

Deploying Icinga2 through Ansible

Monitoring migration through automation

Toshaan Bharvani - VanTosh bvba

<toshaan@vantosh.com>



Icinga Camp Bangalore 2017

13 May 2017

\$ whoami

Toshaan Bharvani - VanTosh

- From Antwerp, Belgium
- Self-employed engineer/trainer (available for hire)
<http://www.vantosh.com>
- Involved with Enterprise OS : PEL, RHEL, CentOS, IBM AIX, OpenBSD, FreeBSD, SLES, ...
- Likes to keep everything secure : SELinux, WebSec, ...
- Lives in a virtual world : KVM, Xen, LXC, PowerVM, z/VM, ...
- Likes automation CfgMgmt / DevOps : Ansible, Foreman, Puppet
- Works on both hardware and software side
- Wants to take over the universe
- Twitter : [@toshywoshy](https://twitter.com/toshywoshy)
- Blog : <http://www.toshaan.com>

Table of contents

Monitoring

Automation

Inventory

Virtual
Machines

Icinga2

Icinga2 Masters

Icinga2 Clients

Conclusion

The End

- 1 Monitoring
- 2 Automation
Inventory
Virtual Machines
- 3 Icinga2
Icinga2 Masters
Icinga2 Clients
- 4 Conclusion

Deploying
Icinga2 through
Ansible

Toshaan
Bharvani -
VanTosh bvba

Monitoring

Automation

Inventory

Virtual
Machines

Icinga2

Icinga2 Masters

Icinga2 Clients

Conclusion

The End

1

Monitoring

Monitoring

Deploying
Icinga2 through
Ansible

Toshaan
Bharvani -
VanTosh bvba

Monitoring

Automation

Inventory

Virtual
Machines

Icinga2

Icinga2 Masters

Icinga2 Clients

Conclusion

The End



Choices

- Nagios
 - Well know, good for smaller install base
- Zenoss
 - Big coupled project without any cohesion
- Zabbix
 - Distributed, all-in-one solution, good
- Ganglia
 - RRD as static, web interface is dated
- Icinga
 - Better web interface with nagios forked backend
- OpsView
 - Fancier web interface which is liked by managers, nagios forked
- Shinken
 - Distributed, modular, good for small install base
- Icinga2
 - Distributed, better scalability, better interface, modular and integrations

Choices

- Nagios
 - Well know, good for smaller install base
- Zenoss
 - Big coupled project without any cohesion
- Zabbix
 - Distributed, all-in-one solution, good
- Ganglia
 - RRD as static, web interface is dated
- Icinga
 - Better web interface with nagios forked backend
- OpsView
 - Fancier web interface which is liked by managers, nagios forked
- Shinken
 - Distributed, modular, good for small install base
- Icinga2
 - Distributed, better scalability, better interface, modular and integrations

Choices

- Nagios
 - Well know, good for smaller install base
- Zenoss
 - Big coupled project without any cohesion
- Zabbix
 - Distributed, all-in-one solution, good
- Ganglia
 - RRD as static, web interface is dated
- Icinga
 - Better web interface with nagios forked backend
- OpsView
 - Fancier web interface which is liked by managers, nagios forked
- Shinken
 - Distributed, modular, good for small install base
- Icinga2
 - Distributed, better scalability, better interface, modular and integrations

Choices

- Nagios
 - Well know, good for smaller install base
- Zenoss
 - Big coupled project without any cohesion
- Zabbix
 - Distributed, all-in-one solution, good
- Ganglia
 - RRD as static, web interface is dated
- Icinga
 - Better web interface with nagios forked backend
- OpsView
 - Fancier web interface which is liked by managers, nagios forked
- Shinken
 - Distributed, modular, good for small install base
- Icinga2
 - Distributed, better scalability, better interface, modular and integrations

Choices

- Nagios
 - Well know, good for smaller install base
- Zenoss
 - Big coupled project without any cohesion
- Zabbix
 - Distributed, all-in-one solution, good
- Ganglia
 - RRD as static, web interface is dated
- Icinga
 - Better web interface with nagios forked backend
- OpsView
 - Fancier web interface which is liked by managers, nagios forked
- Shinken
 - Distributed, modular, good for small install base
- Icinga2
 - Distributed, better scalability, better interface, modular and integrations

Choices

- Nagios
 - Well know, good for smaller install base
- Zenoss
 - Big coupled project without any cohesion
- Zabbix
 - Distributed, all-in-one solution, good
- Ganglia
 - * RRD as static web interface is dated
- Icinga
 - * Better web interface with nagios forked backend
- OpsView
 - * Fancier web interface which is liked by managers, nagios forked
- Shinken
 - * Distributed, modular, good for small install base
- Icinga2
 - * Distributed, better scalability, better interface, modular and integrations

Choices

- Nagios
 - Well know, good for smaller install base
- Zenoss
 - Big coupled project without any cohesion
- Zabbix
 - Distributed, all-in-one solution, good
- Ganglia
 - RRD as static, web interface is dated
- Icinga
 - Better web interface with nagios forked backend
- OpsView
 - Fancier web interface which is liked by managers, nagios forked
- Shinken
 - Distributed, modular, good for small install base
- Icinga2
 - Distributed, better scalability, better interface, modular and integrations

Choices

- Nagios
 - Well know, good for smaller install base
- Zenoss
 - Big coupled project without any cohesion
- Zabbix
 - Distributed, all-in-one solution, good
- Ganglia
 - RRD as static, web interface is dated
- Icinga
 - Better web interface with nagios forked backend
- OpsView
 - Fancier web interface which is liked by managers, nagios forked
- Shinken
 - Distributed, modular, good for small install base
- Icinga2
 - Distributed, better scalability, better interface, modular and integrations

Choices

- Nagios
 - Well know, good for smaller install base
- Zenoss
 - Big coupled project without any cohesion
- Zabbix
 - Distributed, all-in-one solution, good
- Ganglia
 - RRD as static, web interface is dated
- Icinga
 - Better web interface with nagios forked backend
- OpsView
 - Fancier web interface which is liked by managers, nagios forked
- Shinken
 - Distributed, modular, good for small install base
- Icinga2
 - Distributed, better scalability, better interface, modular and integrations

Choices

- Nagios
 - Well know, good for smaller install base
- Zenoss
 - Big coupled project without any cohesion
- Zabbix
 - Distributed, all-in-one solution, good
- Ganglia
 - RRD as static, web interface is dated
- Icinga
 - Better web interface with nagios forked backend
- OpsView
 - Fancier web interface which is liked by managers, nagios forked
- Shinken
 - Distributed, modular, good for small install base
- Icinga2
 - Distributed, better scalability, better interface, modular and integrations

Choices

- Nagios
 - Well know, good for smaller install base
- Zenoss
 - Big coupled project without any cohesion
- Zabbix
 - Distributed, all-in-one solution, good
- Ganglia
 - RRD as static, web interface is dated
- Icinga
 - Better web interface with nagios forked backend
- OpsView
 - Fancier web interface which is liked by managers, nagios forked
- Shinken
 - Distributed, modular, good for small install base
- Icinga2
 - Distributed, better scalability, better interface, modular and integrations

Choices

- Nagios
 - Well know, good for smaller install base
- Zenoss
 - Big coupled project without any cohesion
- Zabbix
 - Distributed, all-in-one solution, good
- Ganglia
 - RRD as static, web interface is dated
- Icinga
 - Better web interface with nagios forked backend
- OpsView
 - Fancier web interface which is liked by managers, nagios forked
- Shinken
 - Distributed, modular, good for small install base
- Icinga2
 - Distributed, better scalability, better interface, modular and integrations

Choices

- Nagios
 - Well know, good for smaller install base
- Zenoss
 - Big coupled project without any cohesion
- Zabbix
 - Distributed, all-in-one solution, good
- Ganglia
 - RRD as static, web interface is dated
- Icinga
 - Better web interface with nagios forked backend
- OpsView
 - Fancier web interface which is liked by managers, nagios forked
- Shinken
 - Distributed, modular, good for smal install base
- Icinga2
 - Distributed, better scalability, better interface, modular and integrations

Choices

- Nagios
 - Well know, good for smaller install base
- Zenoss
 - Big coupled project without any cohesion
- Zabbix
 - Distributed, all-in-one solution, good
- Ganglia
 - RRD as static, web interface is dated
- Icinga
 - Better web interface with nagios forked backend
- OpsView
 - Fancier web interface which is liked by managers, nagios forked
- Shinken
 - Distributed, modular, good for smal install base
- Icinga2
 - Distributed, better stability, better interface, modular and integrations

Choices

- Nagios
 - Well know, good for smaller install base
- Zenoss
 - Big coupled project without any cohesion
- Zabbix
 - Distributed, all-in-one solution, good
- Ganglia
 - RRD as static, web interface is dated
- Icinga
 - Better web interface with nagios forked backend
- OpsView
 - Fancier web interface which is liked by managers, nagios forked
- Shinken
 - Distributed, modular, good for smal install base
- Icinga2
 - Distributed, better scalability, better interface, modular and integrations

Choices

- Nagios
 - Well know, good for smaller install base
- Zenoss
 - Big coupled project without any cohesion
- Zabbix
 - Distributed, all-in-one solution, good
- Ganglia
 - RRD as static, web interface is dated
- Icinga
 - Better web interface with nagios forked backend
- OpsView
 - Fancier web interface which is liked by managers, nagios forked
- Shinken
 - Distributed, modular, good for smal install base
- Icinga2
 - Distributed, better scalability, better interface, modular and integrations

Deploying
Icinga2 through
Ansible

Toshaan
Bharvani -
VanTosh bvba

Monitoring

Automation

Inventory

Virtual
Machines

Icinga2

Icinga2 Masters

Icinga2 Clients

Conclusion

The End

2

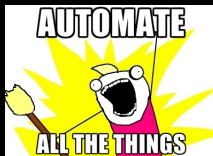
Automation

Introducing



“Ansible is a fictional machine capable of instantaneous or superluminal communication. Typically it is depicted as a lunch-box-sized object with some combination of microphone, speaker, keyboard and display. It can send and receive messages to and from a corresponding device over any distance whatsoever with no delay.”¹

- Configuration Management Tool
- System Orchestration Tool
- Remote Execution/Deployment Tool
- ...



An overview

- Python2 based
 - Server based, agentless²
 - uses SSH protocol (can use accelerated mode uses a daemon and port)
 - host information in flat files, CMDB, scripts, ...
 - executes the task on the host side
 - Playbook : combination of tasks with meta information
 - YAML
 - JSON
 - Roles : abstract playbook
 - Tasks
 - Variables
 - Handlers
 - Templates : Jinja2
 - works where Python2 works³

² Abstraction of SSH / WinRM to connect with the node and in certain setup an agent might be required, but by default it is not necessary

³ Ansible can actually run without Python on the remote host, however it is not fully supported

An overview

- Python2 based
- Server based, agentless²
 - uses SSH protocol (can use accelerated mode uses a daemon and port)
 - host information in flat files, CMDB, scripts, ...
 - executes the task on the host side
 - Playbook : combination of tasks with meta information
 - YAML
 - JSON
 - Roles : abstract playbook
 - Tasks
 - Variables
 - Handlers
 - Templates : Jinja2
 - works where Python2 works³

² Abstraction of SSH / WinRM to connect with the node and in certain setup an agent might be required, but by default it is not necessary

³ Ansible can actually run without Python on the remote host, however it is not fully supported

An overview

- Python2 based
- Server based, agentless²
- uses SSH protocol (can use accelerated mode uses a daemon and port)
 - host information in flat files, CMDB, scripts, ...
 - executes the task on the host side
 - Playbook : combination of tasks with meta information
 - YAML
 - JSON
 - Roles : abstract playbook
 - Tasks
 - Variables
 - Handlers
 - Templates : Jinja2
 - works where Python2 works³

² Abstraction of SSH / WinRM to connect with the node and in certain setup an agent might be required, but by default it is not necessary

³ Ansible can actually run without Python on the remote host, however it is not fully supported

An overview

- Python2 based
- Server based, agentless²
- uses SSH protocol (can use accelerated mode uses a daemon and port)
- host information in flat files, CMDB, scripts, ...
 - executes the task on the host side
 - Playbook : combination of tasks with meta information
 - YAML
 - JSON
 - Roles : abstract playbook
 - Tasks
 - Variables
 - Handlers
 - Templates : Jinja2
 - works where Python2 works³

² Abstraction of SSH / WinRM to connect with the node and in certain setup an agent might be required, but by default it is not necessary

³ Ansible can actually run without Python on the remote host, however it is not fully supported

An overview

- Python2 based
- Server based, agentless²
- uses SSH protocol (can use accelerated mode uses a daemon and port)
- host information in flat files, CMDB, scripts, ...
- executes the task on the host side
 - Playbook : combination of tasks with meta information
 - YAML
 - JSON
 - Roles : abstract playbook
 - Tasks
 - Variables
 - Handlers
 - Templates : Jinja2
 - works where Python2 works³

² Abstraction of SSH / WinRM to connect with the node and in certain setup an agent might be required, but by default it is not necessary

³ Ansible can actually run without Python on the remote host, however it is not fully supported

An overview

- Python2 based
- Server based, agentless²
- uses SSH protocol (can use accelerated mode uses a daemon and port)
- host information in flat files, CMDB, scripts, ...
- executes the task on the host side
- Playbook : combination of tasks with meta information
 - YAML
 - JSON
- Roles : abstract playbook
 - Tasks
 - Variables
 - Handlers
- Templates : Jinja2
- works where Python2 works³

² Abstraction of SSH / WinRM to connect with the node and in certain setup an agent might be required, but by default it is not necessary

³ Ansible can actually run without Python on the remote host, however it is not fully supported

An overview

- Python2 based
- Server based, agentless²
- uses SSH protocol (can use accelerated mode uses a daemon and port)
- host information in flat files, CMDB, scripts, ...
- executes the task on the host side
- Playbook : combination of tasks with meta information
 - YAML
 - JSON
- Roles : abstract playbook
 - Tasks
 - Variables
 - Handlers
- Templates : Jinja2
- works where Python2 works³

² Abstraction of SSH / WinRM to connect with the node and in certain setup an agent might be required, but by default it is not necessary

³ Ansible can actually run without Python on the remote host, however it is not fully supported

An overview

- Python2 based
- Server based, agentless²
- uses SSH protocol (can use accelerated mode uses a daemon and port)
- host information in flat files, CMDB, scripts, ...
- executes the task on the host side
- Playbook : combination of tasks with meta information
 - YAML
 - JSON
- Roles : abstract playbook
 - Tasks
 - Variables
 - Handlers
- Templates : Jinja2
- works where Python2 works³

² Abstraction of SSH / WinRM to connect with the node and in certain setup an agent might be required, but by default it is not necessary

³ Ansible can actually run without Python on the remote host, however it is not fully supported

An overview

- Python2 based
- Server based, agentless²
- uses SSH protocol (can use accelerated mode uses a daemon and port)
- host information in flat files, CMDB, scripts, ...
- executes the task on the host side
- Playbook : combination of tasks with meta information
 - YAML
 - JSON
- Roles : abstract playbook
 - Tasks
 - Variables
 - Handlers
- Templates : Jinja2
- works where Python2 works³

² Abstraction of SSH / WinRM to connect with the node and in certain setup an agent might be required, but by default it is not necessary

³ Ansible can actually run without Python on the remote host, however it is not fully supported

An overview

- Python2 based
- Server based, agentless²
- uses SSH protocol (can use accelerated mode uses a daemon and port)
- host information in flat files, CMDB, scripts, ...
- executes the task on the host side
- Playbook : combination of tasks with meta information
 - YAML
 - JSON
- Roles : abstract playbook
 - Tasks
 - Variables
 - Handlers
 - Templates : Jinja2
 - works where Python2 works³

² Abstraction of SSH / WinRM to connect with the node and in certain setup an agent might be required, but by default it is not necessary

³ Ansible can actually run without Python on the remote host, however it is not fully supported

An overview

- Python2 based
- Server based, agentless²
- uses SSH protocol (can use accelerated mode uses a daemon and port)
- host information in flat files, CMDB, scripts, ...
- executes the task on the host side
- Playbook : combination of tasks with meta information
 - YAML
 - JSON
- Roles : abstract playbook
 - Tasks
 - Variables
 - Handlers
- Templates : Jinja2
- works where Python2 works³

² Abstraction of SSH / WinRM to connect with the node and in certain setup an agent might be required, but by default it is not necessary

³ Ansible can actually run without Python on the remote host, however it is not fully supported

An overview

- Python2 based
- Server based, agentless²
- uses SSH protocol (can use accelerated mode uses a daemon and port)
- host information in flat files, CMDB, scripts, ...
- executes the task on the host side
- Playbook : combination of tasks with meta information
 - YAML
 - JSON
- Roles : abstract playbook
 - Tasks
 - Variables
 - Handlers
- Templates : Jinja2
- works where Python2 works³

² Abstraction of SSH / WinRM to connect with the node and in certain setup an agent might be required, but by default it is not necessary

³ Ansible can actually run without Python on the remote host, however it is not fully supported

An overview

- Python2 based
- Server based, agentless²
- uses SSH protocol (can use accelerated mode uses a daemon and port)
- host information in flat files, CMDB, scripts, ...
- executes the task on the host side
- Playbook : combination of tasks with meta information
 - YAML
 - JSON
- Roles : abstract playbook
 - Tasks
 - Variables
 - Handlers
- Templates : Jinja2
- works where Python2 works³

² Abstraction of SSH / WinRM to connect with the node and in certain setup an agent might be required, but by default it is not necessary

³ Ansible can actually run without Python on the remote host, however it is not fully supported

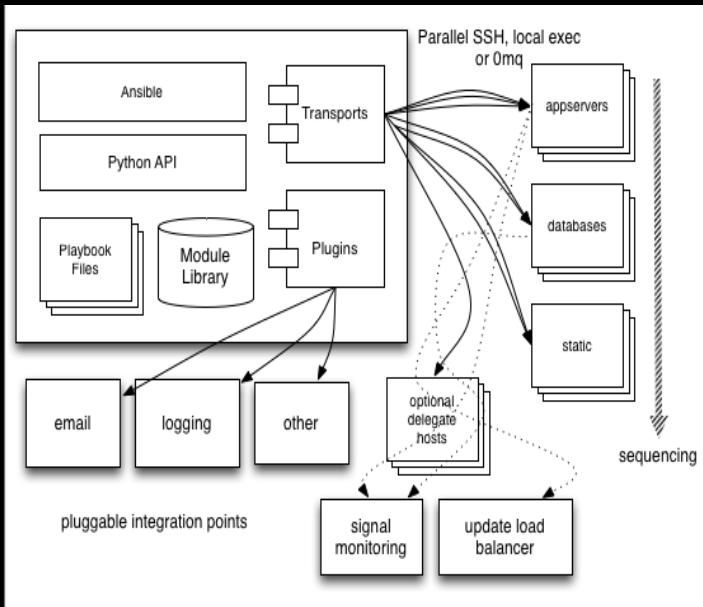
An overview

- Python2 based
- Server based, agentless²
- uses SSH protocol (can use accelerated mode uses a daemon and port)
- host information in flat files, CMDB, scripts, . . .
- executes the task on the host side
- Playbook : combination of tasks with meta information
 - YAML
 - JSON
- Roles : abstract playbook
 - Tasks
 - Variables
 - Handlers
- Templates : Jinja2
- works where Python2 works³

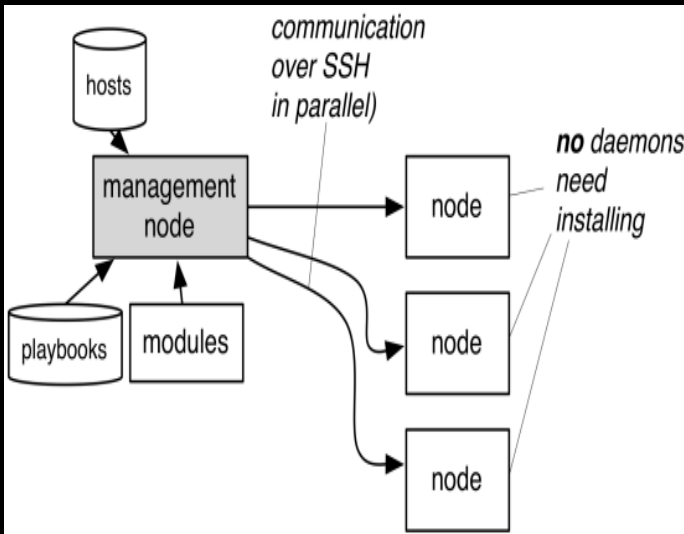
² Abstraction of SSH / WinRM to connect with the node and in certain setup an agent might be required, but by default it is not necessary

³ Ansible can actually run without Python on the remote host, however it is not fully supported

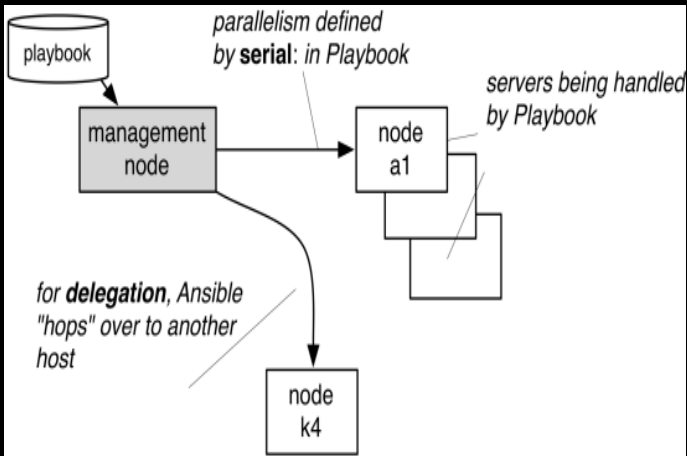
Internal Design



Normal Adhoc Mode



Delegation Mode



Inventory - Hypervisors

- hosts file is by default in ini format

```
1 [hypervisors]
2     hypervisor0
3     hypervisor1
4     hypervisor2
5     hypervisor3
6
7 [virtualmachines]
8     vm0
9     vm1
10    vm2
11    vm3
```

Inventory - Hypervisor (0)

```
1 ---
2 distro: powerel7
3 machinearch: ppc64
4 rootpwd: $1$f0pPKH0e$0xrX07K19DPmpcmJooede7X.
5 language: en_US
6 keyboard: us
7 timezone: "Europe/Brussels"
8 nics:
9   - { type: bridge , name: br0 , model: virtio , bootproto: dhcp , device:
10     eth0 , onboot: on }
11 hostname: hyp0.vantosh.demo
12 sshdrsakeylength: 8192
13 sshdecdsakeylength: 521
14 users:
15   - { name: supervisor , id: 1000 , ssh: yes , sshpubkey: /home/toshywoshy/.
16     keys/id_hyp0.pub }
17 nrpe: { address: hyp0.vantosh.demo , zone: democenter0 , contact: mrnice }
```

Inventory - Hypervisor (1)

```
1 ---
2 distro: powerel7
3 machinearch: ppc64le
4 rootpwd: $1$f0pPKH0e$0xrX07K19DPmpcmJooede7X.
5 language: en_US
6 keyboard: us
7 timezone: "Europe/Brussels"
8 nics:
9   - { type: bridge , name: br0 , model: virtio , bootproto: dhcp , device:
10     eth0 , onboot: on }
11 hostname: hyp1.vantosh.demo
12 sshdrsakeylength: 8192
13 sshdecdsakeylength: 521
14 users:
15   - { name: supervisor , id: 1000 , ssh: yes , sshpubkey: /home/toshywoshy/.
16     keys/id_hyp1.pub }
17 nrpe: { address: hyp1.vantosh.demo , zone: democenter1 , contact: mrnotsonice }
```

Inventory - Hypervisor (2)

```
1 ---
2 distro: centos7
3 machinearch: x86_64
4 rootpwd: $1$f0pPKH0e$0xrX07K19DPmpcmJooede7X.
5 language: en_US
6 keyboard: us
7 timezone: "Europe/Brussels"
8 nics:
9   - { type: bridge , name: br0 , model: virtio , bootproto: dhcp , device:
10     eth0 , onboot: on }
11 hostname: hyp2.vantosh.demo
12 sshdrsakeylength: 8192
13 sshdecdsakeylength: 521
14 users:
15   - { name: supervisor , id: 1000 , ssh: yes , sshpubkey: /home/toshywoshy/.
16     keys/id_hyp2.pub }
17 nrpe: { address: hyp2.vantosh.demo , zone: democenter1 , contact: mrnice }
```

Inventory - Virtual Machine (0)

```
1 ---
2 hyper: hypervisor0
3 distro: powerel7
4 machinearch: ppc64
5 virtualcpus: 32
6 virtualsockets: 2
7 virtualcores: 2
8 virtualthreads: 8
9 ramsize: 2048
10 vmwaittime: 15
11 language: en_US
12 keyboard: us
13 timezone: "Europe/Brussels"
14 rootpwd: $!$!0pPKH0e$0xrX07K19DPmpcmJooede7X.
15 sshdrsakeylength: 8192
16 sshdecsakeylength: 521
17 virtualfilepath: /virtual/
18 disks:
19 - { path: /virtual/vm0-prep0.qcow2 , size: 10M }
20 - { path: /virtual/vm0-boot0.qcow2 , size: 512M }
21 - { path: /virtual/vm0-root0.qcow2 , size: 6G }
22 - { path: /virtual/vm0-swap0.qcow2 , size: 512M }
23 - { path: /virtual/vm0-swap1.qcow2 , size: 512M }
24 bootloader: bootloader --location=mbr --driveorder=vda,vdb,vdc,vdd
25 partitions:
26 - part prepboot --ondisk=vda --asprimary --fstype="prepboot" --fsoptions='
27   prepboot' --size=9
28 - part /boot --ondisk=vdb --asprimary --fstype="xfs" --fsoptions='defaults,
29   noatime,discard' --grow --size=1
30 - part pv.01 --ondisk=vdc --asprimary --grow --size=1
31 - volgroup VolGroupRoot --pesize=131072 pv.01
32 - logvol / --fstype="xfs" --fsoptions='defaults,noatime,discard' --name=
33   LogVolRoot --vgname=VolGroupRoot --grow --size=1
34 - part swap --onpart=vdd --asprimary --fstype="swap" --fsoptions='defaults,
35   discard' --grow --size=1
36 - part swap --onpart=vde --asprimary --fstype="swap" --fsoptions='defaults,
37   discard' --grow --size=1
38 hostname: vm0.vantosh.demo
39 nics:
40 - { type: bridge , name: br0 , model: virtio , bootproto: dhcp , device:
41   eth0 , onboot: on }
42 users:
43 - { name: administrator , id: 1000 , ssh: yes , sshpubkey: /home/toshywoshy
44   /.keys/id_vm0.pub }
45 nrpe: { address: vm0.vantosh.demo , zone: democenter0 , contact: mnrice }
```

Inventory - Virtual Machine (1)

```
1 ---
2 hyper: hypervisor0
3 distro: powerel7
4 machinearch: ppc64le
5 virtualcpus: 32
6 virtualsockets: 2
7 virtualcores: 2
8 virtualthreads: 8
9 ramsize: 2048
10 vmwaittime: 15
11 language: en_US
12 keyboard: us
13 timezone: "Europe/Brussels"
14 rootpwd: $!$!0pPKH0e$0xrX07K19DPmpcmJooede7X.
15 sshdrsakeylength: 8192
16 sshdecsakeylength: 521
17 virtualfilepath: /virtual/
18 disks:
19 - { path: /virtual/vm0-prep0.qcow2 , size: 10M }
20 - { path: /virtual/vm0-boot0.qcow2 , size: 512M }
21 - { path: /virtual/vm0-root0.qcow2 , size: 6G }
22 - { path: /virtual/vm0-swap0.qcow2 , size: 512M }
23 - { path: /virtual/vm0-swap1.qcow2 , size: 512M }
24 bootloader: bootloader --location=mbr --driveorder=vda,vdb,vdc,vdd
25 partitions:
26 - part prepboot --ondisk=vda --asprimary --fstype="prepboot" --fsoptions='
27   prepboot' --size=9
28 - part /boot --ondisk=vdb --asprimary --fstype="xfs" --fsoptions='defaults,
29   noatime,discard' --grow --size=1
30 - part pv.01 --ondisk=vdc --asprimary --grow --size=1
31 - volgroup VolGroupRoot --pesize=131072 pv.01
32 - logvol / --fstype="xfs" --fsoptions='defaults,noatime,discard' --name=
33   LogVolRoot --vgname=VolGroupRoot --grow --size=1
34 - part swap --onpart=vdd --asprimary --fstype="swap" --fsoptions='defaults,
35   discard' --grow --size=1
36 - part swap --onpart=vde --asprimary --fstype="swap" --fsoptions='defaults,
37   discard' --grow --size=1
38 hostname: vm0.vantosh.demo
39 nics:
40 - { type: bridge , name: br0 , model: virtio , bootproto: dhcp , device:
41   eth0 , onboot: on }
42 users:
43 - { name: administrator , id: 1000 , ssh: yes , sshpubkey: /home/toshywoshy
44   /.keys/id_vm0.pub }
45 nrpe: { address: vm0.vantosh.demo , zone: democenter1 , contact: mrnotsonice }
```

Inventory - Virtual Machine (2)

Monitoring

Automation

Inventory

Virtual
Machines

Icinga2

Icinga2 Masters

Icinga2 Clients

Conclusion

The End

```
1 ---
2 hyper: hypervisor0
3 distro: centos7
4 machinearch: x86_64
5 virtualcpus: 32
6 virtualsockets: 2
7 virtualcores: 2
8 virtualthreads: 8
9 ramsize: 2048
10 vmmwaittime: 15
11 language: en_US
12 keyboard: us
13 timezone: "Europe/Brussels"
14 rootpwd: $1$f0pPKH0e$0xrX07K19DPmpcmJooede7X.
15 sshdrsakeylength: 8192
16 sshdcdsakeylength: 521
17 virtualfilepath: /virtual/
18 disks:
19 - { path: /virtual/vm0-boot0.qcow2 , size: 512M }
20 - { path: /virtual/vm0-root0.qcow2 , size: 6G }
21 - { path: /virtual/vm0-swap0.qcow2 , size: 512M }
22 - { path: /virtual/vm0-swap1.qcow2 , size: 512M }
23 bootloader: bootloader --location=mbr --driveorder=vda,vdb,vdc,vdd
24 partitions:
25 - part /boot --ondisk=vda --asprimary --fstype="xfs" --fsoptions='defaults,
26 noatime,discard' --grow --size=1
27 - part pv.01 --ondisk=vdb --asprimary --grow --size=1
28 - volgroup VolGroupRoot --pesize=131072 pv.01
29 - logvol / --fstype="xfs" --fsoptions='defaults,noatime,discard' --name=
30 LogVolRoot --vgname=VolGroupRoot --grow --size=1
31 - part swap --onpart=vdc --asprimary --fstype="swap" --fsoptions='defaults,
32 discard' --grow --size=1
33 - part swap --onpart=vdd --asprimary --fstype="swap" --fsoptions='defaults,
34 discard' --grow --size=1
35 hostname: vm0.vantosh.demo
36 nics:
37 - { type: bridge , name: br0 , model: virtio , bootproto: dhcp , device:
38 eth0 , onboot: on }
39 users:
40 - { name: administrator , id: 1000 , ssh: yes , sshpubkey: /home/toshywoshy
41 /.keys/id_vm0.pub }
42 nrpe: { address: vm0.vantosh.demo , zone: democenter1 , contact: mrnice }
```

Playbooks - vminstaller

Toshaan
Bharvani -
VanTosh bvba

Monitoring

Automation

Inventory

Virtual
Machines

Icinga2

Icinga2 Masters

Icinga2 Clients

Conclusion

The End

```
1 - name: set machine arch variables
2   include_vars: "{{ machinearch }}/console.yml"
3   when: virtconsole is not defined
4
5 - name: set distribution variables
6   include_vars: "{{ distro }}.yml"
7
8 - name: create the qemu disk images
9   action: qemu-img dest={{ item.path }} size={{ item.size }} format="qcow2"
10     options="preallocation=failoc"
11     with_items: "{{ disks }}"
12     delegate_to: "{{ hyper }}"
13
14 ***
15 *** Debian ?
16 - name: start distribution installation - PEL7
17   include: install-vm-pel7.yml
18   when: templatetype == 'pel7'
19
20 *** EL ?
21 - name: start distribution installation - EL7
22   include: install-vm-el7.yml
23   when: templatetype == 'el7'
24
25 - name: execute minimum ram hack where required
26   include: minimum-ram-hack.yml
27   when: minram is defined
28
29
30 - name: start the vm
31   action: virt guest={{ inventory_hostname }} command=start
32   delegate_to: "{{ hyper }}"
33   ignore_errors: yes
34
35 - name: wait for vm to become available
36   local action: wait_for host={{ ansible_host }} port={{ ansible_port }} delay
37     =5 state=started timeout=300
38   become: no
39
40 - name: gather facts
41   action: setup
```


Playbooks - pel7

```
1 - name: create and copy the kickstart file of the machine
2   action: template src="pel7/kickstart.cfg" dest="{{virtualfilepath }}{{
3     inventory_hostname }}.cfg"
4   delegate_to: "{{ hyper }}"
5 - name: make install script
6   action: template src="common/install-vm.sh" dest="{{ virtualfilepath }}{{
7     inventory_hostname }}-create-vm.sh" owner=root group=root mode=770
8   delegate_to: "{{ hyper }}"
9 - name: execute install script
10  action: raw {{ virtualfilepath }}{{ inventory_hostname }}-create-vm.sh
11  delegate_to: "{{ hyper }}"
12  register: createdvm
13
14 - name: wait for vm to install
15  local_action: pause minutes={{ vmwaittime }}
```

```
1 ---
2 distroname: powerel7
3 distrotype: rhel7
4 templatetype: pel7
5 minram: 2048
6 location: http://10.59.39.200/yum/pel7/{{ machinearch }}/os/
7 repos:
8   - name: "PowerEL-base"
9     uri: "http://10.59.39.200/yum/pel7/{{ machinearch }}/base/packages/"
10    cost: 100
11 textrargs: 'text repo={{ location }} ks=file:/{{ inventory_hostname }}.cfg
             console=hvc0,tt5200 headless noshell nofirewire rd_NO_PLYMOUTH plymouth.enable
             =0 biosdevname=0 elevator=noop geoloc=0 nopass kdump_addon=off hpcmia'
```

Deploying
Icinga2 through
Ansible

Toshaan
Bharvani -
VanTosh bvba

Monitoring

Automation

Inventory

Virtual
Machines

Icinga2

Icinga2 Masters

Icinga2 Clients

Conclusion

The End

3

Icinga2

Icinga2 - Main Task

Toshaan
Bharvani -
VanTosh bvba

Monitoring

Automation

Inventory

Virtual
Machines

Icinga2

Icinga2 Masters

Icinga2 Clients

Conclusion

The End

```
1 - name: check whether icinga2 variable are set correctly
2   action: fail msg="Icinga2 variables set incorrectly, please set OK master OK
3     zone, do not set both at the same time."
4   when: icinga2_master is defined and icinga2_zone is defined
5 - name: install icinga2 master
6   include: master.yml
7   when: icinga2_master is defined and icinga2_master == true
8
9 - name: install icinga2 zonemaster
10  include: zonemaster.yml
11  when: icinga2_zonemaster is defined and icinga2_zonemaster == true
```

Icinga2 - Master

```
1  - name: install icinga2 packages
2  action: yum name={{ item }} state=latest
3  with_items:
4  - icinga2-common
5  - icinga2-bin
6  - icinga2-python
7  - icinga2
8  - icinga2-ido-pgsql
9  - icingaweb2
10 - python-carbon
11 - python-whisper
12 - graphite-web
13 - libXrender
14 - libX11
15 - nrpe
16 - nagios-plugins
17 - nagios-plugins-ping
18 - nagios-plugins-nrpe
19 - nagios-plugins-disk
20 - nagios-plugins-load
21 - nagios-plugins-users
22 - nagios-plugins-procs
23 - nagios-plugins-swap
24 - nagios-plugins-ssh
25 - nagios-plugins-tcp
26 - nagios-plugins-check_rbl
27 - nagios-plugins-by_ssh
28 - nagios-plugins-dig
29 - nagios-plugins-http
30 - nagios-plugins-smtp
31 - nagios-plugins-dns
32 - nagios-plugins-snmp
33
34 - name: copy main icinga2 configuration files
35 action: template src={{ item }} dest=/etc/icinga2/{{ item }} owner=icinga
      group=icinga mode=550 setype=etc_t
36 with_items:
37 - icinga2.conf
38 - zones.conf
39 - constants.conf
40
41 - name: copy features configuration files into icinga2 feature activation
  directory
42 action: template src={{ item }} dest=/etc/icinga2/features-available/{{ item
  }}
43 with_items:
44 - api.conf
45 - ido-pgsql.conf
46 - graphite.conf
47
48 - name: activate icinga2 features
49 action: command /usr/sbin/icinga2 feature enable {{ item }}
50 with_items:
```

Icinga2 - Zonemaster

```
1 - name: install icinga2 packages
2 action: yum name={{ item }} state=latest
3 with_items:
4   - icinga2-common
5   - icinga2-bin
6   - icinga2-python
7   - icinga2
8   - nrpe
9   - nagios-plugins
10  - nagios-plugins-ping
11  - nagios-plugins-nrpe
12  - nagios-plugins-disk
13  - nagios-plugins-load
14  - nagios-plugins-users
15  - nagios-plugins-procs
16  - nagios-plugins-swap
17  - nagios-plugins-ssh
18  - nagios-plugins-tcp
19  - nagios-plugins-check_rbl
20  - nagios-plugins-by_ssh
21  - nagios-plugins-dlg
22  - nagios-plugins-http
23  - nagios-plugins-smtp
24  - nagios-plugins-dns
25  - nagios-plugins-check_cups_queue
26  - nagios-plugins-check_sip
27  - nagios-plugins-dhcp
28  - nagios-plugins-snmp
29
30 - name: copy main icinga2 configuration files
31 action: template src={{ item }} dest=/etc/icinga2/{{ item }} owner=icinga
      group=icinga mode=650 setype=etc_t
32 with_items:
33   - icinga2.conf
34   - zones.conf
35   - constants.conf
36
37 - name: copy features configuration files into icinga2 feature activation
      directory
38 action: template src={{ item }} dest=/etc/icinga2/features-available/{{ item
      }}
39 with_items:
40   - api.conf
41
42 - name: activate icinga2 features
43 action: command /sbin/icinga2-enable-feature {{ item }}
44 with_items:
45   - api
46
47 - name: check if a configuration already exists
48 action: stat path=/etc/icinga2/conf.d/templates.conf
49 register: defaultconfig
50
```

Icinga2 - zones.conf

Toshaan
Bharvani -
VanTosh bvba

Monitoring

Automation

Inventory

Virtual

Machines

Icinga2

Icinga2 Masters

Icinga2 Clients

Conclusion

The End

```
1 /*
2 * -----
3 * VanTosh Icinga2 Zone Configuration File
4 * (c) copyleft 2014 VanTosh
5 * Author: Toshaan Bharvani <toshaan@vantosh.com>
6 * -----
7 * {{ ansible_managed }}
8 */
9
10 {% for icinga2_zone in icinga2_zones %}
11 object Endpoint "{{ icinga2_zone.nodename }}" {
12     host = "{{ icinga2_zone.hostname }}"
13     port = {{ icinga2_zone.port }}
14 }
15 object Zone "{{ icinga2_zone.zonename }}" {
16     endpoints = [ "{{ icinga2_zone.nodename }}" ]
17     {% if icinga2_zone.parent is defined %}
18     parent = "{{ icinga2_zone.parent }}"
19     {% endif %}
20 }
21 {% endfor %}
```

Icinga2 Client (0)

Toshaan
Bharvani -
VanTosh bvba

Monitoring

Automation

Inventory

Virtual

Machines

Icinga2

Icinga2 Masters

Icinga2 Clients

Conclusion

The End

```
1 - name: set variables for nrpe - RedHat
2   include_vars: redhat.yml
3   when: ansible_os_family == 'RedHat'
4
5 - name: install nrpe packages and configuration - RedHat
6   include: redhat.yml
7   when: ansible_os_family == 'RedHat'
8
9 - name: set variables for nrpe - FreeBSD
10  include_vars: freebsd.yml
11  when: ansible_os_family == 'FreeBSD'
12
13 - name: install nrpe packages and configuration - FreeBSD
14  include: freebsd.yml
15  when: ansible_os_family == 'FreeBSD'
16
17 - name: add host to monitoring system
18  action: template src=host.conf dest=/etc/icinga2/zones.d/{{ nrpe.zone }}/{{
19         inventory_hostname }}.{{ domain }}.conf owner=icinga group=icinga mode=550
20         setype=etc_t
21  delegate_to: trivikrama
22  notify: reload icinga2
```


Icinga2 Client (1)

Toshaan
Bharvani -
VanTosh bvba

Monitoring

Automation

Inventory

Virtual
Machines

Icinga2

Icinga2 Masters

Icinga2 Clients

Conclusion

The End

```
1 - name: install icinga2 packages and dependencies
2 action: yum pkg={{ item }} state=latest
3 with_items:
4   - icinga2
5   - nagios-plugins-disk
6   - nagios-plugins-load
7   - nagios-plugins-users
8   - nagios-plugins-procs
9   - nagios-plugins-swap
10  - nagios-plugins-check_iostat
11  - bc
12  - sysstat
13
14 - name: base icinga2 config
15 action: template src=nrpe.cfg dest=/etc/nagios/nrpe.cfg
16
17 - name: nrpe commands
18 action: template src={{ item }}.cfg dest=/etc/nrpe/d/
19 with_items:
20   - users
21   - disk
22   - procs
23   - load
24   - swap
25   - iostat
26
27 - name: restart icinga2 daemon
28 action: service name=icinga2 state=restarted enabled=yes
```

```
1  {% for location in ansible_mounts %}
2  {% if location.fstype != 'cifs' and location.fstype != 'nfs' and location.
   fstype != 'fuse' and location.fstype != 'iisprocfs' and location.fstype != '
   idescfs' and location.fstype != 'swap' %}
3  command[check_disk_{{ location.mount }}]={{ nrpe_checkcmddir }}check_disk -w
   15% -c 7% -p {{ location.mount }}
4  {% endif %}
5  {% endfor %}
```

Icinga2 Client - memory.cfg

Monitoring

Automation

Inventory

Virtual
Machines

Icinga2

Icinga2 Masters

Icinga2 Clients

Conclusion

The End

```
1 command[check_memory]={{ nrpe_checkcmdir }}pmp-check-unix-memory -w 50 -c 75
```

Icinga2 Client - iostat.cfg

Toshaan
Bharvani -
VanTosh bvba

Monitoring

Automation

Inventory

Virtual
Machines

Icinga2

Icinga2 Masters

Icinga2 Clients

Conclusion

The End

```
1  {% for disk in ansible_devices %}
2  {% if disk != 'sr0' %}
3  command[check_iostat_{{ disk }}]={{ nrpe_checkcmdir }}check_iostat -d {{ disk
   }} -w 750 -c 1250
4  {% endif %}
5  {% endfor %}
```

Icinga2 Client - host.conf

```
1 /*
2 * -----
3 *   VanTosh Icinga2 Host File
4 *   (c) copyleft 2014 VanTosh
5 *   Author: Toshaan Bharvani <toshaan@vantosh.com>
6 * -----
7 *   {{ ansible_managed }}
8 * -----
9 {% if ddns is defined %}
10 *   {{ ddns.hostname }}.{{ ddns.domain }}
11 {% else %}
12 *   {{ ansible_ssh_host }}
13 {% endif %}
14 *   fqdn: {{ ansible_fqdn }}
15 *   ansible: {{ inventory_hostname }}
16 * -----
17 */
18
19
20 object Host "{{ inventory_hostname }}" {
21     import "generic-host"
22     display_name = "{{ inventory_hostname }}"
23     address = "{{ nrpe.address }}"
24     {% if ansible_virtualization_role is defined %}
25     {% if hyper is not defined and ansible_virtualization_role == 'host' %}
26     {% elif hyper is defined and ansible_virtualization_role == 'guest' %}
27     //parent_host_name = "{{ hyper }}.{{ domain }}"
28     {% endif %}
29     {% endif %}
30     vars.sla = "24x7"
31 }
32
33 object Service "ssh" {
34     import "generic-service"
35     host_name = "{{ inventory_hostname }}"
36     display_name = "SSH {{ inventory_hostname }}"
37     check_command = "ssh"
38     vars.ssh_port = {{ ansible_ssh_port }}
39 }
40
41 object Service "current_users" {
42     import "generic-service"
43     host_name = "{{ inventory_hostname }}"
44     display_name = "Current Users"
45     check_command = "nrpe"
46     vars.nrpe_command = "check_users"
47     {% if nrpe.port is defined %}
48     vars.nrpe_port = {{ nrpe.port }}
49     {% else %}
50     vars.nrpe_port = 5666
51     {% endif %}
52     {% if nrpe_ssl is defined and nrpe_ssl == false %}
```

Deploying
Icinga2 through
Ansible

Toshaan
Bharvani -
VanTosh bvba

Monitoring

Automation

Inventory

Virtual
Machines

Icinga2

Icinga2 Masters

Icinga2 Clients

Conclusion

The End

4

Conclusion

Conclusion

Automation **MUST**
[RFC2119] be easy



More information

- Icinga : <http://www.icinga.org>
- Ansible : <http://www.ansible.com/>
- Ansible Roles Install VM : <https://github.com/toshywoshy/ansible-roles-vminstall>
- Ansible Roles Icinga2 : <https://github.com/toshywoshy/ansible-roles-icinga2>
- Ansible Roles NRPE : <https://github.com/toshywoshy/ansible-roles-nrpe>
- Ansible Roles NCSA-ng : <https://github.com/toshywoshy/ansible-roles-ncsang>

More information

- Icinga : <http://www.icinga.org>
- Ansible : <http://www.ansible.com/>
- Ansible Roles Install VM : <https://github.com/toshywoshy/ansible-roles-vminstall>
- Ansible Roles Icinga2 : <https://github.com/toshywoshy/ansible-roles-icinga2>
- Ansible Roles NRPE : <https://github.com/toshywoshy/ansible-roles-nrpe>
- Ansible Roles NCSA-ng : <https://github.com/toshywoshy/ansible-roles-ncsang>

More information

- Icinga : <http://www.icinga.org>
- Ansible : <http://www.ansible.com/>
- Ansible Roles Install VM : <https://github.com/toshywoshy/ansible-roles-vminstall>
- Ansible Roles Icinga2 :
<https://github.com/toshywoshy/ansible-roles-icinga2>
- Ansible Roles NRPE :
<https://github.com/toshywoshy/ansible-roles-nrpe>
- Ansible Roles NCSA-ng :
<https://github.com/toshywoshy/ansible-roles-ncsang>

More information

- Icinga : <http://www.icinga.org>
- Ansible : <http://www.ansible.com/>
- Ansible Roles Install VM : <https://github.com/toshywoshy/ansible-roles-vminstall>
- Ansible Roles Icinga2 :
<https://github.com/toshywoshy/ansible-roles-icinga2>
- Ansible Roles NRPE :
<https://github.com/toshywoshy/ansible-roles-nrpe>
- Ansible Roles NCSA-ng :
<https://github.com/toshywoshy/ansible-roles-ncsang>

More information

- Icinga : <http://www.icinga.org>
- Ansible : <http://www.ansible.com/>
- Ansible Roles Install VM : <https://github.com/toshywoshy/ansible-roles-vminstall>
- Ansible Roles Icinga2 : <https://github.com/toshywoshy/ansible-roles-icinga2>
- Ansible Roles NRPE : <https://github.com/toshywoshy/ansible-roles-nrpe>
- Ansible Roles NCSA-ng : <https://github.com/toshywoshy/ansible-roles-ncsang>

More information

- Icinga : <http://www.icinga.org>
- Ansible : <http://www.ansible.com/>
- Ansible Roles Install VM : <https://github.com/toshywoshy/ansible-roles-vminstall>
- Ansible Roles Icinga2 : <https://github.com/toshywoshy/ansible-roles-icinga2>
- Ansible Roles NRPE : <https://github.com/toshywoshy/ansible-roles-nrpe>
- Ansible Roles NCSA-ng : <https://github.com/toshywoshy/ansible-roles-ncsang>



Thank You



Toshaan Bharvani - VanTosh bvba <toshaan@vantosh.com>

<http://www.vantosh.com/>

Made with Beamer L^AT_EX
a T_EXbased Presentation program