Transformation of monitoring to 21st century in Livesport

Jakub Štollmann
Network & Systems Engineer

@pershinghar

Livesport

WhoAmI?

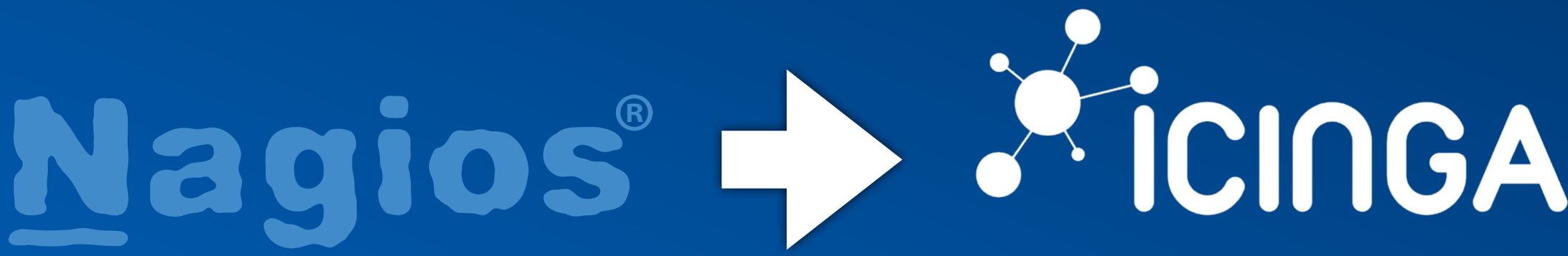
- Network & Systems Engineer
- Livesport s.r.o
- Python & Icinga enthusiast (recently)

- @pershinghar
- jakub@stlm.cz

Livesport

Agenda

Generally:



Agenda

- Introduction
- Motivation
- New conception & deployment
- Future

Livesport ?

- Livescore provider
- We operate 500+ hosts
  - Physical servers
  - Virtual servers
  - Network hosts
- 3 locations
- Lot's of traffic

- Still growing!

Motivation ?

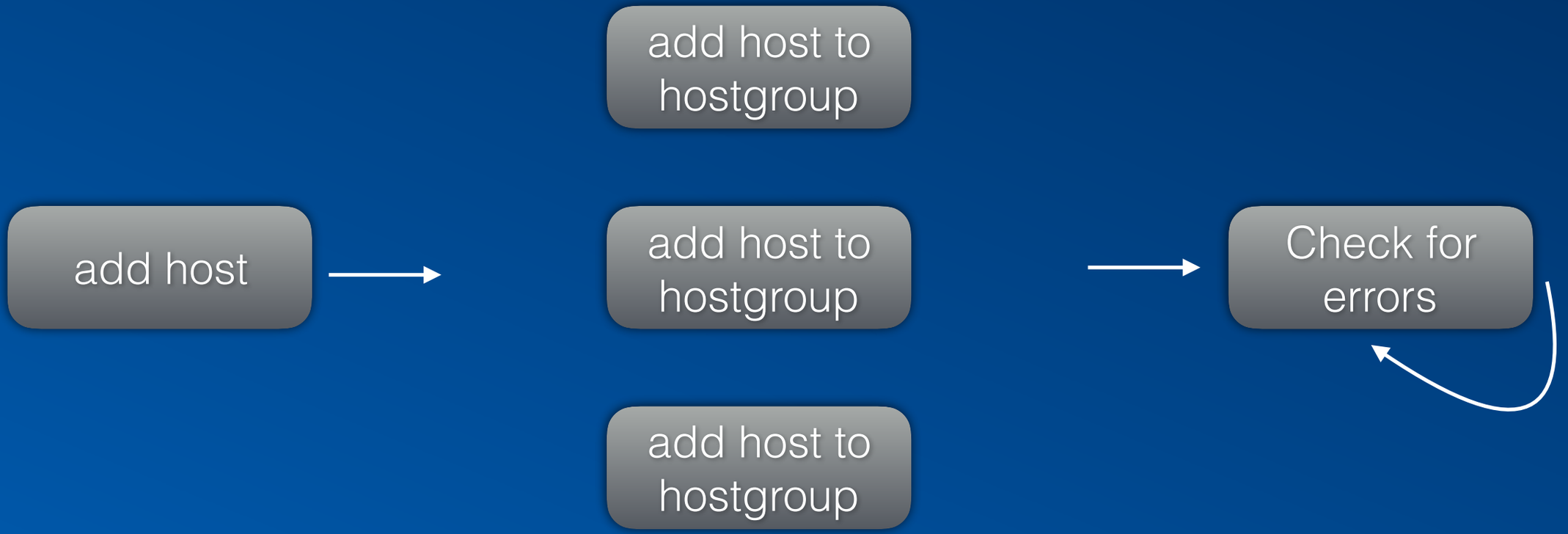
Once upon a time...

Livesport

Motivation

1. Install server
2. Add host to configuration
3. Check for errors
4. Fix errors
5. Restart monitoring system and check it again
6. Fix another mistakes

Manual config

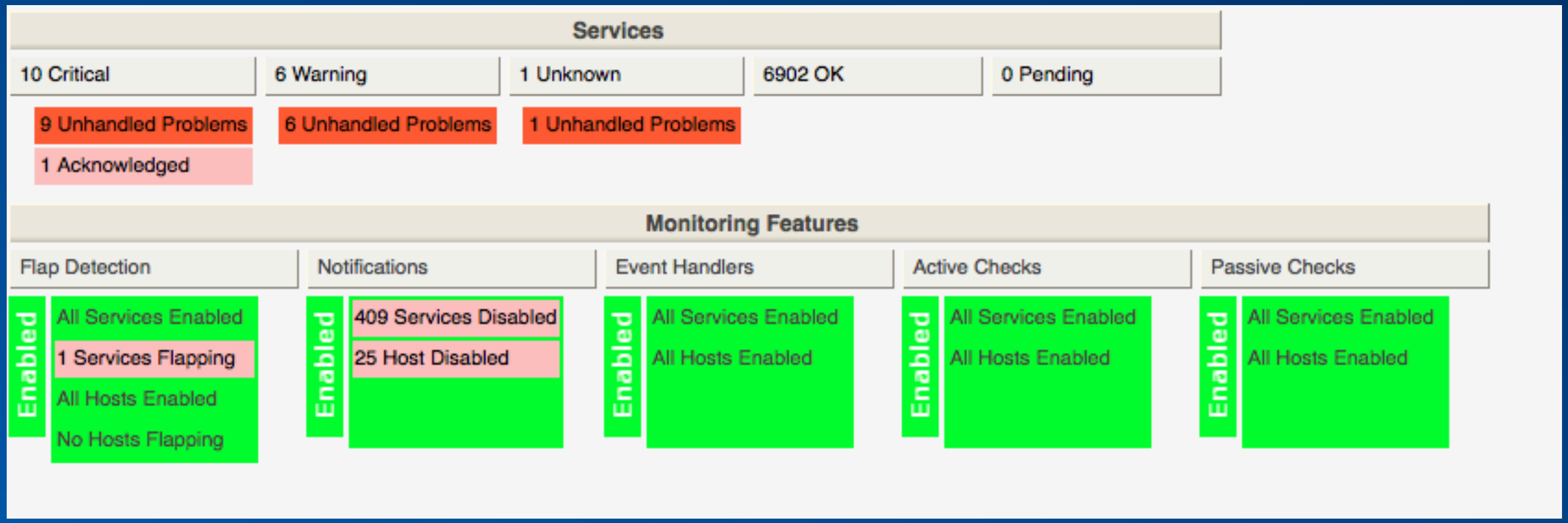


Livesport

```
define host{
   use                     generic-host        ; Name of host template to use
   host_name               varnishYes47
   alias                   varnishYes47.edhost.eu
   address                 143.123.32.236
}

define host{
   use                     generic-host        ; Name of host template to use
   host_name               varnishYes48
   alias                   varnishYes48.edhost.eu
   address                 143.123.32.237
}

define host{
   use                     generic-host        ; Name of host template to use
   host_name               varnishYes49
   alias                   varnishYes49.edhost.eu
   address                 143.123.32.238
}

define host{
   use                     generic-host        ; Name of host template to use
   host_name               varnishYes50
   alias                   varnishYes50.edhost.eu
   address                 143.123.32.239
}
```

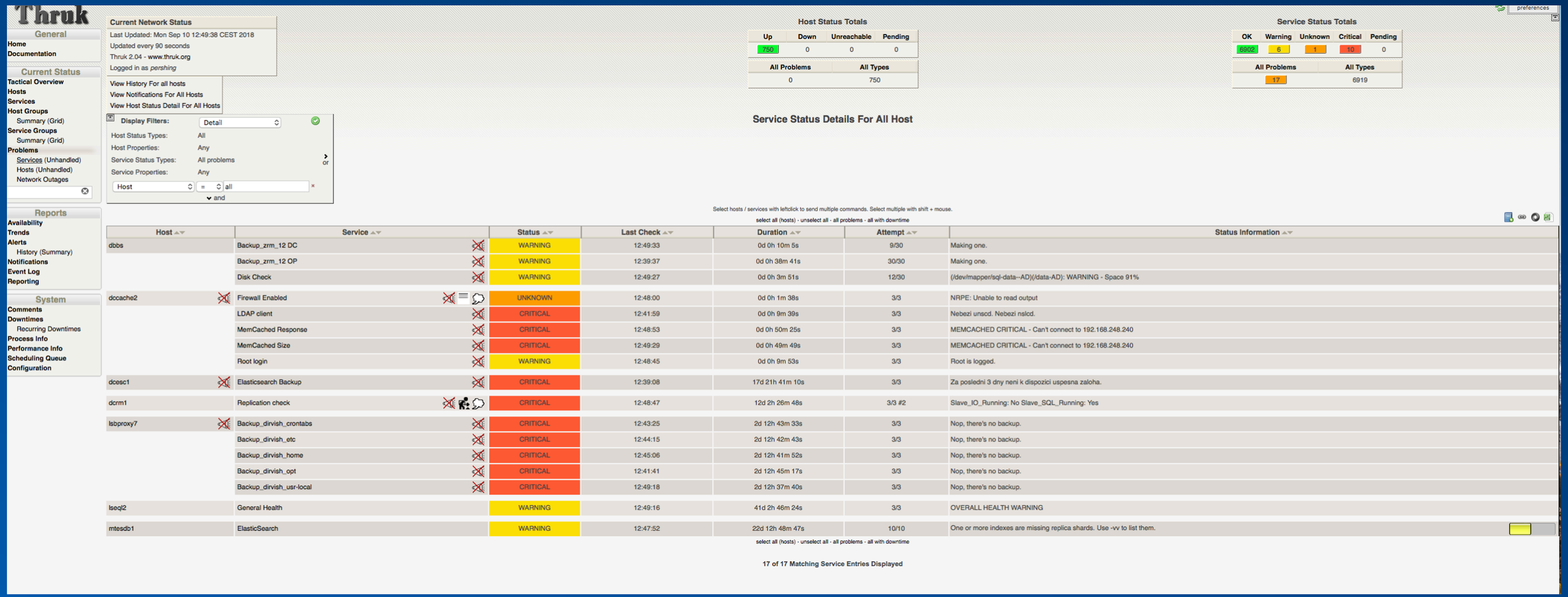
```
### SERVER
define hostgroup{
        hostgroup_name server
        alias Servers
    hostgroup_members server_mysql,server_mysql_slave,server_http,server_memcached
        members ns1,lspush1,lspush2,lspush3,lspush4,lspush5,lspush6,lspush7,lspush8,lsrlbs1,lsrlbs2,lsrlb
w1,lsrlbw2,lscache1,lscache2,lscache3,dcpush,dckvido,dckvido2,lsdns1,lsdns2,lsdns3,lsmongo1,lsmongo2,wata
,lssphinx1,lssphinx2,puppet,lsxen01,lsxen02,lsxen03,lsxen04,lsxen05,lsxen06,lsxen07,lsxen08,lsxen09,lsxen
10,lsxen11,lsxen12,lsxen13,lsxen14,lsxen15,lsxen16,lsxen17,lsxen18,lsxen19,lsxen20,lsxen21,lsxen22,lsxen2
3,lsxen24,lsxen25,lsxen26,lsxen27,management,opadmin,oppush,lsweb3,mongoarbiter1,monitor,lsid1,lsid2,dbbs
,deploy,cartman,ns3,nagios1,fsp2,dccache1,dccache2,fsstat,jenkins,dccml1,dccml2,dccml3,dccml4,dccron2,inv
entory,dcapi1,dcapi2,opcache1,opcache2,opproxy01,opproxy02,mrpaja,dcproxy1,dcproxy2,mail1,mail2,lswiki,dc
apiadmin,session1,session2,videoplayer,mailz,lsc1,infoport,lkm1,lks1,lks2,lks3,lks4,lks5,lsnp1,lsnp2,lsnp
3,lsnp4,lsdocker1,node2,node3,node4,node5,node6,node7,node8,node9,node10,node11,node12,node13,node14,node
15,node16,node17,node18,node19,node20,node21,lsbackup1,bs-kernel,lssearch1,dcarchive,lspush9,lspush10,bui
ldsystem,exapi1,exapi2,exw1,exw2,lstrac2,dcredis1,dcredis2,sso1,sso2,ssodb1,lstrac1,lsopenx1,lsopenx2,lso
penx3,lsopenx4,lsnpm2,lsnpm1,lsidm1,lsidm2,dcapi3,dcapi4,dcapi5,opparser1,opparser2,opparser3,opparser4,o
pparser5,opparser6,oppstor1,oppstor2,adadmin1,adadmin2,oppfp1,oppfp2,gwmedia,cactip1,cactip2,cactip3,sql1
,lswatch,dbb1,dcfix,mtprom1,mtweb1,mtifdb1,mtifdb2,mtesdb1,esdbm1,esdb1,esdb2,esdb3,esdb4,dcesc1,dcesc2,d
cesc3,lsgitlab1,esdbc1,esdbc2,chat1,lsnpredis1,lsnpredis2,dcapi6,dcapi7,dcapi8,pml1,puppet1,opapiredis1,o
pmc1,opmc2,opmc3,opmc4,opmc5,opmc6,opmc7,opmc8,opmcbackup1,opmctest1,opmctest2,opmctest3,opapiw1,opapiw2,
opapiw3,opapiw4,opapir1,opapir2,opapir3,opapir4,certbot1,lsnpproxy1,lsnpproxy2,dccron1,mailx,dcfix2,dcfix
1,lsnp5,lsnp6,kvidomqm1,kvidomqm2,dcapiproxy1,dcapiproxy2,kvidomq1,kvidomq2,esga1,kvxdba1
}
```

Crowded web interface



Livesport

Crowded web interface



Livesport

Motivation

- Manual configuration
  - Config structure - almost no possibility of automation
- Bad incident handling in monitoring system
  - Lot's of false alarms
  - Unreadable frontend (lot's of disabled notifications)
- Approx 10 000+ lines of (manual) configuration in few files (biggest around 5000)
- No **HA**

Livesport

OK, so let’s change it!

Once upon a time...

Livesport

What we need to handle?

- 500+ hosts, growing
- 5000+ services, growing*
- Notifications to various users, teams, via various channels

* - not unique

Requirements:

- Automated configuration
- Automatic on-call set up
- Multi-channel notifications
- Tailored notifications
- HA cluster & possibility of multiple locations
- Readable frontend

Let's prepare

Livesport

Software?

- Do we need another software?
  - Nagios3 - pretty old
  - New versions are not free

Software?

- Do we need another software?
  - Nagios3 - pretty old
  - New versions are not free

- **Yes.**

Software?

- We need something:
  - To match our requirements
  - We can easily migrate to
  - We can use effectively, and we understand it

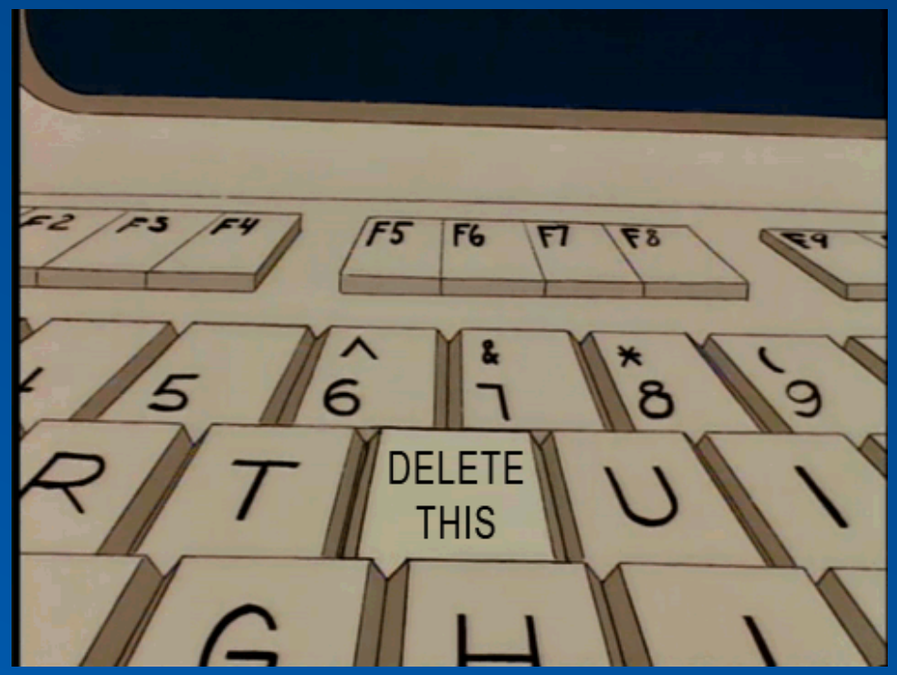
Software!

Software!

- **Icinga2**
  - Json style configurations
  - Base functionality same as Nagios, but more effective
  - Object structure is the same
  - HA options
  - Modular architecture

What about old configs?

- Migration is possible, but we want automation
- Let’s use them only for crosschecking purposes when adding services to new system.



Livesport

Automated configuration

Livesport

We need flow

Some data

- Information about hosts, devices
  - IP addresses
  - Services
- Other info like notification settings

Somewhere

- Internal DB's
  - PuppetDB

PuppetDB

Details

Certname	[redacted]
Facts	Oct 02 2019 - 13:21:05
Catalog	Oct 02 2019 - 13:21:23
Report	Oct 02 2019 - 13:21:33

Reports

Show 10 entries

End time	Status					
Oct 02 2019 - 13:21:33	UNCHANGED	3787	0	0	0	0
Oct 02 2019 - 13:08:48	UNCHANGED	3787	0	0	0	0

Facts

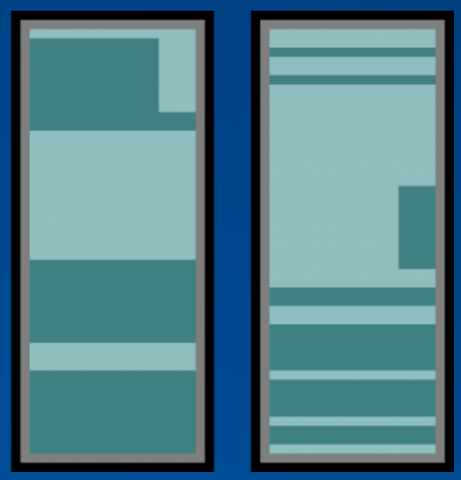
Search:

Name	Value
xtradb_cluster_number_of_nodes	0
volume_groups	{u'virtstorage': {u'uuid': u'v6TMlM-yC0T-GRSs-abjX-W7WR-ssAt-ErBDmU', u'allocation_policy': u'normal', u'free': u'487.63g', u'attr': u'wz--n-', u'size': u'727.63g', u'permissions': u'writeable'}}
virtual	physical
varnish_package	no
varnish_instances	
uuid	4C4C4544-0034-4E10-8046-B1C04F465632
uptime_seconds	17448605
uptime_hours	4846
uptime_days	201
uptime	201 days
	{u'authenticated': u'remote',

Somewhere

- Internal DB's
  - PuppetDB
  - Racktables (MySQL)

PuppetDB



Racktables

sw1

summary

Common name: sw1
Object type: Network switch
Visible label: sw1
Asset tag: sw1
HW type: Huawei CE6851-48S6Q-HI
OEM S/N 1: [redacted]
Explicit tags: Network

Comment

stack so sw2

ports and links

Local name	Visible label	Interface	L2 address	Remote object and port	Cable ID
mgm		1000Base-T			

IP addresses

rackspace allocation

-3PP : 4

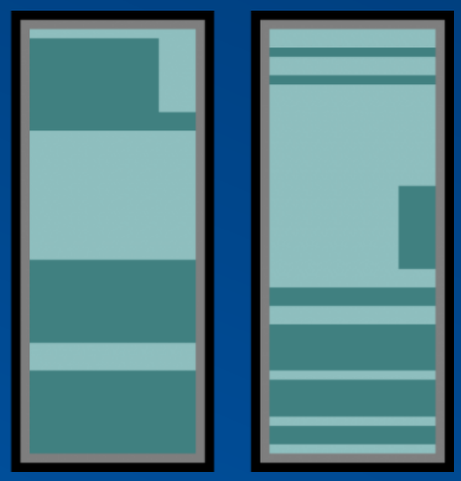
	Front	Interior	Back
42	not-usable-blank	PoE PatchPanel	
41	STS-R4	OpticalPatchpanel	
40			Cable Organizer
39		RJ45 PatchPanel	
38			Cable Organizer
37	sw1		
36	sw2		
35	sensors		Cable Organizer
34	gw1		
33	gw2		
32			Cable Organizer
31			
30			
29			
28			
27			
26			
25			

Livesport

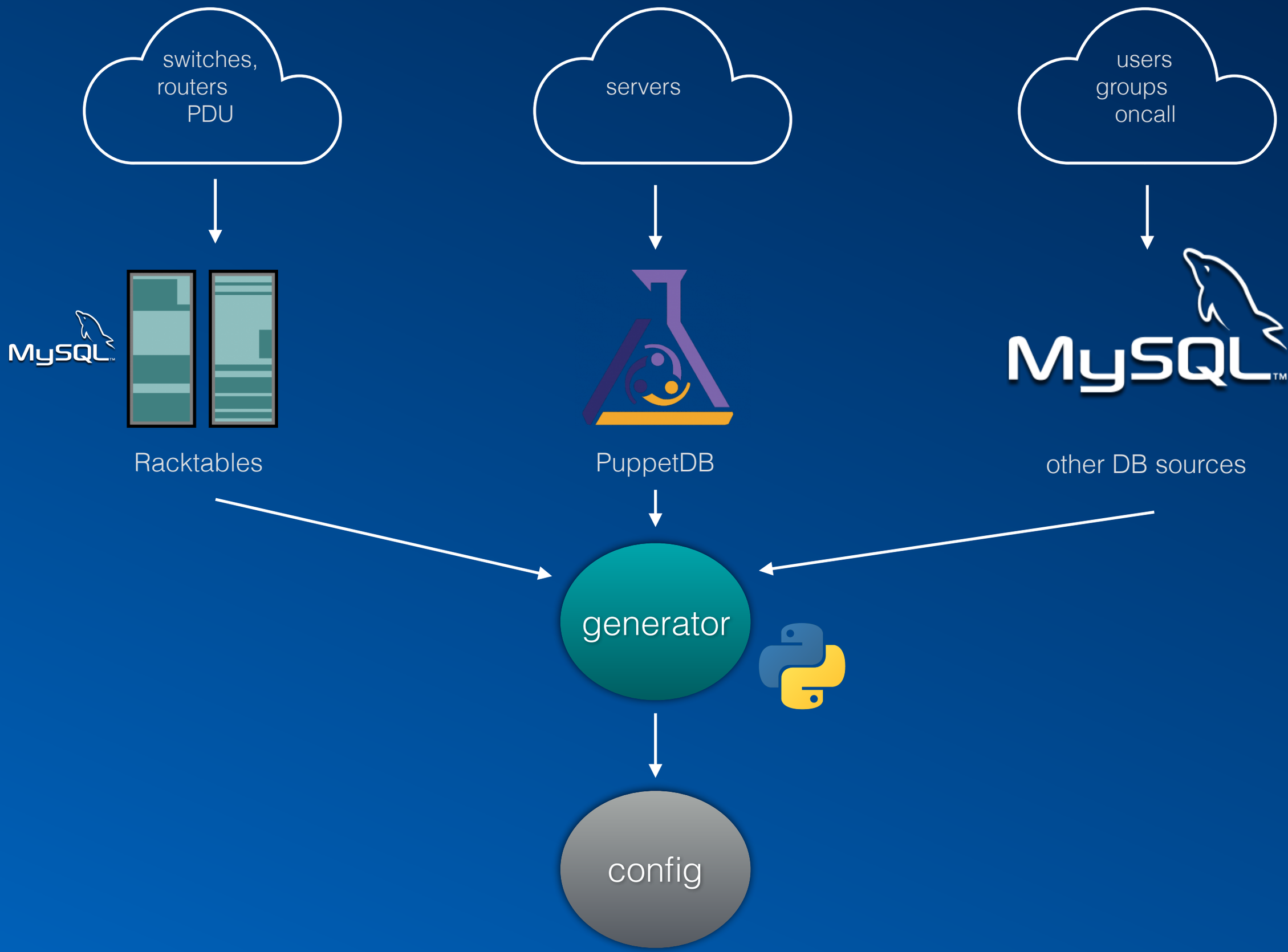
Somewhere

- Internal DB's
  - PuppetDB
  - Racktables (MySQL)
  - MySQL - internal

PuppetDB



Racktables


switches,
routers
PDU
servers
users
groups
oncall
MySQL
MySQL
Racktables
PuppetDB
other DB sources
generator
config

Livesport

What do we want to generate?

- Objects:
  - Hosts
  - Services
  - Users
  - Groups

What do we want to generate?

- Objects:
  - Hosts
  - ~~Services~~ (next section)
  - Users
  - Groups

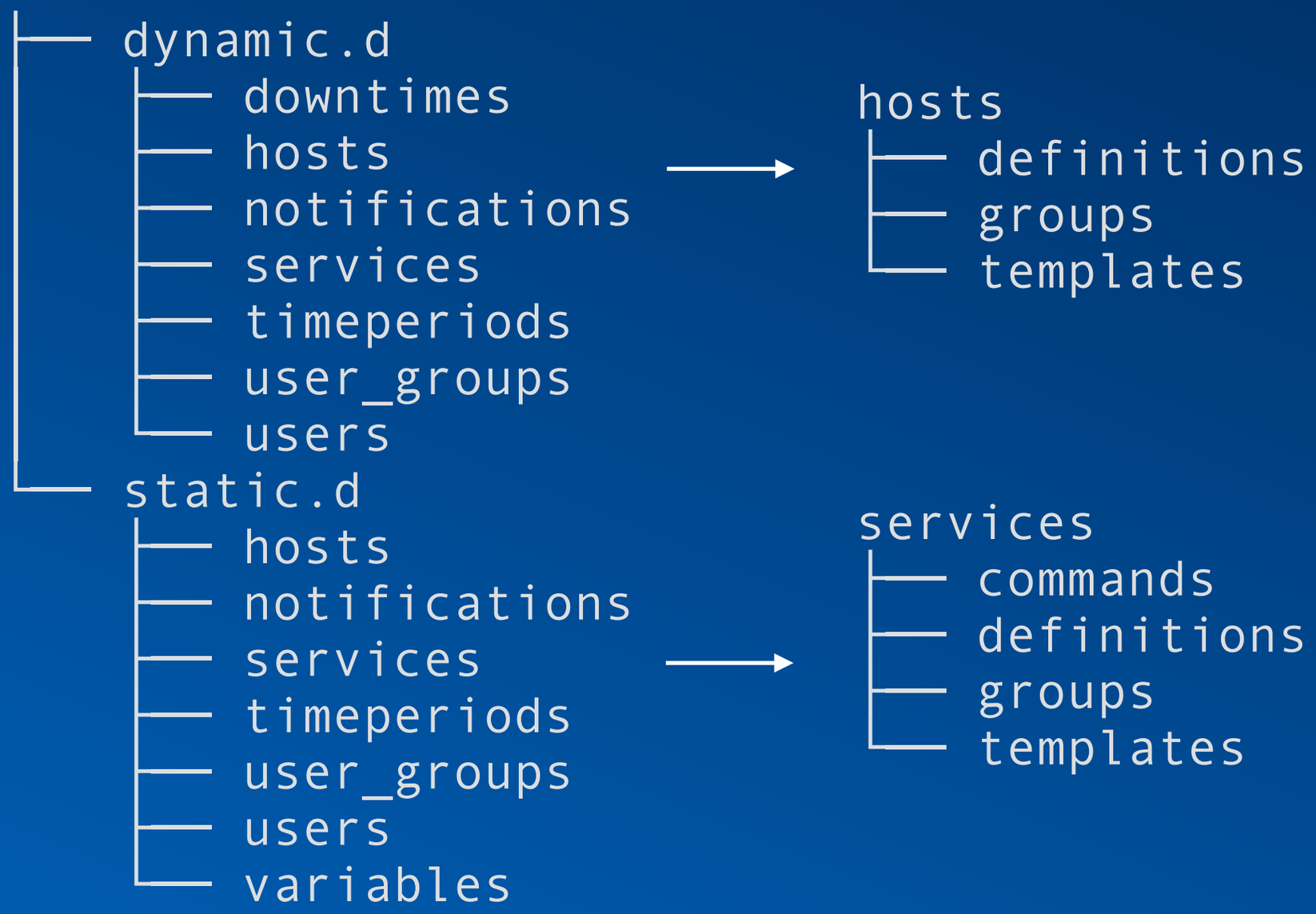
OK, where we should put this?

Livesport

Directory structure

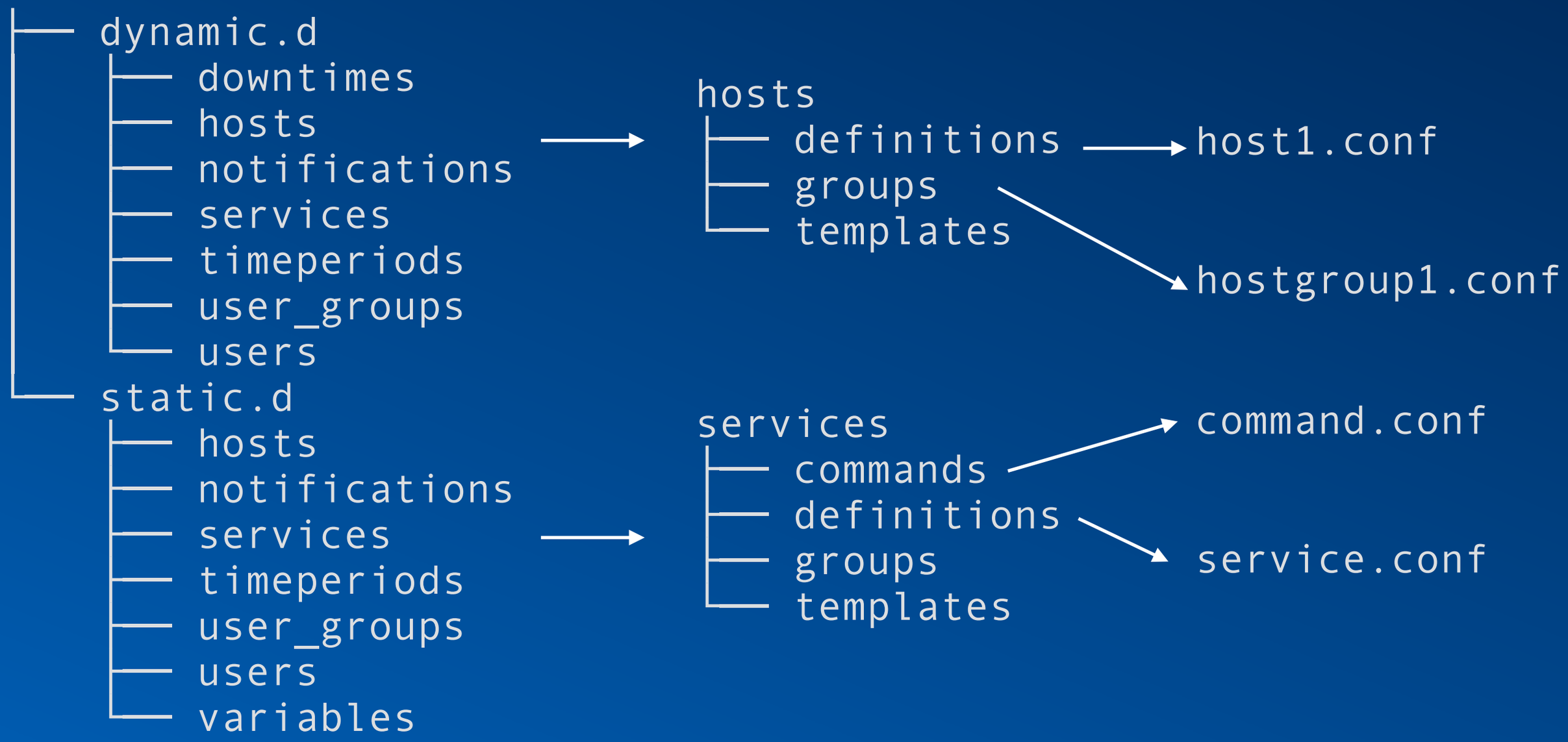
```
├── dynamic.d
│   ├── downtimes
│   ├── hosts
│   ├── notifications
│   ├── services
│   ├── timeperiods
│   ├── user_groups
│   └── users
└── static.d
    ├── hosts
    ├── notifications
    ├── services
    ├── timeperiods
    ├── user_groups
    ├── users
    └── variables
```

Directory structure



Livesport

Directory structure



Livesport

Items to generate

- `somehost.conf`
- `someuser.conf`
- `somegroup.conf`

Hosts

- IP
  - Main (management)
  - Public
  - V6 also
- Services (systemd)
- Tags (notification settings)
- Other variables we need

Hosts

```
object Host "ourhost.somedomain" {
     import "debian-server"
     address = "192.168.240.234"

     # Variables
     vars.addresses = {
          pub = ["90.91.92.93"]
          priv = ["192.168.231.34"]
          pub6 = ["2d0f:7d50:1:240:5644:ff:fe43:52ae"]
     }
     vars.systemd_services = ["apache2", "atd", "autovt@", "bbu-disable-check", "bbu-
disable", "cron", "dsm_sa_ipmi", "fail2ban", "getty@", "instsvcdrv", "ipmidrv",
"lslogbeat", "lvm2-monitor", "monitor-client", "nagios-api", "nagios-nrpe-server"]
     vars.server_tags = {
          "Service" = [ "MySql","Apache","PHP" ]
          "DeploymentState" = [ "Production" ]
          "Server" = [ "Virtual" ]
          "Project" = [ "ED" ]
          "NotificationPeriod" = [ "24x7" ]
          "Type" = [ "Management" ]
     }
     vars.md_raid_active = "no"
     vars.swap_enabled = "no"
     vars.mysqlserver_package_name = ""
     vars.is_virtual = "true"
}
```


Livesport

Users

- Name :-)
- Contact details (pager, email)
- States for notifications
- Types for notifications
- Oncall state
- Team / group

Users

```
object User "jakub.stollmann" {
    import "generic-user"

    display_name = "Jakub Stollmann"
    groups += [ "TECH" ]
    email = "jakub.stollmann@livesport.eu"
    pager = "123456789"

    vars.oncall = "False"

    vars.notification_options = {
        sms = [ "acknowledgement", "critical", "ok" ]
        email = [ "unknown" ]
     }

  vars.notifications_override = "off"


  }
```

Groups

- Name is enough

What about services?

- We won't generate global services automatically
- We cannot automate this easily due to the number of options

What about services?

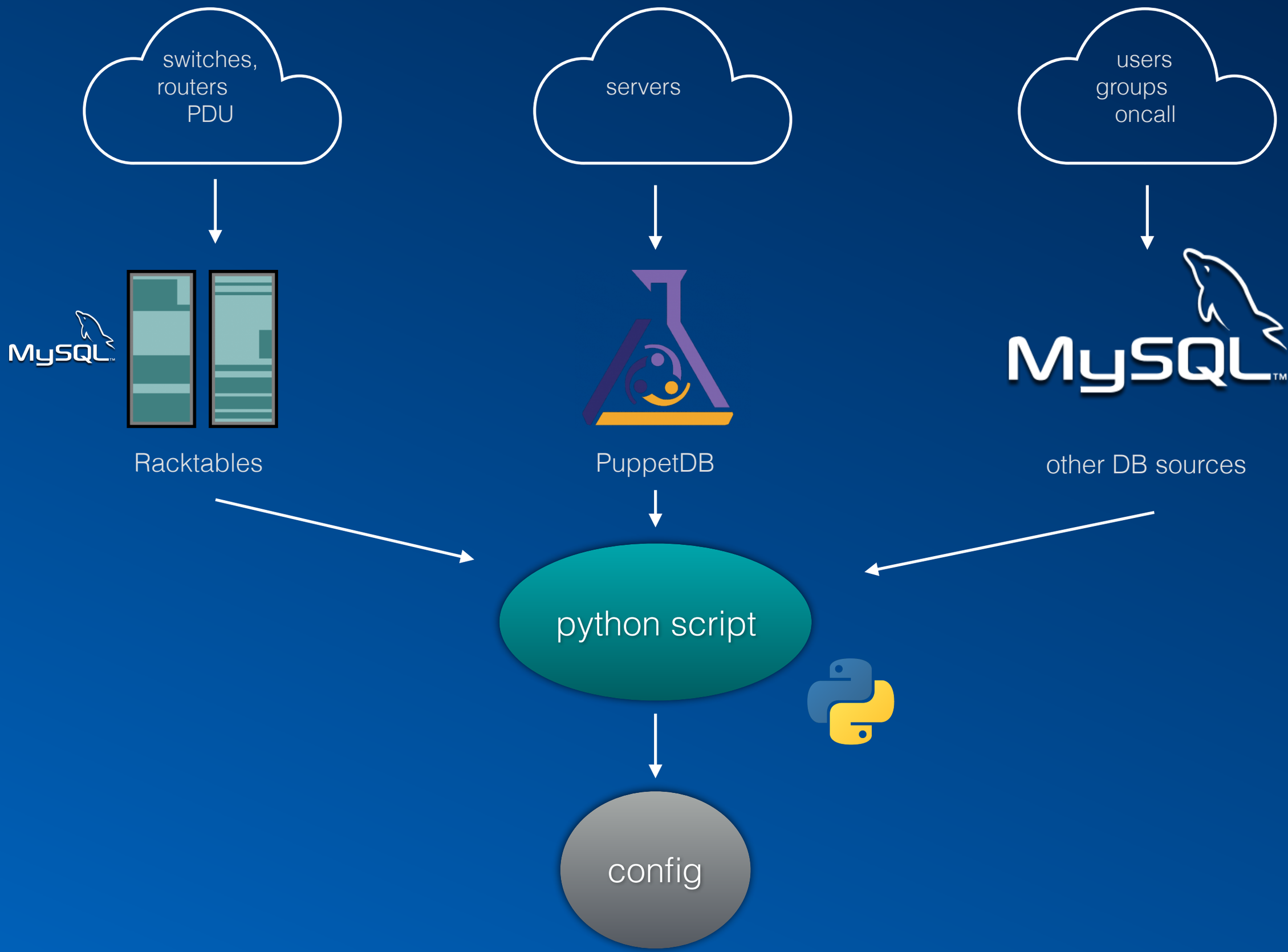
- We won't generate global services automatically
- We cannot automate this easily due to the number of options
- But we can generate simple services - like HTTP check

What about services?

- We define them manually

Service

```
apply Service "HTTP Stack - Nginx" {
    import "generic-tech-service"
    check_command = "nrpe"

    vars.notifications = false
    vars.nrpe_command = "check_http_stack"
    vars.nrpe_arguments = [ "-n" ]

    assign where "nginx" in host.vars.systemd_services
    ignore where "httpstack.disable" in
host.vars.server_tags["Monitoring"]

     notes = ("Checking http://localhost/nginx_status URL.
Command(nrpe): " + vars.nrpe_command + " " +
vars.nrpe_arguments.join(" "))

}
```


switches,
routers
PDU
servers
users
groups
oncall
MySQL
MySQL
Racktables
PuppetDB
other DB sources
python script
config

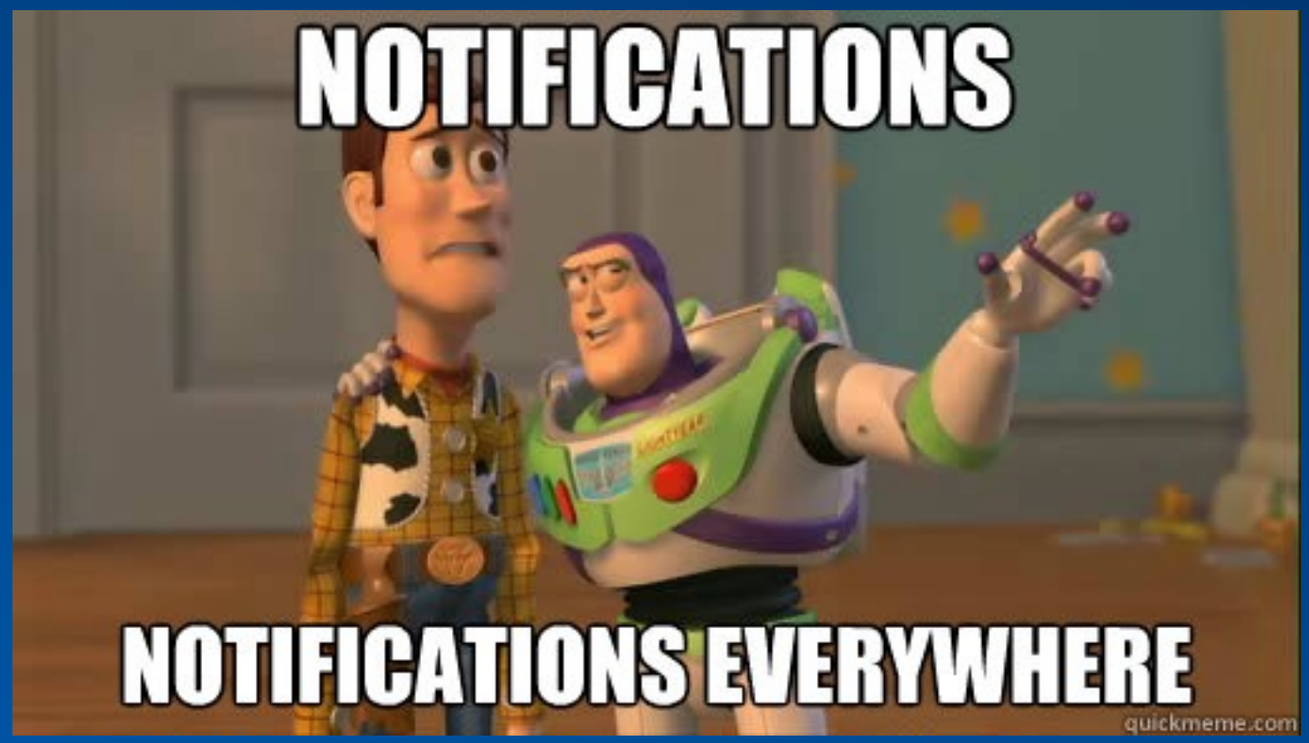
Livesport

Achievements:

- Automated configuration ✓
- Automatic on-call set up ✓
- Multi-channel notifications
- Tailored notifications
- HA cluster & possibility of multiple locations
- Readable frontend

Livesport

Notifications



Notifications

- Icinga has quite simple notification concept

Notifications

- Icinga has quite simple notification concept

- Simple
- Pretty fast
- Partly flexible - you can use anything as your command (also `/bin/true`)
- But...

Livesport

Notifications

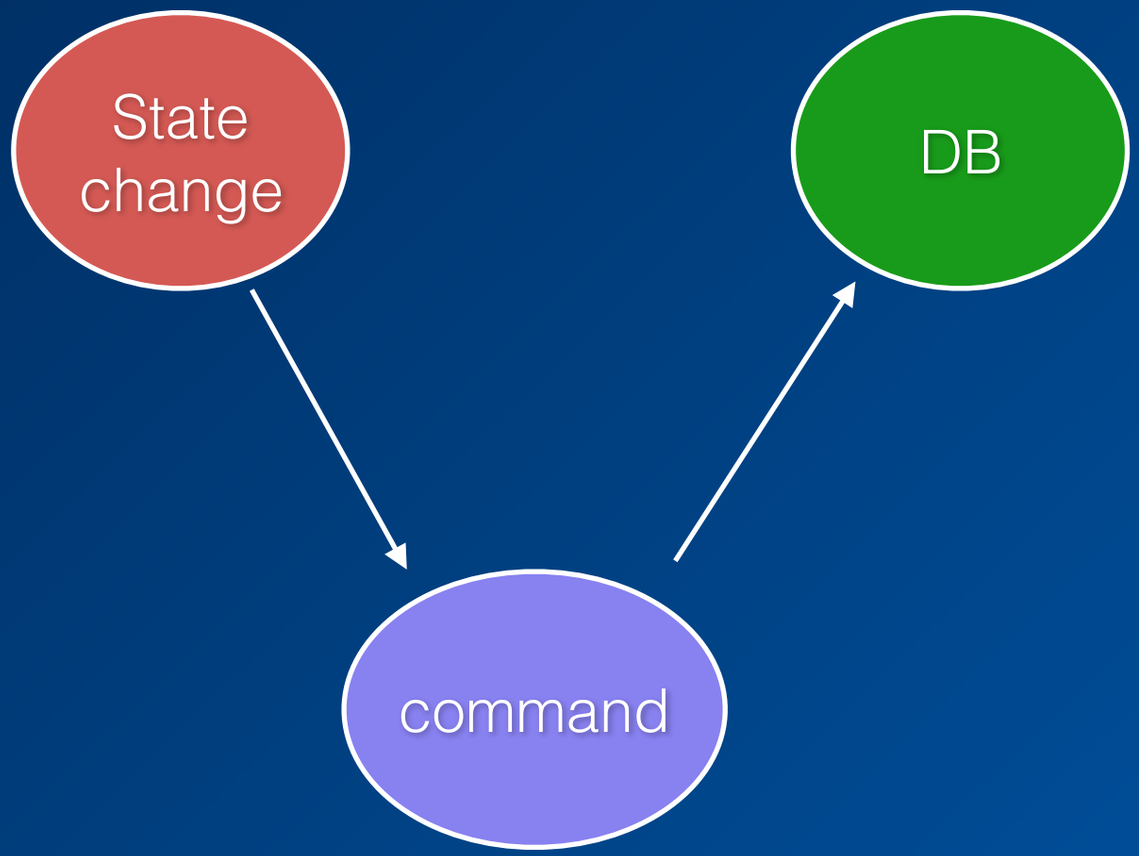
- **No aggregation**
- Every notification is sent per event (state change)

How do we solve that?

- Parser
- Aggregator
- Sender

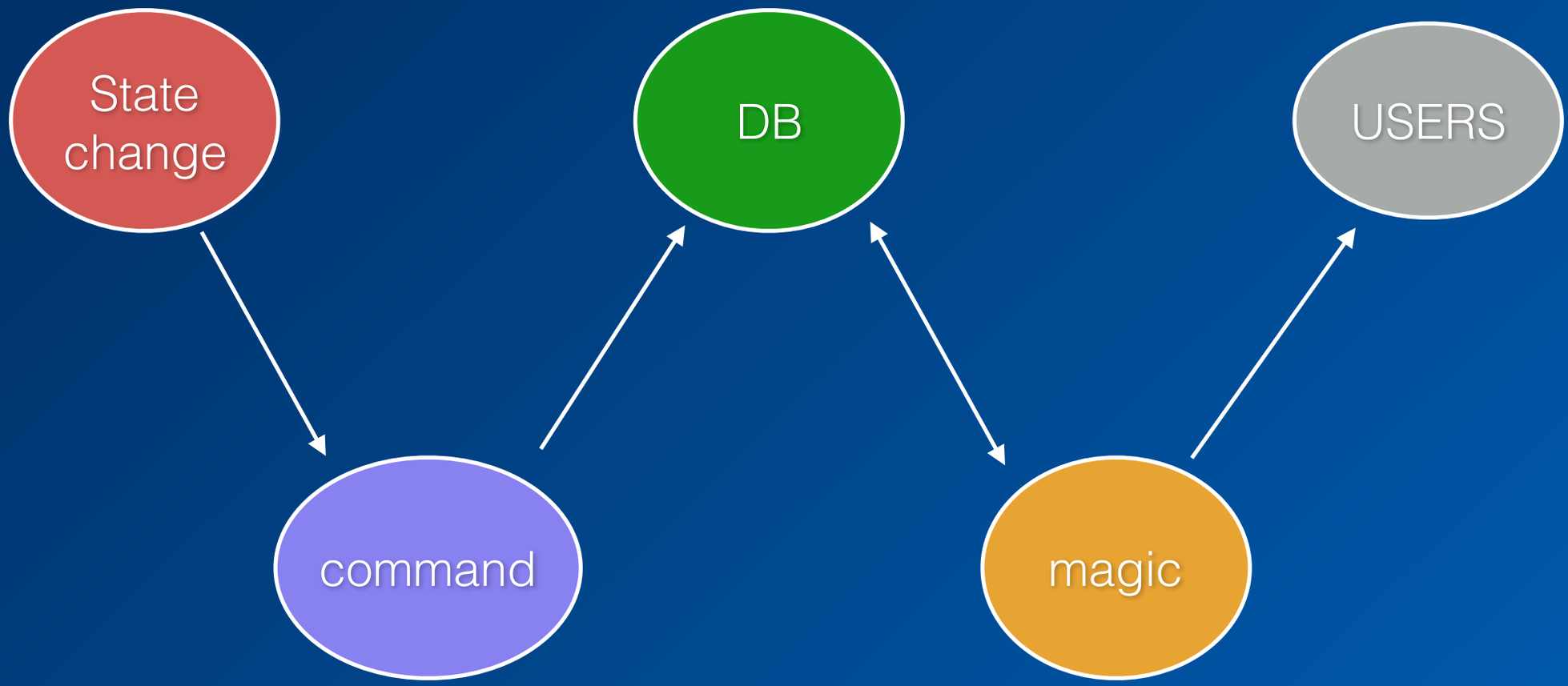
- Command?
- Another level?

How do we solve that?



Livesport

How do we solve that?



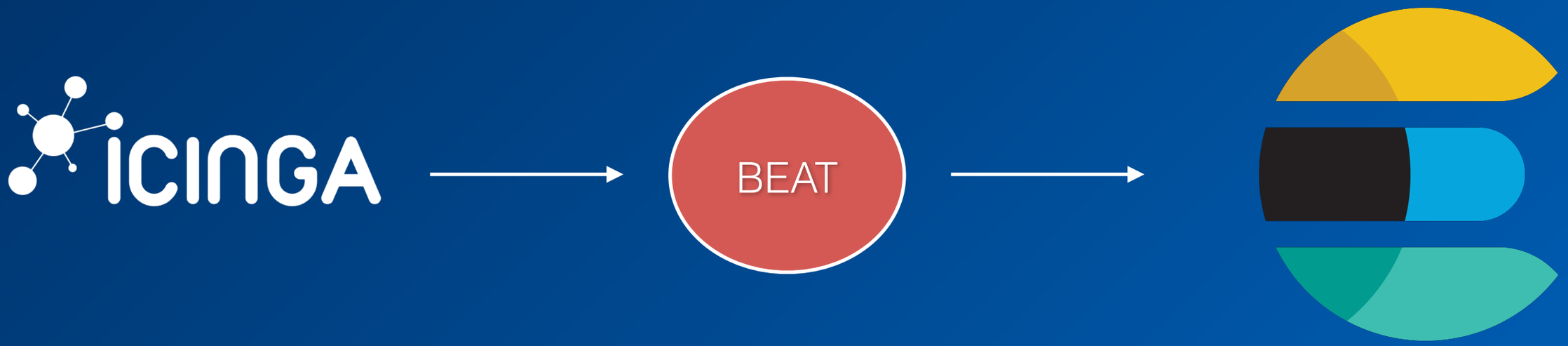
Livesport

In a nutshell

- Raise notification
- Save it to DB
- Read from DB
- Parse, aggregate, generate output
- Send to the user via desired channels

Let's dig into it

- Send notifications to DB? How?
- IcingaBeat!



Let's dig into it

- **Icinga-notificator**
  - Daemon
  - Written in python (by me)
  - Modular
  - Open source
  - https://github.com/pershinghar/icinga-notificator

Let's dig into it

- **Icinga-notificator**
  - Periodically checks for new notifications in DB
    - The key is the correct period
  - If there are, it processes them
  - Can send them via various channels
    - SMS
    - Call (only for a ring)
    - Slack (in beta)
    - Email

Livesport

Achievements:

- Automated configuration
- Automatic on-call set up
- Multi channel notifications ✓
- Tailored notifications ✓
- HA cluster & possibility of multiple locations
- Readable frontend

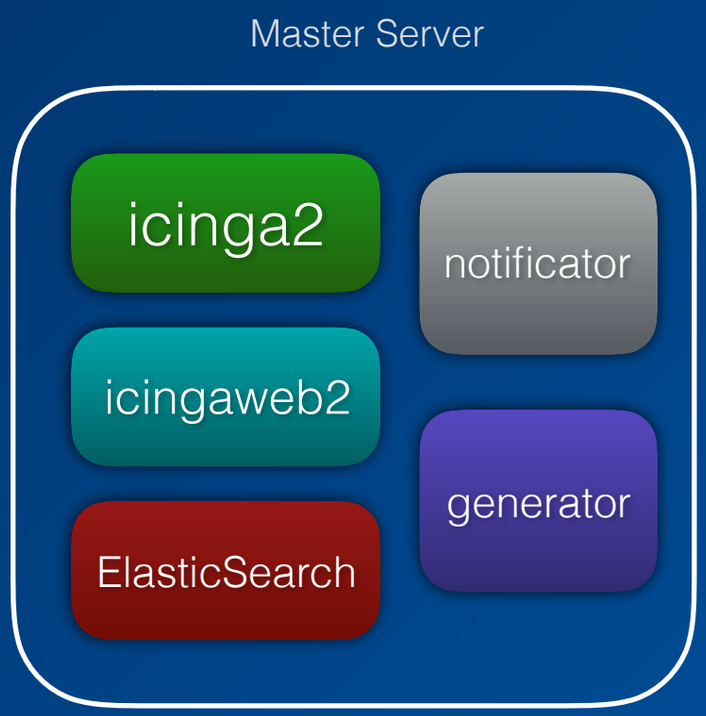
Livesport

HA

Livesport

HA

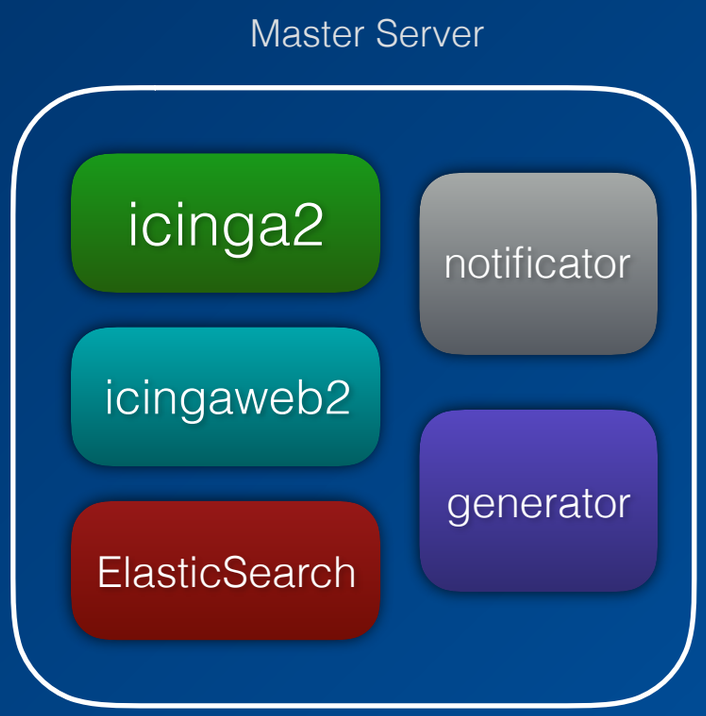
- Actually, Icinga2 natively supports HA
- Concept:
  - 2 "master" server per location - dual master architecture
  - ES on every server



Livesport

HA

- Notificator
  - Handling notifications per server
  - Simple solution
  - Aggregation not so great



Livesport

Multiple locations

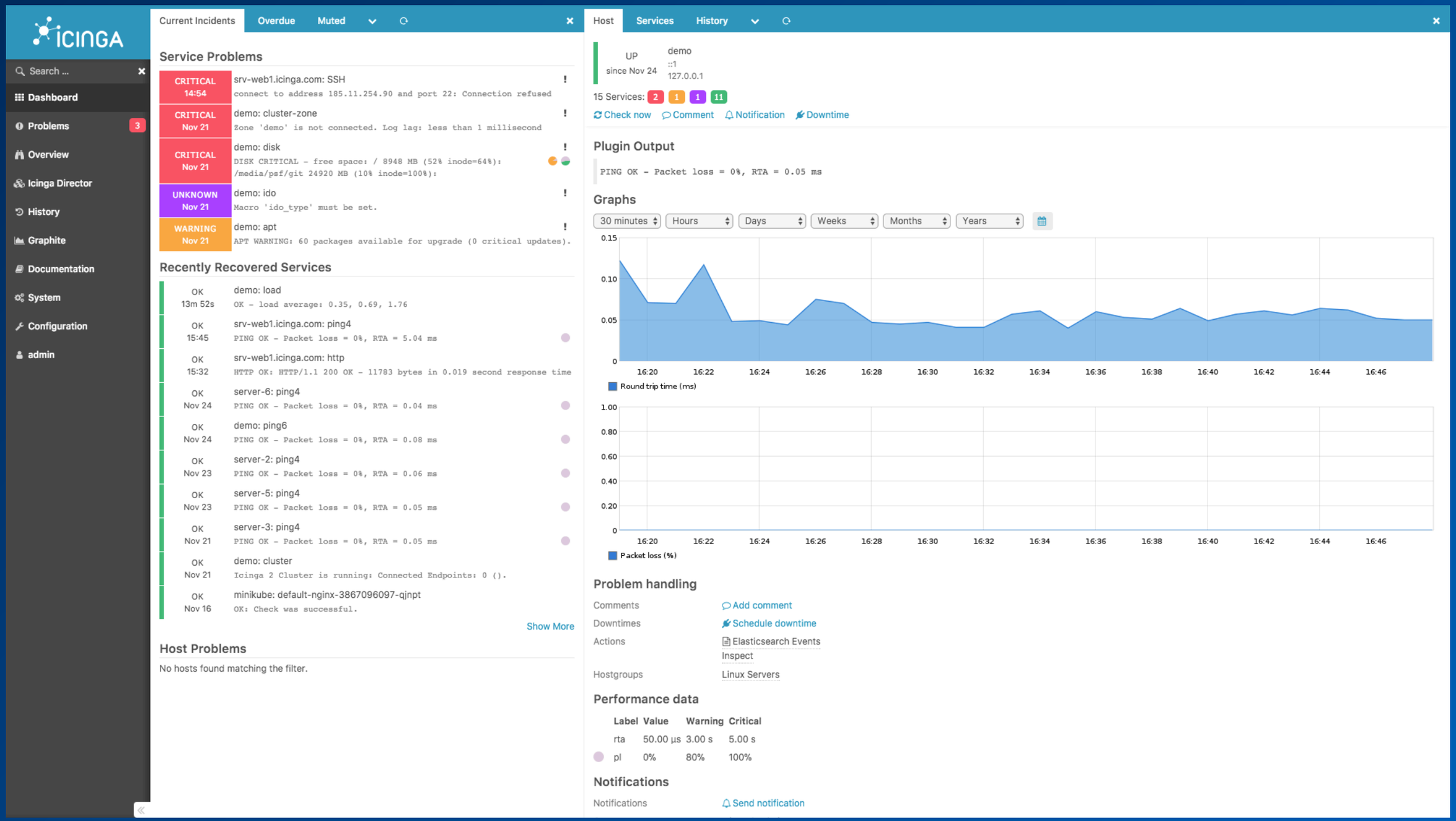
- Easily deploy Icinga anywhere
- Use puppet
  - Possibility to use official module:
    - https://github.com/Icinga/puppet-icinga2
- Summing data?
  - Possible via HA architecture
  - We are not using this

Frontend

Livesport

Frontend

- **Icingaweb2**
  - Default for icinga2
- You can also use others like Thruk


Search ...
Dashboard
Problems 3
Overview
Icinga Director
History
Graphite
Documentation
System
Configuration
admin
Current Incidents
Overdue
Muted
Service Problems
CRITICAL 14:54
srv-web1.icinga.com: SSH
connect to address 185.11.254.90 and port 22: Connection refused
CRITICAL Nov 21
demo: cluster-zone
Zone 'demo' is not connected. Log lag: less than 1 millisecond
CRITICAL Nov 21
demo: disk
DISK CRITICAL - free space: / 8948 MB (52% inode=64%): /media/psf/git 24920 MB (10% inode=100%):
UNKNOWN Nov 21
demo: ido
Macro 'ido_type' must be set.
WARNING Nov 21
demo: apt
APT WARNING: 60 packages available for upgrade (0 critical updates).
Recently Recovered Services
OK 13m 52s
demo: load
OK - load average: 0.35, 0.69, 1.76
OK 15:45
srv-web1.icinga.com: ping4
PING OK - Packet loss = 0%, RTA = 5.04 ms
OK 15:32
srv-web1.icinga.com: http
HTTP OK: HTTP/1.1 200 OK - 11783 bytes in 0.019 second response time
OK Nov 24
server-6: ping4
PING OK - Packet loss = 0%, RTA = 0.04 ms
OK Nov 24
demo: ping6
PING OK - Packet loss = 0%, RTA = 0.08 ms
OK Nov 23
server-2: ping4
PING OK - Packet loss = 0%, RTA = 0.06 ms
OK Nov 23
server-5: ping4
PING OK - Packet loss = 0%, RTA = 0.05 ms
OK Nov 21
server-3: ping4
PING OK - Packet loss = 0%, RTA = 0.05 ms
OK Nov 21
demo: cluster
Icinga 2 Cluster is running: Connected Endpoints: 0 ().
OK Nov 16
minikube: default-nginx-3867096097-qjnpt
OK: Check was successful.
Show More
Host Problems
No hosts found matching the filter.
Host
Services
History
UP since Nov 24
demo
::1
127.0.0.1
15 Services: 2 1 1 11
Check now
Comment
Notification
Downtime
Plugin Output
PING OK - Packet loss = 0%, RTA = 0.05 ms
Graphs
30 minutes
Hours
Days
Weeks
Months
Years
0.15
0.10
0.05
0
16:20 16:22 16:24 16:26 16:28 16:30 16:32 16:34 16:36 16:38 16:40 16:42 16:44 16:46
Round trip time (ms)
1.00
0.80
0.60
0.40
0.20
0
16:20 16:22 16:24 16:26 16:28 16:30 16:32 16:34 16:36 16:38 16:40 16:42 16:44 16:46
Packet loss (%)
Problem handling
Comments
Add comment
Downtimes
Schedule downtime
Actions
Elasticsearch Events
Inspect
Hostgroups
Linux Servers
Performance data
Label Value Warning Critical
rta 50.00 µs 3.00 s 5.00 s
pl 0% 80% 100%
Notifications
Notifications
Send notification

Achievements:

- Automated configuration
- Automatic on-call set up
- Multi-channel notifications
- Tailored notifications
- HA cluster & possibility of multiple locations ✓
- Readable frontend ✓

Livesport

Livesport

Future

- Full HA in notificator
- Automated services generation (for specific types)
- Full deployment with puppet
- More tailored notifications :-)

Useful links

- https://icinga.com
- https://github.com/pershinghar/icinga-notificator
- https://icinga.com/docs/icingabeat/latest/
- https://github.com/Icinga/puppet-icinga2
- https://giphy.com

- slides: https://github.com/pershinghar/linuxdays19

Thanks for attention

Questions?

Livesport