MANAGING ICINGA WITH ANSIBLE

whoami

Matthias Döhler

- Consultant at **NETWAYS**
- **Icinga** at work (and in homelab!)
- HUGE fan of Ansible
- @Donien (GitHub)

whoami

Matthias Döhler

- Consultant at **NETWAYS**
- **Icinga** at work (and in homelab!)
- HUGE fan of Ansible
- @Donien (GitHub)

Agenda

- Setting up Icinga 2
  - Master
  - Satellite
  - Agent
- Icinga DB
- Icinga Web
- Structuring an Ansible inventory

Ansible

- Configuration management
- ~~Agentless~~ uses default tools (SSH/WinRM)
- Push-based
- Tool is written in Python
- Configuration is written in YAML

Ansible

- Do not reinvent the wheel → Use other people's code!
- Ansible roles are like functions, role variables are the arguments passed

Public code exists for a reason.
Use it!

...which brings us to...

Ansible Collection Icinga (by NETWAYS)

- Code for managing official Icinga components
- Roles for:
  - Icinga 2
  - Icinga DB
  - Icinga Web
  - Icinga for Windows

https://github.com/NETWAYS/ansible-collection-icinga

Getting started

Demo setup:

- icinga-**master1**.novalocal
- icinga-**master2**.novalocal
- icinga-**satellite**.novalocal
- icinga-**agent-linux1**.novalocal
- icinga-**agent-linux2**.novalocal
- icinga-**agent-windows**.novalocal

Ansible groups:

- masters
- satellites
- agents (master-agents, satellite-agents)
- linux
- windows

Getting started

Basic playbook:

```yaml
- name: Add Icinga repositories
  become: true
  hosts:
    - linux

  vars_files:
    - "./secrets.yml"

  vars:
    icinga_repo_subscription_username: "ansible-webinar-4qa"
    icinga_repo_subscription_password: "{{ secret_icinga_repo_subscription_password }}"

    # Necessary for plugins on EL
    icinga_repo_epel: true

  roles:
    - netways.icinga.repos
```

Getting started - Icinga installation

```
- name: Base installation Icinga 2
  become: true
  hosts:
    - masters
    - satellites

  roles:
    - netways.icinga.icinga2
```

This only installs Icinga 2 (kinda boring...)

Setting up an Icinga Master

We need:

- Constants (`NodeName` must match CN in certificate)
- Certificate
- CA
- Features

Constants

Constants end up in `/etc/icinga2/constants.conf`

```
icinga2_constants:
  NodeName: "<ideally the FQDN of the system>"
  ZoneName: "<some-zone-name>"
  TicketSalt: "<some-secret-to-create-tickets-later-on>"
```

Features

Features end up in `/etc/icinga2/features-enabled/`

```
icinga2_features:
  - name: checker
  - name: mainlog
    path: "LogDir + /icinga2.log"
```

Many features simply use the same options as listed in the Icinga 2 documentation, while others have added options specific to management approach.

Enabled features outside of `icinga2_features` are purged by default (`icinga2_purge_features: true`).

Features - API

Setting up the node as the (primary) master

```
icinga2_features:
  - ...
  - name: api
    # ca_host: none -> Defines this host as the master (holding the CA)
    ca_host: "none"
    cert_name: "<the-CN-of-the-node>"
    accept_config: true
    accept_commands: true
    endpoints:
      - name: "<CN-of-primary-master>"
        host: "<IP-of-primary-master>"
      - name: "<CN-of-secondary-master>"
        host: "<IP-of-secondary-master>"
    zones:
      - name: "master"
        endpoints:
          - "<CN-of-primary-master>"
          - "<CN-of-secondary-master>"
      - name: "<some-global-zone>"
        global: true
```

Setting up an Icinga Master

```
- name: Icinga infrastructure setup (primary master)
  become: true
  hosts:
    - icinga-master1.novalocal

  roles:
    - netways.icinga.icinga2

  # Continued ...
```

```yaml
vars:
  # Configure Icinga 2 features
  icinga2_features:
    - name: checker
    - name: mainlog
    - name: api
      # ca_host: none -> Defines this host as the master (holding the CA)
      ca_host: "none"
      cert_name: "{{ inventory_hostname }}"
      accept_config: true
      accept_commands: true
      endpoints:
        - name: "icinga-master1.novalocal"
          host: "192.168.56.101"
        - name: "icinga-master2.novalocal"
          host: "192.168.56.102"
      zones:
        - name: "master"
          endpoints:
            - "icinga-master1.novalocal"
            - "icinga-master2.novalocal"
        - name: "director-global"
          global: true
```

Setting up an Icinga Master - Results

/etc/icinga2/constants.conf:

```
const PluginDir = "/usr/lib/nagios/plugins"
const ManubulonPluginDir = "/usr/lib/nagios/plugins"
const PluginContribDir = "/usr/lib/nagios/plugins"
const NodeName = "icinga-master1.novalocal"
const ZoneName = "master"
const TicketSalt = "super-secret-ticket-salt"
```

Setting up an Icinga Master - Results

/etc/icinga2/zones.conf:

```
object Endpoint "icinga-master1.novalocal" {
  host = "192.168.56.101"
}


object Endpoint "icinga-master2.novalocal" {
  host = "192.168.56.102"
}


object Zone "director-global" {
  global = true
}


object Zone "master" {
  endpoints = [ "icinga-master1.novalocal", "icinga-master2.novalocal", ]
}
```

Setting up an Icinga Master - Results

/etc/icinga2/features-enabled/api.conf:

```
object ApiListener "api" {
  accept_config = true
  accept_commands = true
  ticket_salt = TicketSalt
}
```

/etc/icinga2/features-enabled/checker.conf:

```
object CheckerComponent "checker" {
}
```

/etc/icinga2/features-enabled/mainlog.conf:

```
object FileLogger "main-log" {
  path = LogDir + "/icinga2.log"
  severity = "information"
}
```

Setting up an Icinga Master - Results

CA:

- `/var/lib/icinga2/ca/ca.crt`
- `/var/lib/icinga2/ca/ca.key`

Node:

- `/var/lib/icinga2/certs/ca.crt`
- `/var/lib/icinga2/certs/icinga-master1.novalocal.crt`
- `/var/lib/icinga2/certs/icinga-master1.novalocal.key`

Setting up an Icinga Master (second Master)

```
   vars:
     ### Icinga 2
     # Define constants
     icinga2_constants:
       NodeName: "{{ inventory_hostname }}"
       ZoneName: "master"
-      TicketSalt: "super-secret-ticket-salt"

     # Configure Icinga 2 features
     icinga2_features:
       - name: checker
       - name: mainlog
       - name: api
-        ca_host: "none"
+        ca_host: "icinga-master1.novalocal"
         cert_name: "{{ inventory_hostname }}"
         accept_config: true
         accept_commands: true
         ...
```

Getting a Certificate

The role uses Icinga 2 **tickets** (generated from `TicketSalt` + `CN`).

Ansible **delegates** ticket creation to `icinga2_features[]api.ca_host`.

Access to the Master via Ansible is required!
Master must have `TicketSalt` constant set!

Hint:
`icinga2_features[]api.ca_host` is the parent (e.g. potentially a satellite), used to retrieve the parent's certificate.

Delegation to the Master (for ticket creation) can be enforced using
`icinga2_delegate_host: <inventory_hostname-of-the-master>`

Getting a Certificate - Without access to the Master

- Provide the `ticket_salt` manually
- Prevent actual delegation

```
vars:
  ...
  icinga2_features:
    - name: api
      ca_host: "icinga-master1.novalocal"
      cert_name: "{{ inventory_hostname }}"
      accept_config: true
      accept_commands: true
      ...

      # Provide the secret
      ticket_salt: "super-secret-ticket-salt"

  # Delegate to the current host itself
  icinga2_delegate_host: "{{ inventory_hostname }}"
```

Getting a Certificate - Without access to the Master

- Provide the `ticket_salt` manually
- Prevent actual delegation

```
vars:
  ...
  icinga2_features:
    - name: api
      ca_host: "icinga-master1.novalocal"
      cert_name: "{{ inventory_hostname }}"
      accept_config: true
      accept_commands: true
      ...

      # Provide the secret
      ticket_salt: "super-secret-ticket-salt"

  # Delegate to the current host itself
  icinga2_delegate_host: "{{ inventory_hostname }}"
```

It does not matter where the command is being run (when passing `ticket_salt`):

```
icinga2 pki ticket \
    --cn "<cert-name>" \
    --salt "<ticket-salt>"
```

Demo 1

- Add Icinga repository
- Set up primary and secondary Icinga 2 Master

```
ansible-playbook 01_repositories.yml 03_icinga2_master1.yml 04_icinga2_master2.yml
```

Icinga DB

Two roles exist:

- `icingadb` (daemon)
- `icingadb_redis` (Redis)

IcingaDB on Masters

New relevant `vars` on Master 1:

```
vars:
  icinga2_features:
    ...
    - name: icingadb
      password: "super-secret-redis-password"

  ### Icinga DB
  icingadb_redis_binds:
    - "127.0.0.1"
    - "::1"
    - "192.168.56.101"
  icingadb_redis_password: "super-secret-redis-password"

  icingadb_database_type: mysql
  icingadb_database_host: "192.168.56.101"
  icingadb_database_name: "icingadb"
  icingadb_database_user: "icingadb_user"
  icingadb_database_password: "icingadb_password"
  icingadb_database_import_schema: true
```

IcingaDB on Masters

New relevant `vars` on Master 1:

```
vars:
  icinga2_features:
    ...
    - name: icingadb
      password: "super-secret-redis-password"

  ### Icinga DB
  icingadb_redis_binds:
    - "127.0.0.1"
    - "::1"
    - "192.168.56.101"
  icingadb_redis_password: "super-secret-redis-password"

  icingadb_database_type: mysql
  icingadb_database_host: "192.168.56.101"
  icingadb_database_name: "icingadb"
  icingadb_database_user: "icingadb_user"
  icingadb_database_password: "icingadb_password"
  icingadb_database_import_schema: true
```

New relevant `vars` on Master 2:

```
vars:
  icinga2_features:
    ...
    - name: icingadb
      password: "super-secret-redis-password"

  ### Icinga DB
  icingadb_redis_binds:
    - "127.0.0.1"
    - "::1"
    - "192.168.56.102"
  icingadb_redis_password: "super-secret-redis-password"

  icingadb_database_type: mysql
  icingadb_database_host: "192.168.56.101"
  icingadb_database_name: "icingadb"
  icingadb_database_user: "icingadb_user"
  icingadb_database_password: "icingadb_password"
```

Setting up Icinga Web

We need:

- Icinga Web
- Database resources
- Initial user
- Icinga DB Web
  - Connection to both Redis instances
  - `ApiUser` in Icinga 2

Configuring Icinga 2 Objects

Define configuration paths via `icinga2_config_directories` (relative to `/etc/icinga2/`).

```
icinga2_config_directories:
  - zones.d/master/hosts/
  - zones.d/master/services/
```

Define objects via `icinga2_objects`.

```
icinga2_objects:
  - name: "<name-of-object>"
    type: "<type-of-object>"
    file: "<file-to-write-config-in>"

    # Additional mandatory and / or optional attributes based the object type
```

Example Objects

```
icinga2_config_directories:
  - zones.d/master/

icinga2_objects:
  - name: dummy-host
    type: Host
    file: zones.d/master/hosts.conf
    check_command: dummy
    check_interval: 3m
    address: 127.0.0.1

  - name: ping
    type: Service
    file: zones.d/master/services.conf
    apply: true
    check_command: ping4
    assign:
      - host.address
```

ApiUser on Masters

New relevant `vars` on Master 1:

```
vars:
  icinga2_config_directories:
    - zones.d/master/

  icinga2_objects:
    - name: "icingadb-web"
      type: ApiUser
      file: "zones.d/master/api-users.conf"
      password: "super-secret-icingadb-web-password"
      permissions:
        - "actions/*"
        - "objects/query/*"
        - "objects/modify/*"
        - "status/query"
```

YAML ain't enough

YAML is used as an abstraction.
Icinga 2 DSL features are missing.

Solution: Provide your own configuration files.

```
icinga2_custom_config:
  - name: myown_command.conf
    path: zones.d/master/myown_command.conf
```

Setting up Icinga Web - Database

```
vars:
  icingaweb2_db:
    type: mysql
    name: "icingaweb2"
    host: "192.168.56.101"
    user: "icingaweb2_user"
    password: "icingaweb2_password"

  icingaweb2_db_import_schema: true


  icingaweb2_admin_username: "admin"
  icingaweb2_admin_password: "admin"
```

Setting up Icinga Web - Resources

Any additional resources (databases and other) → `icingaweb2_resources`

```
vars:
  icingaweb2_resources:
    icingadb_db:
      type: db
      db: mysql
      host: "192.168.56.101"
      dbname: "icingadb"
      username: "icingadb_user"
      password: "icingadb_password"
      charset: "utf8"
```

Setting up Icinga Web - Generic overview

INI files are generally written using a templating macro.

YAML:

```yaml
icingaweb2_resources:
  <database-name>:
    key1: value1
    key2: value2
    key3: value3
    key4:
      - value4-1
      - value4-2
```

INI (/etc/icingaweb2/resources.ini):

```ini
[<database-name>]
key1 = "value1"
key2 = "value2"
key3 = "value3"
key4 = "value4-1, value4-2"
```

Setting up Icinga Web - Modules

```
icingaweb2_modules:
  <module-name>:
    enabled: true
    source: package
```

Additional properties depend on the given module.

```
icingaweb2_modules:
  <module-name>:
    enabled: true
    source: package
    <filename>: # refers to /etc/icingaweb2/modules/<module-name>/<filename>.ini
      <section>:
        key1: value1
    <module-specific-key>:
      ...
```

We try to support:

- official Icinga Web modules by Icinga (if packaged)
- some Icinga Web modules provided by NETWAYS (e.g. PerfDataGraphs)

Setting up Icinga Web - Module Icinga DB Web

```
icingaweb2_modules:
  icingadb:
    enabled: true
    source: package
    config:
      icingadb:
        resource: "icingadb_db"
    redis:
      redis1:
        host: "192.168.56.101"
        password: "super-secret-redis-password"
      redis2:
        ...
    commandtransports:
      instance01:
        transport: api
        host: "192.168.56.101"
        username: "icingadb-web"
        password: "super-secret-icingadb-web-password"
      instance02:
        ...
```

Demo 2

- Install and configure Icinga DB (Redis and daemon)
- Install and configure Icinga Web (and Icinga DB Web)

```
ansible-playbook 06_icingadb_master2.yml 07_icingadb_and_icingaweb_master1.yml
```

Setting up Icinga Web - Users and Permissions

- Adding users

```
icingaweb2_users:
  - username: user1
    password: some-password
  - username: user2
    password: another-password
```

- Mange roles and permissions

```
icingaweb2_roles:
  watchers:
    users:
      - "user1"
      - "another-user"
    permissions:
      - "module/icingadb"
      - "icingadb/command/downtime/*"
    refusals:
      - "icingadb/object/show-source"
    icingadb/filter/hosts: "host.name=*windows*"
```

Disabling the example configuration

Disable the default configuration (`conf.d/`) by setting `icinga2_confd: false`.

Adding a Satellite

We need:

- Add `vars` for the Satellite
- Adjust `icinga2_features[].api` on the Masters
- (also disable `conf.d/`)

Adding a Satellite

Adjusted API feature:

```
  icinga2_features:
    - name: api
      ...
      endpoints:
        - name: "icinga-master1.novalocal"
          host: "192.168.56.101"
        - name: "icinga-master2.novalocal"
          host: "192.168.56.102"
+       - name: "icinga-satellite.novalocal"
+         host: "192.168.56.103"
      zones:
        - name: "master"
          endpoints:
            - "icinga-master1.novalocal"
            - "icinga-master2.novalocal"
+       - name: "satellite"
+         parent: "master"
+         endpoints:
+           - "icinga-satellite.novalocal"
```

```
  # Disabled on all hosts
+ icinga2_confd: false

  # Master 1 only!
  # Cluster sync will provide config to other nodes
  icinga2_config_directories:
    - "zones.d/master/"
+   - "zones.d/satellite/"
```

Adding Linux Agents

Rather trivial:

```
vars:
  icinga2_constants:
    NodeName: "{{ inventory_hostname }}"
    ZoneName: "{{ inventory_hostname }}"

  icinga2_features:
    - name: mainlog
      # continued on the right →
```

```
- name: api
  ca_host: "icinga-master1.novalocal"
  cert_name: "{{ inventory_hostname }}"
  accept_config: true
  accept_commands: true
  endpoints:
    - name: "icinga-master1.novalocal"
      host: "192.168.56.101"
    - name: "icinga-master2.novalocal"
      host: "192.168.56.102"
  zones:
    - name: "master"
      endpoints:
        - "icinga-master1.novalocal"
        - "icinga-master2.novalocal"
    - name: "{{ inventory_hostname }}"
      parent: "master"
      endpoints:
        - "{{ inventory_hostname }}"
    # global zones as needed
```

Adding Windows Agents

Role `icinga2` does not work on Windows!

Own dedicated role `ifw` for Icinga for Windows!

- Installs Icinga for Windows
- Handles installation and configuration of Icinga 2
- Manages repositories and components

Adding Windows Agents with role ifw

Example:

```
- name: Add repositories
  hosts: agents:&windows

  vars:
    # Just an example
    ifw_icinga2_global_zones:
      - "windows-agents"

    ifw_icinga2_ca_host: icinga-master1.novalocal
    ifw_icinga2_parent_zone: master

    # Already the default
    #ifw_icinga2_cn: "{{ inventory_hostname }}"
    ifw_icinga2_ticket: "{{ inventory_hostname | netways.icinga.icinga2_ticket(ticketsalt='super-secret-ticket-salt') }}"

    ifw_icinga2_parents:
      - cn: icinga-master1.novalocal
        host: 192.168.56.101
      - cn: icinga-master2.novalocal
        host: 192.168.56.102

  roles:
    - netways.icinga.ifw
```

Monitoring Agents

`icinga2_objects` can be a dictionary (in hostvars)!

`icinga2_objects` within hostvars (of some arbitrary host):

```
icinga2_objects:
  <host-to-deploy-config-on>:
    - name: "{{ inventory_hostname }}"
      type: Host
      ...
```

Referenced variables must be available in inventory.

Monitoring Agents

Example hostvars for Agents attatched to the master zone:

```
_icinga2_zone: master

icinga2_objects:
  icinga-master1.novalocal:
    - name: "{{ inventory_hostname }}"
      type: Host
      file: "zones.d/{{ _icinga2_zone }}/hosts.conf"
      check_command: "hostalive"
```

Structuring an inventory

```
inventory/
├── group_vars/
│   ├── <group1>/
│   │   ├── icinga2.yml
│   │   └── meta.yml
│   ├── <group2>/
│   └── ...
└── host_vars/
    ├── <host1>/
    │   ├── icinga2.yml
    │   └── meta.yml
    ├── <host2>/
    └── ...
```

Structuring an inventory

```
inventory/
├── group_vars/
│   ├── all/
│   │   └── icinga2_objects.yml
│   ├── satellite-agents/
│   │   ├── icinga2.yml
│   │   └── meta.yml
│   └── agents/
│       └── icinga2_objects.yml
└── host_vars/
    └── icinga-agent-linux2/
        ├── icinga2.yml
        └── meta.yml
```

group_vars/satellite-agents/meta.yml:

```yaml
_icinga2_zone: "satellite"
```

group_vars/satellite-agents/icinga2.yml:

```yaml
icinga2_constants:
  ...

icinga2_features:
  ...
```

Structuring an inventory

```
inventory/
├── group_vars/
│   ├── all/
│   │   └── icinga2_objects.yml
│   ├── satellite-agents/
│   │   ├── icinga2.yml
│   │   └── meta.yml
│   └── agents/
│       └── icinga2_objects.yml
└── host_vars/
    └── icinga-agent-linux2/
        ├── icinga2.yml
        └── meta.yml
```

host_vars/icinga-agent-linux2/meta.yml:

```yaml
_icinga2_address: "192.168.56.105"
```

host_vars/icinga-agent-linux2/icinga2.yml:

```yaml
individual_icinga2_objects:
  icinga-master1.novalocal:
    - name: "Updates"
      type: Service
      host_name: "{{ inventory_hostname }}"
      file: "zones.d/{{ _icinga2_zone }}/services.conf"
      check_command: "apt"
      command_endpoint: "{{ inventory_hostname }}"
```

Structuring an inventory

```
inventory/
├── group_vars/
│   ├── all/
│   │   └── icinga2_objects.yml
│   ├── satellite-agents/
│   │   ├── icinga2.yml
│   │   └── meta.yml
│   └── agents/
│       └── icinga2_objects.yml
└── host_vars/
    └── icinga-agent-linux2/
        ├── icinga2.yml
        └── meta.yml
```

host_vars/icinga-agent-linux2/meta.yml:

```yaml
_icinga2_address: "192.168.56.105"
```

group_vars/agents/icinga2_objects.yml:

```yaml
agents_icinga2_objects:
  icinga-master1.novalocal:
    - name: "{{ inventory_hostname }}"
      type: Endpoint
      file: "zones.d/{{ _icinga2_zone }}/zones.conf"
    - name: "{{ inventory_hostname }}"
      type: Zone
      file: "zones.d/{{ _icinga2_zone }}/zones.conf"
      parent: "{{ _icinga2_zone }}"
      endpoints:
        - "{{ inventory_hostname }}"
    - name: "Icinga 2"
      type: Service
      file: "zones.d/{{ _icinga2_zone }}/zones.conf"
      host_name: "{{ inventory_hostname }}"
      check_command: "icinga"
      command_endpoint: "{{ inventory_hostname }}"
```

Structuring an inventory

```
inventory/
├── group_vars/
│   ├── all/
│   │   └── icinga2_objects.yml
│   ├── satellite-agents/
│   │   ├── icinga2.yml
│   │   └── meta.yml
│   └── agents/
│       └── icinga2_objects.yml
└── host_vars/
    └── icinga-agent-linux2/
        ├── icinga2.yml
        └── meta.yml
```

group_vars/all/icinga2_objects.yml:

```yaml
generic_icinga2_objects:
  icinga-master1.novalocal:
    - name: "{{ inventory_hostname }}"
      type: Host
      file: "zones.d/{{ _icinga2_zone }}/hosts.conf"
      address: "{{ _icinga2_address }}"
      check_command: "hostalive"

icinga2_objects: "{{
  generic_icinga2_objects |
  combine(agents_icinga2_objects | default({}), recursive=true,
list_merge='append') |
  combine(individual_icinga2_objects | default({}), recursive=true,
list_merge='append')
}}"
```

Thank You!

More Ansible!

Mangage the Icinga Director with the `telekom_mms.icinga_director` collection.

Director

```
- name: Create a host template
  telekom_mms.icinga_director.icinga_host_template:
    state: present
    url: "http://localhost/icingaweb2"
    url_username: "admin"
    url_password: "admin"
    object_name: "Default Host"
    check_command: "hostalive"
    imports:
      - ""
```

Director

```
- name: Create a host object
  telekom_mms.icinga_director.icinga_host:
    state: present
    url: "http://localhost/icingaweb2"
    url_username: "admin"
    url_password: "admin"
    object_name: "myDummyHost"
    address: "127.0.0.1"
    imports:
      - "Default Host"
```

Director

```
- name: Deploy the configuration
  telekom_mms.icinga_director.icinga_deploy:
    url: "http://localhost/icingaweb2"
    url_username: "admin"
    url_password: "admin"
    timeout: 5
```

Icinga 2 as Ansible inventory source

Use `netways.icinga.icinga` to use Icinga 2 as an inventory source.

`inventory-icinga.yml`:

```yaml
---

plugin: netways.icinga.icinga

url: https://icinga-master1.novalocal
validate_certs: false
user: icinga2-api-user
password: super-secret-icinga2-api-user-password

# Create Ansible groups based on Icinga 2 information
keyed_groups:
  - prefix: "icinga_zone"
    key: zone
```