Ansible Collection - OPNSense

AnsibleGuy

Mar 25, 2024

USAGE

Tip: Check out the repository on GitHub

Report missing/incorrect information or broken links

Tip: Check out the repository on GitHub

Report missing/incorrect information or broken links

CHAPTER

ONE

1 - INSTALLATION

1.1 Ansible

See the documentation on how to install Ansible.

1.2 Dependencies

The httpx python module is used for API communications!

```
python3 -m pip install --upgrade httpx
```

1.3 Collection

```
# stable version:
ansible-galaxy collection install ansibleguy.opnsense

# latest version:
ansible-galaxy collection install git+https://github.com/ansibleguy/collection_opnsense.
↪git

# install to specific directory for easier development
cd $PLAYBOOK_DIR
ansible-galaxy collection install git+https://github.com/ansibleguy/collection_opnsense.
↪git -p ./collections
```

Tip: Check out the repository on GitHub

Report missing/incorrect information or broken links

CHAPTER

TWO

2 - BASIC

2.1 Prerequisites

You need to create API credentials as described in the OPNSense documentation.

Menu: System - Access - Users - Edit {admin user} - Add api key

2.1.1 SSL Certificate

If you use your firewall for non-testing purposes - you should **ALWAYS USE SSL VERIFICATION** for your connections!

```
ssl_verify: true
```

To make a connection trusted you need either:

- a valid public certificate for the DNS-Name your firewall has (*LetsEncrypt/ACME*)
- an internal certificate authority that is used to create signed certificates
  - you could create such internal certificates using OPNSense. See the OPNSense documentation for self-signed certificates.
  - if you do so - it is important that the IP-address and/or DNS-Name of your firewall is included in the 'Subject Alternative Name' (*SAN*) for it to be valid

After you got a valid certificate - you need to import and activate it:

- Import: 'System - Trust - Certificates - Import'
- Make sure your DNS-Names are allowed: 'System - Settings - Administration - Alternate Hostnames'
- Activate: 'System - Settings - Administration - SSL Certificate'

If you are using an internal CA for your certificates - you have to provide its public key to the modules:

```
ssl_ca_file: '/path/to/ca.pem'
```

2.2 Basics

2.2.1 Defaults

If some parameters will be the same every time - use 'module_defaults':

```
- hosts: localhost
  gather_facts: no
  module_defaults:
    ansibleguy.opnsense.alias:
        firewall: 'opnsense.template.ansibleguy.net'
        api_credential_file: '/home/guy/.secret/opn.key'
        # if you use an internal certificate:
        #   ssl_ca_file: '/etc/ssl/certs/custom/ca.crt'
        # else you COULD (but SHOULD NOT) use:
        #   ssl_verify: false

  tasks:
    - name: Example
      ansibleguy.opnsense.alias:
        name: 'ANSIBLE_TEST1'
        content: ['1.1.1.1']
```

2.2.2 Inventory

If you are running the modules over hosts in your inventory - you would do it like that:

```
- hosts: firewalls
  connection: local  # execute modules on controller
  gather_facts: no
  tasks:
    - name: Example
      ansibleguy.opnsense.alias:
        firewall: "{{ ansible_host }}"  # or use a per-host variable to store the FQDN..
```

2.2.3 Vault

You may want to use '**ansible-vault**' to **encrypt** your 'api_secret'.

Vault-Encryption of the 'api_credential_file' is not yet supported.

```
ansible-vault encrypt_string 'YOUR_API_SECRET'
```

Then add it as variable to your inventory/config:

```
firewall:
  key: 'test'
  secret: !vault |
    $ANSIBLE_VAULT;1.1;AES256
    38303736393135366562396233353930366631396531613062366365363234363063626365656263
    663764663632343733343735333631633266313331643565 0a36643933666538376337643265373 6
```

(continues on next page)

(continued from previous page)

```
    32313332363032646436626230646461376532666366663265373663316331316664336134366338
    65313633626130393 30a3164363865333936366238376531633335643832323133636663616437 30
    3132
```

And refer to it in the module calls or module-defaults:

```
- hosts: localhost
  gather_facts: no
  module_defaults:
    ansibleguy.opnsense.route:
      firewall: '...'
      api_key: "{{ firewall.key }}"
      api_secret: "{{ firewall.secret }}"
```

To decrypt those secrets at runtime, you need to supply the 'ask-vault-pass' argument:

```
ansible-playbook -D opnsense.yml --ask-vault-pass
```

2.2.4 Running

These modules support check-mode and can show you the difference between existing and configured items:

```
# show difference
ansible-playbook opnsense.yml -D

# run in check-mode (no changes are made)
ansible-playbook opnsense.yml --check
```

Tip: Check out the repository on GitHub

Report missing/incorrect information or broken links

CHAPTER

THREE

3 - TROUBLESHOOT

If you get error messages - you should at first check if there are any errors listed.

Sometimes the error message can be pretty long, therefore you might want to copy its output into an editor of your choice and Strg+F/search for the terms `Error:` or `_content`!

Per example:

```
# OUTPUT:
fatal: [localhost]: FAILED! => {"changed": false, "msg": "API call failed | Error: {
↪'rule.interface': 'option not in list'} | Response: {'status_code': 200, '_request':
↪<Request('POST', 'https://FIREWALL/api/firewall/filter/addRule')>, '_num_bytes_
↪downloaded': 73, '_elapsed': datetime.timedelta(microseconds=189718), '_content': b'{\
↪"result\":\"failed\",\"validations\":{\"rule.interface\":\"option not in list\"}}', '_
↪text': '{\"result\":\"failed\",\"validations\":{\"rule.interface\":\"option not in␣
↪list\"}}'}"}

# ERROR:
{'rule.interface': 'option not in list'}
```

3.1 Verbose output

You can also use the `debug` argument to enable verbose output:

```
- name: Example
  ansibleguy.opnsense.alias:
    debug: true
```

When the debug-mode is enabled some useful log files are created in the directory `/tmp/ansibleguy.opnsense` (*HTTP requests made, profiling of time consumption*)

If you only want the profiling logs written, you can also use the `profiling` argument:

```
- name: Example
  ansibleguy.opnsense.alias:
    profiling: true
```

'Multi' modules also support these parameters on a per-item basis - so you don't get flooded.

3.2 Known errors

- 'option not in list' => an invalid option was provided for this parameter
- 'port only allowed for tcp/udp' => any protocol except 'TCP' or 'UDP' provided
- 'ConnectionError: Got timeout calling' => you can override the used timeout manually:

  Per example:

```
- name: Example
  ansibleguy.opnsense.alias:
    timeout: 60  # seconds
```

3.3 Known issues

- **Module-call taking long**

  Many of the modules need to 'apply' its configuration after a change happened.

  Sometimes this `reload` takes some time as the firewall needs to process some information.

  Per example:

  - URL-Table alias needs to be populated
  - Syslog needs to resolve its DNS-target (*if not able to resolve*)

  What to do about it?

  If you are calling a module **in a loop** for multiple items - it might be faster to use the *ansibleguy.opnsense.reload module* instead.

Tip: Check out the repository on GitHub

Report missing/incorrect information or broken links

CHAPTER

FOUR

4 - DEVELOP

The basic API interaction is handled in 'ansibleguy.opnsense.plugins.module_utils.base.api'.

It is a generic abstraction layer for interacting with the api - therefore all plugins should be able to function with it!

4.1 Install

You can install the collection to a specific directory for easier testing.

```
cd $PLAYBOOK_DIR
ansible-galaxy collection install git+https://github.com/ansibleguy/collection_opnsense.
↪git,<COMMIT/BRANCH> -p ./collections
```

Of course you can always place the repository at `${PLAYBOOK_DIR}/ansible_collections/ansibleguy/opnsense` so it gets picked-up by Ansible.

4.2 API Definition

To get to know the API - you will have to read into the API's XML-config that is linked in the OPNSense docs.

Per example: Alias.xml

As XML isn't the most readable format - I would recommend translating it to YAML or JSON.

Here is a nice online-tool to do so: XML-to-YAML | XML-to-JSON

4.3 Module

There are module-templates that can be copied - so you don't have to re-write the basic structure.

4.4 Abstraction

- **Module Abstraction**

- **Module Action-Abstraction**

4.5 Adding new module

Checklist:

- Create the module-file at:

  '<COLLECTION>/plugins/modules/<MODULE>.py'

  You can copy the template from '<COLLECTION>/plugins/modules/_tmpl_obj.py'

  Note: When adding module-parameters - you can copy/paste the field-description from the OPNSense web-ui! We don't have to reinvent the wheel. (*'full help' toggle*)

- For most modules you should create a sub-file to handle the actual logic so the main module-file is kept clean:

  '<COLLECTION>/plugins/module_utils/main/<MODULE>.py'

  You can copy the template from '<COLLECTION>/plugins/module_utils/main/_tmpl.py'

- Add **ansible-based tests**:

  I personally like to write tests before adding new modules and testing the modules functionality from the start (test-driven-development)

  - You can copy the template from '<COLLECTION>/tests/_tmpl.yml'

    Rename all calls to the new module.

  - Add a cleanup-task in '<COLLECTION>/tests/cleanup.yml' (set state we will expect when re-running the tests)

  - Enable the test once it runs successfully - add it to '<COLLECTION>/scripts/test.sh'

- Add **documentation**:

  - You can copy the template from '<COLLECTION>/docs/source/_tmpl/module_template.rst' and replace '<module>' and links

    reStructuredText is preferred, but markdown is also supported

    Also add important module-specific information.

  - Optional: We should also add **inline module-documentation** as standardized for Ansible

    To keep the main module file clean - the documentation should be placed in '<COLLECTION>/plugins/module_utils/inline_docs/'

    You can copy the template from '<COLLECTION>/plugins/module_utils/inline_docs/_tmpl.py'

    You can then import the documentation inside the main module file.

- Add the module to '<COLLECTION>/meta/runtime.yml'

- Add the module as option to the 'ansibleguy.opnsense.list' module:

  '<COLLECTION>/plugins/modules/list.py'
- Add the module as option to the 'ansibleguy.opnsense.reload' module:

  '<COLLECTION>/plugins/modules/reload.py'
- If you are implementing a new service:

  Add the service as option to the 'ansibleguy.opnsense.service' module:

  '<COLLECTION>/plugins/modules/service.py'

4.5.1 Testing

Run the tests like this:

```
# set these variables:
COL='name-of-new-collection'
COL_PATH="$(pwd)/../collections/ansible_collections/ansibleguy/opnsense"  # path to your␣
↪local collection
TEST_FIREWALL='192.168.0.1'  # ip of your test-firewall
TEST_API_KEY="$(pwd)/opn.txt"  # api credentials-file for your test-firewall
export ANSIBLE_DIFF_ALWAYS=yes  # enable diff-mode for debugging

bash "${COL_PATH}/scripts/test_single.sh" "$TEST_FIREWALL" "$TEST_API_KEY" "$COL_PATH" "
↪$COL" 1
```

4.6 API

One can choose to either:

- create a http-session - faster if multiple calls are needed

  p.e. *check current state => create/update/delete*

  ```
  from ansible_collections.ansibleguy.opnsense.plugins.module_utils.base.api import␣
  ↪Session
  session = Session(module=module)
  session.get(cnf={'controller': 'alias', 'command': 'addItem', 'data': {'name':
  ↪'dummy', ...}})
  session.post(cnf={'controller': 'alias', 'command': 'delItem', 'params': [uuid]})
  session.close()
  ```

  or using a context-manager:

  ```
  from ansible_collections.ansibleguy.opnsense.plugins.module_utils.base.api import␣
  ↪Session
  with Session(module=module) as session:
      session.get(cnf={'controller': 'alias', 'command': 'addItem', 'data': {'name':
  ↪'dummy', ...}})
      session.post(cnf={'controller': 'alias', 'command': 'delItem', 'params': [uuid]}
  ↪)
  ```

- use a single call - if only one is needed

  p.e. toggle a cronjob or restart a service

```
from ansible_collections.ansibleguy.opnsense.plugins.module_utils.base.api import␣
↪single_get, single_post
single_get(
    module=module,
    cnf={'controller': 'alias', 'command': 'addItem', 'data': {'name': 'dummy', ...}
↪}
)
single_post(
    module=module,
    cnf={'controller': 'alias', 'command': 'delItem', 'params': [uuid]}
)
```

For the controller/command/params/data definition - check the OPNSense API Docs!

4.7 Debugging

4.7.1 Verbose output

If you want to output something to ansible's runtime - use 'module.warn':

NOTE: This output is buffered by Ansible until the task has finished.

```
module.warn(f"{before} != {after}")
```

You can also use the `debug` argument to enable verbose output of the api requests.

```
- name: Example
  ansibleguy.opnsense.alias:
    debug: true
```

'Multi' modules also support the `debug` parameter on a per-item basis - so you don't get flooded.

When the debug-mode is enabled some useful log files are created in the directory `/tmp/ansibleguy.opnsense`

```
guy$ ls -l /tmp/ansibleguy.opnsense/
alias.log  # time consumption profiling for the executed module: https://docs.python.org/
↪3/library/profile.html
api_calls.log  # a list api calls that were executed by the debugged module
```

4.7.2 Profiling

To profile a modules time-consumption - you can use the existing profiler function:

For it to work, you need to move your modules processing into a dedicated function or object!

The profiler will wrap around this function call and analyze it.

```
from ansible_collections.ansibleguy.opnsense.plugins.module_utils.utils import profiler
from ansible_collections.ansibleguy.opnsense.plugins.module_utils.target_module import␣
```

(continues on next page)

(continued from previous page)

```
↪process

if module.params['profiling']:
    profiler(
        check=process, kwargs=dict(
            m=module, p=module.params, r=result,
        ),
    )

else:
    process(m=module, p=module.params, r=result)
```

Note: these entries can be interpreted as waiting for the responses of HTTP requests:

- 'read' of '_ssl._SSLSocket'
- 'connect' of '_socket.socket'
- 'do_handshake' of '_ssl._SSLSocket'

One can only try to lower the needed HTTP calls.

Tip: Check out the repository on GitHub

Report missing/incorrect information or broken links

CHAPTER

FIVE

1 - BASIC MODULE ARGUMENTS

5.1 All modules

Table 1: Definition

Parameter	Type	Required	Default	Aliases	Comment
firewall	string	true	-	-	IP-Address or DNS hostname of the target firewall. Must be included as 'common name' or 'subject alternative name' in the firewalls web-certificate to use 'ssl_verify=true'
api_port	integer	false	443	-	Port the target firewall uses for its web-interface
api_key	string	false, true if 'api_crede is not used	-	-	API key used to authenticate, alternative to 'api_credential_file'
api_secret	string	false, true if 'api_crede is not used	-	-	API secret used to authenticate, alternative to 'api_credential_file'. Is set as 'no_log' parameter
api_credential_f	path	false, true if 'api_key' and 'api_secre are not used	-	-	Path to the api-credential file as downloaded through the web-interface. Alternative to 'api_key' and 'api_secret'
ssl_verify	boolean	false	true	-	If the certificate of the target firewall should be validated. RECOMMENDED FOR PRODUCTION USAGE!
ssl_ca_file	path	false	-	-	If you use an internal certificate-authority to create the certificate of the target firewall, provide the path to its public key for validation
debug	boolean	false	false	-	Used to en-/disable the debug mode. All API requests and responses will be shown as Ansible warnings at runtime. Will be hidden if the tasks 'no_log' parameter is set to 'true'
profiling	boolean	false	false	-	Used to en-/disable the profiling mode. Time consumption of the module will be logged to '/tmp/ansibleguy.opnsense'
api_timeout	float	false	-	timeout	Manually override the modules default API-request timeout
api_retries	integer	false	0	connect_retrie	Number of retries on API requests, in case there is an error when ESTABLISHING the connection. This does not handle errors returned by the OP-

5.2 Modules managing multiple entries

Table 2: Definition

Parameter	Type	Required	Default	Aliases	Comment
enabled	boolean	false	true	-	En- or disable the entry
state	string	false	present	-	One of 'present', 'absent'. Add or remove the entry

Tip: Check out the repository on GitHub

Report missing/incorrect information or broken links

CHAPTER

SIX

2 - LIST

STATE: stable

TESTS: Used in multiple ones

6.1 Info

This module can list existing items/entries of a specified part of the OPNSense system.

In most cases the returned type of this module ist a list of dictionaries.

Table 1: Definition

Parameter	Type	Required	Default	Aliases	Comment
target	string	true	-	tgt, t	What part of the running config should be queried/listed. One of: 'alias', 'rule', 'route', 'cron', 'syslog', 'package', 'unbound_general', 'unbound_acl', 'unbound_host', 'unbound_domain', 'unbound_dot', 'unbound_forward', 'unbound_host_alias', 'ipsec_cert', 'shaper_pipe', 'shaper_queue', 'shaper_rule', 'monit_service', 'monit_test', 'monit_alert', 'wireguard_server', 'wireguard_peer', 'interface_vlan', 'interface_vxlan', 'source_nat', 'frr_bfd', 'frr_bgp_general', 'frr_bgp_neighbor', 'frr_bgp_prefix_list', 'frr_bgp_community_list', 'frr_bgp_as_path', 'frr_bgp_route_map', 'frr_ospf_general', 'frr_ospf_prefix_list', 'frr_ospf_interface', 'frr_ospf_route_map', 'frr_ospf_network', 'frr_ospf3_general', 'frr_ospf3_interface', 'frr_rip', 'bind_general', 'bind_blocklist', 'bind_acl', 'bind_domain', 'bind_record', 'interface_vip', 'webproxy_general', 'webproxy_cache', 'webproxy_parent', 'webproxy_traffic', 'webproxy_forward', 'webproxy_acl', 'webproxy_icap', 'webproxy_auth', 'webproxy_remote_acl', 'webproxy_pac_proxy', 'webproxy_pac_match', 'webproxy_pac_rule'

For basic parameters see: *Basic*

6.2 Examples

```yaml
- hosts: localhost
  gather_facts: no
  module_defaults:
    group/ansibleguy.opnsense.all:
      firewall: 'opnsense.template.ansibleguy.net'
      api_credential_file: '/home/guy/.secret/opn.key'

  tasks:
    - name: Pulling aliases
      ansibleguy.opnsense.list:
        target: 'alias'
      register: existing_aliases

    - name: Printing
      ansible.builtin.debug:
        var: existing_aliases.data

    - name: Pulling routes
      ansibleguy.opnsense.list:
        target: 'route'
      register: existing_routes

    - name: Printing
      ansible.builtin.debug:
        var: existing_routes.data
```

Tip: Check out the repository on GitHub

Report missing/incorrect information or broken links

CHAPTER

SEVEN

2 - RELOAD

STATE: stable

TESTS: Playbook

7.1 Info

This module can reload the running/loaded configuration for a specified part of the OPNSense system.

Most modules of this collection will automatically reload its relevant running config on change - but you can speed up mass-management of items when disabling reload on single module-calls (*reload: false*), and do it afterward using THIS module.

Alternatively you can use the *ansibleguy.opnsense.service* module with action `reload` if you like it better.

7.2 Definition

Table 1: Definition

Parameter	Type	Required	Default	Aliases	Comment
target	string	true	-	tgt, t	What part of the running config should be reloaded. One of: 'alias', 'rule', 'route', 'cron', 'unbound', 'syslog', 'ipsec', 'ipsec_legacy', 'shaper', 'monit', 'wireguard', 'interface_vlan', 'interface_vxlan', 'interface_vip', 'frr', 'webproxy', 'bind', 'ids'

For basic parameters see: *Basic*

7.3 Examples

```yaml
- hosts: localhost
  gather_facts: no
  module_defaults:
    group/ansibleguy.opnsense.all:
      firewall: 'opnsense.template.ansibleguy.net'
      api_credential_file: '/home/guy/.secret/opn.key'

  tasks:
    - name: Reloading aliases
      ansibleguy.opnsense.reload:
        target: 'alias'

    - name: Reloading routes
      ansibleguy.opnsense.reload:
        target: 'route'
```

7.3.1 Practical

```yaml
- hosts: localhost
  gather_facts: no
  module_defaults:
    group/ansibleguy.opnsense.all:
      firewall: 'opnsense.template.ansibleguy.net'
      api_credential_file: '/home/guy/.secret/opn.key'

  tasks:
    - name: Adding routes
      ansibleguy.opnsense.route:
        network: "{{ item.nw }}"
        gateway: "{{ item.gw }}"
        reload: false
      loop:
        - {nw: '10.206.0.0/16', gw: 'VPN_GW'}
        - {nw: '10.67.0.0/16', gw: 'VPN2_GW'}

    - name: Adding DNS overrides
      ansibleguy.opnsense.unbound_host:
        hostname: "{{ item.host }}"
        domain: 'opnsense.template.ansibleguy.net'
        value: "{{ item.value }}"
        reload: false
      loop:
        - {host: 'a', value: '192.168.0.1'}
        - {host: 'd', value: '192.168.0.5'}

    - name: Reloading
      ansibleguy.opnsense.reload:
        target: "{{ item }}"
      loop:
```

(continues on next page)

(continued from previous page)

```
        - 'route'
        - 'unbound'
```

Tip: Check out the repository on GitHub

Report missing/incorrect information or broken links

CHAPTER

EIGHT

ALIAS

STATE: stable

TESTS: Playbook

API Docs: Core - Firewall

Service Docs: Aliases

This module allows you to manage single aliases.

8.1 Info

For more detailed information on what alias types are supported - see the documentation.

To use GeoIP alias types - you need to configure a source for it first. See: documentation

8.1.1 Mass-Manage

If you want to mass-manage aliases - take a look at the *ansibleguy.opnsense.alias_multi* module. It is scales better for that use-case!

8.2 Definition

Table 1: Definition

Parameter	Type	Required	Default	Aliases	Comment
name	string	true	-	n	Unique name of the alias
description	string	false	-	desc	Description for the alias
content	list	false for state changes, else true	-	cont, c	Values the alias should hold
type	string	false	'host'	t	Type of value the alias should hold. One of: 'host', 'network', 'port', 'url', 'urltable', 'geoip', 'networkgroup', 'mac', 'dynipv6host', 'internal', 'external'
update-freq_days	float	false	7.0	-	Needed only for the alias-type 'urltable'. Interval to update its content. Per example: 0.5 for every 12 hours
reload	boolean	false	false	-	If the running config should be reloaded on change - this may take some time. For mass-managing items you might want to reload it 'manually' after all changes are done => using the *ansibleguy.opnsense.reload* module.

For basic parameters see: *Basic*

8.3 Examples

```
- hosts: localhost
  gather_facts: no
  module_defaults:
    group/ansibleguy.opnsense.all:
      firewall: 'opnsense.template.ansibleguy.net'
      api_credential_file: '/home/guy/.secret/opn.key'

    ansibleguy.opnsense.list:
      target: 'alias'

    ansibleguy.opnsense.reload:
      target: 'alias'

  tasks:
    - name: Example
      ansibleguy.opnsense.alias:
        name: 'ANSIBLE_TEST1'
        description: 'just a test'
        content: '1.1.1.1'
        state: 'present'
        # type: 'host'  # default
```

(continues on next page)

(continued from previous page)

```
      # updatefreq_days: 3  # used only for type 'urltable'
      # ssl_ca_file: '/etc/ssl/certs/custom/ca.crt'
      # ssl_verify: False
      # api_key: !vault ...  # alternative to 'api_credential_file'
      # api_secret: !vault ...
      # debug: false

  - name: Adding
    ansibleguy.opnsense.alias:
      name: 'ANSIBLE_TEST2'
      content: '192.168.1.1'

  - name: Listing
    ansibleguy.opnsense.list:
    #  target: 'alias'
    register: existing_entries

  - name: Printing aliases
    ansible.builtin.debug:
      var: existing_entries.data  # type = list of dicts

  - name: Changing
    ansibleguy.opnsense.alias:
      name: 'ANSIBLE_TEST2'
      content: ['192.168.1.5', '192.168.10.4']

  - name: Removing
    ansibleguy.opnsense.alias:
      name: 'ANSIBLE_TEST3'
      state: 'absent'

  - name: Disabling
    ansibleguy.opnsense.alias:
      name: 'ANSIBLE_TEST2'
      enabled: false

  - name: Adding ports
    ansibleguy.opnsense.alias:
      name: 'ANSIBLE_TEST3'
      type: 'port'
      content: [80, 443, '9000:9002']

  - name: Adding url-table
    ansibleguy.opnsense.alias:
      name: 'ANSIBLE_TEST4'
      type: 'urltable'
      updatefreq_days: 2.6
      content: 'https://www.spamhaus.org/drop/drop.txt'

  - name: Adding dns-names
    ansibleguy.opnsense.alias:
      name: 'ANSIBLE_TEST5'
```

(continues on next page)

(continued from previous page)

```
      content:
        - 'acme-v02.api.letsencrypt.org'
        - 'staging-v02.api.letsencrypt.org'
        - 'r3.o.lencr.org'

  - name: Adding network
    ansibleguy.opnsense.alias:
      name: 'ANSIBLE_TEST6'
      type: 'network'
      content: '192.168.0.0/24'

  - name: Adding geoips regions
    ansibleguy.opnsense.alias:
      name: 'ANSIBLE_TEST_1_2_GEOIP2'
      type: 'geoip'
      content: ['AT', 'DE', 'CH']

  - name: Reloading running config
    ansibleguy.opnsense.reload:
    #  target: 'alias'
```

Tip: Check out the repository on GitHub

Report missing/incorrect information or broken links

CHAPTER

NINE

ALIAS - MASS MANAGEMENT

STATE: stable

TESTS: Playbook

API Docs: Core - Firewall

Service Docs: Aliases

This module allows you to manage multiple aliases.

It is faster than the 'alias' module as it reduces the needed api/http calls.

9.1 Info

For more detailed information on what alias types are supported - see the OPNSense documentation.

9.2 Multi

- Each alias has the attributes as defined in the *ansibleguy.opnsense.alias* module
- To ensure valid configuration - the attributes of each alias get verified using ansible's built-in verifier

9.3 Definition

For basic parameters see: *Basic*

9.3.1 ansibleguy.opnsense.alias_multi

Table 1: Definition

Parameter	Type	Required	Default	Aliases	Comment
aliases	dictionary	true	-	-	Dictionary of aliases to manage/configure
fail_verification	boolean	false	false	fail_verify	Fail module if single alias fails the verification
fail_processing	boolean	false	true	fail_proc	Fail module if single alias fails to be processed
state	string	false	'present'	-	Options: 'present', 'absent'
enabled	boolean	false	true	-	If all aliases should be en- or disabled
output_info	boolean	false	false	info	Enable to show some information on processing at runtime. Will be hidden if the tasks 'no_log' parameter is set to 'true'.
reload	boolean	false	true	-	If the running config should be reloaded on change - this may take some time. For mass-managing items you might want to reload it 'manually' after all changes are done => using the *ansibleguy.opnsense.reload* module.

9.3.2 ansibleguy.opnsense.alias_purge

Table 2: Definition

Parameter	Type	Required	Default	Aliases	Comment
aliases	dictionary	true	-	-	Configured aliases - to exclude from purging
output_info	boolean	false	false	info	Enable to show some information on processing at runtime. Will be hidden if the tasks 'no_log' parameter is set to 'true'.
action	string	false	'delete'	-	What to do with the matched aliases. One of: 'disable', 'delete'
filters	dictionary	false	-	-	Field-value pairs to filter on - per example: {type: port} - to only purge aliases of type 'port'
filter_invert	boolean	false	false	-	If true - it will purge all but the filtered ones
filter_partial	boolean	false	false	-	If true - the filter will also match if it is just a partial value-match
force_all	boolean	false	false	-	If set to true and neither aliases, nor filters are provided - all non-builtin aliases will be purged
fail_all	boolean	false	false	fail	Fail module if single alias fails to be purged
reload	boolean	false	true	-	If the running config should be reloaded on change - this may take some time. For mass-managing items you might want to reload it 'manually' after all changes are done => using the *ansibleguy.opnsense.reload* module.

9.4 Examples

```
- hosts: localhost
  gather_facts: no
  module_defaults:
    group/ansibleguy.opnsense.all:
      firewall: 'opnsense.template.ansibleguy.net'
      api_credential_file: '/home/guy/.secret/opn.key'

    ansibleguy.opnsense.list:
      target: 'alias'

  tasks:
    - name: Creation
      ansibleguy.opnsense.alias_multi:
        fail_verification: true  # default = false; Fail module if single alias fails␣
↪the verification
        aliases:
          test1:
            content: '1.1.1.1'
          test2:
            content: ['1.1.1.1', '1.1.1.2']
            description: 'to be deleted'
          test3:
            type: 'network'
            content: '10.0.0.0/24'
            description: 'to be disabled'
        # fail_processing: false
        # output_info: false

    - name: Changes
      ansibleguy.opnsense.alias_multi:
        aliases:
          test1:
            content: ['1.1.1.3']
          test2:
            state: 'absent'
          test3:
            enabled: false

    - name: Change state of all
      ansibleguy.opnsense.alias_multi:
        aliases:
          test1:
          test3:
        state: 'absent'
        # enabled: true

    - name: Listing
      ansibleguy.opnsense.list:
      #  target: 'alias'
      register: existing_entries
```

(continues on next page)

(continued from previous page)

```
- name: Printing aliases
  ansible.builtin.debug:
    var: existing_entries.data

- name: Purging all non-configured aliases
  ansibleguy.opnsense.alias_purge:
    aliases: {...}
    # action: 'disable'  # default = remove

- name: Purging all port aliases
  ansibleguy.opnsense.alias_purge:
    filters:  # filtering aliases to purge by alias-parameters
      type: 'port'
    # filter_invert: true  # purge all non-port aliases
```

Tip: Check out the repository on GitHub

Report missing/incorrect information or broken links

CHAPTER

TEN

DNS - BIND

STATE: stable

TESTS: bind_general | bind_blocklist | bind_acl | bind_domain | bind_record | bind_record_multi

API Docs: Plugins - Bind

Service Docs: Bind

10.1 Sponsoring

Thanks to @telmich for sponsoring the development of these modules!

10.2 Prerequisites

You need to install the BIND plugin:

```
os-bind
```

You can also install it using the *ansibleguy.opnsense.package* module.

10.3 Definition

For basic parameters see: *Basic*

10.3.1 ansibleguy.opnsense.bind_general

Table 1: Definition

Parameter	Type	Required	Default	Aliases	Comment
enabled	boolean	false	true	-	En- or disable the BIND service
ipv6	boolean	false	false	-	En- or disable IPv6
response_policy_z	bool	false	false	rpz	En- or disable response policy zones
port	integer	false	53530	p	Port the BIND service should listen on. Integer between 1 and 65535
listen_ipv4	list	false	['127.0.0.	listen_v4, listen	IPv4 addresses the service should listen on
listen_ipv6	list	false	['::1']	listen_v6	IPv6 addresses the service should listen on
query_source_ip	string	false	-	query_ipv query_v4	Specify the IPv4 address used as a source for outbound queries
query_source_ip	string	false	-	query_ipv query_v6	Specify the IPv6 address used as a source for outbound queries
transfer_source_ipv4	string	false	-	transfer_ipv4, transfer_v4	Specify the IPv4 address used as a source for zone transfers
transfer_source_ipv6	string	false	-	transfer_ipv6, transfer_v6	Specify the IPv6 address used as a source for zone transfers
forwarders	list	false	-	fwd	Set one or more hosts to send your DNS queries if the request is unknown
filter_aaaa_v4	bool	false	false	-	En- or disable to filter AAAA records on IPv4 Clients
filter_aaaa_v6	bool	false	false	-	En- or disable to filter AAAA records on IPv6 Clients
log_size	integer	false	5	max_log_	Maximum log file size in MB
cache_size	integer	false	50	max_cach cache_per max_cach	How much memory in percent the cache can use from the system
recursion_acl	list	false	-	recursion	Name of an existing ACL - where you allow which clients can resolve via this service. Usually use your local LAN
transfer_acl	list	false	-	allow_transf transfer	Name of an existing ACL - where you allow which server can retrieve zones
query_acl	list	false	-	allow_query	Name of an existing ACL - where you allow which client are allowed to query this zone
dnssec_validatio	string	false	-	dnssec	One of: 'auto', 'no'. Set to 'auto' to use the static trust anchor configuration by the system
hide_hostname	bool	false	false	-	If the system hostname should be hidden for DNS queries
hide_version	bool	false	true	-	If the local BIND version should be hidden in DNS queries
prefetch	bool	false	true	-	If it should prefetch domains
ratelimit	bool	false	false	-	If DNS replies should be rate limited
rate-limit_count	integer	false	-	-	Set how many replies per second are allowed
rate-limit_except	list	false	['127.0.0. '::1']	-	Except a list of IPs from rate-limiting
reload	boolean	false	true		If the running config should be reloaded on change

10.3.2 ansibleguy.opnsense.bind_blocklist

Table 2: Definition

Parameter	Type	Required	Default	Aliases	Comment
enabled	boolean	false	true	-	En- or disable Blocklists
block	list	false	-	lists	Blocklist's you want to enable. At least one of: 'AdAway List', 'AdGuard List', 'Blocklist.site Ads', 'Blocklist.site Fraud', 'Blocklist.site Phishing', 'Cameleon List', 'Easy List', 'EMD Malicious Domains List', 'Easyprivacy List', 'hpHosts Ads', 'hpHosts FSA', 'hpHosts PSH', 'hpHosts PUP', 'Malwaredomain List', 'NoCoin List', 'PornTop1M List', 'Ransomware Tracker List', 'Simple Ad List', 'Simple Tracker List', 'Steven Black List', 'WindowsSpyBlocker (spy)', 'WindowsSpyBlocker (update)', 'WindowsSpyBlocker (extra)', 'YoYo List'
exclude	list	false	-	safe_list	Domains to exclude from the filter
safe_google	boolean	false	-	safe_searc	-
safe_duckduckg	boolean	false	-	safe_searc	-
safe_youtube	boolean	false	-	safe_searc	-
safe_bing	boolean	false	-	safe_searc	-
reload	boolean	false	true	-	If the running config should be reloaded on change - this may take some time. For mass-managing items you might want to reload it 'manually' after all changes are done => using the *ansibleguy.opnsense.reload* module.

10.3.3 ansibleguy.opnsense.bind_acl

Table 3: Definition

Parameter	Type	Required	Default	Aliases	Comment
name	string	true	-	-	Unique name of the ACL. Some restrictions apply! Length < 32 and neither of: 'any', 'localhost', 'localnets', 'none'
networks	list	false for state changes, else true	-	nets	List of networks to add to the ACL
reload	boolean	false	true	-	If the running config should be reloaded on change - this may take some time. For mass-managing items you might want to reload it 'manually' after all changes are done => using the *ansibleguy.opnsense.reload* module.

10.3.4 ansibleguy.opnsense.bind_domain

Table 4: Definition

Parameter	Type	Required	Default	Aliases	Comment
name	string	true	-	domain_name, domain	Domain name of the zone. Both forward and reverse zones may be specified, i.e. example.com or 0.168.192.in-addr.arpa.
mode	string	false	'primary'	-	Zone operation mode. One of: 'primary', 'secondary'
primary	list	false	-	primary_ip, master, master_ip	Set the IP address of primary server when using secondary mode
transfer_key_algo	string	false	-	-	Set the authentication algorithm for the TSIG key used to transfer domain data from the primary server. One of: 'hmac-sha512', 'hmac-sha384', 'hmac-sha256', 'hmac-sha224', 'hmac-sha1', 'hmac-md5'
transfer_key_name	string	false	-	-	The name of the TSIG key, which must match the value on the primary server
transfer_key	string	false	-	-	The base64-encoded TSIG key
allow_notify	list	false	-	allow_notify, allow_notify	A list of allowed IP addresses to receive notifies from
transfer_acl	list	false	-	allow_transf	Name of an existing ACL - where you allow which server can retrieve zones
query_acl	list	false	-	allow_query	Name of an existing ACL - where you allow which client are allowed to query this zone
ttl	integer	false	86400	-	The general Time To Live for this zone. Between 60 and 86400
refresh	integer	false	21600	-	The time in seconds after which name servers should refresh the zone information. Between 60 and 86400
retry	integer	false	3600	-	The time in seconds after which name servers should retry requests if the primary does not respond. Between 60 and 86400
expire	integer	false	3542400	-	The time in seconds after which name servers should stop answering requests if the primary does not respond. Between 60 and 10000000
negative	integer	false	3600	-	The time in seconds after which an entry for a non-existent record should expire from cache. Between 60 and 86400
admin_mail	string	false	'mail.opns	-	The mail address of zone admin. A @-sign will automatically be replaced with a dot in the zone data
server	string	false	'opnsense.loc	dns_server	The DNS server hosting this file. This should usually be the FQDN of your firewall where the BIND plugin is installed
reload	boolean	false	true	-	If the running config should be reloaded on change - this may take some time. For mass-managing items you might want to reload it 'manually' after all changes are done -> using the *ansibleguy.opnsense.reload* module.

Note:

A domain can only be removed if no records linked to it exist.

Else it will leave the configuration in a state where you'll have to edit the backup-xml and restore it to remove those records as they will not show in the Web-UI and cannot be addressed using the module.

It seems the plugin lacks validation in that case.

10.3.5 ansibleguy.opnsense.bind_record

Table 5: Definition

Parameter	Type	Required	Default	Aliases	Comment
match_fields	list	false	['domain', 'name', 'type']	-	Fields that are used to match configured records with the running config - if any of those fields are changed, the module will think it's a new record. At least one of: 'domain', 'name', 'type', 'value'
name	string	true	-	record	Name of the record
domain	string	true	-	domain_nam	Existing domain/zone for the record
type	string	false	'A'	-	Type of the record. One of: 'A', 'AAAA', 'CAA', 'CNAME', 'DNSKEY', 'DS', 'MX', 'NS', 'PTR', 'RRSIG', 'SRV', 'TLSA', 'TXT'
value	false for state changes, else true	false	''	-	Value the record should hold
round_robin	boolean	false	false	-	If multiple records with the same domain/name/type combination exist - the module will only execute 'state=absent' if set to 'false'. To create multiple ones set this to 'true'. Records will only be created, NOT UPDATED! (no matching is done)
reload	boolean	false	true	-	If the running config should be reloaded on change - this may take some time. For mass-managing items you might want to reload it 'manually' after all changes are done => using the *ansibleguy.opnsense.reload* module.

10.3.6 ansibleguy.opnsense.bind_record_multi

Table 6: Definition

Parameter	Type	Required	Default	Aliases	Comment
records	dictionary	true	-	record	Records to process. Format of the dictionary: {'domain1': [{'name': 'record1', 'value': '192.168.0.1'}, {'name': 'record2', 'type': 'TXT', 'value': 'random'}]} (*dictionary of domains with a list of record-dictionaries*)
match_fields	list	false	['domain', 'name', 'type']	-	Fields that are used to match configured records with the running config - if any of those fields are changed, the module will think it's a new record. At least one of: 'domain', 'name', 'type', 'value'
fail_verification	boolean	false	false	fail_verify	Fail module if single record fails the verification
fail_processing	boolean	false	true	fail_proc	Fail module if single record fails to be processed
state	string	false	'present'	-	Options: 'present', 'absent'
enabled	boolean	false	true	-	If all records should be en- or disabled
output_info	boolean	false	false	info	Enable to show some information on processing at runtime. Will be hidden if the tasks 'no_log' parameter is set to 'true'.
reload	boolean	false	true	-	If the running config should be reloaded on change - this may take some time. For mass-managing items you might want to reload it 'manually' after all changes are done => using the *ansibleguy.opnsense.reload* module.

10.4 Info

10.4.1 Mass-Manage

If you want to mass-manage DNS records - use the ansibleguy.opnsense.bind_record_multi module. It scales better for that use-case!

For other modules:

- If you are mass-managing DNS records or using DNS-Blocklists - you might want to disable `reload: false` on single module-calls!
- This takes a long time, as the service gets reloaded every time!
- You might want to reload it 'manually' after all changes are done => using the *ansibleguy.opnsense.reload* module.

10.4.2 Round-Robin

The management of round-robin records is a harder to manage by the module as a single record cannot be identified!

Therefor the 'bind_record' module has an 'round_robin' argument.

Default mode

With it set to 'false' (*default*) only one record with the exact combination of domain/type/name will be accepted.

Else the module will throw an error!

In this mode the management (*create/update/delete*) of those single records is completely logical.

round-robin mode

If you need to set it to 'true' - its usage changes a little.

Updating the value of a single record within a round-robin is not possible!

Deletion

You could delete a single one of the records by setting the 'match_fields' argument to ['domain', 'name', 'type', 'value'] and therefor matching its value.

But the default behaviour is that you can only delete all of them at once.

If a change is needed, you will have to run the module using 'state=absent' first and then re-create all the records.

10.5 Examples

10.5.1 ansibleguy.opnsense.bind_general

```
- hosts: localhost
  gather_facts: no
  module_defaults:
    group/ansibleguy.opnsense.all:
      firewall: 'opnsense.template.ansibleguy.net'
      api_credential_file: '/home/guy/.secret/opn.key'

    ansibleguy.opnsense.list:
      target: 'bind_general'

  tasks:
    - name: Example
      ansibleguy.opnsense.bind_general:
        # enabled: true
        # ipv6: true
        # response_policy_zones: true
        # port: 53530
        # listen_ipv4: ['127.0.0.1']
        # listen_ipv6: ['::1']
        # query_source_ipv4: ''
```

(continues on next page)

(continued from previous page)

```
      # transfer_source_ipv4: ''
      # query_source_ipv6: ''
      # transfer_source_ipv6: ''
      # forwarders: []
      # filter_aaaa_v4: false
      # filter_aaaa_v6: false
      # filter_aaaa_acl: []
      # log_size: 5
      # cache_size: 50
      # recursion_acl: []
      # transfer_acl: []
      # query_acl: []
      # dnssec_validation: 'no'
      # hide_hostname: false
      # hide_version: true
      # prefetch: true
      # ratelimit: true
      # ratelimit_count:
      # ratelimit_except: ['127.0.0.1', '::1']
      # reload: true

  - name: Configuring BIND
    ansibleguy.opnsense.bind_general:
      enabled: true
      listen_ipv4: ['127.0.0.1', '192.168.0.1']
      query_source_ipv4: '192.168.0.1'
      transfer_source_ipv4: '192.168.0.1'
      filter_aaaa_v4: false
      filter_aaaa_acl: ['192.168.0.2', '192.168.0.4']
      dnssec_validation: 'no'
      hide_hostname: true
      hide_version: true
      ratelimit: true
      prefetch: false
      ratelimit_count: 50
      log_size: 10
      response_policy_zones: false
      ipv6: false

  - name: Pulling settings
    ansibleguy.opnsense.list:
    #  target: 'bind_general'
    register: existing_entries

  - name: Printing settings
    ansible.builtin.debug:
      var: existing_entries.data
```

10.5.2 ansibleguy.opnsense.bind_blocklist

```
- hosts: localhost
  gather_facts: no
  module_defaults:
    group/ansibleguy.opnsense.all:
      firewall: 'opnsense.template.ansibleguy.net'
      api_credential_file: '/home/guy/.secret/opn.key'

    ansibleguy.opnsense.list:
      target: 'bind_blocklist'

  tasks:
    - name: Example
      ansibleguy.opnsense.bind_blocklist:
        # enabled: true
        # block: []
        # exclude: []
        # safe_google: false
        # safe_duckduckgo: false
        # safe_youtube: false
        # safe_bing: false
        # reload: true

    - name: Configuring blocklists
      ansibleguy.opnsense.bind_blocklist:
        block: ['Steven Black List', 'NoCoin List', 'Blocklist.site Phishing', 'AdGuard␣
↪List']
        exclude: ['test.ansibleguy.net', 'ansibleguy.net']
        safe_google: true
        safe_youtube: true

    - name: Disabling blocklists
      ansibleguy.opnsense.bind_blocklist:
        enabled: false
        block: ['Steven Black List', 'NoCoin List', 'Blocklist.site Phishing', 'AdGuard␣
↪List']
        exclude: ['test.ansibleguy.net', 'ansibleguy.net']
        safe_google: true
        safe_youtube: true

    - name: Listing
      ansibleguy.opnsense.list:
      #  target: 'bind_blocklist'
      register: existing_entries

    - name: Printing blocklists
      ansible.builtin.debug:
        var: existing_entries.data
```

10.5.3 ansibleguy.opnsense.bind_acl

```yaml
- hosts: localhost
  gather_facts: no
  module_defaults:
    group/ansibleguy.opnsense.all:
      firewall: 'opnsense.template.ansibleguy.net'
      api_credential_file: '/home/guy/.secret/opn.key'

    ansibleguy.opnsense.list:
      target: 'bind_acl'

  tasks:
    - name: Example
      ansibleguy.opnsense.bind_acl:
        name: 'example'
        # enabled: true
        # networks: []
        # reload: true

    - name: Adding
      ansibleguy.opnsense.bind_acl:
        name: 'test1'
        networks: ['192.168.0.0/24']

    - name: Changing
      ansibleguy.opnsense.bind_acl:
        name: 'test1'
        networks: ['192.168.1.0/25']

    - name: Disabling
      ansibleguy.opnsense.bind_acl:
        name: 'test1'
        networks: ['192.168.1.0/25']
        enabled: false

    - name: Listing
      ansibleguy.opnsense.list:
        # target: 'bind_acl'
      register: existing_entries

    - name: Printing acls
      ansible.builtin.debug:
        var: existing_entries.data

    - name: Removing
      ansibleguy.opnsense.bind_acl:
        name: 'test1'
        state: 'absent'
```

10.5.4 ansibleguy.opnsense.bind_domain

```yaml
- hosts: localhost
  gather_facts: no
  module_defaults:
    group/ansibleguy.opnsense.all:
      firewall: 'opnsense.template.ansibleguy.net'
      api_credential_file: '/home/guy/.secret/opn.key'

    ansibleguy.opnsense.list:
      target: 'bind_domain'

  tasks:
    - name: Example
      ansibleguy.opnsense.bind_domain:
        name: 'example.ansibleguy'
        # enabled: true
        # mode: 'primary'
        # transfer_key_algo: ''
        # transfer_key_name: ''
        # transfer_key: ''
        # allow_notify: []
        # transfer_acl: []
        # query_acl: []
        # ttl: 86400
        # refresh: 21600
        # retry: 3600
        # expire: 3542400
        # negative: 3600
        # admin_mail: 'mail.opnsense.localdomain'
        # server: 'opnsense.localdomain'
        # reload: true

    - name: Adding
      ansibleguy.opnsense.bind_domain:
        name: 'test1.ansibleguy'
        transfer_key_algo: 'hmac-sha512'
        transfer_key_name: 'test'
        transfer_key: "{{ 'randomsecret' | b64encode }}"
        ttl: 14400
        retry: 1800

    - name: Changing
      ansibleguy.opnsense.bind_domain:
        name: 'test1.ansibleguy'
        transfer_key_algo: 'hmac-sha512'
        transfer_key_name: 'test'
        transfer_key: "{{ 'randomsecretNEW' | b64encode }}"
        ttl: 14400
        retry: 1800
        transfer_acl: 'test1_acl'

    - name: Disabling
```

(continues on next page)

(continued from previous page)

```
      ansibleguy.opnsense.bind_domain:
        name: 'test1.ansibleguy'
        enabled: false

    - name: Listing
      ansibleguy.opnsense.list:
        # target: 'bind_domain'
      register: existing_entries

    - name: Printing domains
      ansible.builtin.debug:
        var: existing_entries.data

    - name: Removing
      ansibleguy.opnsense.bind_domain:
        name: 'test1.ansibleguy'
        state: 'absent'
```

10.5.5 ansibleguy.opnsense.bind_record

```
- hosts: localhost
  gather_facts: no
  module_defaults:
    group/ansibleguy.opnsense.all:
      firewall: 'opnsense.template.ansibleguy.net'
      api_credential_file: '/home/guy/.secret/opn.key'

    ansibleguy.opnsense.list:
      target: 'bind_record'

  tasks:
    - name: Example
      ansibleguy.opnsense.bind_record:
        domain: 'template.ansibleguy'
        name: 'example'
        # value: ''
        # type: 'A'
        # round_robin: false
        # enabled: true
        # match_fields: ['domain', 'name', 'type']
        # reload: true

    - name: Adding
      ansibleguy.opnsense.bind_record:
        domain: 'template.ansibleguy'
        name: 'test1'
        value: '192.168.0.1'

    - name: Changing
      ansibleguy.opnsense.bind_record:
```

(continues on next page)

(continued from previous page)

```
        domain: 'template.ansibleguy'
        name: 'test1'
        value: '192.168.1.1'

    - name: Disabling
      ansibleguy.opnsense.bind_record:
        domain: 'template.ansibleguy'
        name: 'test1'
        value: '192.168.1.1'
        enabled: false

    - name: Listing
      ansibleguy.opnsense.list:
        # target: 'bind_record'
      register: existing_entries

    - name: Printing records
      ansible.builtin.debug:
        var: existing_entries.data

    - name: Removing
      ansibleguy.opnsense.bind_record:
        domain: 'template.ansibleguy'
        name: 'test1'
        state: 'absent'
```

10.5.6 ansibleguy.opnsense.bind_record_multi

```
- hosts: localhost
  gather_facts: no
  module_defaults:
    group/ansibleguy.opnsense.all:
      firewall: 'opnsense.template.ansibleguy.net'
      api_credential_file: '/home/guy/.secret/opn.key'

  tasks:
    - name: Example
      ansibleguy.opnsense.bind_record_multi:
        records:
          'template.ansibleguy':  # domain
            - name: 'example'
              value: '192.168.1.1'
        # fail_verification: false
        # fail_processing: false
        # enabled: true
        # match_fields: ['domain', 'name', 'type']
        # reload: true
        # output_info: false

    - name: Adding
```

(continues on next page)

(continued from previous page)

```
    ansibleguy.opnsense.bind_record_multi:
      records:
        'template.ansibleguy':
          - name: 'test1'
            value: '192.168.1.1'
          - name: 'test1'
            type: 'TXT'
            value: 'random'
          - name: 'test2'
            value: '192.168.2.1'
          - name: 'test3'
            value: '192.168.3.1'
          - name: 'test4'
            type: 'CNAME'
            value: 'test1.test3.ansibleguy'

  - name: Changing
    ansibleguy.opnsense.bind_record_multi:
      records:
        'template.ansibleguy':
          - name: 'test1'
            value: '192.168.1.2'
          - name: 'test1'
            type: 'TXT'
            value: 'random_new'
          - name: 'test2'
            value: '192.168.2.1'
            enabled: false
          - name: 'test3'
            state: 'absent'
          - name: 'test4'
            type: 'CNAME'
            value: 'test2.test3.ansibleguy'

  - name: Disabling all
    ansibleguy.opnsense.bind_record_multi:
      records:
        'template.ansibleguy':
          - name: 'test1'
            value: '192.168.1.2'
          - name: 'test1'
            type: 'TXT'
            value: 'random_new'
          - name: 'test2'
            value: '192.168.2.1'
          - name: 'test3'
            state: 'absent'
          - name: 'test4'
            type: 'CNAME'
            value: 'test2.test3.ansibleguy'
      enabled: false
```

(continues on next page)

(continued from previous page)

```
- name: Removing all
  ansibleguy.opnsense.bind_record_multi:
    records:
      'template.ansibleguy':
        - 'test1'
        - name: 'test1'
          type: 'TXT'
        - 'test2'
        - 'test3'
        - name: 'test4'
          type: 'CNAME'
    state: 'absent'
```

Tip: Check out the repository on GitHub

Report missing/incorrect information or broken links

CHAPTER

ELEVEN

CRON JOBS

STATE: stable

TESTS: Playbook

API Docs: Core - Cron

11.1 Definition

Table 1: Definition

Parameter	Type	Required	Default	Aliases	Comment
description	string	true	-	desc	Description for the cron-job. Will be used to identify the job. If changed - the module will think it is a different job!
command	string	false for state changes, else true	-	cmd	One of the pre-defined commands listed in the WEB-GUI. **WARNING** the values seen in the WEB-GUI DO NOT exactly match the ones you need to provide here! Per example: 'automatic firmware update', 'system remote backup' or 'ipsec restart'. Tip: The module will output a list of available commands as error AFTER a first job was created.
parameters	string	false	-	params	Enter parameters for this job if required.
minutes	string	false	'0'	min, m	Value needs to be between 0 and 59; multiple values, ranges, steps and asterisk are supported (ex. 1,10,20,30 or 1-30).
hours	string	false	'0'	hour, h	Value needs to be between 0 and 23; multiple values, ranges, steps and asterisk are supported (ex. 1,2,8 or 0-8).
days	string	false	'*'	day, d	Value needs to be between 1 and 31; multiple values, ranges, L (last day of month), steps and asterisk are supported (ex. 1,2,8 or 1-28).
months	string	false	'*'	month, M	Value needs to be between 1 and 12 or JAN to DEC, multiple values, ranges, steps and asterisk are supported (ex. JAN,2,10 or 3-8).
weekdays	string	false	'*'	wd	Value needs to be between 0 and 7 (Sunday to Sunday), multiple values, ranges, steps and asterisk are supported (ex. 1,2,4 or 0-4).
who	string	false	'root'	-	User who should run the command
reload	boolean	false	true	-	If the running config should be reloaded on change - this may take some time. For mass-managing items you might want to reload it 'manually' after all changes are done => using the *ansible.guy.opnsense.reload* module.

For basic parameters see: *Basic*

11.2 Usage

To add custom cron-job scripts - see: OPNSense Documentation | OPNSense Forum

11.3 Examples

```yaml
- hosts: localhost
  gather_facts: no
  module_defaults:
    group/ansibleguy.opnsense.all:
      firewall: 'opnsense.template.ansibleguy.net'
      api_credential_file: '/home/guy/.secret/opn.key'

    ansibleguy.opnsense.list:
      target: 'cron'

  tasks:
    - name: Example
      ansibleguy.opnsense.cron:
        description: 'test1'
        command: 'system remote backup'
        # parameters
        # minutes: '0'
        # hours: '0'
        # days: '*'
        # months: '*'
        # weekdays: '*'
        # who: 'root'
        # state: 'absent'
        # debug: false

    - name: Adding daily firmware update check
      ansibleguy.opnsense.cron:
        description: 'test2'
        command: 'firmware poll'
        minutes: '0'
        hours: '0'
        days: '*'

    - name: Removing some job
      ansibleguy.opnsense.cron:
        description: 'test3'
        state: 'absent'

    - name: Adding monthly firmware upgrade
      ansibleguy.opnsense.cron:
        description: 'test4'
        command: 'firmware auto-update'
        minutes: '0'
        hours: '4'
        days: '21'
```

(continues on next page)

(continued from previous page)

```
      months: '*'

  - name: Listing
    ansibleguy.opnsense.list:
    #  target: 'cron'
    register: existing_jobs

  - name: Printing jobs
    ansible.builtin.debug:
      var: existing_jobs.data
```

CHAPTER

TWELVE

FRR BFD

STATE: stable

TESTS: frr_bfd_general | frr_bfd_neighbor

API Docs: Plugins - Quagga

Service Docs: Dynamic Routing

FRR Docs: FRRouting (*make sure you are looking at the current OPNSense package version!*)

12.1 Sponsoring

Thanks to @telmich for sponsoring the development of these modules!

12.2 Prerequisites

You need to install the FRR plugin:

```
os-frr
```

You can also install it using the package module.

12.3 Definition

For basic parameters see: Basics

12.3.1 ansibleguy.opnsense.frr_bfd_general

Parameter	Type	Required	Default value	Aliases	Comment
enabled	bool	false	true	-	En- or disable BFD

12.3.2 ansibleguy.opnsense.frr_bfd_neighbor

Parameter	Type	Required	Default value	Aliases	Comment
ip	string	true	-	neighbor, address, peer_ip, peer	The neighbor IP or IP-range to manage. This field will be used to match existing entries with the provided config!
description	string	false	-	desc	Optional description for the neighbor

12.4 Examples

12.4.1 ansibleguy.opnsense.frr_bfd_general

```
- hosts: localhost
  gather_facts: no
  module_defaults:
    group/ansibleguy.opnsense.all:
      firewall: 'opnsense.template.ansibleguy.net'
      api_credential_file: '/home/guy/.secret/opn.key'

  tasks:
    - name: Example
      ansibleguy.opnsense.frr_bfd_general:
        # enabled: true

    - name: Enabling BFD
      ansibleguy.opnsense.frr_bfd_general:
        enabled: true

    - name: Disabling BFD
      ansibleguy.opnsense.frr_bfd_general:
        enabled: false
```

12.4.2 ansibleguy.opnsense.frr_bfd_neighbor

```
- hosts: localhost
  gather_facts: no
  module_defaults:
    group/ansibleguy.opnsense.all:
      firewall: 'opnsense.template.ansibleguy.net'
      api_credential_file: '/home/guy/.secret/opn.key'

    ansibleguy.opnsense.list:
      target: 'frr_bfd_neighbor'
```

(continues on next page)

(continued from previous page)

```yaml
tasks:
  - name: Example
    ansibleguy.opnsense.frr_bfd_neighbor:
      ip: '10.0.0.1'
      # description: 'test1'
      # enabled: true
      # debug: false
      # state: 'present'
      # reload: true

  - name: Adding neighbor
    ansibleguy.opnsense.frr_bfd_neighbor:
      ip: '10.0.0.1'
      description: 'test2'

  - name: Disabling neighbor
    ansibleguy.opnsense.frr_bfd_neighbor:
      ip: '10.0.0.1'
      description: 'test2'
      enabled: false

  - name: Listing
    ansibleguy.opnsense.list:
    #  target: 'frr_bfd_neighbor'
    register: existing_entries

  - name: Printing neighbors
    ansible.builtin.debug:
      var: existing_entries.data

  - name: Removing neighbor 'test3'
    ansibleguy.opnsense.frr_bfd_neighbor:
      ip: '10.0.0.1'
      state: 'absent'
```

CHAPTER

THIRTEEN

FRR BGP

STATE: stable

TESTS: frr_bgp_general | frr_bgp_neighbor | frr_bgp_prefix_list | frr_bgp_route_map | frr_bgp_community_list | frr_bgp_as_path

API Docs: Plugins - Quagga

Service Docs: Dynamic Routing

FRR Docs: FRRouting (*make sure you are looking at the current OPNSense package version!*)

13.1 Sponsoring

Thanks to @telmich for sponsoring the development of these modules!

13.2 Prerequisites

You need to install the FRR plugin:

```
os-frr
```

You can also install it using the package module.

13.3 Definition

For basic parameters see: Basics

13.3.1 ansibleguy.opnsense.frr_bgp_general

Parameter	Type	Require	Default value	Alias	Comment
as_nu	string	true	-	as, as_nr	BGP AS-Number
id	string	false	-	route	In some cases it might be clearer to set a fixed router-id. (*4-byte field/IPv4 Address*)
graceful	boole	false	-	-	BGP graceful restart functionality as defined in RFC-4724 defines the mechanisms that allows BGP speaker to continue to forward data packets along known routes while the routing protocol information is being restored.
networks	list	false	-	nets	Select the network to advertise, you have to set a Null route via System -> Routes
redistribut	list	false	-	-	Select other routing sources, which should be redistributed to the other nodes. Choose from: 'ospf', 'connected', 'kernel', 'rip', 'static'
reload	boole	false	true	-	If the running config should be reloaded on change - this will take some time. You might want to reload it 'manually' after all changes are done => using the reload module.
enabled	boole	false	true	-	En- or disable the service

13.3.2 ansibleguy.opnsense.frr_bgp_neighbor

Parameter	Type	Required	Default value	Aliases	Comment
match_	string	false	['ip', 'description']	-	Fields that are used to match configured neighbor with the running config - if any of those fields are changed, the module will think it's a new entry. At least one of: 'ip', 'as_number', 'weight', 'local_ip', 'source_int', 'ipv6_link_local_int', 'disable_connected_check', 'description', 'prefix_list_in', 'prefix_list_out', 'route_map_in', 'route_map_out'
as_nun	string	false for state changes else true	-	as, as_nr, remote_as	BGP AS-Number of the neighbor
ip	string	false for state changes else true	-	peer, peer_ip, address, neighbor	IP-address of the neighbor
password	string	false	-	pwd	Set a (MD5-hashed) password for BGP authentication
weight	integer	false	-	-	Specify a default weight value for the neighbor's routes. Integer between 0 and 65535
local_ip	string	false	-	local	Set the local IP connecting to the neighbor. This is only required for BGP authentication.
source_	string	false	-	src_int, update_src, update_sou	Physical name of the IPv4 interface facing the peer. You must provide the network port as shown in 'Interface - Assignments - Interface ID (in brackets)'
ipv6_li	string	false	-	v6_ll_int ipv6_ll_i link_loca	Interface to use for IPv6 link-local neighbours. You must provide the network port as shown in 'Interface - Assignments - Interface ID (in brackets)'
next_h	boole	false	false	nhs	
next_h	boole	false	false	nhsa	Add the parameter "all" after next-hop-self command
multi_l	boole	false	false	-	Specifying ebgp-multihop allows sessions with eBGP neighbors to establish when they are multiple hops away. When the neighbor is not directly connected and this knob is not enabled, the session will not establish.
multi_	boole	false	false	-	Is this neighbour multiprotocol capable per RFC 2283
rrclient	boole	false	false	route_ref	
bfd	boole	false	false	-	Enable BFD support for this neighbor
send_d	boole	false	false	default_orig	
as_ove	boole	false	false	asoverride	Override AS number of the originating router with the local AS number. This command is only allowed for eBGP peers
disable_c	boole	false	false	asoverride	Allow peerings between directly connected eBGP peers using loopback addresses
keepali	integer	false	60	keep_aliv	Keepalive timer to check if the neighbor is still up. Integer between 1 and 1000
hold_d	integer	false	180	holddown	The time in seconds when a neighbor is considered dead. This is usually 3 times the keepalive timer. Integer between 3 and 3000

13.3.3 ansibleguy.opnsense.frr_bgp_prefix_list

Parameter	Type	Required	Default value	Aliases	Comment
name	string	true	-	-	Name to identify the prefix-list by - used in combination with its sequence number. Maximum length = 64
seq	integer	true	-	sequence, seq_numb	Sequence number for the prefix-list
network	string	false for state changes, else true	-	net	
action	string	false for state changes, else true	-	-	Set permit for match or deny to negate the rule. One of: 'permit', 'deny'
description	string	false	-	-	Optional description
version	string	false	IPv4	ipv	IP-version to use. One of: IPv4, IPv6
reload	boole	false	true	-	If the running config should be reloaded on change - this will take some time. You might want to reload it 'manually' after all changes are done => using the reload module.

13.3.4 ansibleguy.opnsense.frr_bgp_route_map

Parameter	Type	Required	Default value	Alias	Comment
name	string	true	-	-	Name to identify the route-map by. Maximum length = 64
id	integer	false for state changes, else true	-	-	Route-map ID between 10 and 99. Be aware that the sorting will be done under the hood, so when you add an entry between it get's to the right position
action	string	false for state changes, else true	-	-	Set permit for match or deny to negate the rule. One of: 'permit', 'deny'
description	string	false	-	-	Optional description
as_path	list	false	-	as_pa	List of as-path entries to link
prefix_list	dict	false	-	pre, prefix	Dictionary of prefixes to link. Per example: "{prefix_name: [seq1, seq2]}" or "{'pre1': [5, 6]}" will link prefixes with the name 'pre1' and sequence 5-6
community_li	list	false	-	community	List of community-list entries to link
set	string	false	-	-	Free text field for your set, please be careful! You can set e.g. "local-preference 300" or "community 1:1" (http://www.nongnu.org/quagga/docs/docs-multi/Route-Map-Set-Command.html#Route-Map-Set-Command)
reload	boole	false	true	-	If the running config should be reloaded on change - this will take some time. You might want to reload it 'manually' after all changes are done => using the reload module.

13.3.5 ansibleguy.opnsense.frr_bgp_community_list

Parameter	Type	Required	Default value	Alias	Comment
description	string	true	-	desc	Description used to identify the community-list by
number	integer	false for state changes, else true	-	nr	Set the number of your Community-List. 1-99 are stardard lists while 100-500 are expanded lists
seq	integer	false for state changes, else true	-	sequence	The ACL sequence number (10-99)
action	string	false for state changes, else true	-	-	Set permit for match or deny to negate the rule. One of: 'permit', 'deny'
community	string	false for state changes, else true	-	comm	The community you want to match. You can also regex and it is not validated so please be careful
reload	boolean	false	true	-	If the running config should be reloaded on change - this will take some time. You might want to reload it 'manually' after all changes are done => using the reload module.

13.3.6 ansibleguy.opnsense.frr_bgp_as_path

Parameter	Type	Required	Default value	Alias	Comment
description	string	true	-	desc	Description used to identify the as-path by
number	integer	false for state changes, else true	-	nr	The ACL rule number (10-99); keep in mind that there are no sequence numbers with AS-Path lists. When you want to add a new line between you have to completely remove the ACL
action	string	false for state changes, else true	-	-	Set permit for match or deny to negate the rule. One of: 'permit', 'deny'
as_pa	string	false for state changes, else true	-	as	The AS pattern you want to match, regexp allowed (e.g. .$ or _1$). It's not validated so please be careful!
reload	boole	false	true	-	If the running config should be reloaded on change - this will take some time. You might want to reload it 'manually' after all changes are done => using the reload module.

13.4 Examples

13.4.1 ansibleguy.opnsense.frr_bgp_general

```
- hosts: localhost
  gather_facts: no
  module_defaults:
    group/ansibleguy.opnsense.all:
      firewall: 'opnsense.template.ansibleguy.net'
      api_credential_file: '/home/guy/.secret/opn.key'

    ansibleguy.opnsense.list:
      target: 'frr_bgp_general'

  tasks:
    - name: Example
      ansibleguy.opnsense.frr_bgp_general:
        as_number: 1337
        # id: '10.0.0.1'
        # graceful: false
        # networks: []
        # redistribute: []
        # enabled: true
        # reload: true
```

(continues on next page)

(continued from previous page)

```
    - name: Configuring general settings
      ansibleguy.opnsense.frr_bgp_general:
        as_number: 1337
        id: '10.0.0.1'
        graceful: true
        networks: ['10.0.10.0/24']
        redistribute: ['static']

    - name: Disabling BGP
      ansibleguy.opnsense.frr_bgp_general:
        as_number: 1337
        id: '10.0.0.1'
        graceful: true
        networks: ['10.0.10.0/24']
        redistribute: ['static']
        enabled: false

    - name: Pulling settings
      ansibleguy.opnsense.list:
      #  target: 'frr_bgp_general'
      register: existing_entries

    - name: Printing settings
      ansible.builtin.debug:
        var: existing_entries.data
```

13.4.2 ansibleguy.opnsense.frr_bgp_neighbor

```
- hosts: localhost
  gather_facts: no
  module_defaults:
    group/ansibleguy.opnsense.all:
      firewall: 'opnsense.template.ansibleguy.net'
      api_credential_file: '/home/guy/.secret/opn.key'

    ansibleguy.opnsense.frr_bgp_neighbor:
      match_fields: ['ip']

    ansibleguy.opnsense.list:
      target: 'frr_bgp_neighbor'

  tasks:
    - name: Example
      ansibleguy.opnsense.frr_bgp_neighbor:
        as_number: 1337
        ip: '10.0.0.1'
        # password: "{{ 'random' | hash('md5') }}"
        # weight: 200
        # local_ip: '10.0.0.254'
        # source_int: 'opt1'
```

(continues on next page)

(continued from previous page)

```
      # ipv6_link_local_int: 'opt1'
      # next_hop_self: false
      # next_hop_self_all: false
      # multi_hop: false
      # multi_protocol: false
      # rrclient: false
      # bfd: false
      # send_default_route: false
      # as_override: false
      # disable_connected_check: false
      # keepalive: 60
      # hold_down: 180
      # connect_timer: 30
      # description: 'test1'
      # prefix_list_in: 'prefix1'
      # prefix_list_out: 'prefix2'
      # route_map_in: 'map1'
      # route_map_out: 'map2'
      # enabled: true
      # reload: true
      # match_fields: ['ip', 'description']

  - name: Creating neighbor
    ansibleguy.opnsense.frr_bgp_neighbor:
      description: 'test2'
      as_number: 1337
      ip: '10.0.0.1'
      password: "{{ 'random' | hash('md5') }}"
      weight: 200
      source_int: 'opt1'
      multi_protocol: true
      keepalive: 45
      hold_down: 135
      # match_fields: ['ip']

  - name: Disabling neighbor
    ansibleguy.opnsense.frr_bgp_neighbor:
      description: 'test2'
      as_number: 1337
      ip: '10.0.0.1'
      password: "{{ 'random' | hash('md5') }}"
      weight: 200
      source_int: 'opt1'
      multi_protocol: true
      keepalive: 45
      hold_down: 135
      enabled: false
      # match_fields: ['ip']

  - name: Pulling neighbors
    ansibleguy.opnsense.list:
    #  target: 'frr_bgp_neighbor'
```

(continues on next page)

(continued from previous page)

```
      register: existing_entries

    - name: Printing neighbors
      ansible.builtin.debug:
        var: existing_entries.data

    - name: Removing neighbor
      ansibleguy.opnsense.frr_bgp_neighbor:
        ip: '10.0.0.1'
        state: 'absent'
```

13.4.3 ansibleguy.opnsense.frr_bgp_prefix_list

```
- hosts: localhost
  gather_facts: no
  module_defaults:
    group/ansibleguy.opnsense.all:
      firewall: 'opnsense.template.ansibleguy.net'
      api_credential_file: '/home/guy/.secret/opn.key'

    ansibleguy.opnsense.list:
      target: 'frr_bgp_prefix_list'

  tasks:
    - name: Example
      ansibleguy.opnsense.frr_bgp_prefix_list:
        name: 'test1'
        network: '10.0.0.0/24'
        seq: 10
        action: 'permit'
        # description: 'test1'
        # enabled: true
        # reload: true

    - name: Creating prefix-list
      ansibleguy.opnsense.frr_bgp_prefix_list:
        name: 'test2'
        network: '10.0.10.0/24'
        seq: 55
        action: 'permit'

    - name: Disabling prefix-list
      ansibleguy.opnsense.frr_bgp_prefix_list:
        name: 'test2'
        network: '10.0.10.0/24'
        seq: 55
        action: 'permit'
        enabled: false

    - name: Pulling prefix-lists
```

(continues on next page)

(continued from previous page)

```
      ansibleguy.opnsense.list:
      #  target: 'frr_bgp_prefix_list'
      register: existing_entries

    - name: Printing prefix-lists
      ansible.builtin.debug:
        var: existing_entries.data

    - name: Removing prefix-list
      ansibleguy.opnsense.frr_bgp_prefix_list:
        name: 'test2'
        state: 'absent'
```

13.4.4 ansibleguy.opnsense.frr_bgp_route_map

```
- hosts: localhost
  gather_facts: no
  module_defaults:
    group/ansibleguy.opnsense.all:
      firewall: 'opnsense.template.ansibleguy.net'
      api_credential_file: '/home/guy/.secret/opn.key'

    ansibleguy.opnsense.list:
      target: 'frr_bgp_route_map'

  tasks:
    - name: Example
      ansibleguy.opnsense.frr_bgp_route_map:
        name: 'test1'
        id: 55
        action: 'permit'
        # as_path_list: []
        # prefix_list: {}
        # community_list: []
        # set: ''
        # description: 'test1'
        # enabled: true
        # reload: true

    - name: Creating route-map
      ansibleguy.opnsense.frr_bgp_route_map:
        name: 'test2'
        prefix_list: {'test_prefix': 50}
        id: 55
        action: 'permit'

    - name: Disabling route-map
      ansibleguy.opnsense.frr_bgp_route_map:
        name: 'test2'
        prefix_list: {'test_prefix': 50}
```

(continues on next page)

(continued from previous page)

```
        id: 55
        action: 'permit'
        enabled: false

    - name: Pulling route-maps
      ansibleguy.opnsense.list:
      #  target: 'frr_bgp_route_map'
      register: existing_entries

    - name: Printing route-maps
      ansible.builtin.debug:
        var: existing_entries.data

    - name: Removing route-map
      ansibleguy.opnsense.frr_bgp_route_map:
        name: 'test2'
        state: 'absent'
```

13.4.5 ansibleguy.opnsense.frr_bgp_community_list

```
- hosts: localhost
  gather_facts: no
  module_defaults:
    group/ansibleguy.opnsense.all:
      firewall: 'opnsense.template.ansibleguy.net'
      api_credential_file: '/home/guy/.secret/opn.key'

    ansibleguy.opnsense.list:
      target: 'frr_bgp_community_list'

  tasks:
    - name: Example
      ansibleguy.opnsense.frr_bgp_community_list:
        description: 'test1'
        number: 55
        seq: 55
        action: 'permit'
        community: 'example'
        # enabled: true
        # reload: true

    - name: Creating community-list
      ansibleguy.opnsense.frr_bgp_community_list:
        description: 'test2'
        number: 20
        seq: 25
        action: 'permit'
        community: 'test_community'

    - name: Disabling community-list
```

(continues on next page)

(continued from previous page)

```
      ansibleguy.opnsense.frr_bgp_community_list:
        description: 'test2'
        number: 20
        seq: 25
        action: 'permit'
        community: 'test_community'
        enabled: false

    - name: Pulling community-lists
      ansibleguy.opnsense.list:
      #  target: 'frr_bgp_community_list'
      register: existing_entries

    - name: Printing community-lists
      ansible.builtin.debug:
        var: existing_entries.data

    - name: Removing community-list
      ansibleguy.opnsense.frr_bgp_community_list:
        description: 'test2'
        state: 'absent'
```

13.4.6 ansibleguy.opnsense.frr_bgp_as_path

```
- hosts: localhost
  gather_facts: no
  module_defaults:
    group/ansibleguy.opnsense.all:
      firewall: 'opnsense.template.ansibleguy.net'
      api_credential_file: '/home/guy/.secret/opn.key'

    ansibleguy.opnsense.list:
      target: 'frr_bgp_as_path'

  tasks:
    - name: Example
      ansibleguy.opnsense.frr_bgp_as_path:
        description: 'test1'
        number: 55
        action: 'permit'
        as_pattern: 'example'
        # enabled: true
        # reload: true

    - name: Creating as-path
      ansibleguy.opnsense.frr_bgp_as_path:
        description: 'test2'
        number: 20
        action: 'permit'
        as_pattern: 'test_as'
```

(continues on next page)

(continued from previous page)

```
- name: Disabling as-path
  ansibleguy.opnsense.frr_bgp_as_path:
    description: 'test2'
    number: 20
    action: 'permit'
    as_pattern: 'test_as'
    enabled: false

- name: Pulling as-paths
  ansibleguy.opnsense.list:
  #  target: 'frr_bgp_as_path'
  register: existing_entries

- name: Printing as-paths
  ansible.builtin.debug:
    var: existing_entries.data

- name: Removing as-path
  ansibleguy.opnsense.frr_bgp_as_path:
    description: 'test2'
    state: 'absent'
```

CHAPTER

FOURTEEN

FRR DIAGNOSTIC

STATE: stable

TESTS: frr_diagnostic

API Docs: Plugins - Quagga

Service Docs: Dynamic Routing

FRR Docs: FRRouting (*make sure you are looking at the current OPNSense package version!*)

14.1 Sponsoring

Thanks to @telmich for sponsoring the development of these modules!

14.2 Prerequisites

You need to install the FRR plugin:

```
os-frr
```

You can also install it using the package module.

14.3 Definition

For basic parameters see: Basics

14.3.1 ansibleguy.opnsense.frr_diagnostic

Parameter	Type	Require	Default value	Alias	Comment
target	string	true	-	-	What information to query. One of: 'bgpneighbors', 'bgproute', 'bgproute4', 'bgproute6', 'bgpsummary', 'generalroute', 'generalroute4', 'generalroute6', 'generalrunningconfig', 'ospfdatabase', 'ospfinterface', 'ospfneighbor', 'ospfoverview', 'ospfroute', 'ospfv3database', 'ospfv3interface', 'ospfv3neighbor', 'ospfv3overview', 'ospfv3route'

14.4 Examples

14.4.1 ansibleguy.opnsense.frr_diagnostic

```
- hosts: localhost
  gather_facts: no
  module_defaults:
    group/ansibleguy.opnsense.all:
      firewall: 'opnsense.template.ansibleguy.net'
      api_credential_file: '/home/guy/.secret/opn.key'

  tasks:
    - name: Example
      ansibleguy.opnsense.frr_diagnostic:
        target: 'generalroute'
      register: frr_info

    - name: Printing
      ansible.builtin.debug:
        var: frr_info.data
```

CHAPTER

FIFTEEN

FRR GENERAL

STATE: stable

TESTS: frr_general

API Docs: Plugins - Quagga

Service Docs: Dynamic Routing

FRR Docs: FRRouting (*make sure you are looking at the current OPNSense package version!*)

15.1 Prerequisites

You need to install the FRR plugin:

```
os-frr
```

You can also install it using the package module.

15.2 Definition

For basic parameters see: Basics

15.2.1 ansibleguy.opnsense.frr_general

Parameter	Type	Required	Default value	Aliases	Comment
enabled	bool	false	true	-	En- or disable the FRR service
profile	string	false	'traditional'	-	One of: 'traditional', 'datacenter'. The 'datacenter' profile is more aggressive. Please refer to the FRR documentation for more information
log	bool	false	true	logging	En- or disable (syslog) logging
log_level	string	false	'notifications'	-	One of: 'critical', 'emergencies', 'errors', 'alerts', 'warnings', 'notifications', 'informational', 'debugging'.
carp	bool	false	false	carp_fai	Will activate the routing service only on the primary device
snmp_a	bool	false	false	-	En- or disable support for Net-SNMP AgentX

15.3 Examples

15.3.1 ansibleguy.opnsense.frr_general

```yaml
- hosts: localhost
  gather_facts: no
  module_defaults:
    group/ansibleguy.opnsense.all:
      firewall: 'opnsense.template.ansibleguy.net'
      api_credential_file: '/home/guy/.secret/opn.key'

  tasks:
    - name: Example
      ansibleguy.opnsense.frr_general:
        # enabled: true
        # profile: 'traditional'
        # log: true
        # log_level: 'notifications'
        # snmp_agentx: false
        # carp: false

    - name: Enabling FRR
      ansibleguy.opnsense.frr_general:
        enabled: true
        profile: 'traditional'
        log: true
        log_level: 'notifications'
```

CHAPTER

SIXTEEN

FRR OSPF

STATE: stable

TESTS: frr_ospf_general | frr_ospf_prefix_list | frr_ospf_interface | frr_ospf_route_map | frr_ospf_network | frr_ospf3_general | frr_ospf3_interface

API Docs: Plugins - Quagga

Service Docs: Dynamic Routing

FRR Docs: FRRouting (*make sure you are looking at the current OPNSense package version!*)

16.1 Sponsoring

Thanks to @telmich for sponsoring the development of these modules!

16.2 Prerequisites

You need to install the FRR plugin:

```
os-frr
```

You can also install it using the package module.

16.3 Definition

For basic parameters see: Basics

16.3.1 OSPF

ansibleguy.opnsense.frr_ospf_general

Parameter	Type	Require	Default value	Aliases	Comment
carp	boole	false	false	carp_demote	Register CARP status monitor, when no neighbors are found, consider this node less attractive. This feature needs syslog enabled using "Debugging" logging to catch all relevant status events. This option is not compatible with "Enable CARP Failover"
id	string	false	-	router_id	If you have a CARP setup, you may want to configure a router id in case of a conflict. (*4-byte field/IPv4 Address*)
cost	integer	false	-	reference_cost, ref_cost	Here you can adjust the reference cost in Mbps for path calculation. Mostly needed when you bundle interfaces to higher bandwidth
passive_i	list	false	-	passive_interface	Select the interfaces, where no OSPF packets should be sent to. You must provide the network port as shown in 'Interface - Assignments - Interface ID (in brackets)'
redistribute	list	false	-	-	Select other routing sources, which should be redistributed to the other nodes. Choose from: 'bgp', 'connected', 'kernel', 'rip', 'static'
redistribute	string	false	-	-	Route Map to set for Redistribution
originate	boole	false	false	orig, advertise_default_g	This will send the information that we have a default gateway
originate_a	boole	false	false	orig_always, always_advertis	This will send the information that we have a default gateway, regardless of if it is available
originate_r	integer	false	-	orig_metric	This let you manipulate the metric when advertising default gateway
reload	boole	false	true	-	If the running config should be reloaded on change - this will take some time. You might want to reload it 'manually' after all changes are done => using the reload module.
enabled	boole	false	true	-	En- or disable the service

ansibleguy.opnsense.frr_ospf_network

Parameter	Type	Required	Default value	Aliases	Comment
match	string	false	['ip', 'mask	-	Fields that are used to match configured interface with the running config - if any of those fields are changed, the module will think it's a new entry. At least one of: 'ip', 'mask', 'area', 'area_range'
ip	string	true	-	network_address, nw_address, address	
mask	string	true	-	network_mask, nw_mask	Integer between 0 and 32
area	string	false for state changes, else true	-	-	Area in wildcard mask style like 0.0.0.0 and no decimal 0. Only use Area in Interface tab or in Network tab once
area_r	string	-	-	-	Here you can summarize a network for this area like 192.168.0.0/23
prefix_lis	string	-	-	prefix_in, pre_in	Prefix-List for inbound direction
prefix_lis	string	-	-	prefix_out, pre_out	Prefix-List for outbound direction
reload	boole	false	true	-	If the running config should be reloaded on change - this will take some time. You might want to reload it 'manually' after all changes are done => using the reload module.

ansibleguy.opnsense.frr_ospf_interface

Parameter	Type	Required	Default value	Alias	Comment
match_	string	false	['interface', 'area']	-	Fields that are used to match configured interface with the running config - if any of those fields are changed, the module will think it's a new entry. At least one of: 'interface', 'area', 'passive', 'carp_depend_on', 'network_type'
interface	string	true	-	name int	Interface to configure. You must provide the network port as shown in 'Interface - Assignments - Interface ID (in brackets)'
area	string	false for state changes, else true	-	-	Area in wildcard mask style like 0.0.0.0 and no decimal 0
auth_ty	string	false	-	-	What authentication type to use. Currently only 'message-digest' is supported
auth_ke	string	true if 'auth_type' is set, else false	-	-	The key to authenticate
auth_ke	integer	false	1	-	Integer between 1 and 255
cost	integer	false	-	-	Integer between 1 and 65535
cost_de	integer	false	65535	-	Integer between 1 and 65535
carp_de	string	false	-	-	The carp VHID to depend on, when this virtual address is not in master state, the interface cost will be set to the demoted cost. Integer between 1 and 65535
hello_i	integer	false	-	hello	Integer between 0 and 4294967295
dead_i	integer	false	-	dead	Integer between 0 and 4294967295
retransmit_int	integer	false	-	retransmit	Integer between 0 and 4294967295
transmit_del	integer	false	-	delay	Integer between 0 and 4294967295
priority	integer	false	-	prio	Integer between 0 and 4294967295
network_t	string	false	-	nw_t	One of: 'broadcast', 'non-broadcast', 'point-to-multipoint', 'point-to-point'
reload	boole	false	true	-	If the running config should be reloaded on change - this will take some time. You might want to reload it 'manually' after all changes are done => using the reload module.

ansibleguy.opnsense.frr_ospf_prefix_list

Parameter	Type	Required	Default value	Aliase	Comment
name	string	true	-	-	The name of the prefix-list
seq	string	false for state changes, else true	-	seq_nu	The ACL sequence number (10-99)
network	string	false for state changes, else true	-	net	The network pattern you want to match. It's not validated so please be careful!
action	string	false for state changes, else true	-	-	Set permit for match or deny to negate the rule. One of: 'permit', 'deny'
reload	boole	false	true	-	If the running config should be reloaded on change - this will take some time. You might want to reload it 'manually' after all changes are done => using the reload module.

ansibleguy.opnsense.frr_ospf_route_map

Parameter	Type	Required	Default value	Alias	Comment
name	string	true	-	-	Name to identify the route-map by
id	integer	false for state changes, else true	-	-	Route-map ID between 10 and 99. Be aware that the sorting will be done under the hood, so when you add an entry between it get's to the right position
action	string	false for state changes, else true	-	-	Set permit for match or deny to negate the rule. One of: 'permit', 'deny'
prefix_li	list	false	-	prefix	List of prefix-list entries to link
set	string	false	-	-	Free text field for your set, please be careful! You can set e.g. "local-preference 300" or "community 1:1" (http://www.nongnu.org/quagga/docs/docs-multi/Route-Map-Set-Command.html#Route-Map-Set-Command)
reload	boole	false	true	-	If the running config should be reloaded on change - this will take some time. You might want to reload it 'manually' after all changes are done => using the reload module.

16.3.2 OSPFv3 (*IPv6*)

ansibleguy.opnsense.frr_ospf3_general

Parameter	Type	Require	Default value	Alias	Comment
carp	boole	false	false	carp_	Register CARP status monitor, when no neighbors are found, consider this node less attractive. This feature needs syslog enabled using "Debugging" logging to catch all relevant status events. This option is not compatible with "Enable CARP Failover"
id	string	false	-	router	If you have a CARP setup, you may want to configure a router id in case of a conflict. (*4-byte field/IPv4 Address*)
redistribut	list	false	-	-	Select other routing sources, which should be redistributed to the other nodes. Choose from: 'bgp', 'connected', 'kernel', 'rip', 'static'
reloa	boole	false	true	-	If the running config should be reloaded on change - this will take some time. You might want to reload it 'manually' after all changes are done => using the reload module.
enabled	boole	false	true	-	En- or disable the service

ansibleguy.opnsense.frr_ospf3_interface

Parameter	Type	Required	Default value	Alias	Comment
match_	string	false	['interface', 'area']	-	Fields that are used to match configured interface with the running config - if any of those fields are changed, the module will think it's a new entry. At least one of: 'interface', 'area', 'passive', 'carp_depend_on', 'network_type'
interface	string	true	-	name int	Interface to configure. You must provide the network port as shown in 'Interface - Assignments - Interface ID (in brackets)'
area	string	false for state changes, else true	-	-	Area in wildcard mask style like 0.0.0.0 and no decimal 0
passive	boole	false	false	-	
cost	integer	false	-	-	Integer between 0 and 4294967295
cost_de	integer	false	-	6553	Integer between 1 and 65535
carp_de	string	false	-	-	The carp VHID to depend on, when this virtual address is not in master state, the interface cost will be set to the demoted cost. Integer between 1 and 65535
hello_i	integer	false	-	hello	Integer between 0 and 4294967295
dead_i	integer	false	-	dead	Integer between 0 and 4294967295
retransmit_int	integer	false	-	retransmit	Integer between 0 and 4294967295
transmit_del	integer	false	-	delay	Integer between 0 and 4294967295
priority	integer	false	-	prio	Integer between 0 and 4294967295
network_t	string	false	-	nw_t	One of: 'broadcast', 'point-to-point'
reload	boole	false	true	-	If the running config should be reloaded on change - this will take some time. You might want to reload it 'manually' after all changes are done => using the reload module.

16.4 Examples

16.4.1 OSPF (*IPv4*)

ansibleguy.opnsense.frr_ospf_general

```
- hosts: localhost
  gather_facts: no
  module_defaults:
    group/ansibleguy.opnsense.all:
      firewall: 'opnsense.template.ansibleguy.net'
      api_credential_file: '/home/guy/.secret/opn.key'

    ansibleguy.opnsense.list:
      target: 'frr_ospf_general'

  tasks:
    - name: Example
      ansibleguy.opnsense.frr_ospf_general:
        # id: '10.0.0.1'
        # cost: 200
        # passive_ints: []
        # redistribute: []
        # redistribute_map: ''
        # carp: false
        # originate: false
        # originate_always: false
        # originate_metric: 1000
        # enabled: true

    - name: Configuring general settings
      ansibleguy.opnsense.frr_ospf_general:
        id: '10.0.1.1'
        cost: 300
        passive_ints: ['lan']
        redistribute: ['static', 'bgp']
        originate: true
        originate_metric: 1000

    - name: Disabling OSPF
      ansibleguy.opnsense.frr_ospf_general:
        id: '10.0.1.1'
        cost: 300
        passive_ints: ['lan']
        redistribute: ['static', 'bgp']
        originate: true
        originate_metric: 1000
        enabled: false

    - name: Pulling settings
      ansibleguy.opnsense.list:
      #  target: 'frr_ospf_general'
```

(continues on next page)

(continued from previous page)

```
      register: existing_entries

    - name: Printing settings
      ansible.builtin.debug:
        var: existing_entries.data
```

ansibleguy.opnsense.frr_ospf_prefix_list

```
- hosts: localhost
  gather_facts: no
  module_defaults:
    group/ansibleguy.opnsense.all:
      firewall: 'opnsense.template.ansibleguy.net'
      api_credential_file: '/home/guy/.secret/opn.key'

    ansibleguy.opnsense.list:
      target: 'frr_ospf_prefix_list'

  tasks:
    - name: Example
      ansibleguy.opnsense.frr_ospf_prefix_list:
        name: 'example'
        seq: 10
        action: 'permit'
        network: '10.0.0.0/24'
        # enabled: true

    - name: Configuring prefix-list
      ansibleguy.opnsense.frr_ospf_prefix_list:
        name: 'test2'
        seq: 25
        action: 'permit'
        network: '10.0.1.0/24'

    - name: Disabling prefix-list
      ansibleguy.opnsense.frr_ospf_prefix_list:
        name: 'test2'
        seq: 25
        action: 'permit'
        network: '10.0.1.0/24'
        enabled: false

    - name: Pulling settings
      ansibleguy.opnsense.list:
      #  target: 'frr_ospf_prefix_list'
      register: existing_entries

    - name: Printing settings
      ansible.builtin.debug:
        var: existing_entries.data
```

(continues on next page)

(continued from previous page)

```
    - name: Removing prefix-list
      ansibleguy.opnsense.frr_ospf_prefix_list:
        name: 'test2'
        state: 'absent'
```

ansibleguy.opnsense.frr_ospf_route_map

```
- hosts: localhost
  gather_facts: no
  module_defaults:
    group/ansibleguy.opnsense.all:
      firewall: 'opnsense.template.ansibleguy.net'
      api_credential_file: '/home/guy/.secret/opn.key'

    ansibleguy.opnsense.list:
      target: 'frr_ospf_route_map'

  tasks:
    - name: Example
      ansibleguy.opnsense.frr_ospf_route_map:
        name: 'example'
        id: 10
        action: 'permit'
        # prefix_list: []
        # set: ''
        # enabled: true

    - name: Configuring route-map
      ansibleguy.opnsense.frr_ospf_route_map:
        name: 'test2'
        id: 65
        action: 'permit'
        set: 'local-preference 300'

    - name: Disabling route-map
      ansibleguy.opnsense.frr_ospf_route_map:
        name: 'test2'
        id: 65
        action: 'permit'
        set: 'local-preference 300'
        enabled: false

    - name: Pulling settings
      ansibleguy.opnsense.list:
      #  target: 'frr_ospf_route_map'
      register: existing_entries

    - name: Printing settings
      ansible.builtin.debug:
```

(continues on next page)

(continued from previous page)

```
      var: existing_entries.data

  - name: Removing route-map
    ansibleguy.opnsense.frr_ospf_route_map:
      name: 'test2'
      state: 'absent'
```

ansibleguy.opnsense.frr_ospf_network

```
- hosts: localhost
  gather_facts: no
  module_defaults:
    group/ansibleguy.opnsense.all:
      firewall: 'opnsense.template.ansibleguy.net'
      api_credential_file: '/home/guy/.secret/opn.key'

    ansibleguy.opnsense.frr_ospf_network:
      match_fields: ['ip', 'mask']

    ansibleguy.opnsense.list:
      target: 'frr_ospf_route_map'

  tasks:
    - name: Example
      ansibleguy.opnsense.frr_ospf_network:
        ip: '10.0.0.0'
        mask: 24
        area: '0.0.0.0'
        # area_range: ''
        # enabled: true

    - name: Configuring network
      ansibleguy.opnsense.frr_ospf_network:
        ip: '10.0.1.0'
        mask: 28
        area: '0.0.1.0'

    - name: Disabling network
      ansibleguy.opnsense.frr_ospf_network:
        ip: '10.0.1.0'
        mask: 28
        area: '0.0.1.0'
        enabled: false

    - name: Pulling settings
      ansibleguy.opnsense.list:
      #  target: 'frr_ospf_network'
      register: existing_entries

    - name: Printing settings
```

(continues on next page)

(continued from previous page)

```
      ansible.builtin.debug:
        var: existing_entries.data

    - name: Removing network
      ansibleguy.opnsense.frr_ospf_network:
        ip: '10.0.1.0'
        mask: 28
        state: 'absent'
```

ansibleguy.opnsense.frr_ospf_interface

```
- hosts: localhost
  gather_facts: no
  module_defaults:
    group/ansibleguy.opnsense.all:
      firewall: 'opnsense.template.ansibleguy.net'
      api_credential_file: '/home/guy/.secret/opn.key'

    ansibleguy.opnsense.frr_ospf_interface:
      match_fields: ['interface']

    ansibleguy.opnsense.list:
      target: 'frr_ospf_interface'

  tasks:
    - name: Example
      ansibleguy.opnsense.frr_ospf_interface:
        interface: 'opt1'
        # area: '0.0.0.0'
        # cost: 10
        # cost_demoted: 10
        # hello_interval: 10
        # dead_interval: 10
        # retransmit_interval: 10
        # transmit_delay: 10
        # priority: 10
        # network_type: ''
        # carp_depend_on: ''
        # auth_type: ''
        # auth_key: ''
        # auth_key_id: 1
        # enabled: true
        # match_fields: ['interface', 'area']

    - name: Configuring interface
      ansibleguy.opnsense.frr_ospf_interface:
        interface: 'opt1'
        area: '0.0.0.0'
        cost: 500
        cost_demoted: 2000
```

(continues on next page)

(continued from previous page)

```
        hello_interval: 60
        dead_interval: 30
        retransmit_interval: 60
        transmit_delay: 60
        priority: 30
        network_type: 'point-to-multipoint'
        auth_type: 'message-digest'
        auth_key: "{{ 'random' | hash('md5') }}"

    - name: Disabling interface
      ansibleguy.opnsense.frr_ospf_interface:
        interface: 'opt1'
        area: '0.0.0.0'
        cost: 500
        cost_demoted: 2000
        hello_interval: 60
        dead_interval: 30
        retransmit_interval: 60
        transmit_delay: 60
        priority: 30
        network_type: 'point-to-multipoint'
        auth_type: 'message-digest'
        auth_key: "{{ 'random' | hash('md5') }}"
        enabled: false

    - name: Pulling settings
      ansibleguy.opnsense.list:
      #  target: 'frr_ospf_interface'
      register: existing_entries

    - name: Printing settings
      ansible.builtin.debug:
        var: existing_entries.data

    - name: Removing interface
      ansibleguy.opnsense.frr_ospf_interface:
        interface: 'opt1'
        state: 'absent'
```

16.4.2 OSPFv3 (*IPv6*)

ansibleguy.opnsense.frr_ospf3_general

```
- hosts: localhost
  gather_facts: no
  module_defaults:
    group/ansibleguy.opnsense.all:
      firewall: 'opnsense.template.ansibleguy.net'
      api_credential_file: '/home/guy/.secret/opn.key'
```

(continues on next page)

(continued from previous page)

```
    ansibleguy.opnsense.list:
      target: 'frr_ospf3_general'

  tasks:
    - name: Example
      ansibleguy.opnsense.frr_ospf3_general:
        # id: '10.0.0.1'
        # redistribute: []
        # carp: false
        # enabled: true

    - name: Configuring general settings
      ansibleguy.opnsense.frr_ospf3_general:
        id: '10.0.1.1'
        redistribute: ['static']

    - name: Disabling OSPFv3
      ansibleguy.opnsense.frr_ospf3_general:
        id: '10.0.1.1'
        redistribute: ['static']
        enabled: false

    - name: Pulling settings
      ansibleguy.opnsense.list:
      #  target: 'frr_ospf3_general'
      register: existing_entries

    - name: Printing settings
      ansible.builtin.debug:
        var: existing_entries.data
```

ansibleguy.opnsense.frr_ospf3_interface

```
- hosts: localhost
  gather_facts: no
  module_defaults:
    group/ansibleguy.opnsense.all:
      firewall: 'opnsense.template.ansibleguy.net'
      api_credential_file: '/home/guy/.secret/opn.key'

    ansibleguy.opnsense.frr_ospf3_interface:
      match_fields: ['interface']

    ansibleguy.opnsense.list:
      target: 'frr_ospf3_interface'

  tasks:
    - name: Example
      ansibleguy.opnsense.frr_ospf3_interface:
```

(continues on next page)

(continued from previous page)

```
      interface: 'opt1'
      # area: '0.0.0.0'
      # cost: 10
      # cost_demoted: 10
      # hello_interval: 10
      # dead_interval: 10
      # retransmit_interval: 10
      # transmit_delay: 10
      # priority: 10
      # network_type: ''
      # carp_depend_on: ''
      # passive: false
      # enabled: true
      # match_fields: ['interface', 'area']

  - name: Configuring interface
    ansibleguy.opnsense.frr_ospf3_interface:
      interface: 'opt1'
      area: '0.0.0.0'
      cost: 500
      cost_demoted: 2000
      hello_interval: 60
      dead_interval: 30
      retransmit_interval: 60
      transmit_delay: 60
      priority: 30
      network_type: 'point-to-point'

  - name: Disabling interface
    ansibleguy.opnsense.frr_ospf3_interface:
      interface: 'opt1'
      area: '0.0.0.0'
      cost: 500
      cost_demoted: 2000
      hello_interval: 60
      dead_interval: 30
      retransmit_interval: 60
      transmit_delay: 60
      priority: 30
      network_type: 'point-to-point'
      enabled: false

  - name: Pulling settings
    ansibleguy.opnsense.list:
    #  target: 'frr_ospf3_interface'
    register: existing_entries

  - name: Printing settings
    ansible.builtin.debug:
      var: existing_entries.data

  - name: Removing interface
```

(continues on next page)

(continued from previous page)

```
    ansibleguy.opnsense.frr_ospf3_interface:
      interface: 'opt1'
      state: 'absent'
```

CHAPTER

SEVENTEEN

FRR RIP

STATE: stable

TESTS: frr_rip

API Docs: Plugins - Quagga

Service Docs: Dynamic Routing

FRR Docs: FRRouting (*make sure you are looking at the current OPNSense package version!*)

17.1 Sponsoring

Thanks to @telmich for sponsoring the development of these modules!

17.2 Prerequisites

You need to install the FRR plugin:

```
os-frr
```

You can also install it using the package module.

17.3 Definition

For basic parameters see: Basics

17.3.1 ansibleguy.opnsense.frr_rip

Parameter	Type	Required	Default value	Aliases	Comment
version	integer	false	2	v	RIP version. 1 or 2
metric	integer	false	-	m	Default metric. Integer from 1 to 16
passive_int	list	false	-	passive_interf	Select the interfaces, where no RIP packets should be sent to
networks	list	false	-	nets	Enter your networks in CIDR notation
redistribute	list	false	-	-	Select other routing sources, which should be redistributed to the other nodes. One or more of: 'bgp', 'ospf', 'connected', 'kernel', 'static'

17.4 Examples

17.4.1 ansibleguy.opnsense.frr_rip

```
- hosts: localhost
  gather_facts: no
  module_defaults:
    group/ansibleguy.opnsense.all:
      firewall: 'opnsense.template.ansibleguy.net'
      api_credential_file: '/home/guy/.secret/opn.key'

    ansibleguy.opnsense.list:
      target: 'frr_rip'

  tasks:
    - name: Example
      ansibleguy.opnsense.frr_rip:
        # version: 2
        # metric: 10
        # passive_ints: []
        # redistribute: []
        # networks: []
        # enabled: true

    - name: Pulling settings
      ansibleguy.opnsense.list:
      #  target: 'frr_rip'
      register: existing_entries

    - name: Printing settings
```

(continues on next page)

(continued from previous page)

```
    ansible.builtin.debug:
      var: existing_entries.data

  - name: Enabling & Configuring RIP
    ansibleguy.opnsense.frr_rip:
      passive_ints: ['lan']
      redistribute: ['static']
      networks: ['10.0.10.0/24']
      enabled: true

  - name: Disabling RIP
    ansibleguy.opnsense.frr_rip:
      enabled: false
```

Tip: Check out the repository on GitHub

Report missing/incorrect information or broken links

CHAPTER

EIGHTEEN

INTRUSION PREVENTION SYSTEM

STATE: unstable

TESTS: ansibleguy.opnsense.ids_general | ansibleguy.opnsense.ids_action | ansibleguy.opnsense.ids_policy | ansibleguy.opnsense.ids_policy_rule | ansibleguy.opnsense.ids_rule | ansibleguy.opnsense.ids_ruleset | ansibleguy.opnsense.ids_user_rule

API Docs: IDS

Service Docs: Intrusion Prevention System

18.1 Definition

For basic parameters see: *Basic*

18.1.1 ansibleguy.opnsense.ids_action

Table 1: Definition

Parameter	Type	Required	Default	Aliases	Comment
action	string	true	-	do, a	Action to execute. One of: 'get_alert_info', 'get_alert_logs', 'query_alerts', 'status', 'reconfigure', 'restart', 'start', 'stop', 'drop_alert_log', 'reload_rules', 'update_rules'. These ones return information: 'get_alert_info', 'get_alert_logs', 'query_alerts', 'status'
alert_id	string	false	-	alert	Parameter Alert-ID needed for 'get_alert_info'

18.1.2 ansibleguy.opnsense.ids_general

Interfaces for 'ids_general' must be provided as used in the network config (*p.e. 'opt1' instead of 'DMZ'*)

Table 2: Definition

Parameter	Type	Required	Default	Aliases	Comment
interfaces	list	true	-	ints	Select interface(s) to use. When enabling IPS, only use physical interfaces here (no vlans etc)
enabled	boolean	false	true	-	Enable intrusion detection system
block	boolean	false	false	protection, ips	Enable protection mode (block traffic). Before enabling, please disable all hardware offloading first in advanced network!
promiscuous	boolean	false	-	physical, vlan	For certain setups (like IPS with vlans), this is required to actually capture data on the physical interface
default_packet_siz	int	false	(system default)	packet_siz	With this option, you can set the size of the packets on your network. It is possible that bigger packets have to be processed sometimes. The engine can still process these bigger packets, but processing it will lower the performance. Unset = system default
local_networks	list	false	['192.168. '10.0.0.0/8 '172.16.0.	home_net	Networks to interpret as local
pattern_matcher	string	false	(system default)	algorithm, matcher, algo	One of: 'ac', 'ac-bs', 'ac-ks', 'hs'. Select the multi-pattern matcher algorithm to use. Options: unset = system default, 'ac' = 'Aho-Corasick', 'ac-bs' = 'Aho-Corasick, reduced memory implementation', 'ac-ks' = 'Aho-Corasick, Ken Steele variant', 'hs' = 'Hyperscan'
profile	string	false	(system default)	detect_profil	One of: 'low', 'medium', 'high', 'custom'. The detection engine builds internal groups of signatures. The engine allow us to specify the profile to use for them, to manage memory on an efficient way keeping a good performance. Unset = system default
profile_toclient_gro	integer	true if profile = 'custom'	-	toclient_gro	Between 0 and 65535. If Custom is specified. The detection engine tries to split out separate signatures into groups so that a packet is only inspected against signatures that can actually match. As in large rule set this would result in way too many groups and memory usage similar groups are merged together
profile_toserver_gr	integer	true if profile = 'custom'	-	toserver_g	See 'profile_toclient_groups'
schedule	string	false	'ids rule updates'	update_cron	Name/Description of an existing cron-job that should be used to update IDS
syslog_alerts	boolean	false	-	syslog, log	Send alerts to system log in fast log format. This will not change the alert logging used by the product itself
syslog_output	boolean	false	-	log_stdout	Send alerts in eve format to syslog, using log level info. This will not change the alert logging used by the product itself. Drop logs will only be send to the internal logger, due to restrictions in suricata
log_level	string	false	(system default)	-	One of: 'info', 'perf', 'config', 'debug'. Increase the verbosity of the Suricata application logging by increasing the log level from the default. Unset = system default
log_retention	integer	false	4	log_count	Number of logs to keep
log_payload	boolean	false	-	log_packe	Send packet payload to the log for further analyses
log_rotate	string	false	weekly		One of: 'weekly', 'daily'. Rotate alert logs at pro-

18.1.3 ansibleguy.opnsense.ids_ruleset

The `reload` action will download/update the rulesets. If modifying multiple ones in a loop you might want to disable it on single calls.

Table 3: Definition

Parameter	Type	Required	Default	Aliases	Comment
name	string	true	-	description, desc	Name of the ruleset you want to modify. Will show a list of existing ones if an invalid one is supplied!
enabled	boolean	false	true	-	En- or disable the ruleset
reload	boolean	false	true	-	If the running config should be reloaded on change - this may take some time. For mass-managing items you might want to reload it 'manually' after all changes are done => using the *ansibleguy.opnsense.reload* module.

18.1.4 ansibleguy.opnsense.ids_rule

Table 4: Definition

Parameter	Type	Required	Default	Aliases	Comment
sid	integer	true	-	id	Unique signature-ID of the rule you want to modify
action	string	false	alert	a	One of 'alert', 'drop'. Set action to perform here, only used when in IPS mode
enabled	boolean	false	true	-	En- or disable the rule
reload	boolean	false	true	-	If the running config should be reloaded on change - this may take some time. For mass-managing items you might want to reload it 'manually' after all changes are done => using the *ansibleguy.opnsense.reload* module.

18.1.5 ansibleguy.opnsense.ids_user_rule

Table 5: Definition

Parameter	Type	Required	Default	Aliases	Comment
name	string	true	-	description, desc	Unique rule name
source_ip	string	false	-	source, src_ip, src	Set the source IP or network to match. Leave this field empty for using 'any'
destination_ip	string	false	-	destination, dst_ip, dst	Set the destination IP or network to match. Leave this field empty for using 'any'
ssl_fingerprint	string	false	-	fingerprint, ssl_fp	The SSL fingerprint, for example: 'B5:E1:B3:70:5E:7C:FF:EB:92:C4:29:E5:5B:AC:2F:AE:70:17:E9:9
action	string	false	alert	a"	One of 'alert', 'drop', 'pass'. Set action to perform here, only used when in IPS mode
bypass	boolean	false	false	bp	Set bypass keyword. Increases traffic throughput. Suricata reads a packet, decodes it, checks it in the flow table. If the corresponding flow is local bypassed then it simply skips all streaming, detection and output and the packet goes directly out in IDS mode and to verdict in IPS mode
enabled	boolean	false	true	-	En- or disable the rule
reload	boolean	false	true	-	If the running config should be reloaded on change - this may take some time. For mass-managing items you might want to reload it 'manually' after all changes are done => using the *ansibleguy.opnsense.reload* module.

18.1.6 ansibleguy.opnsense.ids_policy

Table 6: Definition

Parameter	Type	Required	Default	Aliases	Comment
name	string	true	-	description, desc	Unique policy name
priority	integer	false	0	prio	Policies are processed on a first match basis a lower number means more important
rulesets	list	false	-	rs	Rulesets this policy applies to (all when none selected). Rulesets must be enabled beforehand!
action	list	false	-	a	One or multiple of: 'disable', 'alert', 'drop'. Rule configured action
new_action	string	false	alert	na	One or multiple of: 'default', 'disable', 'alert', 'drop'. Action to perform when filter policy applies
rules	dictionary	false	-	-	Key-value pairs of policy-rules as provided by the enabled rulesets. Values must be string or lists. Example: '{rules: {signature_severity: [Minor, Major], tag: Dshield}}'
enabled	boolean	false	true	-	En- or disable the Policy
reload	boolean	false	true	-	If the running config should be reloaded on change - this may take some time. For mass-managing items you might want to reload it 'manually' after all changes are done => using the *ansibleguy.opnsense.reload* module.

18.1.7 ansibleguy.opnsense.ids_policy_rule

Table 7: Definition

Parameter	Type	Required	Default	Aliases	Comment
sid	integer	true	-	id	Unique signature-ID of the rule you want to match
action	string	false	alert	a	One of: 'alert', 'drop'. Rule configured action
enabled	boolean	false	true	-	En- or disable the rule
reload	boolean	false	true	-	If the running config should be reloaded on change - this may take some time. For mass-managing items you might want to reload it 'manually' after all changes are done => using the *ansibleguy.opnsense.reload* module.

18.2 Info

Warning: The `list` module will not return all details of the existing entries as the current implementation does not scale well.

18.3 Examples

18.3.1 ansibleguy.opnsense.ids_action

```
- hosts: localhost
  gather_facts: false
  module_defaults:
    group/ansibleguy.opnsense.all:
      firewall: 'opnsense.template.ansibleguy.net'
      api_credential_file: '/home/guy/.secret/opn.key'

  tasks:
    - name: Example
      ansibleguy.opnsense.ids_action:
        action: 'status'
        # alert_id: ''
        # debug: false

    - name: Pull Alert Logs
      ansibleguy.opnsense.ids_action:
        action: 'get_alert_logs'
      register: ids_logs

    - name: Printing
      ansible.builtin.debug:
        var: ids_logs.data

    - name: Reload Rules
      ansibleguy.opnsense.ids_action:
        action: 'reload_rules'

    - name: Update Rules
      ansibleguy.opnsense.ids_action:
        action: 'update_rules'

    - name: Pull Alert Information
      ansibleguy.opnsense.ids_action:
        action: 'get_alert_info'
        alert_id: 1337
      register: ids_alert

    - name: Printing
      ansible.builtin.debug:
        var: ids_alert.data
```

18.3.2 ansibleguy.opnsense.ids_general

```yaml
- hosts: localhost
  gather_facts: false
  module_defaults:
    group/ansibleguy.opnsense.all:
      firewall: 'opnsense.template.ansibleguy.net'
      api_credential_file: '/home/guy/.secret/opn.key'

    ansibleguy.opnsense.list:
      target: 'ids_general'

  tasks:
    - name: Example
      ansibleguy.opnsense.ids_general:
        interfaces: ['opt1']
        # enabled: true
        # block: true
        # promiscuous: false
        # default_packet_size: ''
        # local_networks: ['192.168.0.0/16', '10.0.0.0/8', '172.16.0.0/12']
        # pattern_matcher: ''
        # profile: ''
        # profile_toclient_groups: ''
        # profile_toserver_groups: ''
        # schedule: 'ids rule updates'
        # syslog_alerts: false
        # syslog_output: false
        # log_level: ''
        # log_retention: 4
        # log_payload: false
        # log_rotate: 'weekly'
        # reload: true
        # debug: false

    - name: Enabling IDS (learning mode)
      ansibleguy.opnsense.ids_general:
        interfaces: ['opt1']
        enabled: true
        pattern_matcher: 'ac'
        profile: 'low'
        local_networks: ['10.0.0.0/16']
        log_rotate: 'daily'
        log_retention: 14
        syslog: true
        log_level: 'info'

    - name: Enabling IPS (blocking)
      ansibleguy.opnsense.ids_general:
        interfaces: ['opt1']
        enabled: true
        block: true
        pattern_matcher: 'ac'
```

(continues on next page)

(continued from previous page)

```
        profile: 'low'
        local_networks: ['10.0.0.0/16']
        log_rotate: 'daily'
        log_retention: 14
        syslog: true
        log_level: 'info'

    - name: Listing
      ansibleguy.opnsense.list:
      #  target: 'ids_general'
      register: existing_settings

    - name: Printing Settings
      ansible.builtin.debug:
        var: existing_settings.data
```

18.3.3 ansibleguy.opnsense.ids_ruleset

```
- hosts: localhost
  gather_facts: false
  module_defaults:
    group/ansibleguy.opnsense.all:
      firewall: 'opnsense.template.ansibleguy.net'
      api_credential_file: '/home/guy/.secret/opn.key'

    ansibleguy.opnsense.list:
      target: 'ids_ruleset'

  tasks:
    - name: Example
      ansibleguy.opnsense.ids_ruleset:
        name: 'ET open/drop'
        # enabled: true
        # reload: true
        # debug: false

    - name: Enabling & downloading ruleset 'ET open/drop'
      ansibleguy.opnsense.ids_ruleset:
        name: 'ET open/compromised'
        reload: true

    - name: Disabling ruleset 'ET open/compromised'
      ansibleguy.opnsense.ids_ruleset:
        name: 'ET open/compromised'
        enabled: false

    - name: Listing
      ansibleguy.opnsense.list:
      #  target: 'ids_ruleset'
      register: existing_rulesets
```

(continues on next page)

(continued from previous page)

```
    - name: Printing Rulesets
      ansible.builtin.debug:
        var: existing_rulesets.data
```

18.3.4 ansibleguy.opnsense.ids_rule

```
- hosts: localhost
  gather_facts: false
  module_defaults:
    group/ansibleguy.opnsense.all:
      firewall: 'opnsense.template.ansibleguy.net'
      api_credential_file: '/home/guy/.secret/opn.key'

    ansibleguy.opnsense.list:
      target: 'ids_rule'

  tasks:
    - name: Example
      ansibleguy.opnsense.ids_rule:
        sid: 2400000
        # enabled: true
        # action: 'alert'
        # reload: true
        # debug: false

    - name: Setting rule with ID 2400000 to drop
      ansibleguy.opnsense.ids_rule:
        sid: 2400000
        action: 'drop'

    - name: Disabling rule with ID 2400011
      ansibleguy.opnsense.ids_rule:
        sid: 2400011
        enabled: false

    - name: Listing
      ansibleguy.opnsense.list:
      #  target: 'ids_rule'
      register: existing_rules

    - name: Printing Rules
      ansible.builtin.debug:
        var: existing_rules.data
```

18.3.5 ansibleguy.opnsense.ids_user_rule

```yaml
- hosts: localhost
  gather_facts: false
  module_defaults:
    group/ansibleguy.opnsense.all:
      firewall: 'opnsense.template.ansibleguy.net'
      api_credential_file: '/home/guy/.secret/opn.key'

    ansibleguy.opnsense.list:
      target: 'ids_user_rule'

  tasks:
    - name: Example
      ansibleguy.opnsense.ids_user_rule:
        name: 'Example'
        # source_ip: ''
        # destination_ip: ''
        # ssl_fingerprint: ''
        # action: 'alert'
        # bypass: false
        # enabled: true
        # reload: true
        # debug: false

    - name: Adding
      ansibleguy.opnsense.ids_user_rule:
        name: 'ANSIBLE_TEST_1_1'
        source_ip: '192.168.10.1'
        destination_ip: '1.1.1.1'
        action: 'alert'
        bypass: false

    - name: Disabling
      ansibleguy.opnsense.ids_user_rule:
        name: 'ANSIBLE_TEST_1_1'
        source_ip: '192.168.10.1'
        destination_ip: '1.1.1.1'
        action: 'alert'
        bypass: false
        enabled: false

    - name: Removing
      ansibleguy.opnsense.ids_user_rule:
        name: 'ANSIBLE_TEST_1_1'
        state: 'absent'

    - name: Listing
      ansibleguy.opnsense.list:
      #  target: 'ids_user_rule'
      register: existing_rules

    - name: Printing Rules
```

(continues on next page)

(continued from previous page)

```
      ansible.builtin.debug:
        var: existing_rules.data
```

18.3.6 ansibleguy.opnsense.ids_policy

```
- hosts: localhost
  gather_facts: false
  module_defaults:
    group/ansibleguy.opnsense.all:
      firewall: 'opnsense.template.ansibleguy.net'
      api_credential_file: '/home/guy/.secret/opn.key'

    ansibleguy.opnsense.list:
      target: 'ids_policy'

  tasks:
    - name: Example
      ansibleguy.opnsense.ids_policy:
        name: 'Example'
        # priority: 0
        # rulesets: []
        # action: []
        # new_action: 'alert'
        # rules: {}
        # enabled: true
        # reload: true
        # debug: false

    - name: Adding
      ansibleguy.opnsense.ids_policy:
        name: 'ANSIBLE_TEST_1_1'
        priority: 1
        rulesets: 'ET open/drop'
        action: ['drop']
        new_action: 'alert'
        rules:
          classtype: ['misc-attack', 'bad-unknown']
          signature_severity: 'Minor'

    - name: Disabling
      ansibleguy.opnsense.ids_policy:
        name: 'ANSIBLE_TEST_1_1'
        priority: 1
        rulesets: 'ET open/drop'
        action: ['drop']
        new_action: 'alert'
        rules:
          classtype: ['misc-attack', 'bad-unknown']
          signature_severity: 'Minor'
        enabled: false
```

(continues on next page)

(continued from previous page)

```
    - name: Removing
      ansibleguy.opnsense.ids_policy:
        name: 'ANSIBLE_TEST_1_1'
        state: 'absent'

    - name: Listing
      ansibleguy.opnsense.list:
      #  target: 'ids_policy'
      register: existing_policies

    - name: Printing Policies
      ansible.builtin.debug:
        var: existing_policies.data
```

18.3.7 ansibleguy.opnsense.ids_policy_rule

```
- hosts: localhost
  gather_facts: false
  module_defaults:
    group/ansibleguy.opnsense.all:
      firewall: 'opnsense.template.ansibleguy.net'
      api_credential_file: '/home/guy/.secret/opn.key'

    ansibleguy.opnsense.list:
      target: 'ids_policy_rule'

  tasks:
    - name: Example
      ansibleguy.opnsense.ids_policy_rule:
        sid: 2400000
        # action: 'alert'
        # enabled: true
        # reload: true
        # debug: false

    - name: Adding
      ansibleguy.opnsense.ids_policy_rule:
        sid: 2400000
        action: 'alert'

    - name: Disabling
      ansibleguy.opnsense.ids_policy_rule:
        sid: 2400000
        action: 'alert'
        enabled: false

    - name: Removing
      ansibleguy.opnsense.ids_policy_rule:
        sid: 2400000
```

(continues on next page)

(continued from previous page)

```
      state: 'absent'

  - name: Listing
    ansibleguy.opnsense.list:
    #  target: 'ids_policy_rule'
    register: existing_rules

  - name: Printing Rules
    ansible.builtin.debug:
      var: existing_rules.data
```

Tip: Check out the repository on GitHub

Report missing/incorrect information or broken links

CHAPTER

NINETEEN

INTERFACE

STATE: stable

TESTS: vlan | vxlan | vip

API Docs: Core - Interfaces

Service Docs: VLAN Docs | VxLAN Docs | VIP Docs

19.1 Info

19.1.1 ansibleguy.opnsense.interface_vlan

This module manages VLAN configuration that can be found in the WEB-UI menu: 'Interfaces - Other Types - VLAN'

19.1.2 ansibleguy.opnsense.interface_vxlan

This module manages VXLAN configuration that can be found in the WEB-UI menu: 'Interfaces - Other Types - VXLAN'

19.1.3 ansibleguy.opnsense.interface_vip

This module manages VIP configuration that can be found in the WEB-UI menu: 'Interfaces - Virtual IPs - Settings'

19.2 Definition

For basic parameters see: *Basic*

19.2.1 ansibleguy.opnsense.interface_vlan

Table 1: Definition

Parameter	Type	Required	Default	Aliases	Comment
description	string	true	-	desc, name	The unique description used to match the configured entries to the existing ones
interface	string	false for state changes, else true	-	parent, port, int, if	The parent interface to add the vlan to. Existing VLAN capable interface - you must provide the network port as shown in 'Interfaces - Assignments - Network port'
vlan	integer	false for state changes, else true	-	tag, id	802.1Q VLAN tag (between 1 and 4094)
priority	integer	false	0	prio	802.1Q VLAN PCP (between 0 and 7)
reload	boolean	false	true	-	If the running config should be reloaded on change - this may take some time. For mass-managing items you might want to reload it 'manually' after all changes are done => using the *ansibleguy.opnsense.reload* module.

19.2.2 ansibleguy.opnsense.interface_vxlan

Table 2: Definition

Parameter	Type	Required	Default	Aliases	Comment
id	integer	true	-	vxlanid, vni	The unique ID of the VxLAN
interface	string	false for state changes, else true	-	vxlan-dev, device, int	Optionally set an interface to bind the VxLAN to. You must provide the network port as shown in 'Interface - Assignments - Interface ID (in brackets)'
local	string	false for state changes, else true	-	source_ad source_ip, vxlan-local, source, src	Source IP for the VxLAN tunnel. The source address used in the encapsulating IPv4/IPv6 header. The address should already be assigned to an existing interface. When the interface is configured in unicast mode, the listening socket is bound to this address.
remote	string	false	-	remote_addı remote_ip, destination, vxlan-remote, dest	Remote IP for the VxLAN tunnel - if unicast is used. The interface can be configured in a unicast, or point-to-point, mode to create a tunnel between two hosts. This is the IP address of the remote end of the tunnel.
group	string	false	-	multicast_grouı multicast_addre multicast_ip, vxlan-group	Remote IP for the VxLAN tunnel - if multicast is used. The interface can be configured in a multicast mode to create a virtual network of hosts. This is the IP multicast group address the interface will join.
reload	boolean	false	true	-	If the running config should be reloaded on change - this may take some time. For mass-managing items you might want to reload it 'manually' after all changes are done => using the *ansibleguy.opnsense.reload* module.

19.2.3 ansibleguy.opnsense.interface_vip

Warning: This feature is only available in OPNSense version >= 23.1

Table 3: Definition

Parameter	Type	Required	Default	Aliases	Comment
match_fields	list	false	['address', 'interface']	-	Fields that are used to match configured VIPs with the running config - if any of those fields are changed, the module will think it's a new entry. At least one of: 'address', 'interface', 'cidr', 'description'
address	string	true	-	addr, ip, network, net	Provide an address and subnet to use. (e.g 192.168.0.1/24)
interface	string	true	-	port, int, if	Existing interface - you must provide the network port as shown in 'Interfaces - Assignments - Network port'
mode	string	false	ipalias	m	One of: 'ipalias', 'carp', 'proxyarp', 'other'
expand	boolean	false	true	-	-
bind	boolean	false	true	-	Assigning services to the virtual IP's interface will automatically include this address. Uncheck to prevent binding to this address instead
gateway	string	false	-	gw	For some interface types a gateway is required to configure an IP Alias (ppp/pppoe/tun), leave this field empty for all other interface types
password	string	false	-	pwd	VHID group password
vhid	integer	false	-	group, grp, id	VHID group that the machines will share
advertising_base	integer	false	1	adv_base, base	The frequency that this machine will advertise. 0 usually means master. Otherwise the lowest combination of both values in the cluster determines the master
advertising_skew	integer	false	0	adv_skew, skew	-
description	string	false	-	desc, name	Optional description
reload	boolean	false	true	-	If the running config should be reloaded on change - this may take some time. For mass-managing items you might want to reload it 'manually' after all changes are done => using the *ansibleguy.opnsense.reload* module.

19.3 Examples

19.3.1 ansibleguy.opnsense.interface_vlan

```yaml
- hosts: localhost
  gather_facts: no
  module_defaults:
    group/ansibleguy.opnsense.all:
      firewall: 'opnsense.template.ansibleguy.net'
      api_credential_file: '/home/guy/.secret/opn.key'

    ansibleguy.opnsense.list:
      target: 'interface_vlan'

  tasks:
    - name: Example
      ansibleguy.opnsense.interface_vlan:
        description: 'example'
        interface: 'vtnet0'
        vlan: 100
        # priority: 0
        # debug: false
        # state: 'present'
        # reload: true

    - name: Adding VLAN
      ansibleguy.opnsense.interface_vlan:
        description: 'test1'
        interface: 'vtnet0'
        vlan: 100

    - name: Listing
      ansibleguy.opnsense.list:
      #  target: 'interface_vlan'
      register: existing_entries

    - name: Printing VLANs
      ansible.builtin.debug:
        var: existing_entries.data

    - name: Removing VLAN
      ansibleguy.opnsense.interface_vlan:
        description: 'test1'
        state: 'absent'
```

19.3.2 ansibleguy.opnsense.interface_vxlan

```yaml
- hosts: localhost
  gather_facts: no
  module_defaults:
    group/ansibleguy.opnsense.all:
      firewall: 'opnsense.template.ansibleguy.net'
      api_credential_file: '/home/guy/.secret/opn.key'

    ansibleguy.opnsense.list:
      target: 'interface_vxlan'

  tasks:
    - name: Example
      ansibleguy.opnsense.interface_vxlan:
        id: 100
        local: '192.168.0.1'
        # remote: ''
        # group: ''
        # interface: 'lan'
        # debug: false
        # state: 'present'
        # reload: true

    - name: Adding VxLAN
      ansibleguy.opnsense.interface_vxlan:
        id: 100
        local: '192.168.0.1'
        interface: 'lan'

    - name: Listing
      ansibleguy.opnsense.list:
      #  target: 'interface_vxlan'
      register: existing_entries

    - name: Printing VxLANs
      ansible.builtin.debug:
        var: existing_entries.data

    - name: Removing VxLAN
      ansibleguy.opnsense.interface_vxlan:
        id: 100
        state: 'absent'
```

19.3.3 ansibleguy.opnsense.interface_vip

```yaml
- hosts: localhost
  gather_facts: no
  module_defaults:
    group/ansibleguy.opnsense.all:
      firewall: 'opnsense.template.ansibleguy.net'
      api_credential_file: '/home/guy/.secret/opn.key'

    ansibleguy.opnsense.list:
      target: 'interface_vip'

  tasks:
    - name: Example
      ansibleguy.opnsense.interface_vip:
        interface: 'opt1'
        address: '192.168.0.100/24'
        # match_fields: ['address', 'interface]
        # mode: 'ipalias'
        # expand: true
        # bind: true
        # gateway: ''
        # password: ''
        # vhid: ''
        # advertising_base: 1
        # advertising_skew: 0
        # description: ''
        # debug: false
        # state: 'present'
        # reload: true

    - name: Adding VIP
      ansibleguy.opnsense.interface_vip:
        interface: 'opt1'
        address: '192.168.0.100/24'
        mode: 'carp'
        vhid: 10
        password: 'secret'

    - name: Listing
      ansibleguy.opnsense.list:
      #  target: 'interface_vip'
      register: existing_entries

    - name: Printing VIPs
      ansible.builtin.debug:
        var: existing_entries.data

    - name: Removing VIP
      ansibleguy.opnsense.interface_vip:
        interface: 'opt1'
        address: '192.168.0.100/24'
        state: 'absent'
```

Tip: Check out the repository on GitHub

Report missing/incorrect information or broken links

CHAPTER

TWENTY

IPSEC

STATE: stable

TESTS: ipsec_cert | ipsec_psk | ipsec_connection | ipsec_pool | ipsec_vti

API Docs: Core - IPSec

Service Docs: IPSec | IPSec Examples | IPSec VTI

20.1 Limitations

> **Warning:** The IPSec modules can only be used on OPNSense version >= 23.1

20.2 Definition

For basic parameters see: *Basic*

20.2.1 ansibleguy.opnsense.ipsec_connection

Module alias: ansibleguy.opnsense.ipsec_tunnel

Table 1: Definition

Parameter	Type	Required	Default	Aliases	Comment
name	string	true	-	description, desc	Unique connection/tunnel name
local_addresses	list	false	-	local_addr, local	Local address[es] to use for IKE communication. Accepts single IPv4/IPv6 addresses, DNS names, CIDR subnets or IP address ranges. As an initiator, the first non-range/non-subnet is used to initiate the connection from. As a responder the local destination address must match at least to one of the specified addresses, subnets or ranges. If FQDNs are assigned, they are resolved every time a configuration lookup is done. If DNS resolution times out, the lookup is delayed for that time. When left empty %any is choosen as default
remote_addresses	list	false	-	remote_addr, remote	Remote address[es] to use for IKE communication. Accepts single IPv4/IPv6 addresses, DNS names, CIDR subnets or IP address ranges. As an initiator, the first non-range/non-subnet is used to initiate the connection to. As a responder the local destination address must match at least to one of the specified addresses, subnets or ranges. If FQDNs are assigned, they are resolved every time a configuration lookup is done. If DNS resolution times out, the lookup is delayed for that time. To initiate a connection, at least one specific address or DNS name must be specified
pools	list	false	-	networks	List of named IP pools to allocate virtual IP addresses and other configuration attributes from. Each name references a pool by name from either the pools section or an external pool. Note that the order in which they are queried primarily depends on the plugin order
proposals	list	false	['default']	props	A proposal is a set of algorithms. For non-AEAD algorithms this includes IKE an encryption algorithm, an integrity algorithm, a pseudo random function (PRF) and a Diffie-Hellman key exchange group. For AEAD algorithms, instead of encryption and integrity algorithms a combined algorithm is used. With IKEv2 multiple algorithms of the same kind can be specified in a single proposal, from which one gets selected. For IKEv1 only one algorithm per kind is allowed per proposal, more algorithms get implicitly stripped. Use multiple proposals to offer different algorithm combinations with IKEv1. Algorithm keywords get separated using dashes. Multiple proposals may be separated by commas. The special value default adds a default proposal of supported algorithms considered safe and is usually a good choice for interoperability.
unique	string	false	no	-	One of: 'no', 'never', 'keep', 'replace'; Connection uniqueness policy to enforce. To avoid multiple connections from the same user, a uniqueness policy can be enforced.
aggressive	boolean	false	false	aggr	Enables IKEv1 Aggressive Mode instead of IKEv1 Main Mode with Identity Protection. Aggressive

20.2.2 ansibleguy.opnsense.ipsec_pool

Module alias: ansibleguy.opnsense.ipsec_network

Table 2: Definition

Parameter	Type	Required	Default	Aliases	Comment
name	string	true	-	-	Unique pool/network name
network	string	false for state changes, else true	-	net, cidr	Pool network in CIDR format
reload	boolean	false	true	-	If the running config should be reloaded on change - this may take some time. For mass-managing items you might want to reload it 'manually' after all changes are done => using the *ansibleguy.opnsense.reload* module.

20.2.3 ansibleguy.opnsense.ipsec_child

Table 3: Definition

Parameter	Type	Required	Default	Aliases	Comment
name	string	true	-	description, desc	Unique name to identify the entry
connection	string	false for state changes, else true	-	tunnel, conn, tun	Connection to link this child to
mode	string	false	tunnel	-	One of: 'tunnel', 'transport', 'pass', 'drop'; IPsec Mode to establish CHILD_SA with. tunnel negotiates the CHILD_SA in IPsec Tunnel Mode whereas transport uses IPsec Transport Mode. pass and drop are used to install shunt policies which explicitly bypass the defined traffic from IPsec processing or drop it, respectively
local_net	list	true	-	local_traffic, local_cidr, local_ts, local	List of local traffic selectors to include in CHILD_SA. Each selector is a CIDR subnet definition
remote_net	list	true	-	remote_traff remote_cidr remote_ts, remote	List of remote traffic selectors to include in CHILD_SA. Each selector is a CIDR subnet definition
sha256_96	boolean	false	false	sha256	HMAC-SHA-256 is used with 128-bit truncation with IPsec. For compatibility with implementations that incorrectly use 96-bit truncation this option may be enabled to configure the shorter truncation length in the kernel. This is not negotiated, so this only works with peers that use the incorrect truncation length (or have this option enabled)
start_action	string	false	start	start	One of: 'none', 'trap|start', 'route', 'start', 'trap'; Action to perform after loading the configuration. The default of none loads the connection only, which then can be manually initiated or used as a responder configuration. The value trap installs a trap policy which triggers the tunnel as soon as matching traffic has been detected. The value start initiates the connection actively. To immediately initiate a connection for which trap policies have been installed, user Trap|start
close_action	string	false	none	close	One of: 'none', 'trap', 'start'; Action to perform after a CHILD_SA gets closed by the peer. The default of none does not take any action. trap installs a trap policy for the CHILD_SA (note that this is redundant if start_action includes trap). start tries to immediately re-create the CHILD_SA. close_action does not provide any guarantee that the CHILD_SA is kept alive. It acts on explicit close messages only but not on negotiation failures. Use trap policies to reliably re-create failed

20.2.4 ansibleguy.opnsense.ipsec_vti

Table 4: Definition

Parameter	Type	Required	Default	Aliases	Comment
name	string	true	-	description, desc	Unique name to identify the entry
request_id	integer	false for state changes, else true	-	req_id, reqid	This might be helpful in some scenarios, like route based tunnels (VTI), but works only if each CHILD_SA configuration is instantiated not more than once. The default uses dynamic reqids, allocated incrementally
local_address	string	false	-	local_addr, local	-
remote_address	string	false	-	remote_addı remote	-
local_tunnel_addr	string	false	-	local_tun_ac tunnel_local, local_tun	-
remote_tunnel_ad	string	false	-	remote_tun_ tunnel_remot remote_tun	-
reload	boolean	false	true	-	If the running config should be reloaded on change - this may take some time. For mass-managing items you might want to reload it 'manually' after all changes are done => using the *ansibleguy.opnsense.reload* module.

20.2.5 ansibleguy.opnsense.ipsec_auth_local

20.2.6 ansibleguy.opnsense.ipsec_auth_remote

See: ansibleguy.opnsense.ipsec_auth_local

20.2.7 ansibleguy.opnsense.ipsec_cert

Table 5: Definition

Parameter	Type	Required	Default	Aliases	Comment
name	string	true	-	-	Name of the key-pair - used to identify the entry.
public_key	string	false for state changes, else true	-	pub_key, pub	-
private_key	string	false for state changes, else true	-	priv_key, priv	-
type	string	false	rsa	-	Type of the key. Currently the only option is 'rsa'
reload	boolean	false	true	-	If the running config should be reloaded on change - this may take some time. For mass-managing items you might want to reload it 'manually' after all changes are done => using the *ansibleguy.opnsense.reload* module.

20.2.8 ansibleguy.opnsense.ipsec_psk

Table 6: Definition

Parameter	Type	Required	Default	Aliases	Comment
identity_local	string	true	-	identity, ident	This can be either an IP address, fully qualified domain name or an email address.
identity_remote	string	false	-	remote_iden	(optional) This can be either an IP address, fully qualified domain name or an email address to identify the remote host.
psk	string	true	-	key, secret	-
type	string	false	-	kind	One of: 'PSK', 'EAP'

20.3 Usage

To apply changes to the keys, you need to set 'reload: true' on each call or use the *ansibleguy.opnsense.reload* module to apply it once you finished modifying all entries!

As far as I can tell - the IPSec service gets restarted one you do so - be aware of that.

20.3.1 Vault

You may want to use '**ansible-vault**' to **encrypt** your 'private_key' content!

```
ansible-vault encrypt_string '-----BEGIN RSA PRIVATE KEY-----\n...-----END RSA PRIVATE␣
↪KEY-----\n'

# or encrypt the private_key file beforehand (might be easier)
ansible-vault encrypt /path/to/private/key/file.pem
```

20.4 Examples

20.4.1 ansibleguy.opnsense.ipsec_cert

```
- hosts: localhost
  gather_facts: no
  module_defaults:
    group/ansibleguy.opnsense.all:
      firewall: 'opnsense.template.ansibleguy.net'
      api_credential_file: '/home/guy/.secret/opn.key'

    ansibleguy.opnsense.list:
      target: 'ipsec_cert'

  tasks:
    - name: Example
      ansibleguy.opnsense.ipsec_cert:
        name: 'example'
        public_key: |
          -----BEGIN PUBLIC KEY-----
          ...
          -----END PUBLIC KEY-----
        private_key: |
          -----BEGIN RSA PRIVATE KEY-----
          ...
          -----END RSA PRIVATE KEY-----

        # reload: false

    - name: Adding key-pair and applying it
      ansibleguy.opnsense.ipsec_cert:
        name: 'test1'
        public_key: |
          -----BEGIN PUBLIC KEY-----
          ...
          -----END PUBLIC KEY-----
        private_key: !vault ...
        reload: true

    - name: Listing
      ansibleguy.opnsense.list:
```

(continues on next page)

(continued from previous page)

```
      #  target: 'ipsec_cert'
      no_log: true  # could log private keys
      register: existing_entries

    - name: Printing Certificates
      ansible.builtin.debug:
        var: existing_entries.data

    - name: Manually reloading/applying config
      ansibleguy.opnsense.reload:
        target: 'ipsec'
```

20.4.2 ansibleguy.opnsense.ipsec_psk

```
- hosts: localhost
  gather_facts: no
  module_defaults:
    group/ansibleguy.opnsense.all:
      firewall: 'opnsense.template.ansibleguy.net'
      api_credential_file: '/home/guy/.secret/opn.key'

    ansibleguy.opnsense.list:
      target: 'ipsec_psk'

  tasks:
    - name: Example
      ansibleguy.opnsense.ipsec_psk:
        identity: 'example'
        psk: 'secret'
        # type: 'PSK'
        # identity_remote: ''

    - name: Adding
      ansibleguy.opnsense.ipsec_psk:
        identity: 'test1'
        psk: 'my-super-secret'

    - name: Removing
      ansibleguy.opnsense.ipsec_psk:
        identity: 'test1'
        state: 'absent'
```

CHAPTER

TWENTYONE

MONIT

STATE: stable

TESTS: Playbook

API Docs: Core - Monit

Service Docs: Monit

21.1 Info

For mail alerts to work:

- Don't forget to configure your mailing settings at the general monit page
- You will also need to set your sender-mail address in the 'format' field using the 'monit_alert' module. See the examples below.

Interfaces for 'monit_services' must be provided as used in the network config (*p.e. 'opt1' instead of 'DMZ'*)

- per example see menu: 'Interface - Assignments - Interface ID (in brackets)'
- this brings problems if the interface-names are not the same on both nodes when using HA-setups

21.2 Definition

For basic parameters see: Basics

21.2.1 ansibleguy.opnsense.monit_alert

Parameter	Type	Require	Default value	Alias	Comment
recipient	string	true	-	email mail	Mail address to send the alert to
not_c	boole	false	true	not	Do not send alerts for the following events but on all others
event	list	false	-	-	Filter event-types to alert on. Invertable using the 'not_on' parameter. One or multiple of: 'action', 'checksum', 'bytein', 'byteout', 'connection', 'content', 'data', 'exec', 'fsflags', 'gid', 'icmp', 'instance', 'invalid', 'link', 'nonexist', 'packetin', 'packetout', 'permission', 'pid', 'ppid', 'resource', 'saturation', 'size', 'speed', 'status', 'timeout', 'timestamp', 'uid', 'uptime'
format	string	false	-	-	The email format for alerts. Subject: $SERVICE on $HOST failed. "Mail format" is a newline-separated list of properties to control the mail formatting. It is also needed to correctly set the From address
remind	int	false	10	-	Send a reminder after some cycles. Integer between 0 and 86400
description	string	false	-	desc	Send a reminder after some cycles
matcl	string	false	['recipient']	-	Fields that are used to match configured alerts with the running config - if any of those fields are changed, the module will think it's a new entry. At least one of: 'recipient', 'not_on', 'events', 'reminder', 'description'

21.2.2 ansibleguy.opnsense.monit_test

Parameter	Type	Required	Default value	Alias	Comment
name	string	true	-	-	Unique name of the test
type	string	false	'Custom'	-	Type of test. 'Custom' will not be idempotent - will be translated on the server-side. See 'list' module output for details. One of: 'Existence', 'SystemResource', 'ProcessResource', 'ProcessDiskIO', 'FileChecksum', 'Timestamp', 'FileSize', 'FileContent', 'FilesystemMountFlags', 'SpaceUsage', 'InodeUsage', 'DiskIO', 'Permisssion', 'UID', 'GID', 'PID', 'PPID', 'Uptime', 'ProgramStatus', 'NetworkInterface', 'NetworkPing', 'Connection', 'Custom'
condition	string	false for state changes, else true	-	-	The test condition. Per example: 'cpu is greater than 50%' or 'failed host 127.0.0.1 port 22 protocol ssh'
action	string	false for state changes, else true	'alert'	-	One of: 'alert', 'restart', 'start', 'stop', 'exec', 'unmonitor'
path	path	false, true if present and type is 'execute'	-	-	The absolute path to the script to execute - if action is set to 'execute'. Make sure the script is executable by the Monit service

21.2.3 ansibleguy.opnsense.monit_service

Parameter	Type	Required	Default value	Aliases	Comment
name	string	true	-	-	Unique service name
type	string	false for state changes, else true	-	-	One of: 'process', 'file', 'fifo', 'filesystem', 'directory', 'host', 'system', 'custom', 'network'
pidfile	path	false	-	-	
match	string	false	-	-	
path	path	false	-	-	According to the service type path can be a file or a directory
service_tim	path	false	-	svc_ti	Integer between 1 and 86400
address	string	false, true if type is one of 'network', 'host'	-	-	The target IP address for 'host' and 'network' checks
interface	string	false, true if type is one of 'network'	-	-	The existing Interface for 'Network' checks. Alternative to 'address'
start	string	false	-	-	Absolute path to the executable with its arguments to run at service-start
stop	string	false	-	-	Absolute path to the executable with its arguments to run at service-stop
tests	list	false	-	-	Name of tests to link to the service. Not all test-types are compatible with all service-types
depends	list	false	-	-	Optionally define a (list of) service(s) which are required before monitoring this one, if any of the dependencies are either stopped or unmonitored this service will stop/unmonitor too
polltime	string	false	-	-	Set the service poll time. Either as a number of cycles 'NUMBER CYCLES' or Cron-style '* 8-19 * * 1-5'
description	string	false	-	-	

21.3 Examples

21.3.1 Alerts

```
- hosts: localhost
  gather_facts: no
  module_defaults:
    group/ansibleguy.opnsense.all:
      firewall: 'opnsense.template.ansibleguy.net'
      api_credential_file: '/home/guy/.secret/opn.key'
```

(continues on next page)

(continued from previous page)

```
    ansibleguy.opnsense.list:
      target: 'monit_alert'

  tasks:
    - name: Example
      ansibleguy.opnsense.monit_alert:
        recipient: 'monit-alert@template.ansibleguy.net'
        # not_on: false
        # events: []
        # format: ''
        # reminder: 10
        # description: 'example'
        # match_fields: ['recipient']
        # enabled: true
        # reload: true

    - name: Adding simple
      ansibleguy.opnsense.monit_alert:
        recipient: 'monit-alert@template.ansibleguy.net'

    - name: Changing
      ansibleguy.opnsense.monit_alert:
        recipient: 'monit-alert@template.ansibleguy.net'
        format: |
          From: monit-alert@template.ansibleguy.net
          Reply-To: netmaster@template.ansibleguy.net
          Subject: $SERVICE at $HOST failed
        not_on: true
        events: ['timestamp']
        description: 'alert1'
        reminder: 500

    - name: Disabling
      ansibleguy.opnsense.monit_alert:
        recipient: 'monit-alert@template.ansibleguy.net'
        format: |
          From: monit-alert@template.ansibleguy.net
          Reply-To: netmaster@template.ansibleguy.net
          Subject: $SERVICE at $HOST failed
        not_on: true
        events: ['timestamp']
        description: 'alert1'
        reminder: 500
        enabled: false

    - name: Removing
      ansibleguy.opnsense.monit_alert:
        recipient: 'monit-alert@template.ansibleguy.net'
        state: 'absent'

    - name: Listing
      ansibleguy.opnsense.list:
```

(continues on next page)

(continued from previous page)

```
        # target: 'monit_alert'
      register: existing_entries

    - name: Printing alerts
      ansible.builtin.debug:
        var: existing_entries.data
```

21.3.2 Tests

```
- hosts: localhost
  gather_facts: no
  module_defaults:
    group/ansibleguy.opnsense.all:
      firewall: 'opnsense.template.ansibleguy.net'
      api_credential_file: '/home/guy/.secret/opn.key'

    ansibleguy.opnsense.list:
      target: 'monit_test'

  tasks:
    - name: Example
      ansibleguy.opnsense.monit_test:
        name: 'example'
        # type: ''
        # condition: ''
        # action: 'alert'
        # path: ''
        # enabled: true
        # reload: true

    - name: Adding memory tests
      ansibleguy.opnsense.monit_test:
        name: 'test1'
        condition: 'memory usage is greater than 90%'
        type: 'SystemResource'
        action: 'alert'

    - name: Changing
      ansibleguy.opnsense.monit_test:
        name: 'test1'
        condition: 'memory usage is greater than 90%'
        type: 'SystemResource'
        action: 'exec'
        path: '/usr/local/bin/test1.sh'

    - name: Disabling
      ansibleguy.opnsense.monit_test:
        name: 'test1'
        condition: 'memory usage is greater than 90%'
        type: 'SystemResource'
```

(continues on next page)

(continued from previous page)

```
        action: 'exec'
        path: '/usr/local/bin/test1.sh'
        enabled: false

    - name: Removing
      ansibleguy.opnsense.monit_test:
        name: 'test1'
        state: 'absent'

    - name: Adding connection tests
      ansibleguy.opnsense.monit_test:
        name: 'test2'
        condition: 'failed host 127.0.0.1 port 22 protocol ssh'
        type: 'Connection'

    - name: Listing
      ansibleguy.opnsense.list:
        # target: 'monit_test'
      register: existing_entries

    - name: Printing tests
      ansible.builtin.debug:
        var: existing_entries.data
```

21.3.3 Services

```
- hosts: localhost
  gather_facts: no
  module_defaults:
    group/ansibleguy.opnsense.all:
      firewall: 'opnsense.template.ansibleguy.net'
      api_credential_file: '/home/guy/.secret/opn.key'

    ansibleguy.opnsense.list:
      target: 'monit_service'

  tasks:
    - name: Example
      ansibleguy.opnsense.monit_service:
        name: 'example'
        # type: ''
        # pidfile: ''
        # match: ''
        # path: ''
        # timeout: 300
        # address: ''
        # interface: ''
        # start: ''
        # stop: ''
        # tests: []
```

(continues on next page)

(continued from previous page)

```
      # depends: []
      # polltime: ''
      # description: 'example'
      # enabled: true
      # reload: true

  - name: Adding simple
    ansibleguy.opnsense.monit_service:
      name: 'service1'
      type: 'custom'
      start: '/usr/local/bin/test1_start.sh'

  - name: Changing
    ansibleguy.opnsense.monit_service:
      name: 'service1'
      type: 'custom'
      start: '/usr/local/bin/service1_start.sh'
      stop: '/usr/local/bin/service1_stop.sh'
      tests: ['test1']

  - name: Adding another
    ansibleguy.opnsense.monit_service:
      name: 'service2'
      type: 'network'
      interface: 'opt2'
      depends: ['service1']

  - name: Disabling
    ansibleguy.opnsense.monit_service:
      name: 'service2'
      type: 'network'
      interface: 'opt2'
      depends: ['service1']
      enabled: false

  - name: Removing
    ansibleguy.opnsense.monit_service:
      name: 'service2'
      state: 'absent'

  - name: Listing
    ansibleguy.opnsense.list:
      # target: 'monit_service'
    register: existing_entries

  - name: Printing services
    ansible.builtin.debug:
      var: existing_entries.data
```

21.3.4 Practical example

Mail notification on IDS alert: see documentation

```yaml
- hosts: localhost
  gather_facts: no
  module_defaults:
    group/ansibleguy.opnsense.all:
      firewall: 'opnsense.template.ansibleguy.net'
      api_credential_file: '/home/guy/.secret/opn.key'

  tasks:
    - name: Adding test
      ansibleguy.opnsense.monit_test:
        name: 'SURICATA_EVE'
        condition: 'content = "blocked"'
        type: 'FileContent'
        action: 'alert'

    - name: Adding service
      ansibleguy.opnsense.monit_service:
        name: 'SURICATA_ALERT'
        type: 'file'
        path: '/var/log/suricata/eve.json'
        tests: ['SURICATA_EVE']
```

Tip: Check out the repository on GitHub

Report missing/incorrect information or broken links

CHAPTER

TWENTYTWO

OPENVPN

STATE: unstable

TESTS: ansibleguy.opnsense.openvpn_client | ansibleguy.opnsense.openvpn_server | ansibleguy.opnsense.openvpn_static_key | ansibleguy.opnsense.openvpn_client_override | ansibleguy.opnsense.openvpn_status

API Docs: OpenVPN

Service Docs: OpenVPN

22.1 Info

You can use the *ansibleguy.opnsense.service* module to interact with the OpenVPN service.

22.2 Definition

For basic parameters see: *Basic*

22.2.1 ansibleguy.opnsense.openvpn_server

Table 1: Definition

Parameter	Type	Required	Default	Aliases	Comment
name	string	true	-	description, desc	The name used to match this config to existing entries
server_ip4	string	true if no server_ip6	-	server, client_net	This directive will set up an OpenVPN server which will allocate addresses to clients out of the given network/netmask. The server itself will take the .1 address of the given network for use as the server-side endpoint of the local TUN/TAP interface

continues on next page

Table 1 – continued from previous page

Parameter	Type	Required	Default	Aliases	Comment
server_ip6	string	true if no server_ip4	-	server6, client_net_	This directive will set up an OpenVPN server which will allocate addresses to clients out of the given network/netmask. The server itself will take the next base address (+1) of the given network for use as the server-side endpoint of the local TUN/TAP interface
protocol	string	false	udp	proto	One of: 'udp', 'udp4', 'udp6', 'tcp', 'tcp4', 'tcp6'. Use protocol for communicating with remote host.
port	integer	false	1194	local_port, bind_port	Port number to use
address	string	false	-	bind_addr bind, ip	Optional IP address for bind. If specified, OpenVPN will bind to this address only. If unspecified, OpenVPN will bind to all interfaces.
mode	string	false	tun	type	One of: 'tun', 'tap'. Choose the type of tunnel, OSI Layer 3 [tun] is the most common option to route IPv4 or IPv6 traffic, [tap] offers Ethernet 802.3 (OSI Layer 2) connectivity between hosts and is usually combined with a bridge.
topology	string	false	subnet	topo	One of: 'net30', 'p2p', 'subnet'. Configure virtual addressing topology when running in –dev tun mode. This directive has no meaning in –dev tap mode, which always uses a subnet topology.
max_connection	integer	false	-	max_conn max_clien	Specify the maximum number of clients allowed to concurrently connect to this server.
log_level	integer	false	-	verbosity, verb	From 0 to 11. Output verbosity level. 0 = no output, 1-4 = normal, 5 = log packets, 6-11 debug
keepalive_interv	integer	false	-	kai	Ping interval in seconds. 0 to disable keep alive
keepalive_timeo	integer	false	-	kat	Causes OpenVPN to restart after n seconds pass without reception of a ping or other packet from remote.
renegotiate_time	integer	false	-	reneg_tim reneg	Renegotiate data channel key after n seconds (default=3600). When using a one time password, be advised that your connection will automatically drop because your password is not valid anymore. Set to 0 to disable, remember to change your client as well.

continues on next page

Table 1 – continued from previous page

Parameter	Type	Required	Default	Aliases	Comment
auth_token_time	integer	false	-	auth_time, token_time	After successful user/password authentication, the OpenVPN server will with this option generate a temporary authentication token and push that to the client. On the following renegotiations, the OpenVPN client will pass this token instead of the users password. On the server side the server will do the token authentication internally and it will NOT do any additional authentications against configured external user/password authentication mechanisms. When set to 0, the token will never expire, any other value specifies the lifetime in seconds.
certificate	string	true if no ca	-	cert	Certificate to use for this service.
ca	string	true if no certificate	-	certificate_authority, authority	Select a certificate authority when it differs from the attached certificate.
crl	string	false	-	certificate_revocation_list, revocation_list	Select a certificate revocation list to use for this service.
key	string	false	-	tls_key, tls_static_	Add an additional layer of HMAC authentication on top of the TLS control channel to mitigate DoS attacks and attacks on the TLS stack. The prefixed mode determines if this measurement is only used for authentication (–tls-auth) or includes encryption (–tls-crypt).
authentication	string	false	-	auth, auth_algo	One of: 'BLAKE2b512', 'BLAKE2s256', 'whirlpool', 'none', 'MD4', 'MD5', 'MD5-SHA1', 'RIPEMD160', 'SHA1', 'SHA224', 'SHA256', 'SHA3-224', 'SHA3-256', 'SHA3-384', 'SHA3-512', 'SHA384', 'SHA512', 'SHA512-224', 'SHA512-256', 'SHAKE128', 'SHAKE256'. Authenticate data channel packets and (if enabled) tls-auth control channel packets with HMAC using message digest algorithm alg.
network_local	list	false	-	local, net_local, push_route	These are the networks accessible on this host, these are pushed via route{-ipv6} clauses in OpenVPN to the client
network_remote	list	false	-	remote, net_remote, route	Remote networks for the server, add route to routing table after connection is established
data_ciphers	list	false	-	ciphers	One or multiple of: 'AES-256-GCM', 'AES-128-GCM', 'CHACHA20-POLY1305'. Restrict the allowed ciphers to be negotiated to the ciphers in this list.

continues on next page

Table 1 – continued from previous page

Parameter	Type	Required	Default	Aliases	Comment
data_cipher_fall	string	false	-	cipher_fallb	One of: 'AES-256-GCM', 'AES-128-GCM', 'CHACHA20-POLY1305'. Configure a cipher that is used to fall back to if we could not determine which cipher the peer is willing to use. This option should only be needed to connect to peers that are running OpenVPN 2.3 or older versions, and have been configured with –enable-small (typically used on routers or other embedded devices).
auth_mode	list	false	-	authentication_mode auth_sourc	Select authentication methods to use, leave empty if no challenge response authentication is needed.
auth_group	string	false	-	group	Restrict access to users in the selected local group. Please be aware that other authentication backends will refuse to authenticate when using this option.
options	list	false	-	opts	One or multiple of: 'client-to-client', 'duplicate-cn', 'passtos', 'persist-remote-ip', 'route-nopull', 'route-noexec', 'remote-random'. Various less frequently used yes/no options which can be set for this instance.
push_options	list	false	-	push_opts	One or multiple of: 'block-outside-dns', 'register-dns'. Various less frequently used yes/no options which can be pushed to the client for this instance.
redirect_gateway	list	false	-	redirect_gw, redir_gw	One or multiple of: 'local', 'autolocal', 'def1', 'bypass_dhcp', 'bypass_dns', 'block_local', 'ipv6', 'notipv4'. Automatically execute routing commands to cause all outgoing IP traffic to be redirected over the VPN.
domain	string	false	-	dns_doma	Set Connection-specific DNS Suffix.
domain_list	list	false	-	dns_doma	Add name to the domain search list. Repeat this option to add more entries. Up to 10 domains are supported
dns_servers	list	false	-	dns	Set primary domain name server IPv4 or IPv6 address. Repeat this option to set secondary DNS server addresses.
ntp_servers	list	false	-	ntp	Set primary NTP server address (Network Time Protocol). Repeat this option to set secondary NTP server addresses.
mtu	integer	false	-	tun_mtu	Take the TUN device MTU to be tun-mtu and derive the link MTU from it.
route_metric	integer	false	-	metric, push_metr	Specify a default metric m for use with –route on the connecting client (push option).
fragment_size	integer	false	-	frag_size	Enable internal datagram fragmentation so that no UDP datagrams are sent which are larger than the specified byte size.

continues on next page

Table 1 – continued from previous page

Parameter	Type	Required	Default	Aliases	Comment
verify_client_cert	string	false	require	verify_client, verify_cert	One of: 'require', 'none'. Specify if the client is required to offer a certificate.
cert_depth	integer	false	-	certificate_depth	From 1 to 5. When a certificate-based client logs in, do not accept certificates below this depth. Useful for denying certificates made with intermediate CAs generated from the same CA as the server.
register_dns	boolean	false	false	-	Run ipconfig /flushdns and ipconfig /registerdns on connection initiation. This is known to kick Windows into recognizing pushed DNS servers.
ocsp	boolean	false	false	use_ocsp, verify_ocsp	When the CA used supplies an authorityInfoAccess OCSP URI extension, it will be used to validate the client certificate.
user_as_cn	boolean	false	false	username_as_c	Use the authenticated username as the common-name, rather than the common-name from the client certificate.
user_cn_strict	boolean	false	false	username_cn_	When authenticating users, enforce a match between the Common Name of the client certificate and the username given at login.
mss_fix	boolean	false	false	mss	Announce to TCP sessions running over the tunnel that they should limit their send packet sizes such that after OpenVPN has encapsulated them, the resulting UDP packet size that OpenVPN sends to its peer will not exceed the recommended size.
reload	boolean	false	true	-	If the running config should be reloaded on change - this may take some time. For mass-managing items you might want to reload it 'manually' after all changes are done => using the *ansiblegguy.opnsense.reload* module.

22.2.2 ansibleguy.opnsense.openvpn_client

Table 2: Definition

Parameter	Type	Required	Default	Aliases	Comment
name	string	true	-	description, desc	The name used to match this config to existing entries
remote	list	true	-	peer, server	Remote host name or IP address with optional port
protocol	string	false	udp	proto	One of: 'udp', 'udp4', 'udp6', 'tcp', 'tcp4', 'tcp6'. Use protocol for communicating with remote host.
port	integer	false	-	local_port, bind_port	Port number to use. Specifies a bind address, or nobind when client does not have a specific bind address.
address	string	false	-	bind_addr bind, ip	Optional IP address for bind. If specified, OpenVPN will bind to this address only. If unspecified, OpenVPN will bind to all interfaces.
mode	string	false	tun	type	One of: 'tun', 'tap'. Choose the type of tunnel, OSI Layer 3 [tun] is the most common option to route IPv4 or IPv6 traffic, [tap] offers Ethernet 802.3 (OSI Layer 2) connectivity between hosts and is usually combined with a bridge.
log_level	integer	false	-	verbosity, verb	From 0 to 11. Output verbosity level. 0 = no output, 1-4 = normal, 5 = log packets, 6-11 debug
keepalive_interv	integer	false	-	kai	Ping interval in seconds. 0 to disable keep alive
keepalive_timeo	integer	false	-	kat	Causes OpenVPN to restart after n seconds pass without reception of a ping or other packet from remote.
renegotiate_time	integer	false	-	reneg_tim reneg	Renegotiate data channel key after n seconds (default=3600). When using a one time password, be advised that your connection will automatically drop because your password is not valid anymore. Set to 0 to disable, remember to change your client as well.
carp_depend_or	string	false	-	vip, vip_depen carp, carp_depe	The CARP VHID to depend on. When this virtual address is not in master state, then the instance will be shutdown.
certificate	string	true if no ca	-	cert	Certificate to use for this service.
ca	string	true if no certificate	-	certificate_autho authority	Select a certificate authority when it differs from the attached certificate.
key	string	false	-	tls_key, tls_static_	Add an additional layer of HMAC authentication on top of the TLS control channel to mitigate DoS attacks and attacks on the TLS stack. The prefixed mode determines if this measurement is only used for authentication (–tls-auth) or includes encryption (–tls-crypt).
authentication	string	false	-	auth, auth_algo	One of: 'BLAKE2b512', 'BLAKE2s256', 'whirlpool', 'none', 'MD4', 'MD5', 'MD5-SHA1', 'RIPEMD160', 'SHA1', 'SHA224', 'SHA256', 'SHA3-224', 'SHA3-256', 'SHA3-384', 'SHA3-512', 'SHA384', 'SHA512', 'SHA512-224', 'SHA512-256', 'SHAKE128', 'SHAKE256'. Authenticate data channel packets

22.2.3 ansibleguy.opnsense.openvpn_static_key

Table 3: Definition

Parameter	Type	Required	Default	Aliases	Comment
name	string	true	-	description, desc	The name used to match this config to existing entries
mode	string	false	crypt	type	One of: 'auth', 'crypt'. Define the use of this key, authentication (–tls-auth) or authentication and encryption (–tls-crypt)
key	string	false	-	-	OpenVPN Static key. If empty - it will be auto-generated.
reload	boolean	false	true	-	If the running config should be reloaded on change - this may take some time. For mass-managing items you might want to reload it 'manually' after all changes are done => using the *ansibleguy.opnsense.reload* module.

22.2.4 ansibleguy.opnsense.openvpn_client_override

Table 4: Definition

Parameter	Type	Required	Default	Aliases	Comment
name	string	true	-	description, desc	The client's X.509 common-name used to match these override to
servers	list	true	-	instances	Select the OpenVPN servers where this override applies to, leave empty for all
description	string	false	-	desc	You may enter a description here for your reference (not parsed).
block	boolean	false	false	block_con block_clie	Block this client connection based on its common name. Don't use this option to permanently disable a client due to a compromised key or password. Use a CRL (certificate revocation list) instead.
push_reset	boolean	false	false	reset	Don't inherit the global push list for a specific client instance. NOTE: –push-reset is very thorough: it will remove almost all options from the list of to-be-pushed options. In many cases, some of these options will need to be re-configured afterwards - specifically, –topology subnet and –route-gateway will get lost and this will break client configs in many cases.
network_tunnel_ip	string	false	-	tun_ip4, tunnel_ip4	Push virtual IP endpoints for client tunnel, overriding dynamic allocation.
network_tunnel_ip	string	false	-	tun_ip6, tunnel_ip6	Push virtual IP endpoints for client tunnel, overriding dynamic allocation.
network_local	list	false	-	net_local, push_rout	These are the networks accessible by the client, these are pushed via route{-ipv6} clauses in OpenVPN to the client.
network_remote	list	false	-	net_remot route	Remote networks for the server, these are configured via iroute{-ipv6} clauses in OpenVPN and inform the server to send these networks to this specific client.
route_gateway	string	false	-	route_gw, rt_gw	Specify a default gateway to use for the connected client. Without one set the first address in the netblock is being offered. When segmenting the tunnel (server) network, this one might not be accessible from the client.
redirect_gateway	list	false	-	redirect_gw, redir_gw	Automatically execute routing commands to cause all outgoing IP traffic to be redirected over the VPN.
register_dns	boolean	false	false	-	Run ipconfig /flushdns and ipconfig /registerdns on connection initiation. This is known to kick Windows into recognizing pushed DNS servers.
domain	string	false	-	dns_doma	Set Connection-specific DNS Suffix.
domain_list	list	false	-	dns_doma	Add name to the domain search list. Repeat this option to add more entries. Up to 10 domains are supported
dns_servers	list	false	-	dns	Set primary domain name server IPv4 or IPv6 address. Repeat this option to set secondary DNS server addresses.
ntp_servers	list	false	-	ntp	Set primary NTP server address (Network Time Protocol). Repeat this option to set secondary NTP server addresses.
wins_servers	list	false		wins	Set primary WINS server address (NetBIOS over

22.2.5 ansibleguy.opnsense.openvpn_status

Table 5: Definition

Parameter	Type	Required	Default	Aliases	Comment
target	string	false	sessions	kind	One of: 'sessions', 'routes'. What information to query

---

22.3 Usage

The instance description/name is used to match your config to the existing entries.

WARNING: The openvpn_server and openvpn_client module share the same namespace! Be aware that you p.e. CANNOT create an openvpn_server with the same name as an existing openvpn_client (*on the same box*)!

Use can create an manage certificates using the OPNSense WebUI!

22.4 Examples

22.4.1 ansibleguy.opnsense.openvpn_server

```
- hosts: localhost
  gather_facts: no
  module_defaults:
    group/ansibleguy.opnsense.all:
      firewall: 'opnsense.template.ansibleguy.net'
      api_credential_file: '/home/guy/.secret/opn.key'

    ansibleguy.opnsense.list:
      target: 'openvpn_instance'

  tasks:
    - name: Example
      ansibleguy.opnsense.openvpn_server:
        name: 'example'
        server_ip4: ''
        server_ip6: ''
        certificate: ''
        # topology: 'subnet'
        # protocol: 'udp'
        # port: ''
        # address: ''
        # mode: 'tun'
        # max_connections: ''
```

(continues on next page)

(continued from previous page)

```
      # ca: ''
      # crl: ''
      # verify_client_cert: 'require'
      # cert_depth: ''
      # data_ciphers: []
      # data_cipher_fallback: ''
      # ocsp: false
      # log_level: 3
      # keepalive_interval: ''
      # keepalive_timeout: ''
      # key: ''
      # authentication: ''
      # auth_mode: []
      # auth_group: ''
      # renegotiate_time: ''
      # auth_token_time: ''
      # network_local: []
      # network_remote: []
      # options: []
      # push_options: []
      # redirect_gateway: []
      # route_metric: ''
      # mtu: ''
      # fragment_size: ''
      # domain: ''
      # domain_list: []
      # dns_servers: []
      # ntp_servers: []
      # register_dns: false
      # user_as_cn: false
      # user_cn_strict: false
      # mss_fix: false
      # reload: true
      # enabled: true

  - name: Adding
    ansibleguy.opnsense.openvpn_server:
      name: 'ANSIBLE_TEST_1_1'
      port: 20000
      protocol: 'udp'
      mode: 'tun'
      server: '192.168.77.0/29'
      network_local: ['192.168.78.128/27']
      ca: 'OpenVPN'
      certificate: 'OpenVPN Server'

  - name: Changing
    ansibleguy.opnsense.openvpn_server:
      name: 'ANSIBLE_TEST_1_1'
      port: 20000
      protocol: 'udp'
      mode: 'tun'
```

(continues on next page)

(continued from previous page)

```
      server: '192.168.77.0/29'
      network_local: ['192.168.78.128/27']
      ca: 'OpenVPN'
      certificate: 'OpenVPN Server'
      cert_depth: 1
      data_ciphers: ['AES-256-GCM', 'CHACHA20-POLY1305']
      max_connections: 100
      user_as_cn: true
      user_cn_strict: true
      push_options: ['block-outside-dns', 'register-dns']
      mtu: 1420

  - name: Disabling
    ansibleguy.opnsense.openvpn_server:
      name: 'ANSIBLE_TEST_1_1'
      port: 20000
      protocol: 'udp'
      mode: 'tun'
      server: '192.168.77.0/29'
      network_local: ['192.168.78.128/27']
      ca: 'OpenVPN'
      certificate: 'OpenVPN Server'
      cert_depth: 1
      data_ciphers: ['AES-256-GCM', 'CHACHA20-POLY1305']
      max_connections: 100
      user_as_cn: true
      user_cn_strict: true
      push_options: ['block-outside-dns', 'register-dns']
      mtu: 1420
      enabled: false

  - name: Listing
    ansibleguy.opnsense.list:
      # target: 'openvpn_instance'
    register: existing_entries

  - name: Printing instances
    ansible.builtin.debug:
      var: existing_entries.data

  - name: Removing
    ansibleguy.opnsense.openvpn_server:
      name: 'test1'
      state: 'absent'
```

22.4.2 ansibleguy.opnsense.openvpn_client

```yaml
- hosts: localhost
  gather_facts: no
  module_defaults:
    group/ansibleguy.opnsense.all:
      firewall: 'opnsense.template.ansibleguy.net'
      api_credential_file: '/home/guy/.secret/opn.key'

    ansibleguy.opnsense.list:
      target: 'openvpn_instance'

  tasks:
    - name: Example
      ansibleguy.opnsense.openvpn_client:
        name: 'example'
        remote: 'example.ovpn.ansibleguy.net:10000'
        certificate: ''
        # ca: ''
        # protocol: 'udp'
        # port: ''
        # address: ''
        # mode: 'tun'
        # log_level: 3
        # keepalive_interval: ''
        # keepalive_timeout: ''
        # carp_depend_on: ''
        # key: ''
        # authentication: ''
        # username: ''
        # password: ''
        # renegotiate_time: ''
        # network_local: []
        # network_remote: []
        # options: []
        # mtu: ''
        # fragment_size: ''
        # mss_fix: false
        # reload: true
        # enabled: true

    - name: Adding
      ansibleguy.opnsense.openvpn_client:
        name: 'test1'
        remote: 'openvpn.test.ansibleguy.net:20000'
        protocol: 'udp'
        mode: 'tun'
        network_remote: ['192.168.77.128/27', '192.168.89.64/27']
        log_level: 2
        ca: 'OpenVPN'
        certificate: 'OpenVPN Client'
        mtu: 1400
```

(continues on next page)

(continued from previous page)

```
    - name: Changing
      ansibleguy.opnsense.openvpn_client:
        name: 'test1'
        remote: 'openvpn.test.ansibleguy.net:10000'
        protocol: 'tcp'
        mode: 'tun'
        network_remote: ['192.168.77.0/24']
        log_level: 5
        ca: 'OpenVPN'
        certificate: 'OpenVPN Client'
        mtu: 1400

    - name: Disabling
      ansibleguy.opnsense.openvpn_client:
        name: 'test1'
        remote: 'openvpn.test.ansibleguy.net:10000'
        protocol: 'tcp'
        mode: 'tun'
        network_remote: ['192.168.77.0/24']
        log_level: 5
        ca: 'OpenVPN'
        certificate: 'OpenVPN Client'
        mtu: 1400
        enabled: false

    - name: Listing
      ansibleguy.opnsense.list:
        # target: 'openvpn_instance'
      register: existing_entries

    - name: Printing instances
      ansible.builtin.debug:
        var: existing_entries.data

    - name: Removing
      ansibleguy.opnsense.openvpn_client:
        name: 'test1'
        state: 'absent'
```

22.4.3 ansibleguy.opnsense.openvpn_static_key

```
- hosts: localhost
  gather_facts: no
  module_defaults:
    group/ansibleguy.opnsense.all:
      firewall: 'opnsense.template.ansibleguy.net'
      api_credential_file: '/home/guy/.secret/opn.key'

    ansibleguy.opnsense.list:
```

(continues on next page)

(continued from previous page)

```
    target: 'openvpn_static_key'

tasks:
  - name: Example
    ansibleguy.opnsense.openvpn_static_key:
      name: 'example'
      # mode: 'crypt'
      # key: ''

  - name: Adding
    ansibleguy.opnsense.openvpn_static_key:
      name: 'test1'
      # key: => will be auto-generated

  - name: Changing
    ansibleguy.opnsense.openvpn_static_key:
      name: 'test1'
      key: '#\n# 2048 bit OpenVPN static key\n#\n
        -----BEGIN OpenVPN Static key V1-----\n
        c07e43dc02829f88184b4fb74243e4ac\
        nb1d24d1d1a74cd21df8ac64a527915ae\n
        9c736c0c219eb33774e40e61f6f660c8\n
        daf44730850fae665f5f609a71e99f3c\n
        8a636b16dff7434ce3b7f9aca896287b\n
        d6c62d2f6d7db4e9cfcfe0f101cc6474\n
        0c98246fbcd203891a0343777c7551c7\n
        aa2ba1e6a6ab4fcf593a894d4da8f180\n
        d44645b5a658e17f5d48408a020430c3\n
        5b768f413a2ec69ead015750cacb53d7\n
        64a19bce04b29f11d3ca7560a99958b6\n
        9203f493fd7e740b5a5a3d1afe1b4185\n
        50043805c5bac513baf2306e42c1c1f8\n
        0fd16661536a3ee72ffbd1d2d1b1f6c0\n
        9683064c9bc044ee0357f4b94f5687ed\n
        67cb013625cfb9b113ecff16674d63e6\n
        -----END OpenVPN Static key V1-----'

  - name: Listing
    ansibleguy.opnsense.list:
      # target: 'openvpn_static_key'
    register: existing_entries

  - name: Printing static-keys
    ansible.builtin.debug:
      var: existing_entries.data

  - name: Removing
    ansibleguy.opnsense.openvpn_static_key:
      name: 'test1'
      state: 'absent'

  - name: Linking key to OpenVPN-client
```

(continues on next page)

(continued from previous page)

```
      ansibleguy.opnsense.openvpn_client:
        name: 'test-client'
        remote: 'openvpn.test.ansibleguy.net'
        ca: 'OpenVPN'
        key: 'test-key'
```

22.4.4 ansibleguy.opnsense.openvpn_client_override

```
- hosts: localhost
  gather_facts: no
  module_defaults:
    group/ansibleguy.opnsense.all:
      firewall: 'opnsense.template.ansibleguy.net'
      api_credential_file: '/home/guy/.secret/opn.key'

  tasks:
    - name: Example
      ansibleguy.opnsense.openvpn_client_override:
        name: 'example'
        # servers: []
        # description: ''
        # block: false
        # push_reset: false
        # network_tunnel_ip4: ''
        # network_tunnel_ip6: ''
        # network_local: []
        # network_remote: []
        # route_gateway: ''
        # redirect_gateway: []
        # register_dns: false
        # domain: ''
        # domain_list: []
        # dns_servers: []
        # ntp_servers: []
        # wins_servers: []
        # reload: true
        # enabled: true

    - name: Adding
      ansibleguy.opnsense.openvpn_client_override:
        name: 'test1'
        servers: 'test-server'
        network_tunnel_ip4: '192.168.77.3/29'
        network_local: ['192.168.78.128/27']
        domain: 'test.vpn'
        dns_servers: ['1.1.1.1', '8.8.8.8']

    - name: Blocking client
      ansibleguy.opnsense.openvpn_client_override:
```

(continues on next page)

(continued from previous page)

```
        name: 'test2'
        block: true

    - name: Listing
      ansibleguy.opnsense.list:
        # target: 'openvpn_client_override'
      register: existing_entries

    - name: Printing client-overrides
      ansible.builtin.debug:
        var: existing_entries.data

    - name: Removing
      ansibleguy.opnsense.openvpn_client_override:
        name: 'test1'
        state: 'absent'
```

22.4.5 ansibleguy.opnsense.openvpn_status

```
- hosts: localhost
  gather_facts: no
  module_defaults:
    group/ansibleguy.opnsense.all:
      firewall: 'opnsense.template.ansibleguy.net'
      api_credential_file: '/home/guy/.secret/opn.key'

  tasks:
    - name: Querying OpenVPN Sessions
      ansibleguy.opnsense.openvpn_status:
        target: 'sessions'
      register: ovpn_sessions

    - name: Printing Sessions
      ansible.builtin.debug:
        var: ovpn_sessions.data

    - name: Querying OpenVPN Routes
      ansibleguy.opnsense.openvpn_status:
        target: 'routes'
      register: ovpn_routes

    - name: Printing Routes
      ansible.builtin.debug:
        var: ovpn_routes.data
```

Tip: Check out the repository on GitHub

Report missing/incorrect information or broken links

CHAPTER

TWENTYTHREE

PACKAGE

STATE: stable

TESTS: Playbook

API Docs: Core - Firmware

Service Docs: Plugins

23.1 Info

If:

- the package cache is too old, it will take some time - as OPNSense automatically checks for updates beforehand
- the target firewall runs an outdated version, the actions 'install' and 'reinstall' will fail as OPNSense prevents it
  - in that case - you should run *ansibleguy.opnsense.system* with action 'upgrade'

Be aware that the list-module with target 'package' will return installed plugins AND base-packages.

23.2 Definition

Table 1: Definition

Parameter	Type	Required	Default	Aliases	Comment
name	list of strings	true	-	Package or list of packages to process	
action	string	true	-	Action to execute. One of: 'install', 'reinstall', 'remove', 'lock', 'unlock'	
post_sleep	int	false	3	Seconds to sleep after executing the action. The firewall needs some time to update package info.	
timeout	float	false	30.0	Seconds until the action request timeout	

For basic parameters see: *Basic*

23.3 Examples

```yaml
- hosts: localhost
  gather_facts: no
  module_defaults:
    group/ansibleguy.opnsense.all:
      firewall: 'opnsense.template.ansibleguy.net'
      api_credential_file: '/home/guy/.secret/opn.key'

    ansibleguy.opnsense.list:
      target: 'package'

  tasks:
    - name: Installing
      ansibleguy.opnsense.package:
        name: 'os-api-backup'
        action: 'install'

    - name: Installing - multiple packages at once
      ansibleguy.opnsense.package:
        name: ['os-api-backup', 'os-dmidecode']
        action: 'install'

    - name: Removing
      ansibleguy.opnsense.package:
        name: 'os-api-backup'
        action: 'remove'

    - name: Re-installing
      ansibleguy.opnsense.package:
        name: 'os-api-backup'
        action: 'reinstall'

    - name: Locking
      ansibleguy.opnsense.package:
        name: 'os-api-backup'
        action: 'lock'

    - name: Unlocking
      ansibleguy.opnsense.package:
        name: 'os-api-backup'
        action: 'unlock'

    - name: Listing
      ansibleguy.opnsense.list:
      #  target: 'package'
      register: existing_entries

    - name: Printing installed packages
      ansible.builtin.debug:
        var: existing_entries.data
```

Tip: Check out the repository on GitHub

Report missing/incorrect information or broken links

CHAPTER

TWENTYFOUR

ROUTE

STATE: stable

TESTS: Playbook

API Docs: Core - Routes

Service Docs: Routes

24.1 Definition

Table 1: Definition

Parameter	Type	Required	Default	Aliases	Comment
gateway	string	true	-	gw	An existing gateway that should be used as target for the network. The network ip protocol (*IPv4/IPv6*) must be the same! **WARNING**: You need to supply the gateways short-name as can be seen in the WEB-UI menu 'System - Gateways - Single - Name'
network	string	true	-	nw, net	Network to route. The network ip protocol (*IPv4/IPv6*) must be the same!
description	string	false	-	desc	Optional description for the route. Could be used as unique-identifier when set as only 'match_field'.
match_fields	list of strings	false	['network', 'gateway']	-	Fields that are used to match configured routes with the running config - if any of those fields are changed, the module will think it's a new route
reload	boolean	false	true	-	If the running config should be reloaded on change - this may take some time. For mass-managing items you might want to reload it 'manually' after all changes are done => using the *ansible.guy.opnsense.reload* module.

For basic parameters see: *Basic*

24.2 Usage

First you will have to know about **route-matching**.

The module somehow needs to link the configured and existing routes to manage them.

You can to set how this matching is done by setting the 'match_fields' parameter!

The default behaviour is that a route is matched by its 'gateway' and 'network'.

However - it is **recommended** to use/set 'description' as **unique identifier** if many routes are used.

24.3 Examples

```yaml
- hosts: localhost
  gather_facts: no
  module_defaults:
    group/ansibleguy.opnsense.all:
      firewall: 'opnsense.template.ansibleguy.net'
      api_credential_file: '/home/guy/.secret/opn.key'

    ansibleguy.opnsense.route:
      match_fields: ['description']

    ansibleguy.opnsense.list:
      target: 'route'

  tasks:
    - name: Example
      ansibleguy.opnsense.route:
        description: 'test1'
        network: '172.16.0.0/12'
        gateway: 'LAN_GW'
        # match_fields: ['description']
        # enabled: true
        # debug: false
        # state: 'present'

    - name: Adding route
      ansibleguy.opnsense.route:
        description: 'test2'
        network: '10.206.0.0/16'
        gateway: 'VPN_GW'
        # match_fields: ['description']

    - name: Disabling route
      ansibleguy.opnsense.route:
        description: 'test3'
        network: '10.55.0.0/16'
        gateway: 'VPN_GW'
        enabled: false
        # match_fields: ['description']
```

(continues on next page)

(continued from previous page)

```
- name: Listing
  ansibleguy.opnsense.list:
  #  target: 'route'
  register: existing_entries

- name: Printing routes
  ansible.builtin.debug:
    var: existing_entries.data

- name: Removing route 'test3'
  ansibleguy.opnsense.route:
    description: 'test3'
    network: '10.55.0.0/16'
    gateway: 'VPN_GW'
    state: 'absent'
    match_fields: ['description']
```

Tip: Check out the repository on GitHub

Report missing/incorrect information or broken links

CHAPTER

TWENTYFIVE

RULE

STATE: unstable

TESTS: Playbook

API Docs: Core - Firewall

Service Docs: Rules

25.1 Prerequisites

You need to install the following plugin as OPNSense has no core-api for managing its firewall rules:

```
os-firewall
```

You can also install it using the *ansibleguy.opnsense.package* module.

25.2 Limitations

This plugin has some limitations you need to know of:

- ports don't support aliases
- each of these parameters only takes ONE value per rule:
  - port
  - protocol (*or 'any'; 'TCP/UDP' is NOT valid*)
  - ip-protocol (*IPv4/IPv6*)
  - direction
- gateway-groups are not valid yet => see OPNSense Forum or OPNSense Issue
- the ruleset managed by this plugin is SEPARATE from the default WEB-UI rules (*Firewall - Rules*) - combined usage might bring complications
- interfaces must be provided as used in the network config (*p.e. 'opt1' instead of 'DMZ'*)
  - per example see menu: 'Interface - Assignments - Interface ID (in brackets)'
  - this brings problems if the interface-names are not the same on both nodes when using HA-setups

25.3 Info

25.3.1 Savepoint

You can prevent lockout-situations using the savepoint systems:

- *ansibleguy.opnsense.savepoint*

25.3.2 Mass-Manage

If you want to mass-manage rules - take a look at the *ansibleguy.opnsense.rule_multi* module. It scales better for that use-case!

25.3.3 Web-UI

These rules are shown in the separate WEB-UI table.

Menu: 'Firewall - Automation - Filter'

25.4 Definition

Table 1: Definition

Parameter	Type	Required	Default	Aliases	Comment
match_fields	list	true	-	-	Fields that are used to match configured rules with the running config - if any of those fields are changed, the module will think it's a new rule. At least one of: 'sequence', 'action', 'interface', 'direction', 'ip_protocol', 'protocol', 'source_invert', 'source_net', 'source_port', 'destination_invert', 'destination_net', 'destination_port', 'gateway', 'description', 'uuid'
sequence	int	false	1	seq	Sequence for rule processing, Integer between 1 and 1000000
action	string	false	'pass'	a	Rule action. One of: 'pass', 'block' or 'reject'
quick	boolean	false	true	q	When set to quick, the rule is handled on "first match" basis, which means that the first rule matching the packet will take precedence over rules following in sequence.
interface	list	false	['lan']	i, int	One or multiple interfaces use this rule on
direction	string	false	'in'	d, dir	Direction of the traffic. Traffic IN is coming into the firewall interface, while traffic OUT is going out of the firewall interface. In visual terms: [Source] -> IN -> [Firewall] -> OUT -> [Destination]. The default policy is to filter inbound traffic, which means the policy applies to the interface on which the traffic is originally received by the firewall from the source. This is more efficient from a traffic processing perspective. In most cases, the default policy will be the most appropriate.
ip_protocol	string	false	'inet'	ipp, ip_proto	IP protocol to match. One of: 'inet', 'inet6' (*IPv4 = 'inet', IPv6 = 'inet6'*)
protocol	string	false	'any'	p, proto	Protocol like 'TCP', 'UDP', 'ICMP' and so on. For options see the WEB-UI. 'TCP/UDP' is NOT valid!
source_invert	boolean	false	false	si, src_inv, src_not	Inverted matching of the source
source_net	string	false	'any'	s, src, source	Host, network, alias or 'any'
source_port	string	false	-	sp, src_port	Leave empty to allow all, alias not supported
destination_invert	boolean	false	false	di, dest_inv, dest_not	Inverted matching of the destination
destination_net	string	false	'any'	d, dest, destination	Host, network, alias or 'any'
destination_port	string	false	-	dp, dest_port	Leave empty to allow all, alias not supported
gateway	string	false	-	g, gw	Existing gateway to use
log	boolean	false	true	l	If rule matches should be shown in the firewall logs
description	string	false	-	desc	Description for the rule
state	string	false	'present'	st	State of the rule. One of: 'present', 'absent'
enabled	boolean	false	true	en	If the rule should be en- or disabled
uuid	string	false	-	-	Optionally you can supply the uuid of an existing rule

For basic parameters see: *Basic*

25.5 Usage

First you will have to know about **rule-matching**.

The module somehow needs to link the configured and existing rules to manage them.

You need to set how this matching is done by setting the 'match_fields' parameter!

It is **recommended** to use/set **unique identifiers** like 'description' to make sure rules can be matched without overlapping.

You could also use the UUID of existing rules as ID - but you would have to pull (*list*) and configure those 'manually'.

25.6 Examples

25.6.1 Basic

```
- hosts: localhost
  gather_facts: no
  module_defaults:
    group/ansibleguy.opnsense.all:
      firewall: 'opnsense.template.ansibleguy.net'
      api_credential_file: '/home/guy/.secret/opn.key'

    ansibleguy.opnsense.list:
      target: 'rule'

  tasks:
    - name: Example
      ansibleguy.opnsense.rule:
        source_net: '192.168.0.0/24'  # host, network, alias or 'any'
        destination_net: '192.168.10.0/24'
        destination_port: 443  # alias not supported, leave unset for 'any'
        protocol: 'TCP'
        description: 'Generic test'
        match_fields: ['description']
        # sequence: 1
        # action: 'pass'
        # quick: true
        # interface: 'lan'
        # direction: 'in'
        # ip_protocol: 'inet' or 'inet6'
        # source_invert: false
        # source_port: ''
        # destination_invert: false
        # log: true
        # gateway: 'LAN_GW'
        # state: 'present'
        # enabled: true
```

(continues on next page)

(continued from previous page)

```
        # uuid: 'a9d85c00-0aa2-4705-b855-96aae16e05d7'  # optionally use uuid to identify␣
↪existing rules
        # debug: true
        # reload: true

    - name: Listing
      ansibleguy.opnsense.list:
      #  target: 'rule'
      register: existing_entries

    - name: Printing rules
      ansible.bultin.debug:
        var: existing_entries.data
```

25.6.2 With inventory config

```
- hosts: localhost
  gather_facts: no
  module_defaults:
    group/ansibleguy.opnsense.all:
      firewall: 'opnsense.template.ansibleguy.net'
      api_credential_file: '/home/guy/.secret/opn.key'

    ansibleguy.opnsense.rule:
      match_fields: ['description']  # setting description as unique-id field

  # you may want to configure your rules inside the inventory
  vars:
    rules:
      wan_deny_tor_exit_nodes_ipv4:
        src: 'ALIAS_URLTABLE_TOR_EXIT_NODES'
        int: 'wan'
        action: 'block'
      wan_deny_tor_exit_nodes_ipv6:
        src: 'ALIAS_URLTABLE_TOR_EXIT_NODES'
        int: 'wan'
        action: 'block'
        ip_proto: 'inet6'
      lan_to_dmz_https:
        src: 'LAN_net'
        dest: 'DMZ_net'
        dest_port: 443
      lan_to_dmz_http:
        src: 'LAN_net'
        dest: 'DMZ_net'
        dest_port: 80
      internal_to_inet_http:
        src: '172.16.0.0/16'
        dest_invert: true
        dest: 'bogons'
```

(continues on next page)

(continued from previous page)

```
        dest_port: 80
      internal_to_inet_https:
        src: '172.16.0.0/16'
        dest_invert: true
        dest: 'bogons'
        dest_port: 443

  tasks:
    - name: Test
      ansibleguy.opnsense.rule:
        description: "{{ rule_id }}"

        action: "{{ rule.action | default(omit) }}"
        interface: "{{ rule.int | default(omit) }}"
        direction: "{{ rule.dir | default(omit) }}"
        ip_protocol: "{{ rule.ip_proto | default(omit) }}"
        protocol: "{{ rule.proto | default(omit) }}"

        source_invert: "{{ rule.src_invert | default(omit) }}"
        source_net: "{{ rule.src | default(omit) }}"
        source_port: "{{ rule.src_port | default(omit) }}"
        destination_invert: "{{ rule.dest_invert | default(omit) }}"
        destination_net: "{{ rule.dest | default(omit) }}"
        destination_port: "{{ rule.dest_port | default(omit) }}"

        sequence: "{{ rule.seq | default(omit) }}"
        quick: "{{ rule.quick | default(omit) }}"
        log: "{{ rule.log | default(omit) }}"
        gateway: "{{ rule.gw | default(omit) }}"
        state: "{{ rule.state | default(omit) }}"
        enabled: "{{ rule.enabled | default(omit) }}"
        # debug: "{{ rule.debug | default(omit) }}"

      vars:
        rule: "{{ rule_item.value }}"
        rule_id: "{{ rule_item.key }}"

      loop_control:
        loop_var: rule_item
      with_dict: "{{ rules }}"
```

25.6.3 Purging

If you want to delete all existing rules that are **NOT CONFIGURED**.

You can also use the *ansibleguy.opnsense.rule_purge* module to do this in a cleaner way.

```
- hosts: localhost
  gather_facts: no
  module_defaults:
    group/ansibleguy.opnsense.all:
```

(continues on next page)

(continued from previous page)

```
      firewall: 'opnsense.template.ansibleguy.net'
      api_credential_file: '/home/guy/.secret/opn.key'

    ansibleguy.opnsense.list:
      target: 'rule'

    ansibleguy.opnsense.rule:
      match_fields: ['description']

  vars:
    rules: {...}

  tasks:
    - name: Pulling existing rules
      ansibleguy.opnsense.list:
      #  target: 'rule'
      register: existing_entries

    - name: Purging unconfigured rules
      ansibleguy.opnsense.rule:
        state: 'absent'
        description: "{{ existing_rule_id }}"

      when: existing_rule_id not in rules

      vars:
        existing_rule_id: "{{ existing_rule_item.value.description }}"

      loop_control:
        loop_var: existing_rule_item
      with_dict: "{{ existing_entries.data }}"
```

Tip: Check out the repository on GitHub

Report missing/incorrect information or broken links

CHAPTER

TWENTYSIX

RULE - MASS MANAGEMENT

STATE: unstable

TESTS: rule_multi | rule_purge

API Docs: Core - Firewall

Service Docs: Rules

26.1 Info

For basic info, limitations and must-know to the rule-handling see the *ansibleguy.opnsense.rule* module!

26.2 Multi

- Each rule has the attributes as defined in the *'single' ansibleguy.opnsense.rule* module
- To ensure valid configuration - the attributes of each rule get verified using ansible's built-in verifier

26.3 Definition

For basic parameters see: *Basic*

26.3.1 ansibleguy.opnsense.rule_multi

Table 1: Definition

Parameter	Type	Required	Default	Aliases	Comment
rules	dictionary	true	-	-	Dictionary of rules to manage/configure
key_field	string	true	-	-	What field is used as key of the provided dictionary. One of: 'sequence', 'description', 'uuid'
match_fields	list	true	-	-	Fields that are used to match configured rules with the running config - if any of those fields are changed, the module will think it's a new rule. At least one of: 'sequence', 'action', 'interface', 'direction', 'ip_protocol', 'protocol', 'source_invert', 'source_net', 'source_port', 'destination_invert', 'destination_net', 'destination_port', 'gateway', 'description', 'uuid'
fail_verification	boolean	false	true	fail_verify	Fail module if single rule fails the verification
fail_processing	boolean	false	true	fail_proc	Fail module if single rule fails to be processed
override	dictionary	false	-	-	Parameters to override for all rules
defaults	dictionary	false	-	-	Default values for all rules
state	string	false	'present'	-	Options: 'present', 'absent'
enabled	boolean	false	true	-	If all rules should be en- or disabled
output_info	boolean	false	false	info	Enable to show some information on processing at runtime. Will be hidden if the tasks 'no_log' parameter is set to 'true'.
reload	boolean	false	true	apply	If the running config should be reloaded on change - this may take some time. For mass-managing items you might want to reload it 'manually' after all changes are done => using the *ansibleguy.opnsense.reload* module.

26.3.2 ansibleguy.opnsense.rule_purge

Table 2: Definition

Parameter	Type	Required	Default	Aliases	Comment
rules	dictionary	true	-	-	Configured rules - to exclude from purging
key_field	string	true	-	-	What field is used as key of the provided dictionary. One of: 'sequence', 'description', 'uuid'
match_fields	list	true	-	-	Fields that are used to match configured rules with the running config - if any of those fields are changed, the module will think it's a new rule. At least one of: 'sequence', 'action', 'interface', 'direction', 'ip_protocol', 'protocol', 'source_invert', 'source_net', 'source_port', 'destination_invert', 'destination_net', 'destination_port', 'gateway', 'description', 'uuid'
output_info	boolean	false	false	info	Enable to show some information on processing at runtime. Will be hidden if the tasks 'no_log' parameter is set to 'true'.
action	string	false	'delete'	-	What to do with the matched rules. One of: 'disable', 'delete'
filters	dictionary	false	-	-	Field-value pairs to filter on - per example: {interface: lan} - to only purge rules that have only lan as interface
filter_invert	boolean	false	false	-	If true - it will purge all but the filtered ones
filter_partial	boolean	false	false	-	If true - the filter will also match if it is just a partial value-match
force_all	boolean	false	false	-	'If set to true and neither rules, nor filters are provided - all rules will be purged
fail_all	boolean	false	false	fail	Fail module if single rule fails to be purged

26.4 Usage

The 'rule_multi' module is meant to manage dictionaries of rules.

You could either invoke this module:

- once for all rules
- once per logical grouping of rules

26.5 Examples

26.5.1 Basics

```yaml
- hosts: localhost
  gather_facts: no
  module_defaults:
    group/ansibleguy.opnsense.all:
      firewall: 'opnsense.template.ansibleguy.net'
      api_credential_file: '/home/guy/.secret/opn.key'

    ansibleguy.opnsense.rule_multi:
      match_fields: ['description']
      key_field: 'description'  # rule-field that is used as key of the 'rules' dictionary

    ansibleguy.opnsense.list:
      target: 'rule'

    ansibleguy.opnsense.rule_purge:
      match_fields: ['description']
      key_field: 'description'

  tasks:
    - name: Changing
      ansibleguy.opnsense.rule_multi:
        rules:
          test1:
            source_net: '192.168.1.0/24'
            destination_invert: true
            destination_net: '10.1.0.0/8'
            action: 'block'
          test2:
            source_net: '192.168.0.0/16'
            destination_net: '10.156.10.0/24'
            destination_port: 8080
            protocol: 'TCP'
            interface: ['lan', 'opt1']
          test3:
            src: 'ALIAS_URLTABLE_TOR_EXIT_NODES'
            int: 'wan'
            action: 'block'
          test4:
            src: 'ALIAS_URLTABLE_TOR_EXIT_NODES'
            int: 'wan'
            action: 'block'
            ip_proto: 'inet6'
            state: 'absent'
        # match_fields: ['description']
        # key_field: 'description'
        # fail_verification: false
        # fail_processing: false
        # output_info: false
```

(continues on next page)

(continued from previous page)

```
      # reload: true

  - name: Pulling existing rules
    ansibleguy.opnsense.list:
    #  target: 'rule'
    register: existing_entries

  - name: Printing rules
    ansible.builtin.debug:
      var: existing_entries.data

  - name: Purging all non-configured rules
    ansibleguy.opnsense.rule_purge:
      rules: {...}
      # action: 'disable'  # default = remove
      # match_fields: ['description']
      # key_field: 'description'

  - name: Purging allow-rules on interface opt2 that use IPv4
    ansibleguy.opnsense.rule_purge:
      filters:  # filtering rules to purge by rule-parameters
        ip_protocol: 'inet'
        action: 'allow'
        interface: ['opt2']
      # filter_invert: true  # purge all non-port rules
      # match_fields: ['description']
      # key_field: 'description'
```

26.5.2 Options

You can also override all rule parameters as needed.

```
- name: Changing
  ansibleguy.opnsense.rule_multi:
    rules: {...}

    # set parameters and/or states to all rules
    override:
      interface: ['lan', 'opt1', 'opt2']
      log: true

    state: 'absent'
    enabled: false

    # or set default values for all rules (override the built-in default values)
    defaults:
      action: 'block'
      sequence: 50

    # match_fields: ['description']
    # key_field: 'description'
```

To simplify the modules usage and config - you can also use shorter parameter aliases.

```
- name: Changing
  ansibleguy.opnsense.rule_multi:
    rules:
      test1:
        src: 'ALIAS_URLTABLE_TOR_EXIT_NODES'
        int: 'wan'
        action: 'block'
      test2:
        src: 'ALIAS_URLTABLE_TOR_EXIT_NODES'
        int: 'wan'
        action: 'block'
        ip_proto: 'inet6'
        state: 'absent'
      test3:
        s: '192.168.0.0/16'  # source
        d: '10.81.53.0/24'  # destination
        dp: 443  # destination_port
        p: 'TCP'  # protocol
        i: ['lan', 'opt1']  # interface
        en: false  # enabled

    # match_fields: ['description']
    # key_field: 'description'
```

26.5.3 Troubleshooting

- info
- debug overall
- debug per rule

To simplify troubleshooting of bad configuration there are some troubleshooting parameters available.

```
- name: Changing
  ansibleguy.opnsense.rule_multi:
    rules: {...}
    fail_verification: true  # if the module should fail if one rule has a bad config␣
↪(default behaviour)
    output_info: true  # to output information of processed rules
    debug: true  # output verbose information about requests and processing
```

26.5.4 Logical grouping

This example shows an option how to manage complexer rule-sets and/or template rules across multiple sites.

Basically we are abstracting the rule-set into interface-groups (*I'll call them zones*)

```
to be done
```

Tip: Check out the repository on GitHub

Report missing/incorrect information or broken links

CHAPTER

TWENTYSEVEN

FIREWALL SAVEPOINT

STATE: unstable

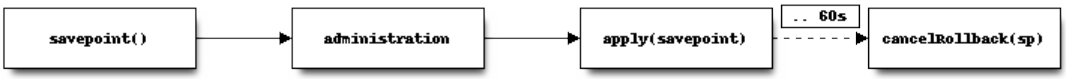
TESTS: Playbook

API Docs: Core - Firewall

27.1 Info

You can use those savepoints to prevent lockout-situations when managing rulesets remotely.

Here is the basic process:



It currently just works with the 'Firewall' plugin:

- *ansibleguy.opnsense.rule*
- *ansibleguy.opnsense.source_nat*

27.2 Definition

Table 1: Definition

Parameter	Type	Required	Default	Aliases	Comment
name	string	false	'create'	Action to execute. One of: 'create', 'revert', 'apply', 'cancel_rollba 'cancel'	
revision	string	false, true if action is one of 'apply', 'revert' or 'cancel_rollba	-	Savepoint revision to apply, revert or cancel_rollba	
controller	string	false	'filter'	Controller to manage the savepoint of. One of: 'source_na 'filter'	
api_module	string	false	'firewall'	Module to manage the savepoint of. Currently only supports 'firewall'	

For basic parameters see: *Basic*

27.3 Examples

```yaml
- hosts: localhost
  gather_facts: no
  module_defaults:
    group/ansibleguy.opnsense.all:
      firewall: 'opnsense.template.ansibleguy.net'
      api_credential_file: '/home/guy/.secret/opn.key'

  tasks:
    - name: Create a savepoint for firewall filters
      ansibleguy.opnsense.savepoint:
        action: 'create'
        controller: 'filter'  # default
      register: filter_savepoint

    - name: Apply savepoint
      ansibleguy.opnsense.savepoint:
        action: 'apply'
        revision: "{{ filter_savepoint.revision }}"

    - name: Revert savepoint
      ansibleguy.opnsense.savepoint:
        action: 'revert'
        revision: "{{ filter_savepoint.revision }}"

    - name: Create a savepoint for firewall source-nat
      ansibleguy.opnsense.savepoint:
        action: 'create'
        controller: 'source_nat'
      register: snat_savepoint

    - name: Remove source-nat savepoint (else it will be reverted automatically)
      ansibleguy.opnsense.savepoint:
        action: 'cancel_rollback'
        controller: 'source_nat'
        revision: "{{ snat_savepoint.revision }}"
```

Tip: Check out the repository on GitHub

Report missing/incorrect information or broken links

CHAPTER

TWENTYEIGHT

SERVICE

STATE: stable

TESTS: Playbook

28.1 Info

This module can interact with a specified service running on the OPNSense system.

28.2 Definition

Table 1: Definition

Parameter	Type	Required	Default	Aliases	Comment
name	string	true	-	service, target, svc, n	Pretty name of the service to interact with. One of: 'acme_client', 'apcupsd', 'bind', 'captive_portal', 'chrony', 'cicap', 'clamav', 'collectd', 'cron', 'crowdsec', 'dns_crypt_proxy', 'dyndns', 'fetchmail', 'freeradius', 'frr', 'ftp_proxy', 'haproxy', 'hwprobe', 'ids', 'iperf', 'ipsec', 'ipsec_legacy', 'lldpd', 'maltrail', 'mdns_repeater', 'monit', 'munin_node', 'netdata', 'netsnmp', 'nginx', 'node_exporter', 'nrpe', 'ntopng', 'nut', 'openconnect', 'openvpn', 'postfix', 'proxy', 'proxysso', 'puppet_agent', 'qemu_guest_agent', 'radsec_proxy', 'redis', 'relayd', 'rspamd', 'shadowsocks', 'shaper', 'siproxd', 'softether', 'sslh', 'stunnel', 'syslog', 'tayga', 'telegraf', 'tftp', 'tinc', 'tor', 'udp_broadcast_relay', 'unbound', 'vnstat', 'wireguard', 'zabbix_agent', 'zabbix_proxy'
action	string	true	-	do, a	What action to execute. Some services may not support all of these actions (*the module will inform you in that case*). One of: 'status', 'start', 'reload', 'restart', 'stop'

For basic parameters see: *Basic*

28.3 Examples

```yaml
- hosts: localhost
  gather_facts: no
  module_defaults:
    group/ansibleguy.opnsense.all:
      firewall: 'opnsense.template.ansibleguy.net'
      api_credential_file: '/home/guy/.secret/opn.key'

  tasks:
    - name: Restarting IPSec service
      ansibleguy.opnsense.service:
        name: 'ipsec'
        action: 'restart'

    - name: Get status of FRR service
      ansibleguy.opnsense.service:
        name: 'frr'
        action: 'status'
      register: frr_svc

    - name: Printing FRR service status
      ansible.builtin.debug:
        var: frr_svc.data

    - name: Stopping Tor service
      ansibleguy.opnsense.service:
        name: 'tor'
        action: 'stop'
```

CHAPTER

TWENTYNINE

TRAFFIC SHAPER

STATE: stable

TESTS: shaper_pipe | shaper_queue | shaper_rule

API Docs: Core - Traffic Shaper

Service Docs: Traffic Shaping

29.1 Info

The description is used to match the configured entries with the existing ones. It must be unique!

Interfaces for 'shaper_rules' must be provided as used in the network config (*p.e. 'opt1' instead of 'DMZ'*)

- per example see menu: 'Interface - Assignments - Interface ID (in brackets)'
- this brings problems if the interface-names are not the same on both nodes when using HA-setups

29.2 Definition

For basic parameters see: Basics

29.2.1 ansibleguy.opnsense.shaper_pipe

Parameter	Type	Required	Default value	Alias	Comment
description	string	true	-	desc	Description for the pipe - will be used to identify the entry. It must be unique!
bandwidth	int	false for state change else true	-	bw	Bandwidth limit for the pipe - used in combination with 'bw_metric'
bandwidth_metric	string	false	Mbit	bw_m	Metric of the provided bandwidth - one of: 'bit', 'Kbit', 'Mbit', 'Gbit'
queue	int	false	-	-	Integer between 2 and 100
mask	string	false	-	-	One of: 'none', 'src-ip', 'dst-ip'. Dynamic pipe creation by source or destination address. Choose destination to give every IP in destination field of rules the specified bandwidth. A pipe with 1Mbit e.g. would let 3 clients lend 1Mbit each so 3Mbit max. Normally this is used for download pipes. Choose source to give every IP in the source field of rules the specified bandwidth. Normally this is used for upload pipes. Leave this value empty if you want to create a pipe with a fixed bandwidth.
scheduler	string	false	-	-	One of: 'fifo', 'rr', 'qfq', 'fq_codel', 'fq_pie'
delay	int	false	-	-	Integer between 1 and 3000
pie_e	boole	false	false	-	Enable PIE active queue management
codel	boole	false	false	-	Enable CoDel active queue management
codel	boole	false	false	-	
codel	int	false	-	-	Integer between 1 and 10000
codel	int	false	-	-	Integer between 1 and 10000
fq-codel	int	false	-	-	Integer between 1 and 65535
fq-codel	int	false	-	-	Integer between 1 and 65535
fq-codel	int	false	-	-	Integer between 1 and 65535
buckets	int	false	-	-	Integer between 1 and 65535
reload	boole	false	true	-	If the running config should be reloaded on change - this will take some time. You might want to reload it 'manually' after all changes are done => using the reload module.
reset	boole	false	false	flush	Can be used instead of 'reload'. If the running config should be flushed and reloaded on change - this will take some time. This might have impact on other services using the same technology underneath (such as Captive portal). You might want to reload it 'manually' after all changes are done => using the service module (*action 'restart'*).

29.2.2 ansibleguy.opnsense.shaper_queue

Parameter	Type	Required	Default value	Alias	Comment
description	string	true	-	desc	Description for the queue - will be used to identify the entry. It must be unique!
pipe	string	false for state change else true	-	-	Pipe to link to the queue
weigl	string	false for state change else true	-	-	Integer between 1 and 100
mask	string	false	-	-	One of: 'none', 'src-ip', 'dst-ip'. Dynamic queue creation by source or destination address. Choose destination to evenly share every IP in destination field of rules the specified bandwidth. A pipe with 1Mbit e.g. would let 4 clients lend 250Kbit each. Normally this is used for download queues. Choose source to evenly share every IP in the source field of rules the specified bandwidth. Normally this is used for upload queues. Leave this value empty if you want to specify multiple queues with different weights.
pipe	string	false for state change else true	-	-	Pipe to link to the queue
pie_e	boole	false	false	-	Enable PIE active queue management
codel	boole	false	false	-	Enable CoDel active queue management
codel	boole	false	false	-	
codel	int	false	-	-	Integer between 1 and 10000
codel	int	false	-	-	Integer between 1 and 10000
buckets	int	false	-	-	Integer between 1 and 65535
reloa	boole	false	true	-	If the running config should be reloaded on change - this will take some time. You might want to reload it 'manually' after all changes are done => using the reload module.
reset	boole	false	false	flush	Can be used instead of 'reload'. If the running config should be flushed and reloaded on change - this will take some time. This might have impact on other services using the same technology underneath (such as Captive portal). You might want to reload it 'manually' after all changes are done => using the service module (*action 'restart'*).

29.2.3 ansibleguy.opnsense.shaper_rule

Parameter	Type	Required	Default value	Aliases	Comment
description	string	true	-	desc	Description for the rule - will be used to identify the entry. It must be unique!
target_p	string	true, false if 'target_queu provided	-	pipe	Pipe to link to the rule, alternative to 'target_queue'
target_q	string	true, false if 'target_pipe' provided	-	queue	Pipe to link to the rule, alternative to 'target_pipe'
sequenc	int	false	-	seq	Integer between 1 and 1000000
interface	string	false	'lan'	int, i	Matching packets traveling to/from interface
interface2	string	false	-	int2, i2	Secondary interface, matches packets traveling to/from interface (1) to/from interface (2). can be combined with direction.
protocol	string	false	-	proto, p	Protocol like 'ip', 'ipv4', 'tcp', 'udp' and so on - for options see the WEB-UI
max_	int	false	-	max_pac packet_le iplen	Integer between 2 and 65535
sourc	boole	false	false	si, src_inv, src_not	Inverted matching of the source
sourc	list	false	['any	s, src, source	Host, network or 'any', alias not supported
sourc	string	false	-	sp, src_port	Leave empty to allow all, alias not supported
destination_i	boole	false	false	di, dest_inv, dest_not	Inverted matching of the destination
destination_	list	false	['any	d, dest, destination	Host, network or 'any', alias not supported
destination_	string	false	-	dp, dest_por	Leave empty to allow all, alias not supported
dscp	list	false	-	-	One or multiple DSCP values - one of: 'be', 'ef', 'af11', 'af12', 'af13', 'af21', 'af22', 'af23', 'af31', 'af32', 'af33', 'af41', 'af42', 'af43', 'cs1', 'cs2', 'cs3', 'cs4', 'cs5', 'cs6', 'cs7'
direc-	string	false	'both	-	One of: 'in', 'out', leave empty for 'both'

29.3 Examples

29.3.1 Pipes

```yaml
- hosts: localhost
  gather_facts: no
  module_defaults:
    group/ansibleguy.opnsense.all:
      firewall: 'opnsense.template.ansibleguy.net'
      api_credential_file: '/home/guy/.secret/opn.key'

    ansibleguy.opnsense.list:
      target: 'shaper_pipe'

  tasks:
    - name: Example
      ansibleguy.opnsense.shaper_pipe:
        description: 'example'
        bandwidth: 50
        # bandwidth_metric: 'Mbit'
        # queue: 50
        # mask: 'none'
        # buckets: 100
        # pie_enable: false
        # codel_enable: false
        # codel_ecn_enable: false
        # codel_target: 100
        # codel_interval: 100
        # fqcodel_quantum: 100
        # fqcodel_limit: 100
        # fqcodel_flows: 100
        # delay: 100
        # enabled: true
        # debug: false
        # state: 'present'
        # reload: true
        # reset: false

    - name: Adding pipe
      ansibleguy.opnsense.shaper_pipe:
        description: 'test1'
        bandwidth: 50

    - name: Disabling pipe
      ansibleguy.opnsense.shaper_pipe:
        description: 'test1'
        bandwidth: 50
        enabled: false

    - name: Listing
      ansibleguy.opnsense.list:
        target: 'shaper_pipe'
```

(continues on next page)

(continued from previous page)

```
      register: existing_entries

    - name: Printing pipes
      ansible.builtin.debug:
        var: existing_entries.data

    - name: Removing pipe
      ansibleguy.opnsense.shaper_pipe:
        description: 'test1'
        state: 'absent'
```

29.3.2 Queues

```
- hosts: localhost
  gather_facts: no
  module_defaults:
    group/ansibleguy.opnsense.all:
      firewall: 'opnsense.template.ansibleguy.net'
      api_credential_file: '/home/guy/.secret/opn.key'

    ansibleguy.opnsense.list:
      target: 'shaper_queue'

  tasks:
    - name: Example
      ansibleguy.opnsense.shaper_queue:
        description: 'example'
        pipe: 'example'
        weight: 50
        # mask: 'none'
        # buckets: 100
        # pie_enable: false
        # codel_enable: false
        # codel_ecn_enable: false
        # codel_target: 100
        # codel_interval: 100
        # enabled: true
        # debug: false
        # state: 'present'
        # reload: true
        # reset: false

    - name: Adding pipe
      ansibleguy.opnsense.shaper_pipe:
        description: 'testPipe1'
        bandwidth: 50

    - name: Adding queue
      ansibleguy.opnsense.shaper_queue:
        description: 'testQueue1'
```

(continues on next page)

(continued from previous page)

```
        pipe: 'testPipe1'
        weight: 50

    - name: Disabling queue
      ansibleguy.opnsense.shaper_queue:
        description: 'testQueue1'
        pipe: 'testPipe1'
        weight: 50
        enabled: false

    - name: Listing
      ansibleguy.opnsense.list:
        target: 'shaper_queue'
      register: existing_entries

    - name: Printing queues
      ansible.builtin.debug:
        var: existing_entries.data

    - name: Removing queues
      ansibleguy.opnsense.shaper_queue:
        description: 'testQueue1'
        state: 'absent'
```

29.3.3 Rules

```
- hosts: localhost
  gather_facts: no
  module_defaults:
    group/ansibleguy.opnsense.all:
      firewall: 'opnsense.template.ansibleguy.net'
      api_credential_file: '/home/guy/.secret/opn.key'

    ansibleguy.opnsense.list:
      target: 'shaper_rule'

  tasks:
    - name: Example
      ansibleguy.opnsense.shaper_rule:
        description: 'example'
        target_pipe: 'example'
        target_queue: 'example'
        # sequence: 1
        # interface: 'lan'
        # interface2: 'wan'
        # max_packet_length: 1024
        # protocol: 'ip'
        # source_invert: false
        # source_net: 'any'
        # source_port: 'any'
```

(continues on next page)

(continued from previous page)

```
      # destination_invert: false
      # destination_net: 'any'
      # destination_port: 'any'
      # dscp: ['be']
      # direction: 'in'
      # enabled: true
      # debug: false
      # state: 'present'
      # reload: true
      # reset: false

  - name: Adding pipe
    ansibleguy.opnsense.shaper_pipe:
      description: 'testPipe1'
      bandwidth: 50

  - name: Adding queue
    ansibleguy.opnsense.shaper_queue:
      description: 'testQueue1'
      pipe: 'testPipe1'
      weight: 50

  - name: Adding rule - link it to queue
    ansibleguy.opnsense.shaper_rule:
      description: 'testRule1'
      target_queue: 'testQueue1'
      protocol: 'tcp'
      destination_port: 80

  - name: Adding rule - link it to pipe
    ansibleguy.opnsense.shaper_rule:
      description: 'testRule2'
      target_pipe: 'testPipe1'
      destination_invert: true
      destination: '172.16.0.0/12'

  - name: Disabling rule and flush-reload
    ansibleguy.opnsense.shaper_rule:
      description: 'testRule1'
      target_queue: 'testQueue1'
      protocol: 'tcp'
      destination_port: 80
      enabled: false

  - name: Listing
    ansibleguy.opnsense.list:
      target: 'shaper_rule'
    register: existing_entries

  - name: Printing rules
    ansible.builtin.debug:
      var: existing_entries.data
```

(continues on next page)

(continued from previous page)

```yaml
- name: Removing rule
  ansibleguy.opnsense.shaper_queue:
    description: 'testRule1'
    state: 'absent'
```

Tip: Check out the repository on GitHub

Report missing/incorrect information or broken links

CHAPTER

THIRTY

SOURCE NAT

STATE: unstable

TESTS: Playbook

API Docs: Core - Firewall

Service Docs: Outbound NAT

30.1 Prerequisites

You need to install the following plugin as OPNSense has no core-api for managing its firewall rules:

```
os-firewall
```

You can also install it using the *ansibleguy.opnsense.package* module.

30.2 Limitations

This plugin has some limitations you need to know of:

- ports don't support aliases
- each of these parameters only takes ONE value per rule:
  - port
  - protocol (*or 'any'; 'TCP/UDP' is NOT valid*)
  - ip-protocol (*IPv4/IPv6*)
- the ruleset managed by this plugin is SEPARATE from the default WEB-UI rules (*Firewall - NAT - Outbound*) - combined usage might bring complications
- interfaces must be provided as used in the network config (*p.e. 'opt1' instead of 'DMZ'*)
  - per example see menu: 'Interface - Assignments - Interface ID (in brackets)'
  - this brings problems if the interface-names are not the same on both nodes when using HA-setups

30.3 Info

30.3.1 Savepoint

You can prevent lockout-situations using the savepoint systems:

- *ansibleguy.opnsense.savepoint*

30.3.2 Web-UI

These rules are shown in the separate WEB-UI table.

Menu: 'Firewall - Automation - Source NAT'

30.4 Definition

Module alias: ansibleguy.opnsense.snat

Table 1: Definition

Parameter	Type	Required	Default	Aliases	Comment
match_fields	list	true	-	-	Fields that are used to match configured rules with the running config - if any of those fields are changed, the module will think it's a new rule. At least one of: 'sequence', 'action', 'interface', 'direction', 'ip_protocol', 'protocol', 'source_invert', 'source_net', 'source_port', 'destination_invert', 'destination_net', 'destination_port', 'gateway', 'description', 'uuid'
sequence	int	false	1	seq	Sequence for rule processing, Integer between 1 and 1000000
interface	string	false for deletion, else true	-	i, int	The interface to match this rule on
ip_protocol	string	false	'inet'	ipp, ip_proto	IP protocol to match. One of: 'inet', 'inet6' (*IPv4 = 'inet', IPv6 = 'inet6'*)
protocol	string	false	'any'	p, proto	Protocol like 'TCP', 'UDP', 'ICMP' and so on. For options see the WEB-UI. 'TCP/UDP' is NOT valid!
source_invert	boolean	false	false	si, src_inv, src_not	Inverted matching of the source
source_net	string	false	'any'	s, src, source	Host, network, alias or 'any'
source_port	string	false	-	sp, src_port	Leave empty to allow all, alias not supported
destination_invert	boolean	false	false	di, dest_inv, dest_not	Inverted matching of the destination
destination_net	string	false	'any'	d, dest, destination	Host, network, alias or 'any'
destination_port	string	false	-	dp, dest_port	Leave empty to allow all, alias not supported
target	string	false for deletion, else true	-	tgt, t	NAT translation target - Packets matching this rule will be mapped to the IP address given here. Host, network or alias
target_port	string	false	-	np, nat_port	
log	boolean	false	true	l	If rule matches should be shown in the firewall logs
description	string	false	-	desc	Description for the rule
state	string	false	'present'	st	State of the rule. One of: 'present', 'absent'
enabled	boolean	false	true	en	If the rule should be en- or disabled
uuid	string	false	-	-	Optionally you can supply the uuid of an existing rule
reload	boolean	false	true	apply	If the running config should be reloaded on change - this may take some time. For mass-managing items you might want to reload it 'manually' after all changes are done => using the *ansible.guy.opnsense.reload* module.

For basic parameters see: *Basic*

30.5 Usage

First you will have to know about **rule-matching**.

The module somehow needs to link the configured and existing rules to manage them.

You need to set how this matching is done by setting the 'match_fields' parameter!

It is **recommended** to use/set **unique identifiers** like 'description' to make sure rules can be matched without overlapping.

You could also use the UUID of existing rules as ID - but you would have to pull (*list*) and configure those 'manually'.

30.6 Examples

```yaml
- hosts: localhost
  gather_facts: no
  module_defaults:
    group/ansibleguy.opnsense.all:
      firewall: 'opnsense.template.ansibleguy.net'
      api_credential_file: '/home/guy/.secret/opn.key'

    ansibleguy.opnsense.source_nat:
      match_fields: ['description']

    ansibleguy.opnsense.list:
      target: 'source_nat'

  tasks:
    - name: Example
      ansibleguy.opnsense.source_nat:
        description: 'example'
        match_fields: ['description']
        target: '192.168.0.1'
        interface: 'opt1'
        # sequence: 1
        # ip_protocol: 'inet'
        # protocol: 'any'
        # source_invert: false
        # source_net: 'any'
        # source_port: 'any'
        # destination_invert: false
        # destination_net: 'any'
        # destination_port: 'any'
        # destination_port: 'any'
        # target_port: none
        # no_nat: false
        # log: true
        # enabled: true
        # debug: false
```

(continues on next page)

(continued from previous page)

```
      # state: 'present'
      # reload: true

  - name: Adding rule
    ansibleguy.opnsense.source_nat:
      description: 'test1'
      source: '192.168.0.0/24'
      destination: '10.0.0.0/24'
      target: '10.0.0.1'
      interface: 'opt1'
      # match_fields: ['description']

  - name: Disabling rule
    ansibleguy.opnsense.source_nat:
      description: 'test1'
      source: '192.168.0.0/24'
      destination: '10.0.0.0/24'
      target: '10.0.0.1'
      interface: 'opt1'
      enabled: false
      # match_fields: ['description']

  - name: Listing
    ansibleguy.opnsense.list:
    #  target: 'source_nat'
    register: existing_entries

  - name: Printing peers
    ansible.builtin.debug:
      var: existing_entries.data

  - name: Removing rule
    ansibleguy.opnsense.source_nat:
      description: 'test1'
      state: 'absent'
      # match_fields: ['description']
```

Tip: Check out the repository on GitHub

Report missing/incorrect information or broken links

CHAPTER

THIRTYONE

SYSLOG

STATE: stable

TESTS: Playbook

API Docs: Core - Syslog

31.1 Definition

Table 1: Definition

Parameter	Type	Required	Default	Aliases	Comment
target	string	true	-	hostname, tgt, server, srv	Server to forward the logs to
port	int	false	514	p	Port to forward the logs to
transport	string	false	udp4	trans, t	Transport protocol to use. One of: 'udp4', 'tcp4', 'udp6', 'tcp6', 'tls4', 'tls6'
level	list	false	['info', 'notice', 'warn', 'err', 'crit', 'alert', 'emerg']	lvl, lv	Log levels to forward. One or multiple of: 'debug', 'info', 'notice', 'warn', 'err', 'crit', 'alert', 'emerg'
program	list	false	-	prog	Limit applications to send logs from. For options see WEB-UI (*value in brackets needed*).
facility	list	false	-	fac	Facility to use. One of multiple of: 'kern', 'user', 'mail', 'daemon', 'auth', 'syslog', 'lpr', 'news', 'uucp', 'cron', 'authpriv', 'ftp', 'ntp', 'security', 'console', 'local0', 'local1', 'local2', 'local3', 'local4', 'local5', 'local6', 'local7'
certificate	string	false, true if transport is TLS	-	cert	Certificate to use for encrypted transport. Provide the certificates ID - not display name.
description	string	false	-	desc	Optional description for the syslog-destination. Could be used as unique-identifier when set as only 'match_field'.
match_fields	list	false	['target', 'facility', 'program']	-	Fields that are used to match configured syslog-destinations with the running config - if any of those fields are changed, the module will think it's a new entry. At least one of: 'target', 'transport', 'facility', 'program', 'level', 'port', 'description'

For basic parameters see: *Basic*

31.2 Examples

```yaml
- hosts: localhost
  gather_facts: no
  module_defaults:
    group/ansibleguy.opnsense.all:
      firewall: 'opnsense.template.ansibleguy.net'
      api_credential_file: '/home/guy/.secret/opn.key'

    ansibleguy.opnsense.syslog:
      match_fields: ['description']

    ansibleguy.opnsense.list:
      target: 'syslog'

  tasks:
    - name: Example
      ansibleguy.opnsense.syslog:
        target: '192.168.0.1'
        # port: 514
        # transport: 'udp4'
        # level: ['info', 'notice', 'warn', 'err', 'crit', 'alert', 'emerg']
        # program: ['firewall', 'openvpn']
        # facility: ['security']
        # certificate: 'certificate-id'
        # rfc5424: false
        # description: 'example'
        # match_fields: ['target', 'facility', 'program']

    - name: Adding 1
      ansibleguy.opnsense.syslog:
        description: 'test1'
        target: '192.168.0.1'
        # match_fields: ['description']

    - name: Listing
      ansibleguy.opnsense.list:
      #  target: 'syslog'
      register: existing_entries

    - name: Printing entries
      ansible.builtin.debug:
        var: existing_entries.data
```

31.2.1 Cleanup

Removing all unwanted (*not configured*) entries.

In this example the description is used as unique identifier!

```
- hosts: localhost
  gather_facts: no
  module_defaults:
    group/ansibleguy.opnsense.all:
      firewall: 'opnsense.template.ansibleguy.net'
      api_credential_file: '/home/guy/.secret/opn.key'

    ansibleguy.opnsense.syslog:
      match_fields: ['description']

    ansibleguy.opnsense.list:
      target: 'syslog'

  vars:
    syslog: {...}

  tasks:
    - name: Listing
      ansibleguy.opnsense.list:
      #  target: 'syslog'
      register: existing_entries

    - name: Purge
      ansibleguy.opnsense.syslog:
        description: "{{ destination.description }}"
        target: "{{ destination.target }}"
        state: 'absent'

      when: destination.description not in syslog | json_query('[*].description')

      loop_control:
        loop_var: destination
      loop: "{{ existing_entries.data }}"
```

Tip: Check out the repository on GitHub

Report missing/incorrect information or broken links

CHAPTER

THIRTYTWO

SYSTEM

STATE: stable

TESTS: Playbook

API Docs: Core - Firmware

32.1 Definition

Table 1: Definition

Parameter	Type	Required	Default	Aliases	Comment
action	string	true	-	Action to execute. One of: 'poweroff' 'reboot', 'update', 'upgrade', 'audit'. **WARNING**: the target firewall will be temporarily unavailable if running action 'upgrade' or 'reboot', or permanently if running action 'poweroff' (;	
wait	boolean	false	true	If the module should wait for the action to finish. Available for 'upgrade' and 'reboot'	
wait_timeout	int	false	90	Seconds to wait for the action to finish - if 'wait' is enabled	

For basic parameters see: *Basic*

32.2 Examples

```yaml
- hosts: localhost
  gather_facts: no
  module_defaults:
    group/ansibleguy.opnsense.all:
      firewall: 'opnsense.template.ansibleguy.net'
      api_credential_file: '/home/guy/.secret/opn.key'

  tasks:
    - name: Reboot the box - will wait until finished
      ansibleguy.opnsense.system:
        action: 'reboot'

    - name: Reboot the box - don't wait
      ansibleguy.opnsense.system:
        action: 'reboot'
        wait: false

    - name: Shutdown the box
      ansibleguy.opnsense.system:
        action: 'poweroff'

    - name: Pull updates
      ansibleguy.opnsense.system:
        action: 'update'

    - name: Start upgrade - will wait until finished
      ansibleguy.opnsense.system:
        action: 'upgrade'
        timeout: 120  # depends on your download speed and firmware-version

    - name: Run audit
      ansibleguy.opnsense.system:
        action: 'audit'
```

Tip: Check out the repository on GitHub

Report missing/incorrect information or broken links

CHAPTER

THIRTYTHREE

DNS - UNBOUND - ACL

STATE: unstable

TESTS: Playbook

API Docs: Core - Unbound

Service Docs: Unbound

33.1 Info

This module manages the ACL settings that can be found in the WEB-UI menu: 'Services - Unbound DNS - Access Lists' (*URL 'ui/unbound/acl'*)

The configured lists are matched by its unique file-name.

> **Warning:** Unbound service actions like `reload` can take long. Please be aware of the **possible downtime**!
>
> You may also need to increase the module `timeout`.

33.2 Definition

Table 1: Definition

Parameter	Type	Required	Default	Aliases	Comment
name	string	true	-	n	Unique name of the ACL
action	string	false	allow	-	What to to with DNS request that match the criteria. One of: 'allow', 'deny', 'refuse', 'allow_snoop', 'deny_non_local', 'refuse_non_local'. **Allow**: Choose what to do with DNS requests that match the criteria specified below. **Deny**: This action stops queries from hosts within the netblock defined below. **Refuse**: This action also stops queries from hosts within the netblock defined below, but sends a DNS rcode REFUSED error message back to the client. **Allow**: This action allows queries from hosts within the netblock defined below. **Allow Snoop**: This action allows recursive and nonrecursive access from hosts within the netblock defined below. Used for cache snooping and ideally should only be configured for your administrative host. **Deny Non-local**: Allow only authoritative local-data queries from hosts within the netblock defined below. Messages that are disallowed are dropped. **Refuse Non-local**: Allow only authoritative local-data queries from hosts within the netblock defined below. Sends a DNS rcode REFUSED error message back to the client for messages that are disallowed.
networks	list	false for state changes, else true	-	nets	List of networks in CIDR notation to apply this ACL on. For example: 192.168.1.0/24
description	string	false	-	desc	The description for the ACL
reload	boolean	false	true	-	If the running config should be reloaded on change - this may take some time. For mass-managing items you might want to reload it 'manually' after all changes are done => using the *ansible.guy.opnsense.reload* module.

For basic parameters see: *Basic*

33.3 Examples

```
- hosts: localhost
  gather_facts: no
  module_defaults:
    group/ansibleguy.opnsense.all:
      firewall: 'opnsense.template.ansibleguy.net'
      api_credential_file: '/home/guy/.secret/opn.key'

    ansibleguy.opnsense.list:
      target: 'unbound_acl'

  tasks:
    - name: Example
      ansibleguy.opnsense.unbound_acl:
        name: 'example'
        # action: "
        # networks: []
        # description: "
        # reload: true
        # enabled: true

    - name: Adding
      ansibleguy.opnsense.unbound_acl:
        name: 'test1'
        action: 'allow'
        networks: ['192.168.0.0/24']

    - name: Changing
      ansibleguy.opnsense.unbound_acl:
        name: 'test1'
        action: 'deny'
        networks: ['192.168.1.0/25']

    - name: Disabling
      ansibleguy.opnsense.unbound_acl:
        name: 'test1'
        action: 'deny'
        networks: ['192.168.1.0/25']
        enabled: false

    - name: Listing
      ansibleguy.opnsense.list:
        # target: 'unbound_acl'
      register: existing_entries

    - name: Printing acls
      ansible.builtin.debug:
        var: existing_entries.data

    - name: Removing
      ansibleguy.opnsense.unbound_acl:
```

(continues on next page)

(continued from previous page)

```
        name: 'test1'
        state: 'absent'
```

Tip: Check out the repository on GitHub

Report missing/incorrect information or broken links

CHAPTER

THIRTYFOUR

DNS - UNBOUND - DOMAIN OVERRIDE

STATE: stable

TESTS: Playbook

API Docs: Core - Unbound

Service Docs: Unbound

34.1 Definition

Table 1: Definition

Parameter	Type	Required	Default	Aliases	Comment
match_fields	string	false	['domain', 'server']	-	Fields that are used to match configured domain-overrides with the running config - if any of those fields are changed, the module will think it's a new entry. At least one of: 'domain', 'server', 'description'
domain	string	true	-	dom, d	Domain to override
server	string	true	-	value, srv	IP address of the authoritative DNS server for this domain. To use a non-default port for communication, append an '@' with the port number
description	string	false	-	desc	Optional description for the domain-override. Could be used as unique-identifier when set as only 'match_field'.
reload	boolean	false	true	-	If the running config should be reloaded on change - this may take some time. For mass-managing items you might want to reload it 'manually' after all changes are done => using the *ansible.guy.opnsense.reload* module.

For basic parameters see: *Basic*

34.2 Info

This module manages DNS domain-overrides configuration that can be found in the WEB-UI menu: 'Services - Unbound DNS - Overrides - Domain overrides'

Entries like these override an entire domain by specifying an authoritative DNS server to be queried for that domain.

> **Warning:** Unbound service actions like `reload` can take long. Please be aware of the **possible downtime**!
>
> You may also need to increase the module `timeout`.

34.3 Usage

First you will have to know about **domain-matching**.

The module somehow needs to link the configured and existing domain-overrides to manage them.

You can to set how this matching is done by setting the 'match_fields' parameter!

The default behaviour is that a domain-override is matched by its 'domain' and 'server' fields.

However - it is **recommended** to use/set 'description' as **unique identifier** if many overrides are used.

34.3.1 Mass-Manage

If you are mass-managing DNS records or using DNS-Blocklists - you might want to disable `reload: false` on single module-calls!

This takes a long time, as the service gets reloaded every time!

You might want to reload it 'manually' after all changes are done => using the *ansibleguy.opnsense.reload* module

34.4 Examples

```yaml
- hosts: localhost
  gather_facts: no
  module_defaults:
    group/ansibleguy.opnsense.all:
      firewall: 'opnsense.template.ansibleguy.net'
      api_credential_file: '/home/guy/.secret/opn.key'

    ansibleguy.opnsense.unbound_domain:
      match_fields: ['description']

    ansibleguy.opnsense.list:
      target: 'unbound_domain'

  tasks:
    - name: Example
      ansibleguy.opnsense.unbound_domain:
        domain: 'opnsense.template.ansibleguy.net'
```

(continues on next page)

(continued from previous page)

```
      server: '192.168.0.1'
      # match_fields: ['description']
      # description: 'example'
      # state: 'present'
      # reload: true
      # enabled: true
      # debug: false

  - name: Adding
    ansibleguy.opnsense.unbound_domain:
      domain: 'opnsense.template.ansibleguy.net'
      server: '192.168.0.1'
      match_fields: ['description']
      description: 'test1'
      # match_fields: ['description']

  - name: Disabling
    ansibleguy.opnsense.unbound_domain:
      domain: 'opnsense.template.ansibleguy.net'
      server: '192.168.0.1'
      match_fields: ['description']
      description: 'test1'
      enabled: false
      # match_fields: ['description']

  - name: Removing
    ansibleguy.opnsense.unbound_domain:
      domain: 'opnsense.template.ansibleguy.net'
      server: '192.168.0.1'
      state: 'absent'
      description: 'test1'
      # match_fields: ['description']

  - name: Listing
    ansibleguy.opnsense.list:
    #  target: 'unbound_domain'
    register: existing_entries

  - name: Printing domains
    ansible.builtin.debug:
      var: existing_entries.data
```

Tip: Check out the repository on GitHub

Report missing/incorrect information or broken links

CHAPTER

THIRTYFIVE

DNS - UNBOUND - DNS-OVER-TLS

STATE: stable

TESTS: Playbook

API Docs: Core - Unbound

Service Docs: Unbound

35.1 Definition

Table 1: Definition

Parameter	Type	Required	Default	Aliases	Comment
domain	string	false	-	dom, d	Provide a domain to limit the DNS-over-TLS to or leave empty to act as a catch-all
target	string	true	-	server, srv, tgt	DNS target server
port	string	false	53	p	DNS port of the target server
verify	string	false	-	common_name, cn, hostname	Verify if CN in certificate matches this value, **if not set - certificate verification will not be performed**! Must be a valid IP-Address or hostname.
reload	boolean	false	true	-	If the running config should be reloaded on change - this may take some time. For mass-managing items you might want to reload it 'manually' after all changes are done => using the *ansible.guy.opnsense.reload* module.

For basic parameters see: *Basic*

35.2 Info

This module manages DNS-over-TLS configuration that can be found in the WEB-UI menu: 'Services - Unbound DNS - DNS over TLS'

35.2.1 Mass-Manage

If you are mass-managing DNS records or using DNS-Blocklists - you might want to disable `reload: false` on single module-calls!

This takes a long time, as the service gets reloaded every time!

You might want to reload it 'manually' after all changes are done => using the *ansibleguy.opnsense.reload* module

> **Warning:** Unbound service actions like `reload` can take long. Please be aware of the **possible downtime**!
>
> You may also need to increase the module `timeout`.

35.3 Examples

```yaml
- hosts: localhost
  gather_facts: no
  module_defaults:
    group/ansibleguy.opnsense.all:
      firewall: 'opnsense.template.ansibleguy.net'
      api_credential_file: '/home/guy/.secret/opn.key'

    ansibleguy.opnsense.list:
      target: 'unbound_dot'

  tasks:
    - name: Example
      ansibleguy.opnsense.unbound_dot:
        target: '1.1.1.1'
        # domain: ''
        # port: 53
        # verify: ''
        # state: 'present'
        # reload: true
        # enabled: true
        # debug: false

    - name: Adding
      ansibleguy.opnsense.unbound_dot:
        domain: 'dot.template.ansibleguy.net'
        target: '1.1.1.1'
        verify: 'dot.template.ansibleguy.net'

    - name: Listing
      ansibleguy.opnsense.list:
```

(continues on next page)

(continued from previous page)

```
    #  target: 'unbound_dot'
    register: existing_entries

  - name: Printing DNS-over-TLS entries
    ansible.builtin.debug:
      var: existing_entries.data
```

Tip: Check out the repository on GitHub

Report missing/incorrect information or broken links

CHAPTER

THIRTYSIX

DNS - UNBOUND - FORWARDING

STATE: stable

TESTS: Playbook

API Docs: Core - Unbound

Service Docs: Unbound

36.1 Definition

Table 1: Definition

Parameter	Type	Required	Default	Aliases	Comment
domain	string	false	-	dom, d	Domain of the host. All queries for this domain will be forwarded to the nameserver specified. Leave empty to catch all queries and forward them to the nameserver
target	string	true	-	server, srv, tgt	Server to forward the dns queries to
port	string	false	53	p	DNS port of the target server
reload	boolean	false	true	-	If the running config should be reloaded on change - this may take some time. For mass-managing items you might want to reload it 'manually' after all changes are done => using the *ansible.guy.opnsense.reload* module.

For basic parameters see: *Basic*

36.2 Info

This module manages DNS-Forwardings that can be found in the WEB-UI menu: 'Services - Unbound DNS - Query Forwardings'

36.2.1 Mass-Manage

If you are mass-managing DNS records or using DNS-Blocklists - you might want to disable `reload: false` on single module-calls!

This takes a long time, as the service gets reloaded every time!

You might want to reload it 'manually' after all changes are done => using the *ansibleguy.opnsense.reload* module

> **Warning:** Unbound service actions like `reload` can take long. Please be aware of the **possible downtime**!
>
> You may also need to increase the module `timeout`.

36.3 Examples

```yaml
- hosts: localhost
  gather_facts: no
  module_defaults:
    group/ansibleguy.opnsense.all:
      firewall: 'opnsense.template.ansibleguy.net'
      api_credential_file: '/home/guy/.secret/opn.key'

    ansibleguy.opnsense.list:
      target: 'unbound_forward'

  tasks:
    - name: Example
      ansibleguy.opnsense.unbound_forward:
        domain: 'dot.template.ansibleguy.net'
        target: '1.1.1.1'
        # port: 53
        # verify: 'dot.template.ansibleguy.net'
        # state: 'present'
        # reload: true
        # enabled: true
        # debug: false

    - name: Adding
      ansibleguy.opnsense.unbound_forward:
        domain: 'dot.template.ansibleguy.net'
        target: '1.1.1.1'

    - name: Listing
      ansibleguy.opnsense.list:
      #  target: 'unbound_forward'
      register: existing_entries

    - name: Printing DNS-Forwardings
      ansible.builtin.debug:
        var: existing_entries.data
```

CHAPTER

THIRTYSEVEN

DNS - UNBOUND GENERAL

STATE: stable

TESTS: unbound_general

API Docs: Core - Unbound

Service Docs: Unbound DNS

37.1 Requirements

This module requires OPNsense 23.7 or later.

37.2 Info

WARNING: Unbound service actions like :code:`reload` can take long. Please be aware of the **possible downtime**!

You may also need to increase the module `timeout`.

37.3 Definition

For basic parameters see: Basics

37.3.1 ansibleguy.opnsense.unbound_general

Parameter	Type	Require	Default value	Alias	Comment
enabled	bool	false	true	-	En- or disable the Unbound DNS service
port	int	false	53	-	The TCP/UDP port used for responding to DNS queries
interfaces	list	false	-	-	The interface(s) used for responding to queries from clients
dnssec	bool	false	false	-	En- or disable DNSSEC
dns64	bool	false	false	-	En- or disable to synthesize AAAA records from A records if no actual AAAA records are present
dns64_prefi	string	false	'64:ff9	-	The DNS64 prefix
aaaa_only_n	bool	false	false	-	En- or disable to remove all A records from the answer section of all responses
register_dhcp_lea	bool	false	false	-	En- or disable to register machines that specify their hostname when requesting a DHCP lease
dhcp_domai	string	false	-	-	The default domain name to use for DHCP lease registration
register_dhcp_sta	bool	false	false	-	En- or disable to register DHCP static mappings
register_ipv6_lin	bool	false	true	-	En- or disable to register IPv6 link-local addresses
register_system_	bool	false	true	-	En- or disable to generate A/AAAA records for the configured listen interfaces
txt_records	bool	false	false	txt	En- or disable to create TXT record for descriptions associated with Host entries and DHCP Static mappings
flush_dns_ca	bool	false	false	-	En- or disable to flush the DNS cache during each daemon reload
local_zone_ty	string	false	'transp	-	The local zone type used for the system domain. One of: 'transparent', 'always_nxdomain', 'always_refuse', 'always_transparent', 'deny', 'inform', 'inform_deny', 'nodefault', 'refuse', 'static', 'typetransparent'
outgoing_interfac	list	false	-	-	The interface(s) that Unbound will use to send queries to authoritative servers and receive their replies
wpad	bool	false	false	-	En- or disable to automatically add CNAME records for the WPAD host of all configured domains as well as overrides for TXT records for domains
reload	boole	false	true	-	If the running config should be reloaded on change - this will take some time. For mass-managing items you might want to reload it 'manually' after all changes are done => using the reload module.

37.4 Examples

37.4.1 ansibleguy.opnsense.unbound_general

```
- hosts: localhost
  gather_facts: no
  module_defaults:
    group/ansibleguy.opnsense.all:
      firewall: 'opnsense.template.ansibleguy.net'
      api_credential_file: '/home/guy/.secret/opn.key'
```

(continues on next page)

(continued from previous page)

```yaml
tasks:
  - name: Example
    ansibleguy.opnsense.unbound_general:
      # enabled: true
      # port: 53
      # interfaces: "
      # dnssec: false
      # dns64: false
      # dns64_prefix: '64:ff9b::/96'
      # aaaa_only_mode: false
      # register_dhcp_leases: false
      # dhcp_domain: "
      # register_dhcp_static_mappings: false
      # register_ipv6_link_local: true
      # register_system_records: true
      # txt_records: false
      # flush_dns_cache: false
      # local_zone_type: 'transparent'
      # outgoing_interfaces: "
      # wpad: false
      # reload: true


  - name: Enabling Unbound
    ansibleguy.opnsense.unbound_general:
      enabled: true
      port: 53
      interfaces: ['lan']
      local_zone_type: 'transparent'
```

Tip: Check out the repository on GitHub

Report missing/incorrect information or broken links

CHAPTER

THIRTYEIGHT

DNS - UNBOUND - HOST OVERRIDE

STATE: stable

TESTS: Playbook

API Docs: Core - Unbound

Service Docs: Unbound

38.1 Definition

Table 1: Definition

Parameter	Type	Required	Default	Aliases	Comment
match_fields	string	false	['hostname', 'domain', 'record_ty 'value', 'prio']	-	Fields that are used to match configured host-overrides with the running config - if any of those fields are changed, the module will think it's a new entry. At least one of: 'hostname', 'domain', 'record_type', 'value', 'prio', 'description'
hostname	string	true	-	host, h	Hostname of the record
domain	string	true	-	dom, d	Domain of the record
record_type	string	false	'A'	type, rr, rt	Record type. One of: 'A', 'AAAA', 'MX'
value	string	false for state changes, else true	-	server, srv, mx	Value the record should hold
prio	int	false	10	mxprio	Priority that is only used for MX record types
description	string	false	-	desc	Optional description for the host-override. Could be used as unique-identifier when set as only 'match_field'.
reload	boolean	false	true	-	If the running config should be reloaded on change - this may take some time. For mass-managing items you might want to reload it 'manually' after all changes are done => using the *ansible.guy.opnsense.reload* module.

For basic parameters see: *Basic*

38.2 Info

This module manages DNS host-overrides configuration that can be found in the WEB-UI menu: 'Services - Unbound DNS - Overrides - Host overrides'

Entries like these override individual results from the forwarders.

Use these for changing DNS results or for adding custom DNS records.

Keep in mind that all resource record types (i.e. A, AAAA, MX, etc. records) of a specified host below are being overwritten.

> **Warning:** Unbound service actions like `reload` can take long. Please be aware of the **possible downtime**!
>
> You may also need to increase the module `timeout`.

38.3 Usage

First you will have to know about **host-matching**.

The module somehow needs to link the configured and existing host-overrides to manage them.

You can to set how this matching is done by setting the 'match_fields' parameter!

The default behaviour is that a host-override is matched by its 'hostname', 'domain', 'record_type', 'value' and 'prio' fields.

However - it is **recommended** to use/set 'description' as **unique identifier** if many overrides are used.

38.3.1 Mass-Manage

If you are mass-managing DNS records or using DNS-Blocklists - you might want to disable `reload: false` on single module-calls!

This takes a long time, as the service gets reloaded every time!

You might want to reload it 'manually' after all changes are done => using the *ansibleguy.opnsense.reload* module

38.4 Examples

```yaml
- hosts: localhost
  gather_facts: no
  module_defaults:
    group/ansibleguy.opnsense.all:
      firewall: 'opnsense.template.ansibleguy.net'
      api_credential_file: '/home/guy/.secret/opn.key'

    ansibleguy.opnsense.unbound_host:
      match_fields: ['description']

    ansibleguy.opnsense.list:
      target: 'unbound_host'
```

(continues on next page)

(continued from previous page)

```yaml
tasks:
  - name: Example
    ansibleguy.opnsense.unbound_host:
      hostname: 'host'
      domain: 'opnsense.template.ansibleguy.net'
      value: '192.168.0.1'
      # match_fields: ['description']
      # record_type: 'A'
      # prio: 10
      # description: 'example'
      # state: 'present'
      # reload: true
      # enabled: true
      # debug: false

  - name: Adding
    ansibleguy.opnsense.unbound_host:
      hostname: 'host'
      domain: 'opnsense.template.ansibleguy.net'
      value: '192.168.0.1'
      description: 'test1'
      # match_fields: ['description']

  - name: Removing
    ansibleguy.opnsense.unbound_host:
      hostname: 'host'
      domain: 'opnsense.template.ansibleguy.net'
      value: '192.168.0.1'
      state: 'absent'
      description: 'test1'
      # match_fields: ['description']

  - name: Adding MX record
    ansibleguy.opnsense.unbound_host:
      hostname: 'mx'
      domain: 'opnsense.template.ansibleguy.net'
      value: 'host.opnsense.template.ansibleguy.net'
      record_type: 'MX'
      description: 'test2'
      # match_fields: ['description']

  - name: Listing
    ansibleguy.opnsense.list:
    #  target: 'unbound_host'
    register: existing_entries

  - name: Printing hosts
    ansible.builtin.debug:
      var: existing_entries.data
```

Tip: Check out the repository on GitHub

Report missing/incorrect information or broken links

CHAPTER

THIRTYNINE

DNS - UNBOUND - HOST ALIAS

STATE: stable

TESTS: Playbook

API Docs: Core - Unbound

Service Docs: Unbound

39.1 Definition

Table 1: Definition

Parameter	Type	Required	Default	Aliases	Comment
match_fields	string	false	['alias', 'domain']	-	Fields that are used to match configured domain-overrides with the running config - if any of those fields are changed, the module will think it's a new entry. At least one of: 'hostname', 'domain', 'alias', 'description'
alias	string	true	-	hostname	Host-alias to create
domain	string	true	-	dom, d	Domain to override
target	string	false for state changes, else true	-	tgt, host	Existing host override record
description	string	false	-	desc	Optional description for the host-alias. Could be used as unique-identifier when set as only 'match_field'.
reload	boolean	false	true	-	If the running config should be reloaded on change - this may take some time. For mass-managing items you might want to reload it 'manually' after all changes are done => using the *ansible.guy.opnsense.reload* module.

For basic parameters see: *Basic*

39.2 Info

This module manages DNS host-alias override configuration that can be found in the WEB-UI menu: 'Services - Unbound DNS - Overrides - Host overrides - Aliases'

Entries like these override individual results from the forwarders.

Use these for changing DNS results or for adding custom DNS records.

Keep in mind that all resource record types (i.e. A, AAAA, MX, etc. records) of a specified host below are being overwritten.

> **Warning:** Unbound service actions like `reload` can take long. Please be aware of the **possible downtime**!
>
> You may also need to increase the module `timeout`.

39.3 Usage

First you will have to know about **alias-matching**.

The module somehow needs to link the configured and existing host-aliases to manage them.

You can to set how this matching is done by setting the 'match_fields' parameter!

The default behaviour is that a host-alias is matched by its 'alias' and 'domain' fields.

However - it is **recommended** to use/set 'description' as **unique identifier** if many aliases are used.

39.3.1 Mass-Manage

If you are mass-managing DNS records or using DNS-Blocklists - you might want to disable `reload: false` on single module-calls!

This takes a long time, as the service gets reloaded every time!

You might want to reload it 'manually' after all changes are done => using the *ansibleguy.opnsense.reload* module

39.4 Examples

```yaml
- hosts: localhost
  gather_facts: no
  module_defaults:
    group/ansibleguy.opnsense.all:
      firewall: 'opnsense.template.ansibleguy.net'
      api_credential_file: '/home/guy/.secret/opn.key'

    ansibleguy.opnsense.unbound_host_alias:
      match_fields: ['description']

    ansibleguy.opnsense.list:
      target: 'unbound_host_alias'
```

(continues on next page)

(continued from previous page)

```yaml
tasks:
  - name: Example
    ansibleguy.opnsense.unbound_host_alias:
      alias: 'test'
      domain: 'opnsense.template.ansibleguy.net'
      target: 'host.opnsense.template.ansibleguy.net'
      # match_fields: ['description']
      # description: 'example'
      # state: 'present'
      # reload: true
      # enabled: true
      # debug: false

  - name: Adding alias 'test1.local' for record 'test.local'
    ansibleguy.opnsense.unbound_host_alias:
      alias: 'test1'
      domain: 'local'
      target: 'test.local'
      description: 'test1'
      # match_fields: ['description']

  - name: Disabling
    ansibleguy.opnsense.unbound_host_alias:
      alias: 'test1'
      domain: 'local'
      target: 'test.local'
      description: 'test1'
      enabled: false
      # match_fields: ['description']

  - name: Removing
    ansibleguy.opnsense.unbound_host_alias:
      alias: 'test1'
      domain: 'local'
      target: 'test.local'
      state: 'absent'
      description: 'test1'
      # match_fields: ['description']

  - name: Listing
    ansibleguy.opnsense.list:
    #  target: 'unbound_host_alias'
    register: existing_entries

  - name: Printing aliases
    ansible.builtin.debug:
      var: existing_entries.data
```

Tip: Check out the repository on GitHub

Report missing/incorrect information or broken links

CHAPTER

FORTY

WEB PROXY

STATE: stable

TESTS: webproxy_general | webproxy_cache | webproxy_parent | webproxy_traffic | webproxy_forward | webproxy_acl | webproxy_icap | webproxy_auth | webproxy_remote_acl | webproxy_pac_proxy | webproxy_pac_match | webproxy_pac_rule

API Docs: Plugins - Proxy

Service Docs: Transparent Proxy | Caching Proxy | Web Proxy/Filter

40.1 Prerequisites

You need to install the following plugin:

```
os-squid
```

You can also install it using the *ansibleguy.opnsense.package* module.

40.2 Info

40.2.1 General

ansibleguy.opnsense.webproxy_general

This module manages the basic Web-Proxy settings that can be found in the WEB-UI menu: 'Services - Web Proxy - Administration - General Proxy Settings' (*URL 'ui/proxy'*)

ansibleguy.opnsense.webproxy_cache

This module manages the Web-Proxy caching-settings that can be found in the WEB-UI menu: 'Services - Web Proxy - Administration - General Proxy Settings - Local Cache Settings (*DropDown*)' (*URL 'ui/proxy#subtab_proxy-general-cache-local'*)

ansibleguy.opnsense.webproxy_parent

This module manages the Web-Proxy parent-proxy settings that can be found in the WEB-UI menu: 'Services - Web Proxy - Administration - General Proxy Settings - Parent Proxy Settings (*DropDown*)' (*URL 'ui/proxy#subtab_proxy-general-parentproxy'*)

ansibleguy.opnsense.webproxy_traffic

This module manages the Web-Proxy traffic-management settings that can be found in the WEB-UI menu: 'Services - Web Proxy - Administration - General Proxy Settings - Traffic Management Settings (*DropDown*)' (*URL 'ui/proxy#subtab_proxy-general-traffic'*)

40.2.2 Forward

ansibleguy.opnsense.webproxy_forward

This module manages the Web-Proxy forwarding settings that can be found in the WEB-UI menu: 'Services - Web Proxy - Administration - Forward Proxy

- General Forward Settings (*DropDown*)' (*URL 'ui/proxy#subtab_proxy-forward-general'*)
- FTP Proxy Settings (*DropDown*)' (*URL 'ui/proxy#subtab_proxy-forward-ftp'*)
- SNMP Agent Settings (*DropDown*)' (*URL 'ui/proxy#subtab_proxy-forward-snmp'*)

ansibleguy.opnsense.webproxy_acl

This module manages the Web-Proxy forwarding ACLs that can be found in the WEB-UI menu: 'Services - Web Proxy - Administration - General Proxy Settings - Access Control List (*DropDown*)' (*URL 'ui/proxy#subtab_proxy-forward-acl'*)

ansibleguy.opnsense.webproxy_icap

This module manages the Web-Proxy ICAP settings that can be found in the WEB-UI menu: 'Services - Web Proxy - Administration - General Proxy Settings - ICAP Settings (*DropDown*)' (*URL 'ui/proxy#subtab_proxy-icap'*)

ansibleguy.opnsense.webproxy_auth

This module manages the Web-Proxy authentication settings that can be found in the WEB-UI menu: 'Services - Web Proxy - Administration - General Proxy Settings - Authentication Settings (*DropDown*)' (*URL 'ui/proxy#subtab_proxy-general-authentication'*)

40.2.3 Remote ACL

ansibleguy.opnsense.webproxy_remote_acl

This module manages the Remote ACL entries that can be found in the WEB-UI menu: 'Services - Web Proxy - Administration - Remote Access Control Lists

The configured lists are matched by its unique file-name.

40.2.4 Proxy Auto-Config

ansibleguy.opnsense.webproxy_pac_proxy

This module manages the Proxy-Auto-Config Proxy entries that can be found in the WEB-UI menu: 'Services - Web Proxy - Administration - Proxy Auto-Config - Proxies (*DropDown*)' (*URL 'ui/proxy#subtab_pac_proxies'*)

ansibleguy.opnsense.webproxy_pac_match

This module manages the Proxy-Auto-Config Match entries that can be found in the WEB-UI menu: 'Services - Web Proxy - Administration - Proxy Auto-Config - Matches (*DropDown*)' (*URL 'ui/proxy#subtab_pac_matches'*)

You need to **provide arguments** for different **match-types**:

- 'url_matches' needs 'url' to be provided
- 'hostname_matches', 'plain_hostname', 'is_resolvable' and 'dns_domain_is' need 'hostname' to be provided
- 'my_ip_in_net' and 'destination_in_net' need 'network' to be provided
- 'date_range' needs 'month_from' and 'month_to' to be provided
- 'time_range' needs 'hour_from' and 'hour_to' to be provided
- 'weekday_range' needs 'weekday_from' and 'weekday_to' to be provided
- 'dns_domain_levels' needs 'domain_level_from' and 'domain_level_to' to be provided

ansibleguy.opnsense.webproxy_pac_rule

This module manages the Proxy-Auto-Config Rule entries that can be found in the WEB-UI menu: 'Services - Web Proxy - Administration - Proxy Auto-Config - Rules (*DropDown*)' (*URL 'ui/proxy#subtab_pac_rules'*)

40.3 Definition

For basic parameters see: *Basic*

40.3.1 General

ansibleguy.opnsense.webproxy_general

Table 1: Definition

Parameter	Type	Required	Default	Aliases	Comment
enabled	boolean	false	true	-	En- or disable the proxy
errors	string	false	opnsense	error_pages	The proxy error pages can be altered, default layout uses OPNsense content, when Squid is selected the content for the selected language will be used (standard squid layout), Custom offers the possibility to upload your own theme content
icp_port	integer	false	-	icp	-
log	boolean	false	true	-	-
log_store	boolean	false	true	-	-
log_target	string	false	file	-	One of: 'file', 'file_extendend', 'file_json', 'syslog', 'syslog_json'. Send log data to the selected target. When syslog is selected, facility local 4 will be used to send messages of info level for these logs
log_ignore	list	false	-	-	Type subnets/addresses you want to ignore for the access.log
dns_servers	list	false	-	-	IPs of alternative DNS servers you like to use
use_via_header	boolean	false	true	-	If set (default), Squid will include a Via header in requests and replies as required by RFC2616
pinger	boolean	false	true	-	Toggles the Squid pinger service. This service is used in the selection of the best parent proxy
handling_forwardec	string	false	default	forwarded_fc forwarded_fc handle_ff	One of: 'default', 'on', 'off', 'transparent', 'delete', 'truncate'. Select what to do with X-Forwarded-For header. If set to: 'on', Squid will append your client's IP address in the HTTP requests it forwards. By default it looks like X-Forwarded-For: 192.1.2.3; If set to: 'off', it will appear as X-Forwarded-For: unknown; 'transparent', Squid will not alter the X-Forwarded-For header in any way; If set to: 'delete', Squid will delete the entire X-Forwarded-For header; If set to: 'truncate', Squid will remove all existing X-Forwarded-For entries, and place the client IP as the sole entry
handling_uri_white	string	false	strip	uri_whites uri_whites handle_uw	One of: 'strip', 'deny', 'allow', 'encode', 'chop'. Select what to do with URI that contain whitespaces. The current Squid implementation of encode and chop violates RFC2616 by not using a 301 redirect after altering the URL
hostname	string	false	-	visible_hostna	The hostname to be displayed in proxy server error messages
email	string	false	admin@loca	visible_email	The email address displayed in error messages to the users
suppress_version	boolean	false	false	-	Suppress Squid version string info in HTTP headers and HTML error pages
connect_timeout	integer	false	-		Between 1 and 120 seconds. This can help you when having connection issues with IPv6 enabled servers.
reload	boolean	false	true	-	If the running config should be reloaded on change - this may take some time. For mass-managing items you might want to reload it 'manually' after all changes are done => using the *ansibleguy.opnsense.reload* module.

ansibleguy.opnsense.webproxy_cache

Table 2: Definition

Parameter	Type	Required	Default	Aliases	Comment
memory_mb	integer	false	256	memory, mem	The cache memory size to use or zero to disable completely
size_mb	integer	false	100	size	The storage size for the local cache
directory	string	false	/var/squid/	dir	The location for the local cache
layer_1	integer	false	16	layer1, l1	The number of first-level subdirectories for the local cache
layer_2	integer	false	256	layer2, l2	The number of second-level subdirectories for the local cache
size_mb_max	integer	false	4	maximum_obje max_size	The maximum object size
memory_kb_max	integer	false	512	maximum_obje max_mem max_mem	The maximum object size
memory_cache_mod	string	false	default	cache_mo mode	One of: 'always', 'disk', 'network', 'default'. Controls which objects to keep in the memory cache (cache_mem) always: Keep most recently fetched objects in memory (default) disk: Only disk cache hits are kept in memory, which means an object must first be cached on disk and then hit a second time before cached in memory. network: Only objects fetched from network is kept in memory
cache_linux_pa	boolean	false	false	-	Enable or disable the caching of packages for linux distributions. This makes sense if you have multiple servers in your network and do not host your own package mirror. This will reduce internet traffic usage but increase disk access
cache_windows_	boolean	false	false	-	Enable or disable the caching of Windows updates. This makes sense if you don't have a WSUS server. If you can setup a WSUS server, this solution should be preferred
reload	boolean	false	true	-	If the running config should be reloaded on change - this may take some time. For mass-managing items you might want to reload it 'manually' after all changes are done => using the *ansibleguy.opnsense.reload* module.

ansibleguy.opnsense.webproxy_parent

Table 3: Definition

Parameter	Type	Required	Default	Aliases	Comment
enabled	boolean	false	true	-	En- or disable the parent-proxy
host	string	false	-	ip	Parent proxy IP address or hostname
auth	boolean	false	false	-	Enable authentication against the parent proxy
user	string	false	-	-	Set a username if parent proxy requires authentication
password	string	false	-	-	Set a username if parent proxy requires authentication
port	integer	false	-	p	-
local_domains	list	false	-	domains	Domains not to be sent via parent proxy
local_ips	list	false	-	ips	IP addresses not to be sent via parent proxy
reload	boolean	false	true	-	If the running config should be reloaded on change - this may take some time. For mass-managing items you might want to reload it 'manually' after all changes are done => using the *ansibleguy.opnsense.reload* module.

ansibleguy.opnsense.webproxy_traffic

Table 4: Definition

Parameter	Type	Required	Default	Aliases	Comment
enabled	boolean	false	true	-	En- or disable the traffic management
download_kb_max	integer	false	2048	download_max, download, dl_max, dl	The maximum size for downloads in kilobytes (leave empty to disable)
upload_kb_max	integer	false	1024	upload_max, upload, ul_max, ul	The maximum size for uploads in kilobytes (leave empty to disable)
throttle_kb_bandwid	integer	false	1024	throttle_bandw throttle_bw, bandwidth, bw	The allowed overall bandwidth in kilobits per second (leave empty to disable)
throttle_kb_host_ban	integer	false	256	throttle_host_b throttle_host_b host_band host_bw	The allowed per host bandwidth in kilobits per second (leave empty to disable)
reload	boolean	false	true	-	If the running config should be reloaded on change - this may take some time. For mass-managing items you might want to reload it 'manually' after all changes are done => using the *ansibleguy.opnsense.reload* module.

40.3.2 Forward

ansibleguy.opnsense.webproxy_forward

Table 5: Definition

Parameter	Type	Required	Default	Aliases	Comment
transparent	boolean	false	false	transparent_mode	Enable transparent proxy mode. You will need a firewall rule to forward traffic from the firewall to the proxy server. You may leave the proxy interfaces empty, but remember to set a valid ACL in that case
ssl_inspection	boolean	false	false	ssl_inspec ssl	Enable SSL inspection mode, which allows to log HTTPS connections information, such as requested URL and/or make the proxy act as a man in the middle between the internet and your clients. Be aware of the security implications before enabling this option. If you plan to use transparent HTTPS mode, you need nat rules to reflect your traffic
ssl_inspection_s	boolean	false	false	ssl_sni_on	Do not decode and/or filter SSL content, only log requested domains and IP addresses. Some old servers may not provide SNI, so their addresses will not be indicated
interfaces	list	false	['lan']	ints	Interface(s) the proxy will bind to
allow_interface_s	boolean	false	true	allow_subne	When enabled the subnets of the selected interfaces will be added to the allow access list
port	integer	false	3128	p	-
port_ssl	integer	false	3129	p_ssl	-
ssl_ca	string	false	-	ca	Select a Certificate Authority to use
ssl_exclude	list	false	-	-	A list of sites which may not be inspected, for example bank sites. Prefix the domain with a . to accept all subdomains (e.g. .google.com)
ssl_cache_mb	integer	false	4	ssl_cache, cache	The maximum size (in MB) to use for SSL certificates
ssl_workers	integer	false	5	workers	The number of ssl certificate workers to use (sslcrtd_children)
snmp	boolean	false	false	-	Enable or disable the squid SNMP Agent
port_snmp	integer	false	3401	p_snmp	-
snmp_password	string	false	public	snmp_con snmp_pwc	The password for access to SNMP agent
interfaces_ftp	list	false	-	ints_ftp	Interface(s) the ftp proxy will bind to
port_ftp	integer	false	2121	p_ftp	-
transparent_ftp	boolean	false	false	-	Enable transparent ftp proxy mode to forward all requests or destination port 21 to the proxy server without any additional configuration
reload	boolean	false	true	-	If the running config should be reloaded on change - this may take some time. For mass-managing items you might want to reload it 'manually' after all changes are done => using the *ansibleguy.opnsense.reload* module.

ansibleguy.opnsense.webproxy_acl

Table 6: Definition

Parameter	Type	Required	Default	Aliases	Comment
allow	list	false	-	allow_subne subnets	IPs and Subnets you want to allow access to the proxy server
exclude	list	false	-	unrestricted, ignore	IPs and Subnets you want to bypass the proxy server
banned	list	false	-	blocked, block, ban	IPs and Subnets you want to deny access to the proxy server
exclude_domains	list	false	-	safe_list, whitelist	You may use a regular expression, use a comma or press Enter for new item. Examples: 'mydomain.com' matches on '*.mydomain.com'; '^https?:\/\/([a-zA-Z]+)\.mydomain\.' matches on 'http(s)://textONLY.mydomain.*'; '\.gif$' matches on '*.gif' but not on '*.gif\test'; '\[0-9]+\.gif$' matches on '\123.gif' but not on '\test.gif'
block_domains	list	false	-	block, block_list, blacklist	You may use a regular expression, use a comma or press Enter for new item. Examples: 'mydomain.com' matches on '*.mydomain.com'; '^https?:\/\/([a-zA-Z]+)\.mydomain\.' matches on 'http(s)://textONLY.mydomain.*'; '\.gif$' matches on '*.gif' but not on '*.gif\test'; '\[0-9]+\.gif$' matches on '\123.gif' but not on '\test.gif'
block_user_agen	list	false	-	block_ua, block_list_	Block user-agents. You may use a regular expression, use a comma or press Enter for new item. Examples: '^(.)+Macintosh(.)+Firefox/37\.0' matches on 'Macintosh version of Firefox revision 37.0'; '^Mozilla' matches on 'all Mozilla based browsers'
block_mime_typ	list	false	-	block_mir block_list_	Block specific MIME type reply. You may use a regular expression, use a comma or press Enter for new item. Examples: 'video/flv' matches on 'Flash Video'; 'application/x-javascript' matches on 'javascripts'
exclude_google	list	false	-	safe_list_g	The domain that will be allowed to use Google GSuite. All accounts that are not in this domain will be blocked to use it
youtube_filter	string	false	-	youtube	One of: 'strict', 'moderate'. Youtube filter level
ports_tcp	list	false	['80:http', '21:ftp', '443:https '70:gopher', '210:wais' '1025-65535:unr ports', '280:http-mgmt', '488:gss-http', '591:file	p_tcp	Allowed destination TCP ports, you may use ranges (ex. 222-226) and add comments with colon (ex. 22:ssh)

ansibleguy.opnsense.webproxy_icap

Table 7: Definition

Parameter	Type	Required	Default	Aliases	Comment
enabled	boolean	false	true	-	If this checkbox is checked, you can use an ICAP server to filter or replace content
request_url	string	false	icap://[::1]	request, request_targ	The url where the REQMOD requests should be sent to
response_url	string	false	icap://[::1]	response, response_tar	The url where the RESPMOD requests should be sent to
ttl	integer	false	60	default_ttl	-
send_client_ip	boolean	false	true	send_clier	If you enable this option, the client IP address will be sent to the ICAP server. This can be useful if you want to filter traffic based on IP addresses
send_username	boolean	false	false	send_user	If you enable this option, the username of the client will be sent to the ICAP server. This can be useful if you want to filter traffic based on usernames. Authentication is required to use usernames
encode_username	boolean	false	false	user_enco encode_user enc_user	Use this option if your usernames need to be encoded
header_usernam	string	false	X-Username	header_us user_head	The header which should be used to send the username to the ICAP server
preview	boolean	false	true	-	If you use previews, only a part of the data is sent to the ICAP server. Setting this option can improve the performance
preview_size	integer	false	1024	-	Size of the preview which is sent to the ICAP server
exclude	list	false	-	-	Exclusion list destination domains.You may use a regular expression, use a comma or press Enter for new item. Examples: 'mydomain.com' matches on '*.mydomain.com'; 'https://({[}a-zA-Z{]}+)\.mydomain\.' matches on 'http(s)://textONLY.mydomain.*'; '\.gif$' matches on '*.gif' but not on '*.gif\test'; '\[0-9]+\.gif$' matches on '\123.gif' but not on '\test.gif'
reload	boolean	false	true	-	If the running config should be reloaded on change - this may take some time. For mass-managing items you might want to reload it 'manually' after all changes are done => using the *ansibleguy.opnsense.reload* module.

ansibleguy.opnsense.webproxy_auth

Table 8: Definition

Parameter	Type	Required	Default	Aliases	Comment
method	string	false	-	type, target	The authentication backend to use - as shown in the WEB-UI at 'System - Access - Servers'. Per example: 'Local Database'
group	string	false	-	local_group	Restrict access to users in the selected (local)group. **NOTE**: please be aware that users (or vouchers) which aren't administered locally will be denied when using this option
prompt	string	false	OPNsense proxy authentication	realm	The prompt will be displayed in the authentication request window
group	string	false	-	local_group	
ttl_h	integer	false	2	ttl, ttl_hours, credential_ttl	This specifies for how long (in hours) the proxy server assumes an externally validated username and password combination is valid (Time To Live). When the TTL expires, the user will be prompted for credentials again
processes	integer	false	5	proc	The total number of authenticator processes to spawn
reload	boolean	false	true	-	If the running config should be reloaded on change - this may take some time. For mass-managing items you might want to reload it 'manually' after all changes are done => using the *ansibleguy.opnsense.reload* module.

40.3.3 Remote ACL

ansibleguy.opnsense.webproxy_remote_acl

Table 9: Definition

Parameter	Type	Required	Default	Aliases	Comment
file	string	true	-	filename	Unique file-name to store the remote acl in. Used to match existing entries with configured ones
url	string	false for state changes, else true	-	-	Url to fetch the acl from
description	string	false for state changes, else true	-	desc	A description to explain what this blacklist is intended for
username	string	false	-	user	Optional user for authentication
password	string	false	-	pwd	Optional password for authentication
categories	list	false	-	cat, filter	Select categories to use, leave empty for all. Categories are visible in the WEB-UI after initial download
verify_ssl	boolean	false	true	verify	If certificate validation should be done - relevant if self-signed certificates are used on the target server!
reload	boolean	false	true	-	If the running config should be reloaded on change - this may take some time. For mass-managing items you might want to reload it 'manually' after all changes are done => using the *ansibleguy.opnsense.reload* module.

40.3.4 Proxy Auto-Config

ansibleguy.opnsense.webproxy_pac_proxy

Table 10: Definition

Parameter	Type	Required	Default	Aliases	Comment
name	string	true	-	-	Unique name for the proxy
url	string	false for state changes, else true	-	-	A proxy URL in the form proxy.example.com:3128
type	string	false	proxy	-	One of: 'proxy', 'direct', 'http', 'https', 'socks', 'socks4', 'socks5'. Usually you should use 'direct' for a direct connection or 'proxy' for a Proxy
description	string	false	-	desc	-
reload	boolean	false	true	-	If the running config should be reloaded on change - this may take some time. For mass-managing items you might want to reload it 'manually' after all changes are done => using the *ansibleguy.opnsense.reload* module.

ansibleguy.opnsense.webproxy_pac_match

Table 11: Definition

Parameter	Type	Required	Default	Aliases	Comment
name	string	true	-	-	Unique name for the match
description	string	false	-	desc	-
negate	boolean	false	false	-	Negate this match. For example you can match if a host is not inside a network
type	string	false	url_match	-	One of: 'url_matches', 'hostname_matches', 'dns_domain_is', 'destination_in_net', 'my_ip_in_net', 'plain_hostname', 'is_resolvable', 'dns_domain_levels', 'weekday_range', 'date_range', 'time_range'. The type of the match. Depending on the match, you will need different arguments
hostname	string	false	-	-	A hostname pattern like *.opnsense.org
url	string	false	-	-	A URL pattern like forum.opnsense.org/index*
network	string	false	-	-	The network address to match in CIDR notation for example like 127.0.0.1/8 or ::1/128
domain_level_from	integer	false	0	domain_from	The minimum amount of dots in the domain name
domain_level_to	integer	false	0	domain_to	The maximum amount of dots in the domain name
hour_from	integer	false	0	time_from	Start hour for match-period
hour_to	integer	false	0	time_to	End hour for match-period
month_from	integer	false	1	date_from	Start month for match-period
month_to	integer	false	1	date_to	End hour month match-period
weekday_from	integer	false	1	day_from	Start weekday for match-period. 1 = monday, 7 = sunday
weekday_to	integer	false	1	day_to	End hour weekday match-period. 1 = monday, 7 = sunday
reload	boolean	false	true	-	If the running config should be reloaded on change - this may take some time. For mass-managing items you might want to reload it 'manually' after all changes are done => using the *ansibleguy.opnsense.reload* module.

ansibleguy.opnsense.webproxy_pac_rule

Table 12: Definition

Parameter	Type	Required	Default	Aliases	Comment
description	string	true	-	desc, name	Unique description used to identify existing rules
matches	list	false for state changes, else true	-	-	Matches you want to use in this rule. This matches are joined using the selected separator
proxies	list	false for state changes, else true	-	-	Proxies you want to use address using this rule
join_type	string	false	and	join	One of: 'and', 'or'. A separator to join the matches. 'or' means any match can be true which can be used to configure the same proxy for multiple networks while 'and' means all matches must be true which can be used to assign the proxy in a more detailed way
match_type	string	false	if	operator	One of: 'if', 'unless'. Choose 'if' in case any case you want to ensure a match to evaluate as is, else choose 'unless' if you want the negated version. Unless is used if you want to use the proxy for every host but not for some special ones
enabled	boolean	false	true	-	En- or disable the rule
reload	boolean	false	true	-	If the running config should be reloaded on change - this may take some time. For mass-managing items you might want to reload it 'manually' after all changes are done => using the *ansibleguy.opnsense.reload* module.

40.4 Examples

40.4.1 General

ansibleguy.opnsense.webproxy_general

```
- hosts: localhost
  gather_facts: no
  module_defaults:
    group/ansibleguy.opnsense.all:
      firewall: 'opnsense.template.ansibleguy.net'
      api_credential_file: '/home/guy/.secret/opn.key'

    ansibleguy.opnsense.list:
      target: 'webproxy_general'
```

(continues on next page)

(continued from previous page)

```
  tasks:
    - name: Example
      ansibleguy.opnsense.webproxy_general:
        # errors: 'opnsense'
        # icp_port: ''
        # log: true
        # log_store: true
        # log_target: 'file'
        # log_ignore: []
        # dns_servers: []
        # use_via_header: true
        # suppress_version: false
        # pinger: true
        # hostname: ''
        # connect_timeout: ''
        # email: 'admin@localhost.local'
        # handling_forwarded_for: 'default'
        # handling_uri_whitespace: 'strip'
        # errors: 'opnsense'
        # enabled: true
        # reload: true
        # debug: false

    - name: Pulling settings
      ansibleguy.opnsense.list:
      #  target: 'webproxy_general'
      register: current_config

    - name: Printing settings
      ansible.builtin.debug:
        var: current_config.data
```

ansibleguy.opnsense.webproxy_cache

```
- hosts: localhost
  gather_facts: no
  module_defaults:
    group/ansibleguy.opnsense.all:
      firewall: 'opnsense.template.ansibleguy.net'
      api_credential_file: '/home/guy/.secret/opn.key'

    ansibleguy.opnsense.list:
      target: 'webproxy_cache'

  tasks:
    - name: Example
      ansibleguy.opnsense.webproxy_cache:
        # memory_mb: 256
        # size_mb: 100
        # directory: '/var/squid/cache'
```

(continues on next page)

(continued from previous page)

```
        # layer_1: 16
        # layer_2: 256
        # size_mb_max: 4
        # memory_kb_max: 512
        # memory_cache_mode: 'default'
        # cache_linux_packages: false
        # cache_windows_updates: false
        # reload: true
        # debug: false

    - name: Pulling settings
      ansibleguy.opnsense.list:
      #  target: 'webproxy_cache'
      register: current_config

    - name: Printing settings
      ansible.builtin.debug:
        var: current_config.data
```

ansibleguy.opnsense.webproxy_parent

```
- hosts: localhost
  gather_facts: no
  module_defaults:
    group/ansibleguy.opnsense.all:
      firewall: 'opnsense.template.ansibleguy.net'
      api_credential_file: '/home/guy/.secret/opn.key'

    ansibleguy.opnsense.list:
      target: 'webproxy_parent'

  tasks:
    - name: Example
      ansibleguy.opnsense.webproxy_parent:
        # host: ''
        # auth: false
        # user: ''
        # password: ''
        # port: ''
        # local_domains: []
        # local_ips: []
        # enabled: true
        # reload: true
        # debug: false

    - name: Pulling settings
      ansibleguy.opnsense.list:
      #  target: 'webproxy_parent'
      register: current_config
```

(continues on next page)

(continued from previous page)

```
    - name: Printing settings
      ansible.builtin.debug:
        var: current_config.data
```

ansibleguy.opnsense.webproxy_traffic

```
- hosts: localhost
  gather_facts: no
  module_defaults:
    group/ansibleguy.opnsense.all:
      firewall: 'opnsense.template.ansibleguy.net'
      api_credential_file: '/home/guy/.secret/opn.key'

    ansibleguy.opnsense.list:
      target: 'webproxy_traffic'

  tasks:
    - name: Example
      ansibleguy.opnsense.webproxy_traffic:
        # download_kb_max: 2048
        # upload_kb_max: 1024
        # throttle_kb_bandwidth: 1024
        # throttle_kb_host_bandwidth: 256
        # enabled: true
        # reload: true
        # debug: false

    - name: Pulling settings
      ansibleguy.opnsense.list:
      #  target: 'webproxy_traffic'
      register: current_config

    - name: Printing settings
      ansible.builtin.debug:
        var: current_config.data
```

40.4.2 Forward

ansibleguy.opnsense.webproxy_forward

```
- hosts: localhost
  gather_facts: no
  module_defaults:
    group/ansibleguy.opnsense.all:
      firewall: 'opnsense.template.ansibleguy.net'
      api_credential_file: '/home/guy/.secret/opn.key'

    ansibleguy.opnsense.list:
      target: 'webproxy_forward'
```

(continues on next page)

(continued from previous page)

```
  tasks:
    - name: Example
      ansibleguy.opnsense.webproxy_forward:
        # interfaces: ['lan']
        # port: 3238
        # port_ssl: 3239
        # transparent: false
        # ssl_inspection: false
        # ssl_inspection_sni_only: false
        # ssl_ca: ''
        # ssl_exclude: []
        # ssl_cache_mb: 4
        # ssl_workers: 5
        # allow_interface_subnets: true
        # snmp: true
        # port_snmp: 3401
        # snmp_password: 'public'
        # interfaces_ftp: []
        # port_ftp: 2121
        # transparent_ftp: false
        # reload: true
        # debug: false
```

ansibleguy.opnsense.webproxy_acl

```
- hosts: localhost
  gather_facts: no
  module_defaults:
    group/ansibleguy.opnsense.all:
      firewall: 'opnsense.template.ansibleguy.net'
      api_credential_file: '/home/guy/.secret/opn.key'

    ansibleguy.opnsense.list:
      target: 'webproxy_acl'

  tasks:
    - name: Example
      ansibleguy.opnsense.webproxy_acl:
        # allow: []
        # exclude: []
        # banned: []
        # exclude_domains: []
        # block_domains: []
        # block_user_agents: []
        # block_mime_types: []
        # exclude_google: []
        # youtube_filter: ''
        # ports_tcp: ['80:http', '21:ftp', '443:https', '70:gopher', '210:wais', '1025-
↪65535:unregistered ports', '280:http-mgmt', '488:gss-http', '591:filemaker', '777:multiling␣
```

(continues on next page)

(continued from previous page)

```
↪http']
        # ports_ssl: ['443:https']
        # reload: true
        # debug: false

    - name: Configuring
      ansibleguy.opnsense.webproxy_acl:
        allow: ['192.168.0.0/24', '172.16.1.0/29', '172.16.0.5']
        exclude: ['192.168.2.0/28', '172.16.1.5']
        banned: ['172.16.3.0/24', '172.16.2.5']
        exclude_domains: ['ansibleguy.net']
        block_domains: ['ansibleguy.com']
        block_user_agents: ['test1', 'test2']
        block_mime_types: ['video/flv', 'test']
        ports_tcp: ['80:http', '21:ftp']
        ports_ssl: ['443:https', '8443:random']
        youtube_filter: 'moderate'

    - name: Pulling settings
      ansibleguy.opnsense.list:
      #  target: 'webproxy_acl'
      register: current_config

    - name: Printing settings
      ansible.builtin.debug:
        var: current_config.data
```

ansibleguy.opnsense.webproxy_icap

```
- hosts: localhost
  gather_facts: no
  module_defaults:
    group/ansibleguy.opnsense.all:
      firewall: 'opnsense.template.ansibleguy.net'
      api_credential_file: '/home/guy/.secret/opn.key'

    ansibleguy.opnsense.list:
      target: 'webproxy_icap'

  tasks:
    - name: Example
      ansibleguy.opnsense.webproxy_icap:
        # request_url: 'icap://[::1]:1344/avscan'
        # response_url: 'icap://[::1]:1344/avscan'
        # ttl: 60
        # send_client_ip: true
        # send_username: false
        # encode_username: false
        # header_username: 'X-Username'
        # preview: true
```

(continues on next page)

(continued from previous page)

```
        # preview_size: 1024
        # exclude: []
        # enabled: true
        # reload: true
        # debug: false

    - name: Pulling settings
      ansibleguy.opnsense.list:
      #  target: 'webproxy_icap'
      register: current_config

    - name: Printing settings
      ansible.builtin.debug:
        var: current_config.data
```

ansibleguy.opnsense.webproxy_auth

```
- hosts: localhost
  gather_facts: no
  module_defaults:
    group/ansibleguy.opnsense.all:
      firewall: 'opnsense.template.ansibleguy.net'
      api_credential_file: '/home/guy/.secret/opn.key'

    ansibleguy.opnsense.list:
      target: 'webproxy_auth'

  tasks:
    - name: Example
      ansibleguy.opnsense.webproxy_auth:
        # method: ''
        # group: ''
        # prompt: 'OPNsense proxy authentication'
        # ttl_h: 2
        # processes: 5
        # reload: true
        # debug: false

    - name: Pulling settings
      ansibleguy.opnsense.list:
      #  target: 'webproxy_auth'
      register: current_config

    - name: Printing settings
      ansible.builtin.debug:
        var: current_config.data
```

40.4.3 Remote ACL

ansibleguy.opnsense.webproxy_remote_acl

```yaml
- hosts: localhost
  gather_facts: no
  module_defaults:
    group/ansibleguy.opnsense.all:
      firewall: 'opnsense.template.ansibleguy.net'
      api_credential_file: '/home/guy/.secret/opn.key'

    ansibleguy.opnsense.list:
      target: 'webproxy_remote_acl'

  tasks:
    - name: Example
      ansibleguy.opnsense.webproxy_remote_acl:
        file: 'example'
        url: 'https://example.ansibleguy.net/rac1'
        description: 'example ACL'
        # categories: []
        # username: ''
        # password: ''
        # verify_ssl: true
        # enabled: true
        # reload: true
        # debug: false

    - name: Adding
      ansibleguy.opnsense.webproxy_remote_acl:
        file: 'test1'
        url: 'https://test.lan/rac1'
        username: 'random'
        password: 'random'
        verify_ssl: true
        description: 'test'

    - name: Disabling
      ansibleguy.opnsense.webproxy_remote_acl:
        file: 'test1'
        url: 'https://test.lan/rac2'
        username: 'random'
        password: 'random2'
        description: 'Custom ACL'
        enabled: false

    - name: Pulling settings
      ansibleguy.opnsense.list:
      #  target: 'webproxy_remote_acl'
      register: existing_entries

    - name: Printing settings
      ansible.builtin.debug:
```

(continues on next page)

(continued from previous page)

```
        var: existing_entries.data

    - name: Removing
      ansibleguy.opnsense.webproxy_remote_acl:
        file: 'test1'
        state: 'absent'
```

40.4.4 Proxy Auto-Config

ansibleguy.opnsense.webproxy_pac_proxy

```
- hosts: localhost
  gather_facts: no
  module_defaults:
    group/ansibleguy.opnsense.all:
      firewall: 'opnsense.template.ansibleguy.net'
      api_credential_file: '/home/guy/.secret/opn.key'

    ansibleguy.opnsense.list:
      target: 'webproxy_pac_proxy'

  tasks:
    - name: Example
      ansibleguy.opnsense.webproxy_pac_proxy:
        name: 'example'
        url: 'example.ansibleguy.net:3128'
        # type: 'proxy'
        # description: ''
        # reload: true
        # debug: false

    - name: Adding
      ansibleguy.opnsense.webproxy_pac_proxy:
        name: 'test1'
        url: 'test.lan:3128'
        description: 'test'

    - name: Pulling settings
      ansibleguy.opnsense.list:
      #  target: 'webproxy_pac_proxy'
      register: existing_entries

    - name: Printing settings
      ansible.builtin.debug:
        var: existing_entries.data

    - name: Removing
      ansibleguy.opnsense.webproxy_pac_proxy:
        file: 'test1'
        state: 'absent'
```

ansibleguy.opnsense.webproxy_pac_match

```yaml
- hosts: localhost
  gather_facts: no
  module_defaults:
    group/ansibleguy.opnsense.all:
      firewall: 'opnsense.template.ansibleguy.net'
      api_credential_file: '/home/guy/.secret/opn.key'

    ansibleguy.opnsense.list:
      target: 'webproxy_pac_match'

  tasks:
    - name: Example
      ansibleguy.opnsense.webproxy_pac_match:
        name: 'example'
        # type: 'url_matches'
        # description: ''
        # negate: false
        # hostname: ''
        # url: ''
        # network: ''
        # domain_level_from: 0
        # domain_level_to: 0
        # hour_from: 0
        # hour_to: 0
        # month_from: 1
        # month_to: 1
        # weekday_from: 1
        # weekday_to: 1
        # reload: true
        # debug: false

    - name: Adding hostname match
      ansibleguy.opnsense.webproxy_pac_match:
        hostname: 'test.ansibleguy.net'
        type: 'hostname_matches'
        description: 'test'

    - name: Adding time match
      ansibleguy.opnsense.webproxy_pac_match:
        hostname: 'test.ansibleguy.net'
        description: 'working hours'
        type: 'time_range'
        hour_from: 6
        hour_to: 18

    - name: Pulling settings
      ansibleguy.opnsense.list:
      #  target: 'webproxy_pac_match'
      register: existing_entries

    - name: Printing settings
```

(continues on next page)

(continued from previous page)

```
      ansible.builtin.debug:
        var: existing_entries.data

    - name: Removing
      ansibleguy.opnsense.webproxy_pac_match:
        file: 'test1'
        state: 'absent'
```

ansibleguy.opnsense.webproxy_pac_rule

```
- hosts: localhost
  gather_facts: no
  module_defaults:
    group/ansibleguy.opnsense.all:
      firewall: 'opnsense.template.ansibleguy.net'
      api_credential_file: '/home/guy/.secret/opn.key'

    ansibleguy.opnsense.list:
      target: 'webproxy_pac_rule'

  tasks:
    - name: Example
      ansibleguy.opnsense.webproxy_pac_rule:
        description: 'example'
        matches: []
        proxies: []
        # join_type: 'and'
        # match_type: 'if'
        # reload: true
        # debug: false

    - name: Adding - linking to existing match & proxy
      ansibleguy.opnsense.webproxy_pac_rule:
        description: 'test_rule'
        matches: ['test_match']
        proxies: ['test_proxy']
        join_type: 'and'
        match_type: 'unless'

    - name: Pulling settings
      ansibleguy.opnsense.list:
      #  target: 'webproxy_pac_rule'
      register: existing_entries

    - name: Printing settings
      ansible.builtin.debug:
        var: existing_entries.data

    - name: Removing
      ansibleguy.opnsense.webproxy_pac_rule:
```

(continues on next page)

(continued from previous page)

```
        file: 'test_rule'
        state: 'absent'
```

CHAPTER

FORTYONE

WIREGUARD

STATE: stable

TESTS: wireguard_server | wireguard_peer | wireguard_general | wireguard_show

API Docs: Plugin - Wireguard

Service Docs: WireGuard - Site to Site | WireGuard - Client to Site

41.1 Definition

For basic parameters see: Basics

41.1.1 ansibleguy.opnsense.wireguard_server

Parameter	Type	Require	Default value	Aliases	Comment
name	string	true	-	-	The unique name of the local WireGuard server instance
peers	list	false	-	clients	List of existing peers that
allowed	list	false	-	tunnel_ips, tunnel_ip, tunneladdress, tunnel_adresses, tunnel_address, addresses, address, allowed	One or multiple IP addresses that are used inside the tunnel
public_ke	string	false	-	pubkey, pub	Optionally provide an existing WireGuard Public Key. If none is provided - a key-pair will be generated automatically or the existing one will be used.
private_	string	false	-	privkey, priv	Optionally provide an existing WireGuard Private Key. If none is provided - a key-pair will be generated automatically or the existing one will be used.
port	integer	false	-	-	Optionally provide a port for the server instance. Needed if dynamic peers will connect to this instance!
mtu	integer	false	1420	-	Integer between 1 and 9300
dns_s	list	false	-	dns	List of DNS servers that will be used to resolve peer endpoint-names
disable_	boole	false	false	disableroutes	If automatically created routes should be disabled. Needs to be set if you want to use policy-based routing, dynamic routing or 'manually' created static routes
gateway	string	false	-	gw	IP address to use as gateway. Can only be used if you enable the 'disable_routes' option.
vip	string	false	-	vip_depend, carp, carp_depend	The Virtual-CARP-IP (CARP VHID) to depend on. When this virtual address is not in master state, then the instance will be shutdown
reload	boole	false	true	-	If the running config should be reloaded on change - this will take some time. For mass-managing items you might want to reload it 'manually' after all changes are done => using the reload module.

41.1.2 ansibleguy.opnsense.wireguard_peer

Parameter	Type	Required	Default value	Aliases	Comment
name	string	true	-	-	The unique name of the local WireGuard peer
endpoint	string	false	-	server_address, serveraddress, target, server	Peer endpoint IP address or DNS-hostname
allowed	list	false for state changes, else true	-	tunnel_ips, tunnel_ip, tunneladdress, tunnel_adresses, tunnel_address, addresses, address, allowed	One or multiple IP addresses used by the peer inside the tunnel
public_k	string	false for state changes, else true	-	pubkey, pub	Provide the WireGuard Public Key of the peer. Used to identify the peer
psk	string	false	-	-	Optionally provide an PSK. The pre-shared key (PSK) is an optional security improvement as per the WireGuard protocol and should be a unique PSK per client for highest security.
port	integer	false	-	-	Optionally provide the port of the peer instance
keepa	integer	false	-	-	Integer between 1 and 86400. Should be used if one of the connection-members is behind NAT
reloa	boole	false	true	-	If the running config should be reloaded on change - this will take some time. For mass-managing items you might want to reload it 'manually' after all changes are done => using the reload module.

41.1.3 ansibleguy.opnsense.wireguard_show

Will return the information seen at the `VPN - Wireguard - Diagnostics` page

41.1.4 ansibleguy.opnsense.wireguard_general

Parameter	Type	Required	Default value	Aliases	Comment
enabled	boolean	false	true	-	Used to enable or disable the wireguard service

41.2 Usage

To make a dynamic WireGuard endpoint to re-connect you may want to create a gateway monitoring (*dpinger*) targeting the remote tunnel-address.

41.3 Examples

41.3.1 ansibleguy.opnsense.wireguard_general

```yaml
- hosts: localhost
  gather_facts: no
  module_defaults:
    group/ansibleguy.opnsense.all:
      firewall: 'opnsense.template.ansibleguy.net'
      api_credential_file: '/home/guy/.secret/opn.key'

  tasks:
    - name: Example
      ansibleguy.opnsense.wireguard_general:
        # enabled: true

    - name: Enabling WireGuard service
      ansibleguy.opnsense.wireguard_general:
        enabled: true
```

41.3.2 ansibleguy.opnsense.wireguard_show

```yaml
- hosts: localhost
  gather_facts: no
  module_defaults:
    group/ansibleguy.opnsense.all:
      firewall: 'opnsense.template.ansibleguy.net'
      api_credential_file: '/home/guy/.secret/opn.key'

  tasks:
    - name: Example
      ansibleguy.opnsense.wireguard_show:
      register: wg_status

    - name: Printing
      ansible.builtin.debug:
        var: wg_status.data
```

41.3.3 ansibleguy.opnsense.wireguard_peer

```yaml
- hosts: localhost
  gather_facts: no
  module_defaults:
    group/ansibleguy.opnsense.all:
      firewall: 'opnsense.template.ansibleguy.net'
      api_credential_file: '/home/guy/.secret/opn.key'

    ansibleguy.opnsense.list:
      target: 'wireguard_peer'

  tasks:
    - name: Example
      ansibleguy.opnsense.wireguard_peer:
        name: 'example'
        # allowed_ips: []
        # enpoint: ''
        # port: ''
        # public_key: ''
        # psk: ''
        # keepalive: ''
        # enabled: true
        # debug: false
        # state: 'present'
        # reload: true

    - name: Adding peer
      ansibleguy.opnsense.wireguard_peer:
        name: 'test1'
        endpoint: 'wg.template.ansibleguy.net'
        allowed_ips: ['10.200.0.1/32']
        public_key: 'gTuhGXA28/qRSLPnH3szr2+A4l3C4tKlUsOORV63+SE='

    - name: Disabling peer
      ansibleguy.opnsense.wireguard_peer:
        name: 'test1'
        enabled: false

    - name: Listing
      ansibleguy.opnsense.list:
      #  target: 'wireguard_peer'
      register: existing_entries

    - name: Printing peers
      ansible.builtin.debug:
        var: existing_entries.data

    - name: Removing peer
      ansibleguy.opnsense.wireguard_peer:
        name: 'test1'
        state: 'absent'
```

41.3.4 ansibleguy.opnsense.wireguard_server

```yaml
- hosts: localhost
  gather_facts: no
  module_defaults:
    group/ansibleguy.opnsense.all:
      firewall: 'opnsense.template.ansibleguy.net'
      api_credential_file: '/home/guy/.secret/opn.key'

    ansibleguy.opnsense.list:
      target: 'wireguard_server'

  tasks:
    - name: Example
      ansibleguy.opnsense.wireguard_server:
        name: 'example'
        # allowed_ips: []
        # peers: []
        # port: ''
        # public_key: ''
        # private_key: ''
        # mtu: 1420
        # dns_servers: []
        # disable_routes: false
        # vip: ''
        # gateway: ''
        # enabled: true
        # debug: false
        # state: 'present'
        # reload: true

    - name: Adding server
      ansibleguy.opnsense.wireguard_server:
        name: 'test1'
        allowed_ips: ['10.200.0.1/32']
        peers: ['peer1']
        port: 51820
        vip: '192.168.2.1'

    - name: Disabling server
      ansibleguy.opnsense.wireguard_server:
        name: 'test1'
        enabled: false

    - name: Listing
      ansibleguy.opnsense.list:
      #  target: 'wireguard_server'
      register: existing_entries

    - name: Printing servers
      ansible.builtin.debug:
        var: existing_entries.data
```

(continues on next page)

(continued from previous page)

```
- name: Removing server
  ansibleguy.opnsense.wireguard_server:
    name: 'test1'
    state: 'absent'
```