

---

# **Ansible Collection - OPNSense**

**AnsibleGuy**

**Mar 19, 2023**



## USAGE

<b>1</b>	<b>1 - Installation</b>	<b>3</b>
<b>2</b>	<b>2 - Basic</b>	<b>5</b>
<b>3</b>	<b>3 - Troubleshoot</b>	<b>9</b>
<b>4</b>	<b>4 - Develop</b>	<b>11</b>
<b>5</b>	<b>1 - Basic module arguments</b>	<b>15</b>
<b>6</b>	<b>2 - List</b>	<b>17</b>
<b>7</b>	<b>2 - Reload</b>	<b>19</b>
<b>8</b>	<b>Alias</b>	<b>21</b>
<b>9</b>	<b>Alias - Mass Management</b>	<b>25</b>
<b>10</b>	<b>DNS - BIND</b>	<b>29</b>
<b>11</b>	<b>Cron Jobs</b>	<b>45</b>
<b>12</b>	<b>FRR BFD</b>	<b>49</b>
<b>13</b>	<b>FRR BGP</b>	<b>53</b>
<b>14</b>	<b>FRR Diagnostic</b>	<b>67</b>
<b>15</b>	<b>FRR OSPF</b>	<b>69</b>
<b>16</b>	<b>FRR RIP</b>	<b>85</b>
<b>17</b>	<b>Interface</b>	<b>89</b>
<b>18</b>	<b>IPSec</b>	<b>97</b>
<b>19</b>	<b>Monit</b>	<b>107</b>
<b>20</b>	<b>Package</b>	<b>117</b>
<b>21</b>	<b>Route</b>	<b>121</b>
<b>22</b>	<b>Rule</b>	<b>125</b>

<b>23 Rule - Mass Management</b>	<b>133</b>
<b>24 Firewall Savepoint</b>	<b>139</b>
<b>25 Service</b>	<b>143</b>
<b>26 Traffic Shaper</b>	<b>145</b>
<b>27 Source NAT</b>	<b>155</b>
<b>28 Syslog</b>	<b>161</b>
<b>29 System</b>	<b>165</b>
<b>30 DNS - Unbound - Domain Override</b>	<b>169</b>
<b>31 DNS - Unbound - DNS-over-TLS</b>	<b>173</b>
<b>32 DNS - Unbound - Forwarding</b>	<b>175</b>
<b>33 DNS - Unbound - Host Override</b>	<b>177</b>
<b>34 DNS - Unbound - Host Alias</b>	<b>181</b>
<b>35 Web Proxy</b>	<b>185</b>
<b>36 WireGuard</b>	<b>211</b>

---

**Tip:** Check out the repository on [GitHub](#)

---

---

**Tip:** Check out the repository on [GitHub](#)

---



## 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 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](#)

---





## 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
    6637646636323437333437353336316332663133316435650a366439336665383763376432653736
```

(continues on next page)

(continued from previous page)

```
32313332363032646436626230646461376532666366663265373663316331316664336134366338
6531363362613039330a316436386533393636623837653163333564383232313363666361643730
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](#)

---



## 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, 'headers':
↳ Headers({'content-type': 'application/json; charset=UTF-8', 'content-length': '73',
↳ 'date': 'Tue, 30 Aug 2022 15:17:57 GMT', 'server': 'OPNsense'}), '_request': <Request(
↳ 'POST', 'https://FIREWALL/api/firewall/filter/addRule')>, 'next_request': None,
↳ 'extensions': {'http_version': b'HTTP/1.1', 'reason_phrase': b'OK', 'network_stream':
↳ <httpcore.backends.sync.SyncStream object at 0x7f7efa1975b0>}, 'history': [], 'is_
↳ closed': True, 'is_stream_consumed': True, 'default_encoding': 'utf-8', 'stream':
↳ <httpx._client.BoundSyncStream object at 0x7f7efa1b28e0>, '_num_bytes_downloaded': 73,
↳ '_decoder': <httpx._decoders.IdentityDecoder object at 0x7f7efa139190>, '_elapsed':
↳ datetime.timedelta(microseconds=189718), '_content': b'{"result":"failed",\
↳ "validations":{"rule.interface":"option not in list"}}', '_encoding': 'UTF-8', '_
↳ text': '{"result":"failed","validations":{"rule.interface":"option not in
↳ list"}}'}"}

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

### 3.1 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
```

## 3.2 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](#)

---

## 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 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](#)

### 4.2 Module

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

### 4.3 Adding new module

#### 4.3.1 Testing

Copy the test-template ‘`_tmpl.yml`’ and rename all calls to the new module.

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
```

## 4.4 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()
```

- 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!](#)

## 4.5 Debugging

### 4.5.1 Verbose output

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

```
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
```

## 4.5.2 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 _
↳process

PROFILE = True

if PROFILE:
    profiler(
        check=process, kwargs=dict(
            m=module, p=module.params, r=result,
        ),
        log_file='target_module.log' # in folder: /tmp/ansibleguy.opnsense/
    )
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](#)

---



## 1 - BASIC MODULE ARGUMENTS

### 5.1 All modules

Table 1: Definition

Parameter	Type	Re-quired	Default	Comment
firewall	string	true	-	IP-Address or DNS hostname of the target firewall. Must be included as 'common name' in the firewalls web-certificate to use 'ssl_verify=true'
api_port	int	false	443	Port the target firewall uses for its web-interface
api_key	string	false, true if 'api_credential_file' is not used	-	API key used to authenticate, alternative to 'api_credential_file'
api_secret	string	false, true if 'api_credential_file' is not used	-	API secret used to authenticate, alternative to 'api_credential_file'. Is set as 'no_log' parameter.
api_credential_file_path	path	false, true if 'api_key' and 'api_secret' are not used	-	Path to the api-credential file as downloaded through the web-interface. Alternative to 'api_key' and 'api_secret'.
ssl_verify	bool	false	true	If the certificate of the target firewall should be validated. <b>RECOMMENDED FOR PRODUCTION USAGE!</b>
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'.
timeout	float	false	-	Manually override the modules default timeout

## 5.2 Modules managing multiple entries

Table 2: Definition

Parameter	Type	Re-quired	Default	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](#)

---

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	Re-quired	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_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

```
- 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](#)

---

## 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	Re-quired	Default	Aliases	Comment
target	string	true	-	tgt, t	What part of the running config should be reloaded. One of: 'alias', 'route', 'cron', 'unbound', 'syslog', 'ipsec', 'shaper', 'monit', 'wireguard', 'interface_vlan', 'interface_vxlan', 'frr', 'webproxy'

For basic parameters see: [Basic](#)

### 7.3 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'
```

(continues on next page)

(continued from previous page)

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

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

### 7.3.1 Practical

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

---

**Tip:** Check out the repository on GitHub

---



**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 scales better for that use-case!

## 8.2 Definition

Table 1: Definition

Parameter	Type	Re-quired	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', 'dynip6host', '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 <i>ansibleguy.opnsense.reload</i> 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: 'geoup'
    content: ['AT', 'DE', 'CH']

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

---

**Tip:** Check out the repository on [GitHub](#)

---

## 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	Re-quired	Default	Aliases	Comment
aliases	dictio-nary	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 <i>ansibleguy.opnsense.reload</i> module.

### 9.3.2 ansibleguy.opnsense.alias\_purge

Table 2: Definition

Parameter	Type	Re-quired	Default	Aliases	Comment
aliases	dictio-nary	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: 'dis-able', 'delete'
filters	dictio-nary	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 <i>ansibleguy.opnsense.reload</i> 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](#)

---



## DNS - BIND

**STATE:** unstable

**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.opsense.package` module.

### 10.3 Definition

For basic parameters see: *Basic*



### 10.3.1 ansibleguy.opnsense.bind\_general

Table 1: Definition

Parameter	Type	Re-quired	Default	Aliases	Comment
enabled	boolean	false	true	-	En- or disable the BIND service
ipv6	boolean	false	false	-	En- or disable IPv6
re- sponse_policy_zones	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	['127.0.0.1']	lis- ten_v4, listen	IPv4 addresses the service should listen on
listen_ipv6	list	false	['::1']	lis- ten_v6	IPv6 addresses the service should listen on
query_source_ipv4	string	false	-	query_ipv4, query_v4	Specify the IPv4 address used as a source for out-bound queries
query_source_ipv6	string	false	-	query_ipv6, query_v6	Specify the IPv6 address used as a source for out-bound queries
trans- fer_source_ipv4	string	false	-	trans- fer_ipv4, trans- fer_v4	Specify the IPv4 address used as a source for zone transfers
trans- fer_source_ipv6	string	false	-	trans- fer_ipv6, trans- fer_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_size	Maximum log file size in MB
cache_size	integer	false	50	max_cache_percentage, cache_percentage, max_cache_size	How much memory in percent the cache can use in the system
recursion_acl	string	false	-	recur- sion	Name of an existing ACL - where you allow which clients can resolve via this service. Usually use your local LAN
transfer_acl	string	false	-	al- low_transfer, transfer	Name of an existing ACL - where you allow which server can retrieve zones
dnssec_validation	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
<b>10.3.1- Definition</b> limit_except	list	false	['127.0.0.1', '::1']	-	Except a list of IPs from rate-limiting
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 add it to a task

### 10.3.2 `ansibleguy.opnsense.bind_blocklist`

Table 2: Definition

Parameter	Type	Re-quired	Default	Aliases	Comment
<code>enabled</code>	boolean	false	true	-	En- or disable Blocklists
<code>block</code>	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'
<code>exclude</code>	list	false	-	<code>safe_list</code>	Domains to exclude from the filter
<code>safe_google</code>	boolean	false	-	<code>safe_search-google</code>	
<code>safe_duckduckgo</code>	boolean	false	-	<code>safe_search-duckduckgo</code>	
<code>safe_youtube</code>	boolean	false	-	<code>safe_search-youtube</code>	
<code>safe_bing</code>	boolean	false	-	<code>safe_search-bing</code>	
<code>reload</code>	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 <i>ansibleguy.opnsense.reload</i> module.

### 10.3.3 `ansibleguy.opnsense.bind_acl`

Table 3: Definition

Parameter	Type	Re-quired	Default	Aliases	Comment
<code>name</code>	string	true	-	-	Unique name of the ACL. Some restrictions apply! Length < 32 and neither of: 'any', 'localhost', 'localnets', 'none'
<code>networks</code>	list	false for state changes, else true	-	nets	List of networks to add to the ACL
<code>reload</code>	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 <i>ansibleguy.opnsense.reload</i> module.

## 10.3.4 ansibleguy.opnsense.bind\_domain

Table 4: Definition

Parameter	Type	Re-quired	Default	Aliases	Comment
name	string	true	-	do-main_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	'master'	-	Zone operation mode. One of: 'master', 'slave'
master	list	false	-	mas-ter_ip	Set the IP address of master server when using slave mode
trans-fer_key_algo	string	false	-	-	Set the authentication algorithm for the TSIG key used to transfer domain data from the master server. One of: 'hmac-sha512', 'hmac-sha384', 'hmac-sha256', 'hmac-sha224', 'hmac-sha1', 'hmac-md5'
trans-fer_key_name	string	false	-	-	The name of the TSIG key, which must match the value on the master server
transfer_key	string	false	-	-	The base64-encoded TSIG key
allow_notify	list	false	-	al-low_notify	A list of allowed IP addresses to receive notifies
transfer_acl	string	false	-	al-low_transfe	Name of an existing ACL - where you allow which server can retrieve zones
query_acl	string	false	-	al-low_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 master 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 master 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.opnsense.localdomain'	-	The email address of zone admin. A @-sign will automatically be replaced with a dot in the zone data
server	string	false	'opnsense.localdomain'	dns_server domain	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 <i>ansibleguy.opnsense.reload</i> 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	Re-quired	Default	Aliases	Comment
<code>match_fields</code>	list	false	['do-main', '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'
<code>name</code>	string	true	-	record	Name of the record
<code>domain</code>	string	true	-	do-main_name	Existing domain/zone for the record
<code>type</code>	string	false	'A'	-	Type of the record. One of: 'A', 'AAAA', 'CAA', 'CNAME', 'DNSKEY', 'DS', 'MX', 'NS', 'PTR', 'RRSIG', 'SRV', 'TLSA', 'TXT'
<code>value</code>	false for state changes, else true	false	''	-	Value the record should hold
<code>round_robin</code>	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)
<code>reload</code>	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 <a href="#">ansibleguy.opnsense.reload</a> module.

### 10.3.6 ansibleguy.opnsense.bind\_record\_multi

Table 6: Definition

Parameter	Type	Re-quired	Default	Aliases	Comment
records	dictio-nary	true	-	record	Records to process. Format of the dictionary: {'domain1': [{'name': 'record1', 'value': '192.168.0.1'}, {'name': 'record2', 'type': 'TXT', 'value': 'random'}]} ( <i>dictionary of domains with a list of record-dictionaries</i> )
match_fields	list	false	['do-main', '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 <i>ansibleguy.opnsense.reload</i> 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! Therefore 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: "  
# 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 blocklists
  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

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

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

## 10.5.4 ansibleguy.opnsense.bind\_domain

```
- 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: 'master'
    # 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 tests
  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 tests
  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](#)

---



## CRON JOBS

**STATE:** unstable

**TESTS:** [Playbook](#)

**API Docs:** [Core - Cron](#)

## 11.1 Definition

Table 1: Definition

Parameter	Type	Re-quired	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. <b>WARNING</b> 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 <i>ansible.opnsense.reload</i> module.

For basic parameters see: *Basic*

## 11.2 Usage

To add custom cron-job scripts - see: [OPNSense Documentation](#) | [OPNSense Forum](#)

## 11.3 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: '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 jobs
  ansibleguy.opnsense.list:
    # target: 'cron'
    register: existing_jobs

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

**STATE:** unstable

**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 More FRR modules

- [Docs](#)

## 12.3 Prerequisites

You need to install the FRR plugin:

```
os-frr
```

You can also install it using the [package module](#).

## 12.4 Definition

For basic parameters see: [Basics](#)

### 12.4.1 ansibleguy.opnsense.frr\_bfd\_general

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

### 12.4.2 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

## 12.5 Examples

### 12.5.1 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

```

## 12.5.2 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'

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 neighbors
  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'
```





**STATE:** unstable

**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 More FRR modules

- [Docs](#)

## 13.3 Prerequisites

You need to install the FRR plugin:

```
os-frr
```

You can also install it using the `package` module.

## 13.4 Definition

For basic parameters see: [Basics](#)

### 13.4.1 ansibleguy.opnsense.frr\_bgp\_general

Parameter	Type	Required	Default value	Aliases	Comment
as_number	string	true	-	as, as_nr	BGP AS-Number
id	string	false	-	router_id	In some cases it might be clearer to set a fixed router-id. ( <i>4-byte field/IPv4 Address</i> )
graceful	boolean	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
redistribute	list	false	-	-	Select other routing sources, which should be redistributed to the other nodes. Choose from: 'ospf', 'connected', 'kernel', 'rip', 'static'
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 <a href="#">reload module</a> .
enabled	boolean	false	true	-	En- or disable the service



### 13.4.2 ansibleguy.opnsense.frr\_bgp\_neighbor

Parameter	Type	Required	Default value	Aliases	Comment
match_fields	list of strings	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_number	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_int	string	false	-	src_int, update_src, update_source	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_link_local_int	string	false	-	v6_ll_int, ipv6_ll_int, link_local_int	Interface to use for IPv6 link-local neighbours. You must provide the network port as shown in 'Interface - Assignments - Interface ID (in brackets)'
next_hop	boolean	false	false	nhs	
next_hop_all	boolean	false	false	nhsa	Add the parameter "all" after next-hop-self command
multi_hop	boolean	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_protocol	boolean	false	false	-	Is this neighbour multiprotocol capable per RFC 2283
rrclient	boolean	false	false	route_reflector_client	
bfd	boolean	false	false	-	Enable BFD support for this neighbor
send_default_originate	boolean	false	false	default_originate	
as_override	boolean	false	false	asoverride	Override AS number of the originating router with the local AS number. This command is only allowed for eBGP peers
disable_connected_check	boolean	false	false	asoverride	Allow peerings between directly connected eBGP peers using loopback addresses
keepalive	integer	false	60	keep_alive	Keepalive timer to check if the neighbor is still up. Integer between 1 and 1000
56	integer				<b>Chapter 13. FRR BGP</b>
hold_down	integer	false	180	hold-down	The time in seconds when a neighbor is considered dead. This is usually 3 times the keepalive timer. Integer between 3 and 3000

### 13.4.3 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_number	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	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 <a href="#">reload module</a> .

### 13.4.4 ansibleguy.opnsense.frr\_bgp\_route\_map

Parameter	Type	Required	Default value	Aliases	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	list	false	-	as_path	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_list	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" ( <a href="http://www.nongnu.org/quagga/docs/docs-multi/Route-Map-Set-Command.html#Route-Map-Set-Command">http://www.nongnu.org/quagga/docs/docs-multi/Route-Map-Set-Command.html#Route-Map-Set-Command</a> )
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 <a href="#">reload module</a> .

### 13.4.5 ansibleguy.opnsense.frr\_bgp\_community\_list

Parameter	Type	Required	Default value	Aliases	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 standard 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 <a href="#">reload module</a> .

### 13.4.6 ansibleguy.opnsense.frr\_bgp\_as\_path

Parameter	Type	Required	Default value	Aliases	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_pattern	string	false for state changes, else true	-	as	The AS pattern you want to match, regexp allowed (e.g. <code>.\$</code> or <code>_1\$</code> ). It's 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 <a href="#">reload module</a> .

## 13.5 Examples

### 13.5.1 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

- 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
```



## 13.5.2 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'
    # 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

```

(continues on next page)

(continued from previous page)

```
    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'
    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'
```

### 13.5.3 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'
```

(continues on next page)

(continued from previous page)

```

    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
  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'

```

### 13.5.4 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

```

(continues on next page)

(continued from previous page)

```
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}
    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'
```

### 13.5.5 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'
```

(continues on next page)

(continued from previous page)

```
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
  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'
```

### 13.5.6 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'

- 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'
```

## FRR DIAGNOSTIC

**STATE:** unstable

**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 More FRR modules

- [Docs](#)

### 14.3 Prerequisites

You need to install the FRR plugin:

```
os-frr
```

You can also install it using the [package module](#).

### 14.4 Definition

For basic parameters see: [Basics](#)

### 14.4.1 ansibleguy.opnsense.frr\_diagnostic

Parameter	Type	Required	Default value	Aliases	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'

## 14.5 Examples

### 14.5.1 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

```



## FRR OSPF

**STATE:** unstable

**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!*)

### 15.1 Sponsoring

Thanks to [@telmich](#) for sponsoring the development of these modules!

### 15.2 More FRR modules

- [Docs](#)

### 15.3 Prerequisites

You need to install the FRR plugin:

```
os-frr
```

You can also install it using the [package module](#).

### 15.4 Definition

For basic parameters see: [Basics](#)

## 15.4.1 OSPF

### ansibleguy.opnsense.frr\_ospf\_general

Parameter	Type	Required	Default value	Aliases	Comment
carp	boolean	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. ( <i>4-byte field/IPv4 Address</i> )
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_ints	list	false	-	passive_interfaces	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_map	string	false	-	-	Route Map to set for Redistribution
originate	boolean	false	false	orig, advertise_default_gw	This will send the information that we have a default gateway
originate_always	boolean	false	false	orig_always, always_advertise_default_gw	This will send the information that we have a default gateway, regardless of if it is available
originate_metric	integer	false	-	orig_metric	This let you manipulate the metric when advertising default gateway
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 <a href="#">reload module</a> .
enabled	boolean	false	true	-	En- or disable the service

## ansibleguy.opnsense.frr\_ospf\_network

Parameter	Type	Required	Default value	Aliases	Comment
match_fields	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_range	string	-	-	-	Here you can summarize a network for this area like 192.168.0.0/23
prefix_list_in	string	-	-	prefix_in, pre_in	Prefix-List for inbound direction
prefix_list_out	string	-	-	prefix_out, pre_out	Prefix-List for outbound direction
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 <a href="#">reload module</a> .

ansibleguy.opnsense.frr\_ospf\_interface

Parameter	Type	Required	Default value	Aliases	Comment
match_fields	list of 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_type	string	false	-	-	What authentication type to use. Currently only 'message-digest' is supported
auth_key	string	true if 'auth_type' is set, else false	-	-	The key to authenticate
auth_key_id	integer	false	1	-	Integer between 1 and 255
cost	integer	false	-	-	Integer between 1 and 65535
cost_demoted	integer	false	65535	-	Integer between 1 and 65535
carp_depend_on	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_interval	integer	false	-	hello	Integer between 0 and 4294967295
dead_interval	integer	false	-	dead	Integer between 0 and 4294967295
retransmit_interval	integer	false	-	retransmit	Integer between 0 and 4294967295
transmit_delay	integer	false	-	delay	Integer between 0 and 4294967295
priority	integer	false	-	prio	Integer between 0 and 4294967295
network_type	string	false	-	nw_type	One of: 'broadcast', 'non-broadcast', 'point-to-multipoint', 'point-to-point'
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 <a href="#">reload</a> module.

**ansibleguy.opnsense.frr\_ospf\_prefix\_list**

Parameter	Type	Required	Default value	Aliases	Comment
name	string	true	-	-	The name of the prefix-list
seq	string	false for state changes, else true	-	seq_number	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	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 <a href="#">reload module</a> .

**ansibleguy.opnsense.frr\_ospf\_route\_map**

Parameter	Type	Required	Default value	Aliases	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_list	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" ( <a href="http://www.nongnu.org/quagga/docs/docs-multi/Route-Map-Set-Command.html#Route-Map-Set-Command">http://www.nongnu.org/quagga/docs/docs-multi/Route-Map-Set-Command.html#Route-Map-Set-Command</a> )
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 <a href="#">reload module</a> .

### 15.4.2 OSPFv3 (IPv6)

#### ansibleguy.opnsense.frr\_ospf3\_general

Parameter	Type	Required	Default value	Aliases	Comment
carp	boolean	false	false	carp_debug	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)
re-distribute	list	false	-	-	Select other routing sources, which should be redistributed to the other nodes. Choose from: 'bgp', 'connected', 'kernel', 'rip', 'static'
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 <a href="#">reload module</a> .
enabled	boolean	false	true	-	En- or disable the service

## ansibleguy.opnsense.frr\_ospf3\_interface

Parameter	Type	Required	Default value	Aliases	Comment
match_fields	list of 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	boolean	false	false	-	
cost	integer	false	-	-	Integer between 0 and 4294967295
cost_demoted	integer	false	-	65535	Integer between 1 and 65535
carp_depend_on	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_interval	integer	false	-	hello	Integer between 0 and 4294967295
dead_interval	integer	false	-	dead	Integer between 0 and 4294967295
retransmit_interval	integer	false	-	retransmit	Integer between 0 and 4294967295
transmit_delay	integer	false	-	delay	Integer between 0 and 4294967295
priority	integer	false	-	prio	Integer between 0 and 4294967295
network_type	string	false	-	nw_type	One of: 'broadcast', 'point-to-point'
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 <a href="#">reload module</a> .

## 15.5 Examples

### 15.5.1 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'

```

## 15.5.2 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'
```



**STATE:** unstable

**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!*)

## 16.1 Sponsoring

Thanks to [@telmich](#) for sponsoring the development of these modules!

## 16.2 More FRR modules

- [Docs](#)

## 16.3 Prerequisites

You need to install the FRR plugin:

```
os-frr
```

You can also install it using the [package module](#).

## 16.4 Definition

For basic parameters see: [Basics](#)

### 16.4.1 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_ints	list	false	-	passive_interfaces	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'

## 16.5 Examples

### 16.5.1 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
      ansible.builtin.debug:

```

(continues on next page)

(continued from previous page)

```
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](#)

---



## INTERFACE

**STATE:** unstable

**TESTS:** [vlan](#) | [vxlan](#) | [vip](#)

**API Docs:** [Core - Interfaces](#)

**Service Docs:** [VLAN Docs](#) | [VxLAN Docs](#) | [VIP Docs](#)

### 17.1 Info

#### 17.1.1 `ansibleguy.opnsense.interface_vlan`

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

#### 17.1.2 `ansibleguy.opnsense.interface_vxlan`

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

#### 17.1.3 `ansibleguy.opnsense.interface_vip`

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

### 17.2 Definition

For basic parameters see: *Basic*

## 17.2.1 ansibleguy.opnsense.interface\_vlan

Table 1: Definition

Parameter	Type	Re-quired	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 <i>ansibleguy.opnsense.reload</i> module.

## 17.2.2 ansibleguy.opnsense.interface\_vxlan

Table 2: Definition

Parameter	Type	Re-quired	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_address, 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	-	re- remote_address, re- remote_ip, desti- nation, 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	-	multi- cast_group, multi- cast_address, multi- cast_ip, vxlan- group	Remote IP for the VxLAN tunnel - if multicast is used. The interface can be configured in a multi-cast 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 <i>ansibleguy.opnsense.reload</i> module.

### 17.2.3 ansibleguy.opnsense.interface\_vip

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

Table 3: Definition

Parameter	Type	Re-quired	Default	Aliases	Comment
match_fields	list	false	['ad- dress', 'inter- face']	-	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	-
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'
cidr	integer	false	32	sub- net_bits, subnet	CIDR of the VIP network
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
advertis- ing_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
advertis- ing_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 <i>ansibleguy.opnsense.reload</i> module.



## 17.3 Examples

### 17.3.1 ansibleguy.opnsense.interface\_vlan

```
- 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 VLANs
  ansibleguy.opnsense.list:
    # target: 'interface_vlan'
    register: existing_entries

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

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

### 17.3.2 ansibleguy.opnsense.interface\_vxlan

```
- 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 VxLANs
  ansibleguy.opnsense.list:
    # target: 'interface_vxlan'
    register: existing_entries

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

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

### 17.3.3 ansibleguy.opnsense.interface\_vip

```

- 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'
    # match_fields: ['address', 'interface']
    # mode: 'ipalias'
    # cidr: 32
    # 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'
    cidr: 24
    mode: 'carp'
    vhid: 10
    password: 'secret'

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

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

- name: Removing VIP
  ansibleguy.opnsense.interface_vip:
    interface: 'opt1'

```

(continues on next page)

(continued from previous page)

```
address: '192.168.0.100'  
state: 'absent'
```

---

**Tip:** Check out [the repository on GitHub](#)

---

**STATE:** unstable

**TESTS:** [ipsec\\_cert](#) | [ipsec\\_connection](#) | [ipsec\\_pool](#) | [ipsec\\_vti](#)

**API Docs:** [Core - IPsec](#)

**Service Docs:** [IPsec](#) | [IPsec Examples](#) | [IPsec VTI](#)

## 18.1 Limitations

**Warning:** The IPsec modules can only be used on OPNSense version  $\geq 23.1$

## 18.2 Definition

For basic parameters see: *Basic*

## 18.2.1 ansibleguy.opnsense.ipsec\_connection

Module alias: ansibleguy.opnsense.ipsec\_tunnel

Table 1: Definition

Parameter	Type	Re-quired	Default	Aliases	Comment
name	string	true	-	descrip- tion, desc	Unique connection/tunnel name
lo- cal_addresses	list	false	-	lo- cal_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 chosen as default
re- mote_addresses	list	false	-	re- mote_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	-	net- works	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	['de- fault']	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.
<b>18.2. Definition</b>					
aggressive	boolean	false	false	aggr	Enables IKEv1 Aggressive Mode instead of IKEv1 Main Mode with Identity Protection. Ag-

## 18.2.2 ansibleguy.opnsense.ipsec\_pool

Module alias: ansibleguy.opnsense.ipsec\_network

Table 2: Definition

Parameter	Type	Re-quired	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 <i>ansibleguy.opnsense.reload</i> module.





### 18.2.3 ansibleguy.opnsense.ipsec\_child

Table 3: Definition

Parameter	Type	Re-quired	Default	Aliases	Comment
name	string	true	-	descrip- tion, 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	-	lo- cal_traffic lo- cal_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	-	re- mote_traffic re- mote_cidr, re- mote_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.
102					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

## 18.2.4 ansibleguy.opnsense.ipsec\_vti

Table 4: Definition

Parameter	Type	Re-quired	Default	Aliases	Comment
name	string	true	-	descrip- tion, 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	-	lo- cal_addr, local	-
re- mote_address	string	false	-	re- mote_addr, remote	-
lo- cal_tunnel_address	string	false	-	lo- cal_tun_addr, tun- nel_local, lo- cal_tun	-
re- mote_tunnel_address	string	false	-	re- mote_tun_addr, tun- nel_remote, re- mote_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 <i>ansibleguy.opnsense.reload</i> module.

## 18.2.5 ansibleguy.opnsense.ipsec\_auth\_local

## 18.2.6 ansibleguy.opnsense.ipsec\_auth\_remote

See: `ansibleguy.opnsense.ipsec_auth_local`

## 18.2.7 ansibleguy.opnsense.ipsec\_cert

Table 5: Definition

Parameter	Type	Re-quired	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 <i>ansibleguy.opnsense.reload</i> module.

## 18.3 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.

### 18.3.1 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_\n↵KEY-----\n'
```

*# or encrypt the private\_key file beforehand (might be easier)*

```
ansible-vault encrypt /path/to/private/key/file.pem
```

## 18.4 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'
```

(continues on next page)

(continued from previous page)

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

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

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



**STATE:** unstable

**TESTS:** [Playbook](#)

**API Docs:** [Core - Monit](#)

**Service Docs:** [Monit](#)

## 19.1 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

## 19.2 Definition

For basic parameters see: [Basics](#)

### 19.2.1 ansibleguy.opnsense.monit\_alert

Parameter	Type	Required	Default value	Aliases	Comment
recipient	string	true	-	email, mail	Mail address to send the alert to
not_on_events	boolean	false	true	not	Do not send alerts for the following events but on all others
events	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
reminder	int	false	10	-	Send a reminder after some cycles. Integer between 0 and 86400
description	string	false	-	desc	Send a reminder after some cycles
match_fields	list	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'



## 19.2.2 ansibleguy.opnsense.monit\_test

Parameter	Type	Required	Default value	Aliases	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', 'Permission', '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

### 19.2.3 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'
pid-file	path	false	-	-	
match	string	false	-	-	
path	path	false	-	-	According to the service type path can be a file or a directory
service_timeout	path	false	-	svc_timeout	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
poll-time	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	-	-	

## 19.3 Examples

### 19.3.1 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
```

## 19.3.2 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

```

### 19.3.3 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)

```
# 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
```

### 19.3.4 Practical example

Mail notification on IDS alert: see documentation

```
- 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](#)

---





**STATE:** stable

**TESTS:** [Playbook](#)

**API Docs:** [Core - Firmware](#)

**Service Docs:** [Plugins](#)

## 20.1 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.

## 20.2 Definition

Table 1: Definition

Parameter	Type	Re-quired	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 times-out	

For basic parameters see: *Basic*

## 20.3 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: '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](#)

---

**STATE:** unstable

**TESTS:** [Playbook](#)

**API Docs:** [Core - Routes](#)

**Service Docs:** [Routes](#)

## 21.1 Definition

Table 1: Definition

Parameter	Type	Re-quired	Default	Aliases	Comment
gateway	string	true	-	gw	An existing gateway that should be used as target for the network. The network ip protocol ( <i>IPv4/IPv6</i> ) must be the same! <b>WARNING:</b> 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 ( <i>IPv4/IPv6</i> ) 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	['net-work', 'gate-way']	-	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 <i>ansible.guypense.reload</i> module.

For basic parameters see: [Basic](#)

## 21.2 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.

## 21.3 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.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 routes
  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](#)

---





**STATE:** unstable

**TESTS:** [Playbook](#)

**API Docs:** [Core - Firewall](#)

**Service Docs:** [Rules](#)

## 22.1 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.

## 22.2 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

## 22.3 Info

### 22.3.1 Savepoint

You can prevent lockout-situations using the savepoint systems:

- *ansibleguy.opnsense.savepoint*

### 22.3.2 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!

### 22.3.3 Web-UI

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

Menu: 'Firewall - Automation - Filter'



## 22.4 Definition

Table 1: Definition

Parameter	Type	Re-quired	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' ( <i>IPv4 = 'inet', IPv6 = 'inet6'</i> )
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*

## 22.5 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'.

## 22.6 Examples

### 22.6.1 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

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

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

## 22.6.2 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'
      dest_port: 80
```

(continues on next page)

(continued from previous page)

```

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 }}"

```

### 22.6.3 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:
      firewall: 'opnsense.template.ansibleguy.net'

```

(continues on next page)

(continued from previous page)

```
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](#)

---



## RULE - MASS MANAGEMENT

**STATE:** unstable

**TESTS:** rule\_multi | rule\_purge

**API Docs:** Core - Firewall

**Service Docs:** Rules

### 23.1 Info

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

### 23.2 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

### 23.3 Definition

For basic parameters see: *Basic*

### 23.3.1 ansibleguy.opnsense.rule\_multi

Table 1: Definition

Parameter	Type	Re-quired	Default	Aliases	Comment
rules	dictio-nary	true	-	-	Dictionary of rules to manage/configure
key_field	string	true	-	-	What field is used as key of the provided dictio-nary. 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', 'di-rection', '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	dictio-nary	false	-	-	Parameters to override for all rules
defaults	dictio-nary	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'.

### 23.3.2 ansibleguy.opnsense.rule\_purge

Table 2: Definition

Parameter	Type	Re-quired	Default	Aliases	Comment
rules	dictio-nary	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	dictio-nary	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

## 23.4 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

## 23.5 Examples

### 23.5.1 Basics

```
- 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)

```

- 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'

```

## 23.5.2 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'

```

### 23.5.3 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

```

### 23.5.4 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

---

## FIREWALL SAVEPOINT

**STATE:** unstable

**TESTS:** [Playbook](#)

**API Docs:** [Core - Firewall](#)

### 24.1 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*

## 24.2 Definition

Table 1: Definition

Parameter	Type	Re-quired	Default	Aliases	Comment
name	string	false	'create'	Action to execute. One of: 'create', 'revert', 'apply', 'cancel_rollback', 'cancel'	
revision	string	false, true if action is one of 'apply', 'revert' or 'cancel_rollback'	-	Save-point revision to apply, revert or cancel_rollback	
controller	string	false	'filter'	Controller to manage the save-point of. One of: 'source_nat', 'filter'	
api_module	string	false	'fire-wall'	Module to manage the save-point of. Currently only supports 'fire-wall'	

For basic parameters see: *Basic*



## 24.3 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'

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](#)

---



## SERVICE

**STATE:** unstable

**TESTS:** [Playbook](#)

### 25.1 Info

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

### 25.2 Definition

Table 1: Definition

Parameter	Type	Re-quired	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', '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 ( <i>the module will inform you in that case</i> ). One of: 'status', 'start', 'reload', 'restart', 'stop'

For basic parameters see: [Basic](#)

## 25.3 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'

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'
```

## TRAFFIC SHAPER

**STATE:** unstable

**TESTS:** [shaper\\_pipe](#) | [shaper\\_queue](#) | [shaper\\_rule](#)

**API Docs:** [Core - Traffic Shaper](#)

**Service Docs:** [Traffic Shaping](#)

### 26.1 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

### 26.2 Definition

For basic parameters see: [Basics](#)

### 26.2.1 ansibleguy.opnsense.shaper\_pipe

Parameter	Type	Required	Default value	Aliases	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 changes, else true	-	bw	Bandwidth limit for the pipe - used in combination with 'bw_metric'
bandwidth_metric	string	false	Mbit	bw_metric	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_enable	boolean	false	false	-	Enable PIE active queue management
codel_enable	boolean	false	false	-	Enable CoDel active queue management
codel_enable_ecn	boolean	false	false	-	
codel_target	int	false	-	-	Integer between 1 and 10000
codel_interval	int	false	-	-	Integer between 1 and 10000
fq_codel_quantum	int	false	-	-	Integer between 1 and 65535
fq_codel_limit	int	false	-	-	Integer between 1 and 65535
fq_codel_flows	int	false	-	-	Integer between 1 and 65535
buckets	int	false	-	-	Integer between 1 and 65535

## 26.2.2 ansibleguy.opnsense.shaper\_queue

Parameter	Type	Required	Default value	Aliases	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 changes, else true	-	-	Pipe to link to the queue
weight	string	false for state changes, 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 changes, else true	-	-	Pipe to link to the queue
pie_enabled	boolean	false	false	-	Enable PIE active queue management
codeled_enabled	boolean	false	false	-	Enable CoDel active queue management
codeled_enabled	boolean	false	-	-	
codeled_target	integer	false	-	-	Integer between 1 and 10000
codeled_interval	integer	false	-	-	Integer between 1 and 10000
buckets	int	false	-	-	Integer between 1 and 65535

### 26.2.3 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_pipe	string	true, false if 'target_queue' provided	-	pipe	Pipe to link to the rule, alternative to 'target_queue'
target_queue	string	true, false if 'target_pipe' provided	-	queue	Pipe to link to the rule, alternative to 'target_pipe'
sequence	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_packet_length	int	false	-	max_packet_len, packet_len, iplen	Integer between 2 and 65535
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 or 'any', alias not supported
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 or 'any', alias not supported
destination_port	string	false	-	dp, dest_port	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'
direction	string	false	'both'	-	One of: 'in', 'out', leave empty for 'both'



## 26.3 Examples

### 26.3.1 Pipes

```

- 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'

- 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 pipes
  ansibleguy.opnsense.list:
    target: 'shaper_pipe'
    register: existing_entries

```

(continues on next page)

(continued from previous page)

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

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

## 26.3.2 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'

    - 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: Disabling queue
```

(continues on next page)

(continued from previous page)

```

ansibleguy.opnsense.shaper_queue:
  description: 'testQueue1'
  pipe: 'testPipe1'
  weight: 50
  enabled: false

- name: Listing queue
  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'

```

### 26.3.3 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'
        # destination_invert: false
        # destination_net: 'any'
        # destination_port: 'any'
        # dscp: ['be']

```

(continues on next page)

```
# direction: 'in'
# enabled: true
# debug: false
# state: 'present'

- 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
  ansibleguy.opnsense.shaper_rule:
    description: 'testRule1'
    target_queue: 'testQueue1'
    protocol: 'tcp'
    destination_port: 80
    enabled: false

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

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

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

---

**Tip:** Check out the repository on [GitHub](#)

---



**STATE:** unstable

**TESTS:** [Playbook](#)

**API Docs:** [Core - Firewall](#)

**Service Docs:** [Outbound NAT](#)

## 27.1 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.

## 27.2 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

## 27.3 Info

### 27.3.1 Savepoint

You can prevent lockout-situations using the savepoint systems:

- *ansibleguy.opnsense.savepoint*

### 27.3.2 Web-UI

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

Menu: 'Firewall - Automation - Source NAT'

## 27.4 Definition

Module alias: `ansibleguy.opnsense.snat`



Table 1: Definition

Parameter	Type	Re-quired	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' ( <i>IPv4 = 'inet', IPv6 = 'inet6'</i> )
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

For basic parameters see: *Basic*

## 27.5 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'.

## 27.6 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.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
      # state: 'present'

  - name: Adding rule
```

(continues on next page)

(continued from previous page)

```
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 peers
  ansibleguy.opnsense.list:
    # target: 'source_nat'
    register: existing_entries

- name: Printing
  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](#)

---



---

CHAPTER  
**TWENTYEIGHT**

---

**SYSLOG**

**STATE:** stable

**TESTS:** [Playbook](#)

**API Docs:** [Core - Syslog](#)

## 28.1 Definition

Table 1: Definition

Parameter	Type	Re-quired	Default	Aliases	Comment
target	string	true	-	host-name, 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 ( <i>value in brackets needed</i> ).
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*

## 28.2 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.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
```

## 28.2.1 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](#)

---



---

CHAPTER  
**TWENTYNINE**

---

**SYSTEM**

**STATE:** unstable

**TESTS:** [Playbook](#)

**API Docs:** [Core - Firmware](#)



## 29.1 Definition

Table 1: Definition

Parameter	Type	Re-quired	Default	Aliases	Comment
action	string	true	-	Action to execute. One of: 'poweroff', 'reboot', 'update', 'upgrade', 'audit'. <b>WARNING:</b> 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	
<b>29.1. Definition</b>				action to finish - if 'wait' is enabled	<b>167</b>

For basic parameters see: *Basic*

## 29.2 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'

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](#)

---

## DNS - UNBOUND - DOMAIN OVERRIDE

**STATE:** unstable

**TESTS:** [Playbook](#)

**API Docs:** [Core - Unbound](#)

**Service Docs:** [Unbound](#)

### 30.1 Definition

Table 1: Definition

Parameter	Type	Re-quired	Default	Aliases	Comment
match_fields	string	false	['do-main', '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	Target server
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 <i>ansible.guy.opnsense.reload</i> module.

For basic parameters see: [Basic](#)

## 30.2 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.

## 30.3 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.

### 30.3.1 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

## 30.4 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.unbound_domain:
    match_fields: ['description']

  ansibleguy.opnsense.list:
    target: 'unbound_domain'

tasks:
  - name: Example
    ansibleguy.opnsense.unbound_domain:
      domain: 'opnsense.template.ansibleguy.net'
      server: '192.168.0.1'
      # match_fields: ['description']
      # description: 'example'
      # state: 'present'
      # enabled: true
```

(continues on next page)

(continued from previous page)

```
# 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 domains
  ansibleguy.opnsense.list:
    # target: 'unbound_domain'
    register: existing_entries

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

---

**Tip:** Check out the repository on [GitHub](#)

---





## DNS - UNBOUND - DNS-OVER-TLS

**STATE:** stable

**TESTS:** [Playbook](#)

**API Docs:** [Core - Unbound](#)

**Service Docs:** [Unbound](#)

### 31.1 Definition

Table 1: Definition

Parameter	Type	Re-quired	Default	Aliases	Comment
domain	string	true	-	dom, d	Domain of the DNS-over-TLS entry
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, host-name	Verify if CN in certificate matches this value, <b>if not set - certificate verification will not be performed!</b> 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 <i>ansibleguy.opnsense.reload</i> module.

For basic parameters see: [Basic](#)

### 31.2 Info

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

### 31.2.1 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

## 31.3 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_dot'

tasks:
- name: Example
  ansibleguy.opnsense.unbound_dot:
    domain: 'dot.template.ansibleguy.net'
    target: '1.1.1.1'
    # port: 53
    # verify: 'dot.template.ansibleguy.net'
    # state: 'present'
    # 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 dots
  ansibleguy.opnsense.list:
    # 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](#)

---

## DNS - UNBOUND - FORWARDING

**STATE:** stable

**TESTS:** [Playbook](#)

**API Docs:** [Core - Unbound](#)

**Service Docs:** [Unbound](#)

### 32.1 Definition

Table 1: Definition

Parameter	Type	Re-quired	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 <i>ansible-guy.opnsense.reload</i> module.

For basic parameters see: [Basic](#)

### 32.2 Info

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

### 32.2.1 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

### 32.3 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_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'
    # enabled: true
    # debug: false

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

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

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

---

**Tip:** Check out [the repository on GitHub](#)

---

## DNS - UNBOUND - HOST OVERRIDE

**STATE:** stable

**TESTS:** [Playbook](#)

**API Docs:** [Core - Unbound](#)

**Service Docs:** [Unbound](#)

### 33.1 Definition

Table 1: Definition

Parameter	Type	Re-quired	Default	Aliases	Comment
match_fields	string	false	['host-name', 'do-main', 'record_type', '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	true	'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 <i>ansible-guy.opnsense.reload</i> module.

For basic parameters see: [Basic](#)

## 33.2 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.

## 33.3 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.

### 33.3.1 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

## 33.4 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.unbound_host:
    match_fields: ['description']

  ansibleguy.opnsense.list:
    target: 'unbound_host'

tasks:
  - name: Example
    ansibleguy.opnsense.unbound_host:
      hostname: 'host'
```

(continues on next page)

(continued from previous page)

```
domain: 'opnsense.template.ansibleguy.net'
value: '192.168.0.1'
# match_fields: ['description']
# record_type: 'A'
# prio: 10
# description: 'example'
# state: 'present'
# 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 hosts
  ansibleguy.opnsense.list:
    # target: 'unbound_host'
    register: existing_entries

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

---

**Tip:** Check out the repository on GitHub

---





## DNS - UNBOUND - HOST ALIAS

**STATE:** stable

**TESTS:** Playbook

**API Docs:** Core - Unbound

**Service Docs:** Unbound

### 34.1 Definition

Table 1: Definition

Parameter	Type	Re-quired	Default	Aliases	Comment
match_fields	string	false	['alias', 'do-main']	-	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	-	host-name	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 <i>ansible.opnsense.reload</i> module.

For basic parameters see: *Basic*

## 34.2 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.

## 34.3 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.

### 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

```
- 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'

tasks:
  - name: Example
    ansibleguy.opnsense.unbound_host_alias:
      alias: 'test'
      domain: 'opnsense.template.ansibleguy.net'
```

(continues on next page)

(continued from previous page)

```
target: 'host.opnsense.template.ansibleguy.net'
# match_fields: ['description']
# description: 'example'
# state: 'present'
# 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 aliases
  ansibleguy.opnsense.list:
    # target: 'unbound_host_alias'
    register: existing_entries

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

---

**Tip:** Check out the repository on [GitHub](#)

---



## WEB PROXY

**STATE:** unstable

**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:** [Core - Proxy](#)

**Service Docs:** [Transparent Proxy](#) | [Caching Proxy](#) | [Web Proxy/Filter](#)

### 35.1 Info

#### 35.1.1 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*’)

## **35.1.2 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*’)

## **35.1.3 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.

## 35.1.4 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*’)

## 35.2 Definition

For basic parameters see: *Basic*

### 35.2.1 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
dns_prio_ipv4	boolean	false	false	dns_ipv4	This option reverses the order of preference to make Squid contact dual-stack websites over IPv4 first. Squid will still perform both IPv6 and IPv4 DNS lookups before connecting. This option will restrict the situations under which IPv6 connectivity is used (and tested) and will hide network problems which would otherwise be detected and warned about
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_forwarded_for	string	false	default	forwarded_for_handling, forwarded_for_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_whitespace	string	false	strip	uri_whitespace_handling, uri_whitespace_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_hostname	The hostname to be displayed in proxy server error messages
email	string	false	admin@localhost	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



## ansibleguy.opnsense.webproxy\_cache

Table 2: Definition

Parameter	Type	Re-quired	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/cache	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_object_size, max_size	The maximum object size
memory_kb_max	integer	false	512	maximum_object_size_in_memory, max_memory, max_mem	The maximum object size
memory_cache_mode	string	false	default	cache_mode, 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_packages	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_updates	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 <i>ansibleguy.opnsense.reload</i> module.

ansibleguy.opnsense.webproxy\_parent

Table 3: Definition

Parameter	Type	Re-quired	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	-
lo-cal_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 <i>ansibleguy.opnsense.reload</i> module.

## ansibleguy.opnsense.webproxy\_traffic

Table 4: Definition

Parameter	Type	Re-quired	Default	Aliases	Comment
enabled	boolean	false	true	-	En- or disable the traffic management
down-load_kb_max	integer	false	2048	down-load_max, down-load, dl_max, dl	The maximum size for downloads in kilobytes (leave empty to disable)
up-load_kb_max	integer	false	1024	up-load_max, upload, ul_max, ul	The maximum size for uploads in kilobytes (leave empty to disable)
throt-tle_kb_bandwidth	integer	false	1024	throt-tle_bandwidth, throt-tle_bw, band-width, bw	The allowed overall bandwidth in kilobits per second (leave empty to disable)
throt-tle_kb_host_bandwidth	integer	false	256	throt-tle_host_bandwidth, throt-tle_host_bw, host_bandwidth, 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 <i>ansibleguy.opnsense.reload</i> module.

## 35.2.2 Forward

ansibleguy.opnsense.webproxy\_forward

Table 5: Definition

Parameter	Type	Re-quired	Default	Aliases	Comment
transparent	boolean	false	false	trans- par- ent_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_inspect, 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_sni_only	boolean	false	false	ssl_sni_only	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_subnets	boolean	false	true	al- low_subnets	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 (sslcrtid_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_community, snmp_pwd	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 <a href="#">ansibleguy.opnsense.reload</a> module.



ansibleguy.opnsense.webproxy\_acl

Table 6: Definition

Parameter	Type	Re-quired	Default	Aliases	Comment
allow	list	false	-	al-low_subnets, subnets	IPs and Subnets you want to allow access to the proxy server
exclude	list	false	-	unre-stricted, 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
ex-clude_domains	list	false	-	safe_list, whitelist	You may use a regular expression, use a comma or press Enter for new item. Examples: 'my-domain.com' matches on '*.mydomain.com'; '^https?:\W([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: 'my-domain.com' matches on '*.mydomain.com'; '^https?:\W([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_agents	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\.' matches on 'Macintosh version of Firefox revision 37.0'; '^Mozilla' matches on 'all Mozilla based browsers'
block_mime_types	list	false	-	block_mime, 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 'javascrip\ts'
ex-clude_google	list	false	-	safe_list, google	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:go-pher', '210:wais', '1025-65535:unregistered 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)
194					<b>Chapter 35. Web Proxy</b>

## ansibleguy.opnsense.webproxy\_icap

Table 7: Definition

Parameter	Type	Re-quired	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]:1344/avscan	request_target	The url where the REQMOD requests should be sent to
response_url	string	false	icap://[::1]:1344/avscan	response_target	The url where the RESPMOD requests should be sent to
ttl	integer	false	60	default_ttl	-
send_client_ip	boolean	false	true	send_client	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
en-code_username	boolean	false	false	user_encode, encode_user, enc_user	Use this option if your usernames need to be encoded
header_username	string	false	X-Username	header_user, user_header	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: <code>'my-domain.com'</code> matches on <code>'*.mydomain.com'</code> ; <code>'https://([a-zA-Z]+)\.mydomain.'</code> matches on <code>'http(s)://textONLY.mydomain.*'</code> ; <code>'\*.gif\$'</code> matches on <code>'*.gif'</code> but not on <code>'*.gif/test'</code> ; <code>'\[0-9]+\*.gif\$'</code> matches on <code>'123.gif'</code> but not on <code>'test.gif'</code>
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 <code>ansibleguy.opnsense.reload</code> module.

ansibleguy.opnsense.webproxy\_auth

Table 8: Definition

Parameter	Type	Re-quired	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	-	lo-cal_group	Restrict access to users in the selected (lo-cal)group. <b>NOTE:</b> please be aware that users (or vouchers) which aren't administered locally will be denied when using this option
prompt	string	false	OP-Nsense proxy authentication	realm	The prompt will be displayed in the authentication request window
group	string	false	-	lo-cal_group	
tth_h	integer	false	2	tth, tth_hours, creden-tial_tth	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 <i>ansibleguy.opnsense.reload</i> module.

35.2.3 Remote ACL



## ansibleguy.opnsense.webproxy\_remote\_acl

Table 9: Definition

Parameter	Type	Re-quired	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 <i>ansibleguy.opnsense.reload</i> module.

## 35.2.4 Proxy Auto-Config

## ansibleguy.opnsense.webproxy\_pac\_proxy

Table 10: Definition

Parameter	Type	Re-quired	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 <i>ansibleguy.opnsense.reload</i> module.

ansibleguy.opnsense.webproxy\_pac\_match

Table 11: Definition

Parameter	Type	Re-quired	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_matches	-	One of: 'url_matches', 'hostname_matches', 'dns_domain_is', 'destination_in_net', 'my_ip_in_net', 'plain_hostname', 'is_resolvable', 'dns_domain_levels', 'week-day_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
do-main_level_from	integer	false	0	do-main_from	The minimum amount of dots in the domain name
do-main_level_to	integer	false	0	do-main_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
week-day_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 <i>ansibleguy.opnsense.reload</i> module.

**ansibleguy.opnsense.webproxy\_pac\_rule**

Table 12: Definition

Parameter	Type	Re-quired	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 <i>ansibleguy.opnsense.reload</i> module.

## 35.3 Examples

### 35.3.1 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: []
    # dns_prio_ipv4: false
    # 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
```

(continues on next page)

(continued from previous page)

```

# directory: '/var/squid/cache'
# 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
```

## 35.3.2 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:
```

(continues on next page)

(continued from previous page)

```
target: 'webproxy_forward'
```

**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-
```

(continues on next page)

(continued from previous page)

```

↪65535:unregistered ports', '280:http-mgmt', '488:gss-http', '591:filemaker', '777:multilingu
↪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'

```

(continues on next page)



(continued from previous page)

```

# preview: true
# 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

```

### 35.3.3 Remote ACL

#### ansibleguy.opnsense.webproxy\_remote\_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_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'
```

### 35.3.4 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

```
- 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'
```

## WIREGUARD

**STATE:** unstable

**TESTS:** [wireguard\\_server](#) | [wireguard\\_peer](#) | [wireguard\\_general](#) | [wireguard\\_show](#)

**API Docs:** [Plugin - Wireguard](#)

**Service Docs:** [WireGuard - Site to Site](#) | [WireGuard - Client to Site](#)

### 36.1 Prerequisites

You need to install the WireGuard plugin:

```
os-wireguard
```

You can also install it using the [package](#) module.

### 36.2 Definition

For basic parameters see: [Basics](#)

### 36.2.1 ansibleguy.opnsense.wireguard\_server

Parameter	Type	Required	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_ips	list	false	-	tunnel_ips, tunnel_ip, tunneladdress, tunnel_addresses, tunnel_address, address, allowed	One or multiple IP addresses that are used inside the tunnel
public_key	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_key	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_servers	list	false	-	dns	List of DNS servers that will be used to resolve peer endpoint-names
disable_routes	boolean	false	false	disableroutes	If automatically created routes should be disabled. Needs to be set if you want to use <a href="#">policy-based routing</a> , <a href="#">dynamic routing</a> 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.
reload	boolean	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 <a href="#">reload module</a> .



### 36.2.2 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_ips	list	false for state changes, else true	-	tunnel_ips, tunnel_ip, tunneladdress, tunnel_addresses, tunnel_address, addresses, address, allowed	One or multiple IP addresses used by the peer inside the tunnel
public_key	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
keepalive	integer	false	-	-	Integer between 1 and 86400. Should be used if one of the connection-members is behind NAT
reload	boolean	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 <a href="#">reload module</a> .

### 36.2.3 ansibleguy.opnsense.wireguard\_show

Parameter	Type	Required	Default value	Aliases	Comment
target	string	false	handshake	-	What information to query. One of: 'handshake', 'config'

### 36.2.4 ansibleguy.opnsense.wireguard\_general

Parameter	Type	Required	Default value	Aliases	Comment
enabled	boolean	false	true	-	Used to enable or disable the wireguard service

## 36.3 Usage

To make a dynamic WireGuard endpoint to re-connect you may want to create a gateway monitoring (*dpinger*) targeting the remote tunnel-address.

## 36.4 Examples

### 36.4.1 ansibleguy.opnsense.wireguard\_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.wireguard_general:
      # enabled: true

  - name: Enabling WireGuard service
    ansibleguy.opnsense.wireguard_general:
      enabled: true
```

### 36.4.2 ansibleguy.opnsense.wireguard\_show

```
- 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:
      # target: 'handshake'

  - name: Querying the current WireGuard handshakes
    ansibleguy.opnsense.wireguard_show:
      target: 'handshake'
      register: wg_hands

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

### 36.4.3 ansibleguy.opnsense.wireguard\_peer

```
- 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: []
    # endpoint: ""
    # 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/qRSLPnH3sizr2+A4l3C4tKlUs00RV63+SE='

- name: Disabling peer
  ansibleguy.opnsense.wireguard_peer:
    name: 'test1'
    enabled: false

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

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

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

### 36.4.4 ansibleguy.opnsense.wireguard\_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: '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
    # 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

- name: Disabling server
  ansibleguy.opnsense.wireguard_server:
    name: 'test1'
    enabled: false

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

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

- name: Removing server
  ansibleguy.opnsense.wireguard_server:
```

(continues on next page)

(continued from previous page)

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