing
  - last_flush_time
  - replset_state
  - index_miss_ratio
  - collections
  - database_size
  - database_indexes
  - replica_primary

https://github.com/mzupan/nagios-plugin-mongodb

- check_postgres.pl
  - archive_ready
  - autovac_freeze
  - backends
  - bloat
  - checkpoint
  - cluster_id
  - commitratio
  - connection
  - custom_query
  - disabled_triggers
  - disk_space
  - fsm_pages
  - prepared_txns
  - query_runtime
  - query_time
  - replicate_row
  - same_schema
  - sequence
  - settings_checksum

https://bucardo.org/check_postgres/check_postgres.pl.html

- check_mysql_health
  - connection-time
  - uptime
  - threads-connected
  - threadcache-hitrate
  - qcache-hitrate
  - qcache-lowmem-prunes
  - bufferpool-hitrate
  - bufferpool-wait-free
  - log-waits
  - tablecache-hitrate
  - table-lock-contention
  - index-usage
  - tmp-disk-tables
  - slow-queries

https://labs.consol.de/de/nagios/check_mysql_health/

Monitoring Databases

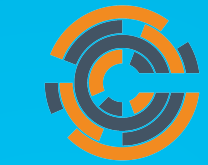
What needs to be monitored?

- Availability
- Troubleshooting
- Replication-Status
- Capacity
- Metrics and performance data

Monitoring Databases

How does it work?

It is all about automation



CHEF

SALTSTACK

ANSIBLE

Icinga2 - API

- HTTP with RESTful Url Schema
- X.509 and/or Basic Auth
- Create, Modify and Delete objects
- Event Stream based on Types and Filters

Configuration

- Objects
- Rule based
- Conditions
- Loops
- Custom Functions

```
object Host "demo.icinga.com" {
  import "generic-host"

  address = "127.0.0.1"
  address6 = "::1"

  vars.os = "Linux"
}
```

Rules

```
apply Service "ssh" {
  import "generic-service"

  check_command = "ssh"

  assign where host.vars.os == "Linux"
  ignore where host.vars.dev == true
}
```


Search...
service = *ssh* | service_display_name = *ssh*
CRITICAL since Aug 1
exchange.icinga.org: ssh
CRITICAL - Socket timeout after 10 seconds
OK for 6m 45s
dummy-host-1: ssh
SSH OK - OpenSSH_6.6.1p1 Ubuntu-2ubuntu2.8 (protocol 2.0)
OK for 6m 45s
dummy-host-6: ssh
SSH OK - OpenSSH_6.6.1p1 Ubuntu-2ubuntu2.8 (protocol 2.0)
OK for 6m 46s
dummy-host-10: ssh
SSH OK - OpenSSH_6.6.1p1 Ubuntu-2ubuntu2.8 (protocol 2.0)
OK for 6m 46s
dummy-host-2: ssh
SSH OK - OpenSSH_6.6.1p1 Ubuntu-2ubuntu2.8 (protocol 2.0)
OK for 6m 46s
dummy-host-3: ssh
SSH OK - OpenSSH_6.6.1p1 Ubuntu-2ubuntu2.8 (protocol 2.0)
OK for 6m 46s
dummy-host-4: ssh
SSH OK - OpenSSH_6.6.1p1 Ubuntu-2ubuntu2.8 (protocol 2.0)
OK for 6m 46s
dummy-host-5: ssh
SSH OK - OpenSSH_6.6.1p1 Ubuntu-2ubuntu2.8 (protocol 2.0)
OK for 6m 46s
dummy-host-7: ssh
SSH OK - OpenSSH_6.6.1p1 Ubuntu-2ubuntu2.8 (protocol 2.0)
OK for 6m 46s
dummy-host-8: ssh
SSH OK - OpenSSH_6.6.1p1 Ubuntu-2ubuntu2.8 (protocol 2.0)
OK for 6m 46s
dummy-host-9: ssh
SSH OK - OpenSSH_6.6.1p1 Ubuntu-2ubuntu2.8 (protocol 2.0)
OK since 10:00
demo: ssh
SSH OK - OpenSSH_6.6.1p1 Ubuntu-2ubuntu2.8 (protocol 2.0)

More Rules

```
object Host "demo.icinga.com" {
  import "generic-host"

  address = "127.0.0.1"
  address6 = "::1"

  vars.http_vhosts["Icinga Web 2"] = {
    http_uri = "/icingaweb2"
  }

  vars.http_vhosts["Grafana"] = {
    http_port = 3000
  }
}
```

```
apply Service for (
  http_vhost => config in host.vars_http_vhosts
){
  import "generic-service"

  display_name = "http " + http_vhost
  check_command = "http"

  vars += config
}
```

OK Oct 13 17:50	http Grafana `HTTP OK: HTTP/1.0 302 Found - 266 bytes in 0.001 second response time`
OK 56m 54s	http Icinga Web 2 `HTTP OK: HTTP/1.1 301 Moved Permanently - 534 bytes in 0.000 second response time`

```
const PostgreSQLUser = "root“
const PostgreSQLPass = “root“


template Host "base-host-pgsql" {
   vars.postgres_dbuser = PostgreSQLUser
   vars.postgres_dbpass = PostgreSQLPass
   vars.os = "Linux"
}
```

```
object Host "live-pgsql-1" {
    import "base-host-pgsql"
    check_command = "hostalive"
    address = "127.0.0.1"
    vars.dbtype = “pgsql“
}
```

```
apply Service “PostgreSQL Connection" {
    import "generic-service"
    check_interval = 10s
    retry_interval = 30s

    check_command = "postgres"

    vars.postgres_dbuser = host.vars.postgres_dbuser
    vars.postgres_dbpass = host.vars.postgres_dbpass
    vars.postgres_action = "connection"

    assign where host.vars.dbtype == "pgsql"
}
```

```
object Host "live-pgsql-1" {
    import "base-host-pgsql"
    check_command = "hostalive"
    address = "127.0.0.1"
    vars.dbtype = "pgsql"

    vars.databases["icinga"] = {
        postgres_warning = 4096 //MB
        postgres_critical = 8192 //MB
    }
    vars.databases["icingaweb2"] = {
        postgres_warning = 2048 //MB
        postgres_critical = 4096 //MB
    }
}
```

```
apply Service ”PostgreSQL Size " for (database => config in host.vars.databases) {
      import "generic-service"
      check_command = "postgres"

      display_name = "PostgreSQL Size " + database

      vars += config

      assign where host.vars.dbtype == "postgres"
}
```

Conditions

```
apply Service "dummy" {
  import "generic-service"

  check_command = "dummy"

  if (host.vars.environment == "dev") {
    check_interval = 30m
  } else {
    check_interval = 5m
  }

  assign where match("srv-*", host.name)
}
```

Functions

```
object Service ”Load" {
  check_command = "load”
  host_name = ”backup.abc.com”

  vars.load_warning = {{
    if (get_time_period("backup").is_inside) {
      return 20
    } else {
      return 5
    }
  }}

}
```

ICINGA

Template Library

```
object CheckCommand "mysql_health" {
      import "ipv4-or-ipv6"
      command = [ PluginContribDir + "/check_mysql_health" ]
            arguments = {
                  "--hostname" = {
                        value = "$mysql_health_hostname$"
                        description = "the database server's hostname"
                  }
                  "--port" = {
                        value = "$mysql_health_port$"
                        description = "the database's port“

...
```

- ITL (Icinga Template Library)
  - mysql_health
  - postgres
  - mongodb
  - mssql_health
  - db2_health
  - oracle_health
  - elasticsearch
  - redis

/usr/share/icinga2/include/plugins-contrib.d/databases.conf

Conclusion

- Download Icinga 2 and Icinga Web 2
- Play with Icinga2 on Vagrant
- Rethink your configuration
- Give us feedback

Thank You!

www.icinga.com

github.com/icinga

@icinga

/icinga

+icinga

ICINGA