
phpipam-ansible-modules Documentation

Christian Meißner

Feb 04, 2023

USER DOCUMENTATION

1	phpIPAM Ansible Modules	1
1.1	Installation	1
1.2	Documentation	1
1.3	Dependencies	2
1.4	Need help?	2
2	codeaffen.phpipam Release Notes	3
2.1	v1.6.1	4
2.2	v1.6.0	4
2.3	v1.5.0	5
2.4	v1.4.0	5
2.5	v1.3.1	6
2.6	v1.3.0	6
2.7	v1.2.0	6
2.8	v1.1.1	6
2.9	v1.1.0	6
2.10	v1.0.0	6
2.11	v0.5.0	7
2.12	v0.4.0	7
2.13	v0.3.0	7
2.14	v0.2.0	7
2.15	v0.1.0	7
3	Codeaffen.Phipam	9
3.1	Description	9
3.2	Plugin Index	9
4	Filters	45
4.1	is_subnet	45
5	Contributor Covenant Code of Conduct	47
5.1	Our Pledge	47
5.2	Our Standards	47
5.3	Enforcement Responsibilities	48
5.4	Scope	48
5.5	Enforcement	48
5.6	Enforcement Guidelines	48
5.7	Attribution	49
6	How to contribute to phpipam-ansible-modules	51
6.1	Did you found a bug?	51

6.2	Did you write a patch for an open bug?	51
6.3	Do you want to add a new feature?	51
6.4	Do you want to contribute to documentation?	53
6.5	Thank you for any contribution	53
7	Releasing the collection	55
8	Indices and tables	57

PHPIPAM ANSIBLE MODULES

This collection provides modules to manage entities in a [phpIPAM](#). This is neither a collection of roles nor playbooks. It provides modules to write your own roles and/or playbooks.

We get a lot of inspiration from [foreman-ansible-modules](#) for our modules.

1.1 Installation

The collection is available via [Ansible Galaxy](#). So you can run

```
ansible-galaxy collection install codeaffen.phpipam
```

Alternatively you can build and install the collection from source.

```
make dist
ansible-galaxy collection install codeaffen-phpipam-<version>.tar.gz
```

1.2 Documentation

1.2.1 readthedocs.io

Current documentation can be found on [readthedocs.io](#).

1.2.2 ansible-doc

If you have installed the collection you can facilitate `ansible-doc` to display documentation for a given module.

```
ansible-doc codeaffen.phpipam.section
```

1.2.3 repository folder

A last option to read the docs is the docs folder in this repository.

1.3 Dependencies

The following dependencies have to be fulfilled by the Ansible controller.

- colour
- geopy
- inflection
- ipaddress
- phypam>=1.0.0

1.4 Need help?

If you've found any issues in this release please head over to github and open a bug so we can take a look.

CODEAFFEN.PHPIPAM RELEASE NOTES

Topics

- *codeaffen.phpipam Release Notes*
 - *v1.6.1*
 - * *Bugfixes*
 - * *Enhancements*
 - *v1.6.0*
 - * *Minor Changes*
 - * *Bugfixes*
 - *v1.5.0*
 - * *Minor Changes*
 - * *Bugfixes*
 - *v1.4.0*
 - * *Minor Changes*
 - * *Bugfixes*
 - * *New Modules*
 - *v1.3.1*
 - * *Bugfixes*
 - *v1.3.0*
 - * *Bugfixes*
 - *v1.2.0*
 - * *Minor Changes*
 - *v1.1.1*
 - * *Bugfixes*
 - *v1.1.0*
 - * *Bugfixes*
 - *v1.0.0*

- * *Major Changes*
- v0.5.0
- * *New Modules*
- v0.4.0
- * *New Modules*
- v0.3.0
- * *New Modules*
- v0.2.0
- * *New Modules*
- v0.1.0

2.1 v1.6.1

2.1.1 Bugfixes

- fix #90 - booleans in subnet module aren't working
- fix #93 - trouble creating subnet with a vrf

2.1.2 Enhancements

- Enhance test suite by running test againsts main pphpipam versions as matrix build
- Update test playbooks to meet best practices for ansible 2.10.x
- move to nodejs 16 for *checkout* and *setup-python* action

2.2 v1.6.0

2.2.1 Minor Changes

- Fix #84 - Allow vlans with same vlan id in different l2 routing domains
- fix #85 - Add *routing_domain* parameter to subnet module to allow subnet with same vlan id in different l2domains

2.2.2 Bugfixes

- Fix #77 - hostname parameter missing in task for address test case
- Fix documentation toolchain to link to external content automatically
- fix #80 - Can't add VLAN to subnet through to phpipam implementation differences in different entities

2.3 v1.5.0

2.3.1 Minor Changes

- fix #68 - add automatic testing facility for all modules
- fix #69 - add facility to setup local phpipam environment
- fix #70 - provide environment variable support for connection data

2.3.2 Bugfixes

- fix *ModuleNotFoundError* while using the collection with ansible >= version 5

2.4 v1.4.0

2.4.1 Minor Changes

- Minor formatting and spelling fixes.
- Switch sphinx from recommonmark to myst_parser.

2.4.2 Bugfixes

- fix #57 - tag lookups failed when specified in an *address* task
- fix #61 - Device type examples
- with [AHH538](<https://issues.redhat.com/browse/AAH-538>) *requires_ansible* is mandatory in *meta/runtime.yml*. So we add the minimum version for our collection here.

2.4.3 New Modules

- codeaffen.phpipam.location - Manage locations
- codeaffen.phpipam.tag - Manage tags

2.5 v1.3.1

2.5.1 Bugfixes

- fix *KeyError 'section'* bug (<https://github.com/codeaffen/phpipam-ansible-modules/issues/41>)

2.6 v1.3.0

2.6.1 Bugfixes

- Fixing *Creating same subnet in different sections isn't possible* bug (<https://github.com/codeaffen/phpipam-ansible-modules/issues/33>)

2.7 v1.2.0

2.7.1 Minor Changes

- Add *is_subnet* filter
- Add filter documentation

2.8 v1.1.1

2.8.1 Bugfixes

- Fix *validate_certs* is not used for api connection (<https://github.com/codeaffen/phpipam-ansible-modules/issues/27>)

2.9 v1.1.0

2.9.1 Bugfixes

- Fixing *There is no 'validate_certs' parameter in modules`* bug (<https://github.com/codeaffen/phpipam-ansible-modules/issues/25>)

2.10 v1.0.0

2.10.1 Major Changes

- Adapt documentation to reflect the final requirements
- Running tests against the new version and do some changes to work as expected
- Switch to stable version of *phpypam* module

2.11 v0.5.0

2.11.1 New Modules

- codeaffen.phpipam.device - Manage devices
- codeaffen.phpipam.device_type - Manage device types

2.12 v0.4.0

2.12.1 New Modules

- codeaffen.phpipam.vrf - Manage virtual routers and forwarders

2.13 v0.3.0

2.13.1 New Modules

- codeaffen.phpipam.domain - Manage L2 routing domains
- codeaffen.phpipam.nameserver - Manage nameservers
- codeaffen.phpipam.vlan - Manage vlans

2.14 v0.2.0

2.14.1 New Modules

- codeaffen.phpipam.address - Manage addresses

2.15 v0.1.0

CODEAFFEN.PHPIPAM

Collection version 1.6.1

- *Description*
- *Plugin Index*

3.1 Description

Ansible Modules to manage phpIPAM installations

Authors:

- Andreas Klamke <andy@klamke.net>
- Billy Longman <bill.longman@summit-broadband.com>
- Christian Meißner <cme@codeaffen.org>
- Gerald-Markus Zabos <gmzabos@users.noreply.github.com>
- Mario Fritschen <github@fritschen.net>
- Scott Arthur <scott@scottatron.com>
- lush <porous@web.de>

3.2 Plugin Index

These are the plugins in the codeaffen.phpipam collection:

3.2.1 Modules

- *address module* – Manage addresses
- *device module* – Manage devices
- *device_type module* – Manage device types
- *domain module* – Manage L2 routing domains
- *location module* – Manage locations

- *nameserver module* – Manage nameservers
- *section module* – Manage sections
- *subnet module* – Manage subnets
- *tag module* – Manage tags
- *vlan module* – Manage vlans
- *vrf module* – Manage virtual routers and forwarders

See also:

List of [collections](#) with docs hosted here.

codeaffen.phpipam.address module – Manage addresses

Note: This module is part of the [codeaffen.phpipam collection](#) (version 1.6.1).

You might already have this collection installed if you are using the `ansible` package. It is not included in `ansible-core`. To check whether it is installed, run `ansible-galaxy collection list`.

To install it, use: `ansible-galaxy collection install codeaffen.phpipam`.

To use it in a playbook, specify: `codeaffen.phpipam.address`.

New in version 0.2.0: of codeaffen.phpipam

- *Synopsis*
 - *Requirements*
 - *Parameters*
 - *Examples*

Synopsis

- create, update and delete addresses

Requirements

The below requirements are needed on the host that executes this module.

- inflection
- ipaddress
- phpypam>=1.0.0

Parameters

Parameter	Comments
app_id string	API app name Default: “ansible”
description string	Address description
device string	Device address belongs to
exclude_ping boolean	Exclude this address from status update scans Choices: <ul style="list-style-type: none"> no yes
hostname string	Address hostname
ignore_ptr boolean	Controls if PTR should not be created Choices: <ul style="list-style-type: none"> no ← (default) yes
ipaddress aliases: ip, address string / required	IP address to handle
is_gateway boolean	Defines if address is presented as gateway Choices: <ul style="list-style-type: none"> no ← (default) yes
mac_address aliases: mac string	Mac address
note string	Note
owner string	Address owner
password string / required	Password of the user to access phpIPAM server
port string	Port
ptr	DNS PTR record

Examples

```
- name: "Reserve an IP address"
  codeaffen.phpipam.address:
    username: "admin"
    password: "s3cr3t"
    server_url: "https://ipam.example.com"
    address: "192.0.2.1"
    section: "Customers"
    description: "Default router of sunet"
    subnet: "192.0.2.0/24"
    is_gateway: yes
    state: present

- name: "Remove address reservation"
  codeaffen.phpipam.address:
    username: "admin"
    password: "s3cr3t"
    server_url: "https://ipam.example.com"
    address: "192.0.2.1"
    subnet: "192.0.2.0/24"
    section: "Customers"
    state: absent
```

Authors

- Christian Meißner (@cmeissner)

Collection links

codeaffen.phpipam.device module – Manage devices

Note: This module is part of the [codeaffen.phpipam collection](#) (version 1.6.1).

You might already have this collection installed if you are using the `ansible` package. It is not included in `ansible-core`. To check whether it is installed, run `ansible-galaxy collection list`.

To install it, use: `ansible-galaxy collection install codeaffen.phpipam`.

To use it in a playbook, specify: `codeaffen.phpipam.device`.

New in version 0.5.0: of `codeaffen.phpipam`

- *Synopsis*
- *Requirements*
- *Parameters*
- *Examples*

Synopsis

- create, update and delete devices

Requirements

The below requirements are needed on the host that executes this module.

- inflection
- ipaddress
- phypam>=1.0.0

Parameters

Parameter	Comments
app_id string	API app name Default: “ansible”
description string	A descriptive text for that entity
hostname string / required	Hostname of the given device
ipaddress string	IP address of the given device
password string / required	Password of the user to access phpIPAM server
rack string	Rack where the device belongs to. If set <i>starting_rack_unit</i> and <i>rack_units</i> are also required.
rack_units integer	Size of device in <i>U</i> . If set <i>rack</i> and <i>starting_rack_unit</i> are also required.
sections list / elements=string	List of sections where the device belongs to
server_url string / required	URL of the phpIPAM server
snmp_community string	The SNMP community string
snmp_port string	The used SNMP port Default: 161
snmp_queries string	
snmp_timeout string	The SNMP connection timeout
snmp_v3_auth_pass string	The password to authenticate via SNMPv3
3.2. Plugin Index	
snmp_v3_auth_protocol string	The used SNMPv3 auth protocol

Examples

```
- name: "Create device"
  codeaffen.phpipam.device:
    username: "admin"
    password: "s3cr3t"
    server_url: "https://ipam.example.com"
    hostname: "leaf-example-01"
    ipaddress: "192.0.2.222"
    sections:
      - Example Inc.
      - DEVOPS department
    state: present

- name: "Remove device"
  codeaffen.phpipam.device:
    username: "admin"
    password: "s3cr3t"
    server_url: "https://ipam.example.com"
    name: "leaf-example-001"
    state: absent
```

Authors

- Christian Meißner (@cmeissner)

Collection links

codeaffen.phpipam.device_type module – Manage device types

Note: This module is part of the [codeaffen.phpipam collection](#) (version 1.6.1).

You might already have this collection installed if you are using the `ansible` package. It is not included in `ansible-core`. To check whether it is installed, run `ansible-galaxy collection list`.

To install it, use: `ansible-galaxy collection install codeaffen.phpipam`.

To use it in a playbook, specify: `codeaffen.phpipam.device_type`.

New in version 0.5.0: of `codeaffen.phpipam`

- *Synopsis*
- *Requirements*
- *Parameters*
- *Examples*

Synopsis

- create, update and delete device types

Requirements

The below requirements are needed on the host that executes this module.

- inflection
- ipaddress
- phypam>=1.0.0

Parameters

Parameter	Comments
app_id string	API app name Default: “ansible”
description string	A descriptive text for that entity
name string / required	Name of the given device type
password string / required	Password of the user to access phpIPAM server
server_url string / required	URL of the phpIPAM server
state string	State of the entity Choices: <ul style="list-style-type: none"> • present ← (default) • absent
username string / required	Username to access phpIPAM server
validate_certs boolean	Is the TLS certificate of the phpIPAM server verified or not. Choices: <ul style="list-style-type: none"> • no • yes ← (default)

Examples

```
- name: "Create device type"
  codeaffen.phpipam.device_type:
    username: "admin"
    password: "s3cr3t"
    server_url: "https://ipam.example.com"
    name: "USP"
    description: "universal power supply"
    state: present

- name: "Remove device type"
  codeaffen.phpipam.device_type:
    username: "admin"
    password: "s3cr3t"
    server_url: "https://ipam.example.com"
    name: "USP"
    state: absent
```

Authors

- Christian Meißner (@cmeissner)

Collection links

codeaffen.phpipam.domain module – Manage L2 routing domains

Note: This module is part of the [codeaffen.phpipam collection](#) (version 1.6.1).

You might already have this collection installed if you are using the `ansible` package. It is not included in `ansible-core`. To check whether it is installed, run `ansible-galaxy collection list`.

To install it, use: `ansible-galaxy collection install codeaffen.phpipam`.

To use it in a playbook, specify: `codeaffen.phpipam.domain`.

New in version 0.3.0: of `codeaffen.phpipam`

- *Synopsis*
- *Requirements*
- *Parameters*
- *Examples*

Synopsis

- create, update and delete L2 routing domains

Requirements

The below requirements are needed on the host that executes this module.

- inflection
- ipaddress
- phypam>=1.0.0

Parameters

Parameter	Comments
app_id string	API app name Default: “ansible”
description string	A descriptive text for that entity
name string / required	Name of the given domain
password string / required	Password of the user to access phpIPAM server
sections list / elements=string	List of sections where the nameserver appears
server_url string / required	URL of the phpIPAM server
state string	State of the entity Choices: <ul style="list-style-type: none"> • present ← (default) • absent
username string / required	Username to access phpIPAM server
validate_certs boolean	Is the TLS certificate of the phpIPAM server verified or not. Choices: <ul style="list-style-type: none"> • no • yes ← (default)

Examples

```
- name: "Create domain"
  codeaffen.phpipam.domain:
    username: "admin"
    password: "s3cr3t"
    server_url: "https://ipam.example.com"
    name: "my domain"
    sections:
```

(continues on next page)

(continued from previous page)

```
- Example Inc.  
- DEVOPS department  
state: present  
  
- name: "Remove domain"  
codeaffen.phpipam.domain:  
  username: "admin"  
  password: "s3cr3t"  
  server_url: "https://ipam.example.com"  
  name: "my domain"  
  state: absent
```

Authors

- Christian Meißner (@cmeissner)

Collection links

codeaffen.phpipam.location module – Manage locations

Note: This module is part of the [codeaffen.phpipam collection](#) (version 1.6.1).

You might already have this collection installed if you are using the `ansible` package. It is not included in `ansible-core`. To check whether it is installed, run `ansible-galaxy collection list`.

To install it, use: `ansible-galaxy collection install codeaffen.phpipam`.

To use it in a playbook, specify: `codeaffen.phpipam.location`.

New in version 1.4.0: of `codeaffen.phpipam`

- *Synopsis*
- *Requirements*
- *Parameters*
- *Examples*

Synopsis

- create, update and delete locations

Requirements

The below requirements are needed on the host that executes this module.

- geopy
- inflection
- ipaddress
- phypam>=1.0.0

Parameters

Parameter	Comments
address string	Address of the given location if <i>resolve_location</i> is set to True, this address will be used to resolve the latitude and longitude Mutually exclusive with latitude and longitude
app_id string	API app name Default: “ansible”
description string	Description of the given location
latitude float	Latitude of the given location if <i>resolve_location</i> is set to True, this latitude will be used to resolve the address This parameter is mutually exclusive with address This parameter is required if <i>longitude</i> is given
longitude float	Longitude of the given location if <i>resolve_location</i> is set to True, this longitude will be used to resolve the address This parameter is mutually exclusive with address This parameter is required if <i>latitude</i> is given
name string / required	Name of the given location
password string / required	Password of the user to access phpIPAM server
resolve_location boolean	Resolve the given location Requires network connectivity to https://nominatim.org/ to resolve the given address If address can not be resolved, latitude and longitude will not be set Choices: <ul style="list-style-type: none"> no ← (default) yes
server_url string / required	URL of the phpIPAM server
state string	State of the entity Choices: <ul style="list-style-type: none"> present ← (default) absent
username string / required	Username to access phpIPAM server
3.2. Plugin Index	23
validate_certs boolean	Is the TLS certificate of the phpIPAM server verified or not. Choices:

Examples

```
- name: "Create with address"
  codeaffen.phpipam.location:
    username: "admin"
    password: "s3cr3t"
    server_url: "https://ipam.example.com"
    name: "my location"
    description: "my location description"
    address: "my location address"
    state: present

- name: "Create location with geo coordinates"
  codeaffen.phpipam.location:
    username: "admin"
    password: "s3cr3t"
    server_url: "https://ipam.example.com"
    name: "my location"
    description: "my location description"
    latitude: 123.456
    longitude: 123.456
    state: present

- name: "Remove location"
  codeaffen.phpipam.location:
    username: "admin"
    password: "s3cr3t"
    server_url: "https://ipam.example.com"
    name: "my location"
    state: absent
```

Authors

- Christian Meißner (@cmeissner)

Collection links

codeaffen.phpipam.nameserver module – Manage nameservers

Note: This module is part of the [codeaffen.phpipam collection](#) (version 1.6.1).

You might already have this collection installed if you are using the `ansible` package. It is not included in `ansible-core`. To check whether it is installed, run `ansible-galaxy collection list`.

To install it, use: `ansible-galaxy collection install codeaffen.phpipam`.

To use it in a playbook, specify: `codeaffen.phpipam.nameserver`.

New in version 0.3.0: of `codeaffen.phpipam`

- *Synopsis*
- *Requirements*
- *Parameters*
- *Examples*

Synopsis

- create, update and delete nameservers

Requirements

The below requirements are needed on the host that executes this module.

- inflection
- ipaddress
- phypam>=1.0.0

Parameters

Parameter	Comments
addresses list / elements=string	List of IP addresses the nameserver can be reached on
app_id string	API app name Default: “ansible”
description string	A descriptive text for that entity
name string / required	Name of the given nameserver
password string / required	Password of the user to access phpIPAM server
sections list / elements=string	List of sections where the nameserver appears
server_url string / required	URL of the phpIPAM server
state string	State of the entity Choices: <ul style="list-style-type: none">• present ← (default)• absent
username string / required	Username to access phpIPAM server
validate_certs boolean	Is the TLS certificate of the phpIPAM server verified or not. Choices: <ul style="list-style-type: none">• no• yes ← (default)

Examples

```
- name: "Create nameserver"
  codeaffen.phpipam.nameserver:
    username: "admin"
    password: "s3cr3t"
    server_url: "https://ipam.example.com"
    name: "cloudflare dns"
    addresses:
      - 1.1.1.1
      - 1.0.0.1
    permissions: 1;2
    state: present

- name: "Remove nameserver"
  codeaffen.phpipam.nameserver:
    username: "admin"
    password: "s3cr3t"
    server_url: "https://ipam.example.com"
    name: "cloudflare dns"
    state: absent
```

Authors

- Christian Meißner (@cmeissner)

Collection links

codeaffen.phpipam.section module – Manage sections

Note: This module is part of the [codeaffen.phpipam collection](#) (version 1.6.1).

You might already have this collection installed if you are using the `ansible` package. It is not included in `ansible-core`. To check whether it is installed, run `ansible-galaxy collection list`.

To install it, use: `ansible-galaxy collection install codeaffen.phpipam`.

To use it in a playbook, specify: `codeaffen.phpipam.section`.

New in version 0.0.1: of `codeaffen.phpipam`

- *Synopsis*
- *Requirements*
- *Parameters*
- *Examples*

Synopsis

- create, update and delete sections

Requirements

The below requirements are needed on the host that executes this module.

- inflection
- ipaddress
- phypam>=1.0.0

Parameters

Parameter	Comments
app_id string	API app name Default: “ansible”
description string	Short descriptive text Default: “None”
dns_resolver string	The NS resolver to be used for this section
list_order integer	Order in sections list view
name string / required	Name of the section
parent aliases: master, master_section string	Name of the parent section Default: “None”
password string / required	Password of the user to access phpIPAM server
permissions json	JSON object that represents the permissions for each user Default: “None”
server_url string / required	URL of the phpIPAM server
show_supernets_only boolean	Show only supernets in subnet list view Choices: <ul style="list-style-type: none"> no ← (default) yes
show_vlan boolean	Show/hide VLANs in subnet list view Choices: <ul style="list-style-type: none"> no ← (default) yes
show_vrf boolean	Show/hide VRFs in subnet list view Choices: <ul style="list-style-type: none"> no ← (default) yes
state string	State of the entity Choices: <ul style="list-style-type: none"> present ← (default) absent
strict	If set to true, consistency of subnets and IP addresses

Examples

```
- name: "Create a section"
  codeaffen.phpipam.section:
    username: "admin"
    password: "s3cr3t"
    server_url: "https://ipam.example.com"
    name: "EXAMPLE INC"
    description: "Section for company EXAMPLE INC"
    state: present

- name: "Create a section with parent"
  codeaffen.phpipam.section:
    username: "admin"
    password: "s3cr3t"
    server_url: "https://ipam.example.com"
    name: "DEVOPS department"
    parent: "EXAMPLE INC"
    description: "Section for devops department in EXAMPLE INC"
    state: present
```

Authors

- Christian Meißner (@cmeissner)

Collection links

codeaffen.phpipam.subnet module – Manage subnets

Note: This module is part of the `codeaffen.phpipam` collection (version 1.6.1).

You might already have this collection installed if you are using the `ansible` package. It is not included in `ansible-core`. To check whether it is installed, run `ansible-galaxy collection list`.

To install it, use: `ansible-galaxy collection install codeaffen.phpipam`.

To use it in a playbook, specify: `codeaffen.phpipam.subnet`.

New in version 0.0.1: of `codeaffen.phpipam`

- *Synopsis*
- *Requirements*
- *Parameters*
- *Examples*
- *Return Values*

Synopsis

- create, update and delete subnets

Requirements

The below requirements are needed on the host that executes this module.

- inflection
- ipaddress
- phypam>=1.0.0

Parameters

Parameter	Comments
allow_requests boolean	Controls if IP requests are allowed for subnet Choices: <ul style="list-style-type: none"> • no ← (default) • yes
app_id string	API app name Default: “ansible”
cidr string	Network in CIDR format. Mutually exclusive with <i>subnet</i> and <i>mask</i> .
description string	Text which is shown in side bar if ‘show as name’ is selected
discover_subnet boolean	Controls if new hosts should be discovered for new host scans Choices: <ul style="list-style-type: none"> • no ← (default) • yes
dns_records boolean	Controls whether hostname DNS records are displayed Choices: <ul style="list-style-type: none"> • no ← (default) • yes
dns_recursive boolean	Controls if PTR records should be created for subnet Choices: <ul style="list-style-type: none"> • no ← (default) • yes

continues on next page

Table 1 – continued from previous page

Parameter	Comments
is_folder boolean	Controls if we are adding subnet or folder can't be changed after subnet was created Choices: <ul style="list-style-type: none"> • no ← (default) • yes
is_full boolean	Marks subnet as used Choices: <ul style="list-style-type: none"> • no ← (default) • yes
linked_subnet string	Linked ipv6 subnet in CIDR format
location string	Subnet location
mask string	Prefix length (bits) for ipv4 and ipv6 subnets. Mutually exclusive with <i>cidr</i> . If set, <i>subnet</i> is required.
nameserver string	Name of the DNS server which should attach to subnet
parent aliases: master_subnet_cidr string	CIDR of parent subnet
password string / required	Password of the user to access phpIPAM server
permissions json	JSON object that represent the permissions for each user Default: "None"
ping_subnet boolean	Controls if subnet should be included in status checks Choices: <ul style="list-style-type: none"> • no ← (default) • yes
routing_domain string added in 1.6.0 of codeaffen.phpipam	Layer-2 routing domain the vlan belongs to If you have <i>vlan</i> with same number in different routing domains it need to be set on desired value Default: "default"
scan_agent string	Name of scanagent which should be used for subnet
section integer / required	Name of the section under which the subnet is located

continues on next page

Table 1 – continued from previous page

Parameter	Comments
server_url string / required	URL of the phpIPAM server
show_as_name boolean	If this is set to 'true' description is shown in side bar instead of CIDR Choices: <ul style="list-style-type: none"> no ← (default) yes
state string	State of the entity Choices: <ul style="list-style-type: none"> present ← (default) absent
subnet string	Network address Mutually exclusive with <i>cidr</i> . If set, <i>mask</i> is required.
subnet_state string	Assigned tag of the subnet.
threshold integer	Subnet threshold
username string / required	Username to access phpIPAM server
validate_certs boolean	Is the TLS certificate of the phpIPAM server verified or not. Choices: <ul style="list-style-type: none"> no yes ← (default)
vlan string	VLAN which the subnet should belongs to
vrf string	VRF which the sunet should belongs to

Examples

```
- name: "Create a subnet"
  codeaffen.phpipam.subnet:
    username: "admin"
    password: "s3cr3t"
    server_url: "https://ipam.example.com"
    cidr: "192.0.2.128/26"
    section: "EXAMPLE INC"
```

(continues on next page)

(continued from previous page)

```
state: present

- name: "Create a subnet with parent"
  codeaffen.phpipam.subnet:
    username: "admin"
    password: "s3cr3t"
    server_url: "https://ipam.example.com"
    cidr: "192.0.2.128/28"
    parent: "192.0.2.128/25"
    section: "DEVOPS department"
    state: present
```

Return Values

Common return values are documented [here](#), the following are the fields unique to this module:

Key	Description
entity dictionary	Final state of the affected entities grouped by their type. Returned: success
subnets list / elements=dictionary	List of subnets. Returned: success

Authors

- Christian Meißner (@cmeissner)

Collection links

codeaffen.phpipam.tag module – Manage tags

Note: This module is part of the [codeaffen.phpipam collection](#) (version 1.6.1).

You might already have this collection installed if you are using the `ansible` package. It is not included in `ansible-core`. To check whether it is installed, run `ansible-galaxy collection list`.

To install it, use: `ansible-galaxy collection install codeaffen.phpipam`.

To use it in a playbook, specify: `codeaffen.phpipam.tag`.

New in version 1.4.0: of `codeaffen.phpipam`

- [Synopsis](#)
- [Requirements](#)

- *Parameters*
- *Examples*

Synopsis

- create, update and delete tags

Requirements

The below requirements are needed on the host that executes this module.

- colour
- inflection
- ipaddress
- phypam>=1.0.0

Parameters

Parameter	Comments
app_id string	API app name Default: “ansible”
bg_color string / required	Background color of the given tag Can be a valid color name or a hex code
compress_range boolean	Compress range or not Choices: <ul style="list-style-type: none"> no ← (default) yes
fg_color string / required	Foreground color of the given tag Can be a valid color name or a hex code
is_locked boolean	Lock tag or not Choices: <ul style="list-style-type: none"> no ← (default) yes
name aliases: type string / required	Name of the given tag
password string / required	Password of the user to access phpIPAM server
server_url string / required	URL of the phpIPAM server
show_tag boolean	Show tag or not Choices: <ul style="list-style-type: none"> no ← (default) yes
state string	State of the entity Choices: <ul style="list-style-type: none"> present ← (default) absent
update_tags boolean	Update tags or not Choices: <ul style="list-style-type: none"> no ← (default) yes
username string / required	Username to access phpIPAM server
36 validate_certs boolean	Chapter 3. Codeaffen.Phipipam Is the TLS certificate of the phpIPAM server verified or not. Choices: <ul style="list-style-type: none"> no

Examples

```
- name: "Create tag"
  codeaffen.phpipam.tag:
    username: "admin"
    password: "s3cr3t"
    server_url: "https://ipam.example.com"
    name: "my tag"
    bg_color: #ffffff
    fg_color: #000000
    state: present

- name: "Remove tag"
  codeaffen.phpipam.tag:
    username: "admin"
    password: "s3cr3t"
    server_url: "https://ipam.example.com"
    name: "my tag"
    state: absent
```

Authors

- Christian Meißner (@cmeissner)

Collection links

codeaffen.phpipam.vlan module – Manage vlans

Note: This module is part of the [codeaffen.phpipam collection](#) (version 1.6.1).

You might already have this collection installed if you are using the `ansible` package. It is not included in `ansible-core`. To check whether it is installed, run `ansible-galaxy collection list`.

To install it, use: `ansible-galaxy collection install codeaffen.phpipam`.

To use it in a playbook, specify: `codeaffen.phpipam.vlan`.

New in version 0.3.0: of `codeaffen.phpipam`

- *Synopsis*
- *Requirements*
- *Parameters*
- *Examples*

Synopsis

- create, update and delete vlans

Requirements

The below requirements are needed on the host that executes this module.

- inflection
- ipaddress
- phypam>=1.0.0

Parameters

Parameter	Comments
app_id string	API app name Default: “ansible”
description string	A descriptive text for that entity
name string / required	Name of the given vlan
password string / required	Password of the user to access phpIPAM server
routing_domain string	Name of the L2 routing domain of that VLAN
server_url string / required	URL of the phpIPAM server
state string	State of the entity Choices: <ul style="list-style-type: none"> • present ← (default) • absent
username string / required	Username to access phpIPAM server
validate_certs boolean	Is the TLS certificate of the phpIPAM server verified or not. Choices: <ul style="list-style-type: none"> • no • yes ← (default)
vlan_id string / required	The VLAN-ID. Must be a number. Due to implementation of api the value has to be <i>string</i> here but is <i>int</i> in phpIPAM

Examples

```
- name: "Create vlan"
  codeaffen.phpipam.vlan:
    username: "admin"
    password: "s3cr3t"
    server_url: "https://ipam.example.com"
    name: "my vlan"
    vlan_id: 1337
    state: present

- name: "Remove vlan"
  codeaffen.phpipam.vlan:
    username: "admin"
    password: "s3cr3t"
    server_url: "https://ipam.example.com"
    name: "my vlan"
    state: absent
```

Authors

- Christian Meißner (@cmeissner)

Collection links

codeaffen.phpipam.vrf module – Manage virtual routers and forwarders

Note: This module is part of the [codeaffen.phpipam collection](#) (version 1.6.1).

You might already have this collection installed if you are using the `ansible` package. It is not included in `ansible-core`. To check whether it is installed, run `ansible-galaxy collection list`.

To install it, use: `ansible-galaxy collection install codeaffen.phpipam`.

To use it in a playbook, specify: `codeaffen.phpipam.vrf`.

New in version 0.4.0: of `codeaffen.phpipam`

- *Synopsis*
- *Requirements*
- *Parameters*
- *Examples*

Synopsis

- create, update and delete L2 routing vrfs

Requirements

The below requirements are needed on the host that executes this module.

- inflection
- ipaddress
- phypam>=1.0.0

Parameters

Parameter	Comments
app_id string	API app name Default: “ansible”
description string	A descriptive text for that entity
distinguisher string	VRF route distinguisher
name string / required	Name of the given vrf
password string / required	Password of the user to access phpIPAM server
sections list / elements=string	List of sections where the nameserver appears
server_url string / required	URL of the phpIPAM server
state string	State of the entity Choices: <ul style="list-style-type: none">• present ← (default)• absent
username string / required	Username to access phpIPAM server
validate_certs boolean	Is the TLS certificate of the phpIPAM server verified or not. Choices: <ul style="list-style-type: none">• no• yes ← (default)

Examples

```
- name: "Create vrf"
  codeaffen.phpipam.vrf:
    username: "admin"
    password: "s3cr3t"
    server_url: "https://ipam.example.com"
    name: "my vrf"
    sections:
      - Example Inc.
      - DEVOPS department
    state: present

- name: "Remove vrf"
  codeaffen.phpipam.vrf:
    username: "admin"
    password: "s3cr3t"
    server_url: "https://ipam.example.com"
    name: "my vrf"
    state: absent
```

Authors

- Christian Meißner (@cmeissner)

Collection links

FILTERS

Filters are used to manipulate data if needed.

4.1 `is_subnet`

This filter checks whether a given subnet belongs to the second one. It returns *true* if first subnet belongs to the second and *false* if not, or if both subnets are the same.

since version: 1.2.0

<code>192.0.2.0/25 codeaffen.phpipam.is_subnet(192.0.2.0/24)</code>

CONTRIBUTOR COVENANT CODE OF CONDUCT

5.1 Our Pledge

We as members, contributors, and leaders pledge to make participation in our community a harassment-free experience for everyone, regardless of age, body size, visible or invisible disability, ethnicity, sex characteristics, gender identity and expression, level of experience, education, socio-economic status, nationality, personal appearance, race, caste, color, religion, or sexual identity and orientation.

We pledge to act and interact in ways that contribute to an open, welcoming, diverse, inclusive, and healthy community.

5.2 Our Standards

Examples of behavior that contributes to a positive environment for our community include:

- Demonstrating empathy and kindness toward other people
- Being respectful of differing opinions, viewpoints, and experiences
- Giving and gracefully accepting constructive feedback
- Accepting responsibility and apologizing to those affected by our mistakes, and learning from the experience
- Focusing on what is best not just for us as individuals, but for the overall community

Examples of unacceptable behavior include:

- The use of sexualized language or imagery, and sexual attention or advances of any kind
- Trolling, insulting or derogatory comments, and personal or political attacks
- Public or private harassment
- Publishing others' private information, such as a physical or email address, without their explicit permission
- Other conduct which could reasonably be considered inappropriate in a professional setting

5.3 Enforcement Responsibilities

Community leaders are responsible for clarifying and enforcing our standards of acceptable behavior and will take appropriate and fair corrective action in response to any behavior that they deem inappropriate, threatening, offensive, or harmful.

Community leaders have the right and responsibility to remove, edit, or reject comments, commits, code, wiki edits, issues, and other contributions that are not aligned to this Code of Conduct, and will communicate reasons for moderation decisions when appropriate.

5.4 Scope

This Code of Conduct applies within all community spaces, and also applies when an individual is officially representing the community in public spaces. Examples of representing our community include using an official e-mail address, posting via an official social media account, or acting as an appointed representative at an online or offline event.

5.5 Enforcement

Instances of abusive, harassing, or otherwise unacceptable behavior may be reported to the community leaders responsible for enforcement at dispute@codeaffen.org. All complaints will be reviewed and investigated promptly and fairly.

All community leaders are obligated to respect the privacy and security of the reporter of any incident.

5.6 Enforcement Guidelines

Community leaders will follow these Community Impact Guidelines in determining the consequences for any action they deem in violation of this Code of Conduct:

5.6.1 1. Correction

Community Impact: Use of inappropriate language or other behavior deemed unprofessional or unwelcome in the community.

Consequence: A private, written warning from community leaders, providing clarity around the nature of the violation and an explanation of why the behavior was inappropriate. A public apology may be requested.

5.6.2 2. Warning

Community Impact: A violation through a single incident or series of actions.

Consequence: A warning with consequences for continued behavior. No interaction with the people involved, including unsolicited interaction with those enforcing the Code of Conduct, for a specified period of time. This includes avoiding interactions in community spaces as well as external channels like social media. Violating these terms may lead to a temporary or permanent ban.

5.6.3 3. Temporary Ban

Community Impact: A serious violation of community standards, including sustained inappropriate behavior.

Consequence: A temporary ban from any sort of interaction or public communication with the community for a specified period of time. No public or private interaction with the people involved, including unsolicited interaction with those enforcing the Code of Conduct, is allowed during this period. Violating these terms may lead to a permanent ban.

5.6.4 4. Permanent Ban

Community Impact: Demonstrating a pattern of violation of community standards, including sustained inappropriate behavior, harassment of an individual, or aggression toward or disparagement of classes of individuals.

Consequence: A permanent ban from any sort of public interaction within the community.

5.7 Attribution

This Code of Conduct is adapted from the Contributor Covenant, version 2.0, available at https://www.contributor-covenant.org/version/2/0/code_of_conduct.html.

Community Impact Guidelines were inspired by Mozilla's code of conduct enforcement ladder.

For answers to common questions about this code of conduct, see the FAQ at <https://www.contributor-covenant.org/faq>. Translations are available at <https://www.contributor-covenant.org/translations>.

HOW TO CONTRIBUTE TO PHPIPAM-ANSIBLE-MODULES

6.1 Did you found a bug?

- Do not open a Github issue if the bug concerns `{php}IPAM` and not the ansible modules.
- Make sure the bug is not already opened by another user.
- If you can't find an open issue which reflects your observed problem, go ahead and [open a new bug](#).
- Provide as much information as mentioned in the bug report template.

6.2 Did you write a patch for an open bug?

- Open new pull request containing the patch.
- Provide a clear description which describes the problem and the solution. Link the existing bug to the PR.

6.3 Do you want to add a new feature?

- Make sure there isn't already a feature request.
- If you can't find an open feature request which describes your feature idea or parts of it, feel free to [open a new feature request](#).
- Suggest your feature idea within the created feature request.
- Provide as much description as possible to enable others to have a good understanding of what you are doing.
- Point out that you want to start to work on the new feature
- If you want to start to work on a new module we suggest to read [How to write new phpipam modules@codeaffen.org](#).

6.3.1 Create your feature in a branch of your fork

Fork our repository and create a new branch for your feature. Then start to work on your feature.

6.3.2 Add tests for your new feature

To make sure that your feature is working as expected and to make sure that it is not breaking any existing functionality, you should add tests for your new feature. To do so, you

1. need to have a working phpipam installation.
2. export the following environment variables:
 - PHPIPAM_URL - the URL of your phpipam installation (e.g. `https://localhost`)
 - PHPIPAM_USERNAME - the username to login to phpipam
 - PHPIPAM_PASSWORD - the password to login to phpipam
 - PHPIPAM_APPID - the appid to login to phpipam
3. add a new task definition for your new feature in `tests/test_playbooks/tasks/`
4. add a vars definition for your new feature in `tests/test_playbooks/vars/`
5. add a crud playbook for your new feature in `tests/test_playbooks/` The playbook should contain the following steps:
 - import vars files `vars/server.yml` and `vars/<yourtest>.yml`
 - tasks for
 - create a new entity
 - create a new entity again to make sure that it will not changed
 - update the entity
 - delete the entity
6. run your test with `ansible-playbook --inventory=tests/inventory/hosts tests/test_playbooks/<yourtest>.yml -vvv` to make sure that your feature is working as expected
7. run your test again with `make test-<yourtest>`
8. run all tests with `make test-all` to make sure that your feature does not break any existing functionality

6.3.3 Create pull request for your new feature

When you are done, push your changes to your fork and create a pull request. Please make sure that you have squashed your changes before you create the pull request. Provide a clear description of your feature and the changes you made.

6.4 Do you want to contribute to documentation?

- Fork me.
- Create a documentation branch.
- Write your documentation changes.
- Open a PR with your changes.
- Discuss with the team about your changes.

6.5 Thank you for any contribution

We will thank you for heeding the contribution guidelines and we encourage you to contribute and join the team.

RELEASING THE COLLECTION

To release the collection the following steps are required:

- Create a pull request with the following changes:
 - Updated `galaxy.yml`:
 - * Version bump
 - * Updated list of authors.
Generate that list via `git shortlog -e -s | sed -E 's/^\s+[[[:digit:]]]+\s+(.*)$/ - "\1"/`.
 - Updated changelog by running `antsibull-changelog release` – it will pick up the version from `galaxy.yml`
- After merging, tag the merge commit with `v<version number>` (i.e. `v0.1.1`).

INDICES AND TABLES

- `genindex`
- `search`