TCPWave

Network Automation with Ansible

Publication Date: 09/21/2020

Contents

Scope

This guide describes how TCPWave integrates with Ansible to orchestrate the tasks, including the usage of the command lines.

About Ansible

Ansible is an open-source IT Configuration Management, Orchestration, and Deployment tool. The Ansible platform makes system administrators, network administrators, and developers to automate many tasks, which includes updates to machines on the network to managing devices on the network. It uses SSH to connect to servers and run the configured tasks.

Software Used

- Ansible version 2.7.0

TCPWave Module for Ansible Integration

You can automate TCPWave IP Address Management System (TIMS) using Ansible playbooks using the secure and powerful REST APIs that are used by TIMS GUI and CLI interfaces, and for integration into cloud orchestration layers. The TIMS REST APIs are designed to be secure and allow only encrypted access to the system without the need for any plain text user ID or password.

TIMS supports two mechanisms for handling REST API Authentication:

- **Session Token Based Authentication**: A long-lived session token is generated in TIMS. This session token is associated with a given admin user and inherits all the permissions of that user. The session token is also associated with a source IP and can be used only from that IP. The life of the session token is set as per the global policy "*Maximum Concurrent Sessions per Admin*". The session token can be revoked or extended at any time. This token is set on the request header as the TIMS-Session-Token parameter. All the API calls with this token are subjected to the same permission checks as the associated user and are audited against that user.
- **Certificate-Based Authentication:** In this protocol, access to TIMS is provided using a certificate signed by a trusted authority. The certificate-based mechanism provides a stateless interface that can be leveraged by automation clients that interact with more than one system. User certificates can be imported to TIMS and associated with a particular admin. All the service calls made using that certificate are authorized and audited against the associated admin.

Please add "validate_certs: no" under uri section in the playbook yaml files if you are using self-signed certificates in the TCPWave IPAM.

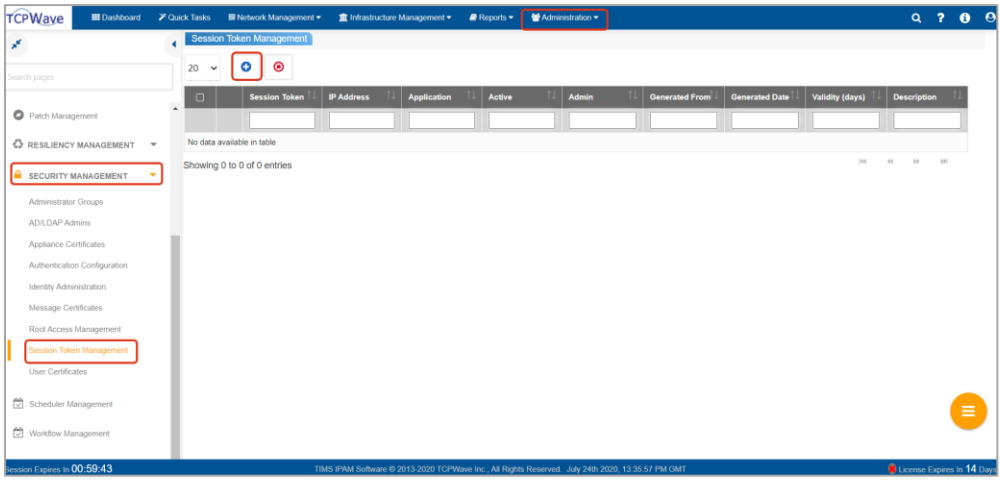
Invoking TCPWave Rest API using Ansible

This section describes a few examples of how TCPWave's Rest APIs are invoked using Ansible playbooks.

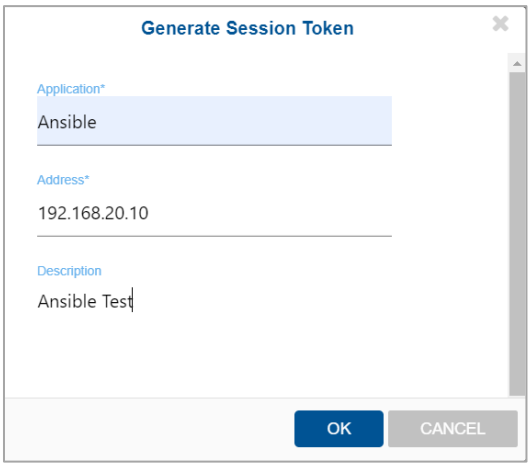
Session Token Authentication

This section explains the steps to generate session token in TIMS GUI along with screenshots.

1. Click **Administration** tab
2. Select **Security Management** from the drop-down
3. Click the **Session Token** Management label as shown:

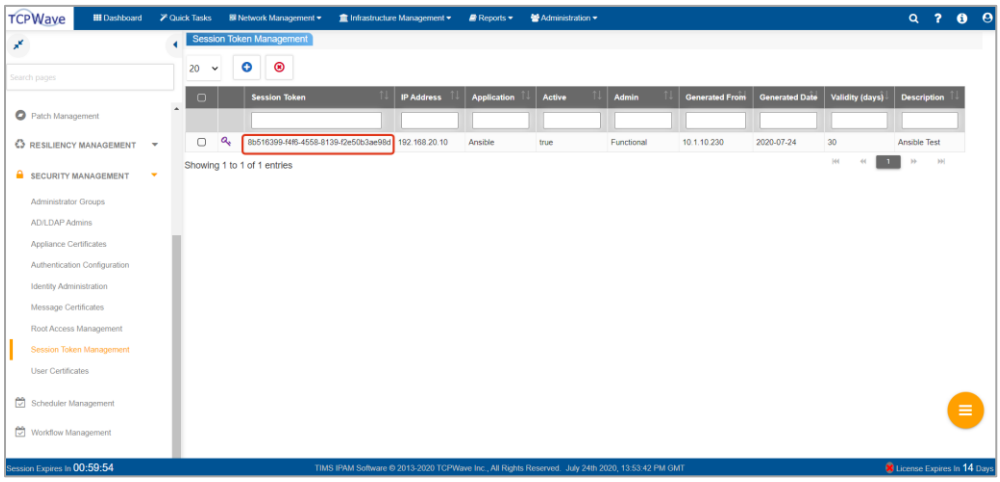


4. Select **Security Management** from the drop-down
5. Click **Add** in the Session Token Management grid
6. System displays **Generate Session Token** pop-up window
7. Enter the **Application**, **Address,** and **Description** fields as shown:



8. Select **Security Management** from the drop-down
9. Click **OK**

10. A new token is generated which can only be accessed from the given IP address as shown:



Example

The following example explains the Session Token Authentication method to invoke the TCPWave REST API using Ansible Playbooks:

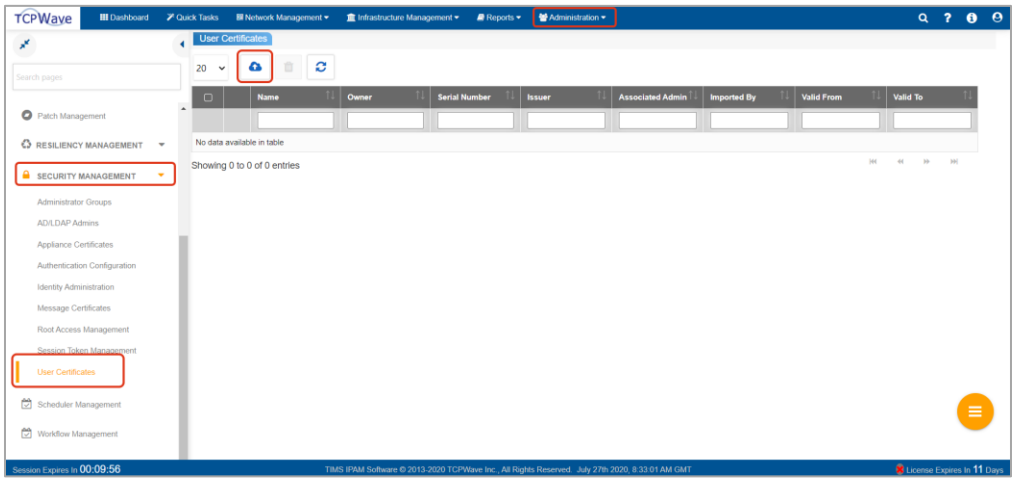
```
---
- name: Tasks to interact with TCPWave IPAM
  hosts: localhost
  tasks:
    - name: Create an organization
      uri:
        url: "https://10.1.10.240:7443/tims/rest/organization/add"
        method: POST
        headers:
          Content-Type: application/json
          TIMS-Session-Token: 6f708f8c-fac6-4e51-9901-c84c4c1b2843
        body: "{{ lookup('file','org.json') }}"
        status_code: 204
        body_format: json
        validate_certs: no
```

Certificate Authentication

This section explains the steps on how to create user certificates in TIMS GUI along with screenshots.

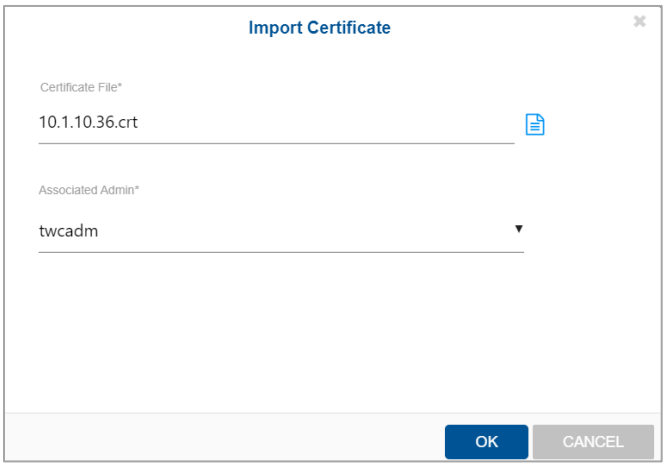
To import the user certificates into the TIMS GUI, follow the given steps:

1. Click **Administration** tab
2. Select **Security Management** from the drop-down
3. Click the **User Certificates** label as shown:



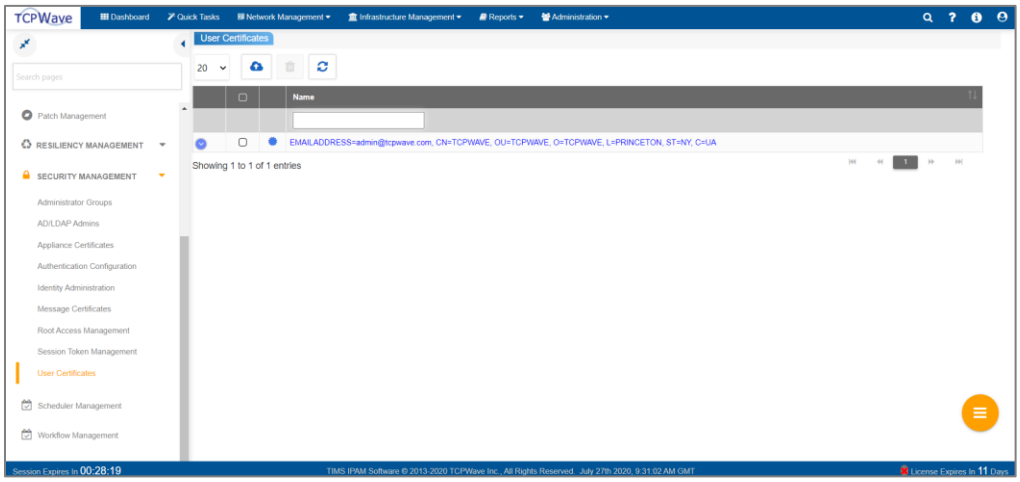
4. Click **Upload** in the User Certificates grid

   The system displays the **Import Certificate** pop-up window as shown:



5. Upload the Certificate File by clicking 
6. Select the **Associate Admin** from the drop-down
7. Click **OK**

   System imports and lists the Certificate in User Certificates grid



Example

The following example explains invoking the TCPWave REST API using Ansible Playbooks via Certification Authentication method:

```
---
- name: Tasks to interact with TCPWave IPAM
  hosts: localhost
  tasks:
    - name: Create an organization
      uri:
        url: "https://10.1.10.240:7443/tims/rest/organization/add"
        method: POST
        headers:
          Content-Type: application/json
        client_cert: "/opt/ansible/keys/ipam_client.crt"
              client_key: "/opt/ansible/keys/ipam_client.key"
        body: "{{ lookup('file','org.json') }}"
        status_code: 204
        body_format: json
        validate_certs: no
```

IPAM Functionalities

This document covers the following functionalities using the Ansible Playbooks

- Create Object RR A
- Create Object RR CNAME
- Create Object RR MX
- Create Object RR TXT
- Create Object RR SRV
- Create Object RR NAPTR
- Create Zone
- Create Zone RR A Record Type
- Create Zone RR CNAME Record Type
- Create Zone RR MX Record Type
- Create Zone RR TXT Record Type
- Create Zone RR AAAA Record Type
- Create Zone RR SRV Record Type
- Create Zone RR NAPTR Record Type
- Create Next Available Subnet
- Create a DHCP Scope

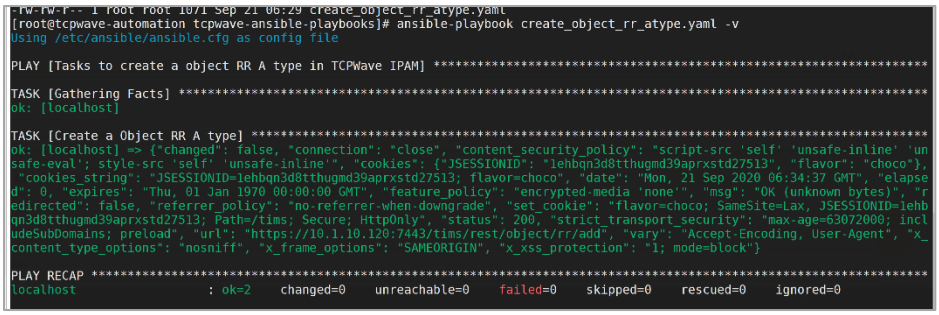
Create Object RR

```
---
- name: Tasks to create an Object RR A type in TCPWave IPAM
hosts: localhost
vars:
user_cert_file: /opt/ansible/keys/ansible.crt
user_cert_key_file: /opt/ansible/keys/ansible.key
ipam_ip: 10.1.10.120
organization_name: Dunkin
owner: ObjectArec
data: 1.0.0.9

 tasks:
- name: Create an Object RR A type
uri:
url: 'https://{{ ipam_ip }}:7443/tims/rest/zone/rr/add'
client_cert: '{{ user_cert_file }}'
client_key: '{{ user_cert_key_file }}'
method: POST
headers:
Content-Type: application/json
body: {
{
            'owner': '{{ owner }}',
            'rrclass': 'IN',
            'rrtype': 'A',
            'ttl': 1200,
            'data': '{{ data }}',
            'ipAddr': '{{ data }}',
            'is_ad_rr': 0,
            'organization_name': '{{ organization_name }}'
}
status_code: 204
body_format: json
validate_certs: no
```

To create an Object RR A type in the TCPWave IPAM, you are required to execute the following command:

```
ansible-playbook create_object_rr_atype.yaml -v
```

Command-Line Output



Create Object RR CNAME

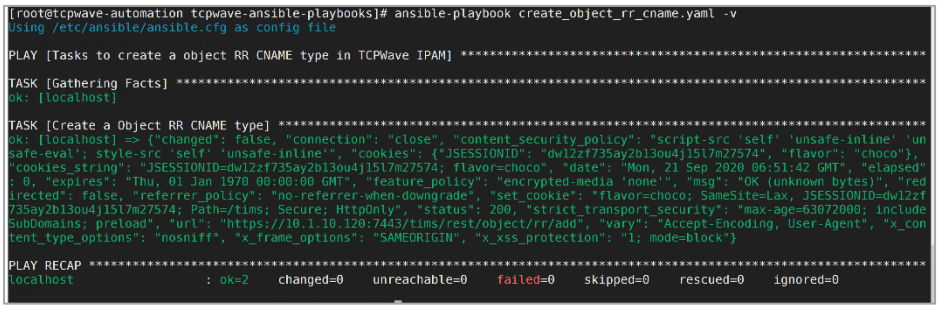
```
--
- name: Tasks to create an Object RR CNAME type in TCPWave IPAM
hosts: localhost
vars:
user_cert_file: /opt/ansible/keys/ansible.crt
user_cert_key_file: /opt/ansible/keys/ansible.key
ipam_ip: 10.1.10.120
organization_name: Dunkin
owner: ObjectC
data: ObjectA
ip: 1.0.0.9

 tasks:
- name: Create an Object RR CNAME type
uri:
url: 'https://{{ ipam_ip }}:7443/tims/rest/object/rr/add'
client_cert: '{{ user_cert_file }}'
client_key: '{{ user_cert_key_file }}'
method: POST
headers:
Content-Type: application/json
body: {
        'owner': '{{ owner }}',
        'rrclass': 'IN',
        'rrtype': 'CNAME',
        'ttl': 1200,
        'data': '{{ data }}',
        'ipAddr': '{{ ip }}',
        'is_ad_rr': 0,
        'organization_name': '{{ organization_name }}'
}
status_code: 200
```

body_format: json
validate_certs: no

To create an Object RR CNAME type in the TCPWave IPAM, you are required to execute the following command:

```
ansible-playbook create_object_rr_cname.yaml -v
```

Command-Line Output



Create Object RR MX

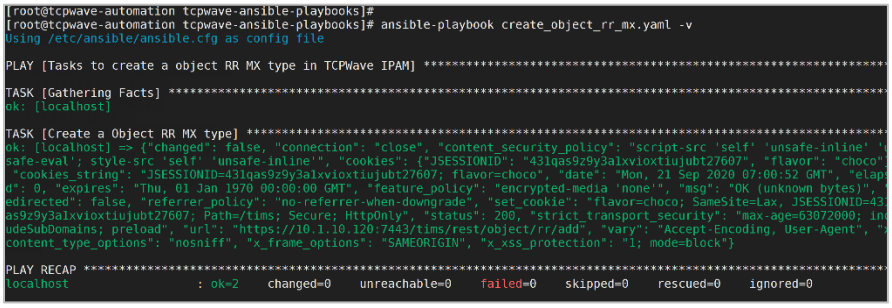
```
---
- name: Tasks to create an Object RR MX type in TCPWave IPAM
hosts: localhost
vars:
user_cert_file: /opt/ansible/keys/ansible.crt
user_cert_key_file: /opt/ansible/keys/ansible.key
ipam_ip: 10.1.10.120
organization_name: Dunkin
owner: ObjectMX
data: 1 ObjectA
ip: 1.0.0.9

 tasks:
- name: Create an Object RR MX type
uri:
url: 'https://{{ ipam_ip }}:7443/tims/rest/object/rr/add'
client_cert: '{{ user_cert_file }}'
client_key: '{{ user_cert_key_file }}'
method: POST
headers:
Content-Type: application/json
body: {
        'owner': '{{ owner }}',
        'rrclass': 'IN',
        'rrtype': 'MX',
```

```
        'ttl': 1200,
        'data': '{{ data }}',
        'ipAddr': '{{ ip }}',
        'is_ad_rr': 0,
        'organization_name': '{{ organization_name }}'
}
status_code: 200
body_format: json
validate_certs: no
```

To create an Object RR MX type in the TCPWave IPAM, you are required to execute the following command:

```
ansible-playbook create_object_rr_mx.yaml -v
```

Command-Line Output



Create Object RR TXT

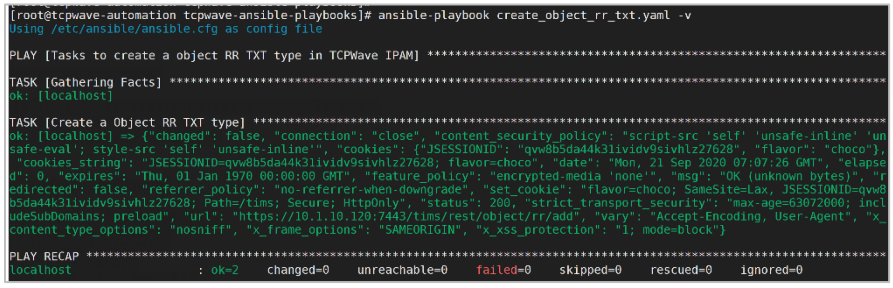
```
---
- name: Tasks to create an Object RR TXT type in TCPWave IPAM
hosts: localhost
vars:
user_cert_file: /opt/ansible/keys/ansible.crt
user_cert_key_file: /opt/ansible/keys/ansible.key
ipam_ip: 10.1.10.120
organization_name: Dunkin
owner: ObjectTXT
data: This is a text rec
ip: 1.0.0.9

 tasks:
- name: Create an Object RR TXT type
uri:
url: 'https://{{ ipam_ip }}:7443/tims/rest/object/rr/add'
client_cert: '{{ user_cert_file }}'
```

```
client_key: '{{ user_cert_key_file }}'
method: POST
headers:
Content-Type: application/json
body: {
        'owner': '{{ owner }}',
        'rrclass': 'IN',
        'rrtype': 'TXT',
        'ttl': 1200,
        'data': '{{ data }}',
        'ipAddr': '{{ ip }}',
        'is_ad_rr': 0,
        'organization_name': '{{ organization_name }}'
}
status_code: 200
body_format: json
validate_certs: no
```

To create an Object RR TXT type in the TCPWave IPAM, you are required to execute the following command:

```
ansible-playbook create_object_rr_txt.yaml -v
```

Command-Line Output



Create Object RR SRV

```
---
- name: Tasks to create an Object RR SRV type in TCPWave IPAM
hosts: localhost
vars:
user_cert_file: /opt/ansible/keys/ansible.crt
user_cert_key_file: /opt/ansible/keys/ansible.key
ipam_ip: 10.1.10.120
organization_name: Dunkin
owner: _1._tcp.ObjectA
data: 1 100 7443 ObjectA
```

```
ip: 1.0.0.9

 tasks:
- name: Create a Object RR SRV type
uri:
url: 'https://{{ ipam_ip }}:7443/tims/rest/object/rr/add'
client_cert: '{{ user_cert_file }}'
client_key: '{{ user_cert_key_file }}'
method: POST
headers:
Content-Type: application/json
body: {
        'owner': '{{ owner }}',
        'rrclass': 'IN',
        'rrtype': 'SRV',
        'ttl': 1200,
        'data': '{{ data }}',
        'ipAddr': '{{ ip }}',
        'is_ad_rr': 0,
        'organization_name': '{{ organization_name }}'
}
status_code: 200
body_format: json
validate_certs: no
```

To create an Object RR SRV type in the TCPWave IPAM, you are required to execute the following command:

```
ansible-playbook create_object_rr_srv.yaml -v
```

Command-Line Output

```
[root@tcpwave-automation tcpwave-ansible-playbooks]# ansible-playbook create_object_rr_srv.yaml -v
Using /etc/ansible/ansible.cfg as config file

PLAY [Tasks to create a object RR SRV type in TCPWave IPAM] ********************************************************************

TASK [Gathering Facts] *********************************************************************************************************
ok: [localhost]

TASK [Create a Object RR SRV type] *********************************************************************************************
ok: [localhost] => {"changed": false, "connection": "close", "content_security_policy": "script-src 'self' 'unsafe-inline' 'un
safe-eval'; style-src 'self' 'unsafe-inline'", "cookies": {"JSESSIONID": "1fz2al36s97d41a6giogwpn8qm27696", "flavor": "choco"}
, "cookies_string": "JSESSIONID=1fz2al36s97d41a6giogwpn8qm27696; flavor=choco", "date": "Mon, 21 Sep 2020 07:28:22 GMT", "elap
sed": 0, "expires": "Thu, 01 Jan 1970 00:00:00 GMT", "feature_policy": "encrypted-media 'none'", "msg": "OK (unknown bytes)",
"redirected": false, "referrer_policy": "no-referrer-when-downgrade", "set_cookie": "flavor=choco; SameSite=Lax, JSESSIONID=1f
z2al36s97d41a6giogwpn8qm27696; Path=/tims; Secure; HttpOnly", "status": 200, "strict_transport_security": "max-age=63072000; i
ncludeSubDomains; preload", "url": "https://10.1.10.120:7443/tims/rest/object/rr/add", "vary": "Accept-Encoding, User-Agent",
"x_content_type_options": "nosniff", "x_frame_options": "SAMEORIGIN", "x_xss_protection": "1; mode=block"}

PLAY RECAP *********************************************************************************************************************
localhost                  : ok=2    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

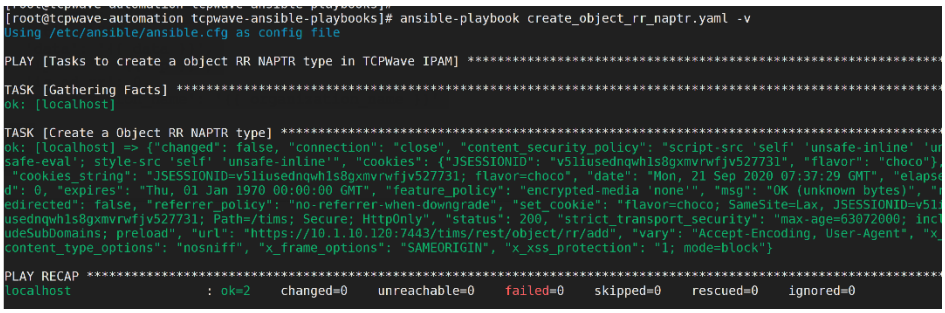
Create Object RR NAPTR

```
---
- name: Tasks to create an Object RR NAPTR type in TCPWave IPAM
hosts: localhost
vars:
user_cert_file: /opt/ansible/keys/ansible.crt
user_cert_key_file: /opt/ansible/keys/ansible.key
ipam_ip: 10.1.10.120
organization_name: Dunkin
owner: NAPTR
data: 100 100 U E2U+si example.com.
ip: 1.0.0.9

 tasks:
- name: Create an Object RR NAPTR type
uri:
url: 'https://{{ ipam_ip }}:7443/tims/rest/object/rr/add'
client_cert: '{{ user_cert_file }}'
client_key: '{{ user_cert_key_file }}'
method: POST
headers:
Content-Type: application/json
body: {
            'owner': '{{ owner }}',
            'rrclass': 'IN',
            'rrtype': 'NAPTR',
            'ttl': 1200,
            'data': '{{ data }}',
            'ipAddr': '{{ ip }}',
            'is_ad_rr': 0,
            'organization_name': '{{ organization_name }}'
}
status_code: 200
body_format: json
validate_certs: no
```

To create an Object RR NAPTR type in the TCPWave IPAM, you are required to execute the following command:

```
ansible-playbook create_object_rr_naptr.yaml -v
```

Command-Line Output



Create Zone

```
---
- name: Tasks to create a Zone in TCPWave IPAM
  hosts: localhost
  vars:
    user_cert_file: /opt/ansible/keys/ansible.crt
    user_cert_key_file: /opt/ansible/keys/ansible.key
    ipam_ip: 10.1.10.120
    organization_name: Dunkin
    zone_name: test.in
    template_name: 'test'

  tasks:
    - name: Create a Zone
      uri:
        url: 'https://{{ ipam_ip }}:7443/tims/rest/zone/add'
        client_cert: '{{ user_cert_file }}'
        client_key: '{{ user_cert_key_file }}'
        method: POST
        headers:
          Content-Type: application/json
        body: {
        {
                    'name': '{{ zone_name }}',
                    'template_name': '{{ template_name }}',
                    'importCloudRR': true,
                    'description': '',
                    'dnssec_enable': 'no',
                    'restrictedZone': 'no',
                    'monitoringService': 'yes',
                    'extensions': [],
                    'secureActiveDirectoryEnable': 'no',
                    'addedARRs': [],
```

```
            'deletedARRs': [],
            'zoneRR': [],
            'organization_name': '{{ organization_name }}'
    }
    status_code: 204
    body_format: json
    validate_certs: no
```

To create a Zone in the TCPWave IPAM, you are required to execute the following command:

```
$ ansible-playbook create_zone.yaml  -v
```

Command-Line Output

Create Zone RR A Record Type

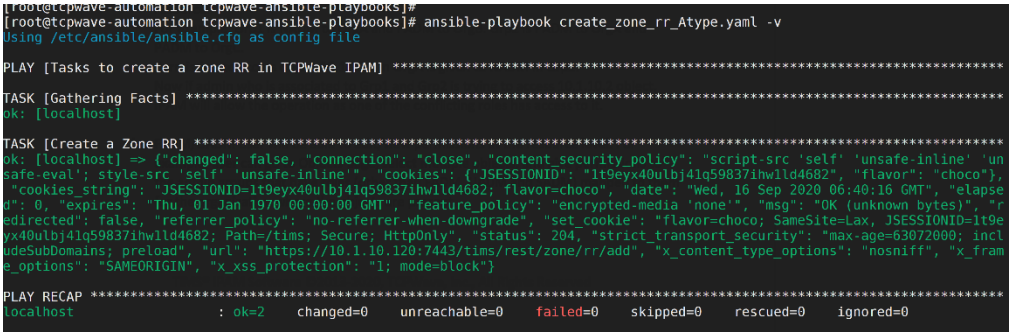
```
 ---

- name: Tasks to create a Zone RR in TCPWave IPAM
  hosts: localhost
  vars:
    user_cert_file: /opt/ansible/keys/ansible.crt
    user_cert_key_file: /opt/ansible/keys/ansible.key
    ipam_ip: 10.1.10.120
    organization_name: Dunkin
    zone_name: test.in
    owner_name: ARecord
    data: 1.0.0.2

  tasks:
    - name: Create a Zone RR
      uri:
        url: 'https://{{ ipam_ip }}:7443/tims/rest/zone/rr/add'
        client_cert: '{{ user_cert_file }}'
        client_key: '{{ user_cert_key_file }}'
        method: POST
        headers:
```

```
  Content-Type: application/json
body: {
{
   'zoneName': '{{ zone_name }}',
   'owner': '{{owner_name}}',
   'rrclass': 'IN',
   'rrtype': 'A',
   'ttl': '1200',
   'data': '{{ data }}',
   'description': '',
   'is_external_rr': 0,
   'status': 1,
   'organization_name': '{{ organization_name }}'
}
status_code: 204
body_format: json
validate_certs: no
```

To create a Zone RR in the TCPWave IPAM, you are required to execute the following command:

```
ansible-playbook create_zone_rr_Atype.yaml -v
```

Command-Line Output



Create Zone RR CNAME Record Type

```
---
- name: Tasks to create a Zone RR CNAME in TCPWave IPAM
  hosts: localhost
  vars:
    user_cert_file: /opt/ansible/keys/ansible.crt
    user_cert_key_file: /opt/ansible/keys/ansible.key
    ipam_ip: 10.1.10.120
    organization_name: Dunkin
    zone_name: test.in
    owner_name: CRecord.test.in.
```

```
  data: ARecord.Ansible.in.

 tasks:
  - name: Create a Zone RR CNAME
    uri:
      url: 'https://{{ ipam_ip }}:7443/tims/rest/zone/rr/add'
      client_cert: '{{ user_cert_file }}'
      client_key: '{{ user_cert_key_file }}'
      method: POST
      headers:
        Content-Type: application/json
      body: {
      {
         'zoneName': '{{ zone_name }}',
         'owner': '{{owner_name}}',
         'rrclass': 'IN',
         'rrtype': 'CNAME',
         'ttl': '1200',
         'data': '{{data}}',
         'description': '',
         'is_external_rr': 0,
         'status': 1,
         'organization_name': '{{ organization_name }}'
      }
      status_code: 204
      body_format: json
      validate_certs: no
```

To create a Zone RR CNAME in the TCPWave IPAM, you are required to execute the following command:

```
ansible-playbook create_zone_rr_cname.yaml -v
```

Command-Line Output

```
[root@tcpwave-automation tcpwave-ansible-playbooks]# ansible-playbook create_zone_rr_cname.yaml -v
Using /etc/ansible/ansible.cfg as config file

PLAY [Tasks to create a zone RR CNAME in TCPWave IPAM] ***************************************************************

TASK [Gathering Facts] ***********************************************************************************************
ok: [localhost]

TASK [Create a Zone RR CNAME] ****************************************************************************************
ok: [localhost] => {"changed": false, "connection": "close", "content_security_policy": "script-src 'self' 'unsafe-inline' 'un
safe-eval'; style-src 'self' 'unsafe-inline'", "cookies": {"JSESSIONID": "1t4wbd4chv6npwtqnvth960la4769", "flavor": "choco"},
"cookies_string": "JSESSIONID=1t4wbd4chv6npwtqnvth960la4769; flavor=choco", "date": "Wed, 16 Sep 2020 07:06:14 GMT", "elapsed"
: 0, "expires": "Thu, 01 Jan 1970 00:00:00 GMT", "feature_policy": "encrypted-media 'none'", "msg": "OK (unknown bytes)", "red
irected": false, "referrer_policy": "no-referrer-when-downgrade", "set_cookie": "flavor=choco; SameSite=Lax, JSESSIONID=1t4wbd
4chv6npwtqnvth960la4769; Path=/tims; Secure; HttpOnly", "status": 204, "strict_transport_security": "max-age=63072000; include
SubDomains; preload", "url": "https://10.1.10.120:7443/tims/rest/zone/rr/add", "x_content_type_options": "nosniff", "x_frame_o
ptions": "SAMEORIGIN", "x_xss_protection": "1; mode=block"}

PLAY RECAP ***********************************************************************************************************
localhost                  : ok=2    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

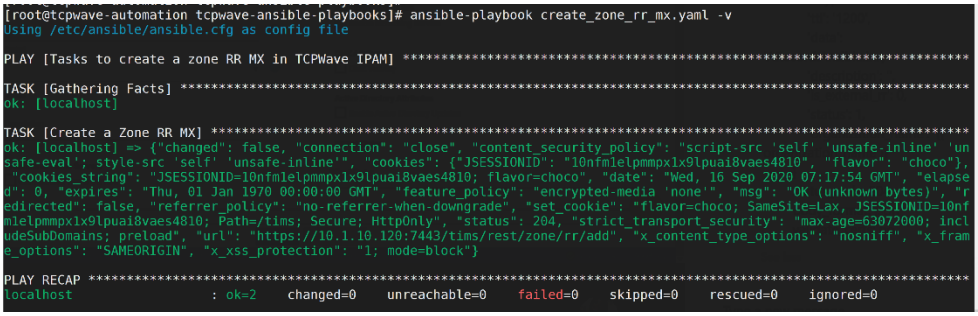
Create Zone RR MX Record Type

```
---
- name: Tasks to create a Zone RR MX in TCPWave IPAM
  hosts: localhost
  vars:
    user_cert_file: /opt/ansible/keys/ansible.crt
    user_cert_key_file: /opt/ansible/keys/ansible.key
    ipam_ip: 10.1.10.120
    organization_name: Dunkin
    zone_name: test.in
    owner_name: MXRecord.test.in.
    data: 100 ARecord.Ansible.in.


  tasks:
    - name: Create a Zone RR MX
      uri:
        url: 'https://{{ ipam_ip }}:7443/tims/rest/zone/rr/add'
        client_cert: '{{ user_cert_file }}'
        client_key: '{{ user_cert_key_file }}'
        method: POST
        headers:
          Content-Type: application/json
        body: {
        {
           'zoneName': '{{ zone_name }}',
           'owner': '{{owner_name}}',
           'rrclass': 'IN',
           'rrtype': 'MX',
           'ttl': '1200',
           'data': '{{data}}',
           'description': '',
           'is_external_rr': 0,
           'status': 1,
           'organization_name': '{{ organization_name }}'
        }
        status_code: 204
        body_format: json
        validate_certs: no
```

To create a Zone RR MX in the TCPWave IPAM, you are required to execute the following command:

```
ansible-playbook create_zone_rr_mx.yaml -v
```

Command-Line Output



Create Zone RR TXT Record Type

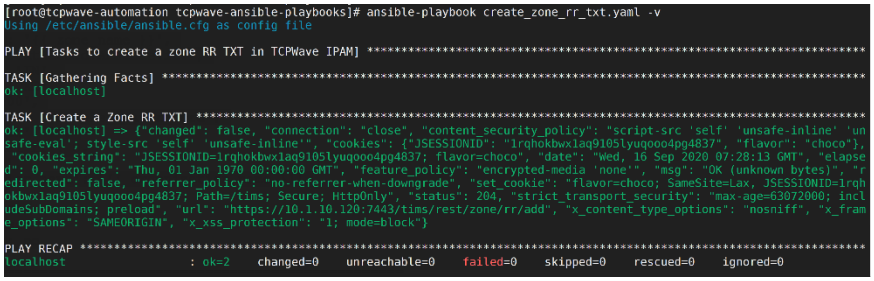
```
---
- name: Tasks to create a Zone RR TXT in TCPWave IPAM
  hosts: localhost
  vars:
    user_cert_file: /opt/ansible/keys/ansible.crt
    user_cert_key_file: /opt/ansible/keys/ansible.key
    ipam_ip: 10.1.10.120
    organization_name: Dunkin
    zone_name: test.in
    owner_name: TextRecord.test.in.
    data: This a sample text record

  tasks:
    - name: Create a Zone RR TXT
      uri:
        url: 'https://{{ ipam_ip }}:7443/tims/rest/zone/rr/add'
        client_cert: '{{ user_cert_file }}'
        client_key: '{{ user_cert_key_file }}'
        method: POST
        headers:
          Content-Type: application/json
        body: {
        {
           'zoneName': '{{ zone_name }}',
           'owner': '{{owner_name}}',
           'rrclass': 'IN',
           'rrtype': 'TXT',
           'ttl': '1200',
           'data': '{{data}}',
           'description': '',
           'is_external_rr': 0,
           'status': 1,
```

```
    'organization_name': '{{ organization_name }}'
  }
  status_code: 204
  body_format: json
  validate_certs: no
```

To create a Zone RR TXT in the TCPWave IPAM, you are required to execute the following command:

```
ansible-playbook create_zone_rr_txt.yaml -v
```

Command-Line Output



Create Zone RR AAAA Record Type

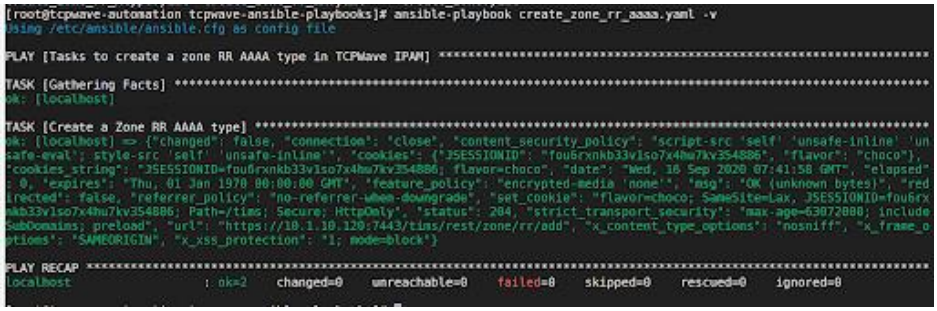
```
---
- name: Tasks to create a Zone RR AAAA type in TCPWave IPAM
  hosts: localhost
  vars:
    user_cert_file: /opt/ansible/keys/ansible.crt
    user_cert_key_file: /opt/ansible/keys/ansible.key
    ipam_ip: 10.1.10.120
    organization_name: Dunkin
    zone_name: test.in
    owner_name: ARecord
    data: 2001:0db8:85a3:0000:0000:8a2e:0370:7334


  tasks:
    - name: Create a Zone RR AAAA type
      uri:
        url: 'https://{{ ipam_ip }}:7443/tims/rest/zone/rr/add'
        client_cert: '{{ user_cert_file }}'
        client_key: '{{ user_cert_key_file }}'
        method: POST
        headers:
          Content-Type: application/json
        body: {
        {
```

```
        'zoneName': '{{ zone_name }}',
        'owner': '{{owner_name}}',
        'rrclass': 'IN',
        'rrtype': 'AAAA',
        'ttl': '1200',
        'data': '{{data}}',
        'description': '',
        'is_external_rr': 0,
        'status': 1,
        'organization_name': '{{ organization_name }}'
    }
    status_code: 204
    body_format: json
    validate_certs: no
```

To create a Zone RR AAAA in the TCPWave IPAM, you are required to execute the following command:

```
ansible-playbook create_zone_rr_aaaa.yaml -v
```

Command-Line Output



Create Zone RR SRV Record Type

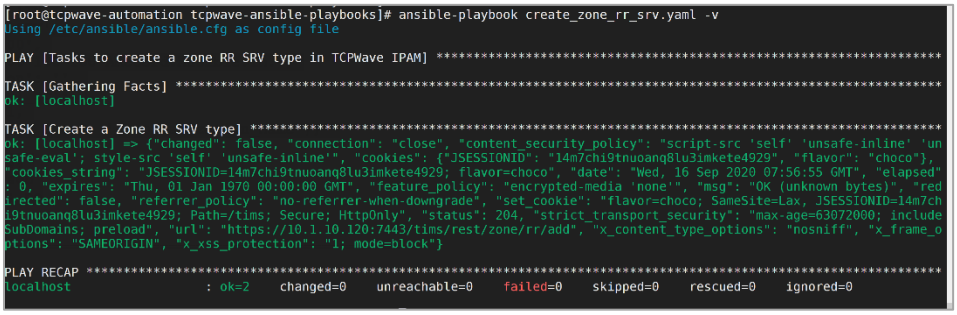
```
---
- name: Tasks to create a Zone RR SRV type in TCPWave IPAM
  hosts: localhost
  vars:
    user_cert_file: /opt/ansible/keys/ansible.crt
    user_cert_key_file: /opt/ansible/keys/ansible.key
    ipam_ip: 10.1.10.120
    organization_name: Dunkin
    zone_name: Ansible.in
    owner_name: _http._TCP.test.in.
    data: 100 100 100 ARecord.test.in.


  tasks:
```

```
- name: Create a Zone RR SRV type
  uri:
    url: 'https://{{ ipam_ip }}:7443/tims/rest/zone/rr/add'
    client_cert: '{{ user_cert_file }}'
    client_key: '{{ user_cert_key_file }}'
    method: POST
    headers:
      Content-Type: application/json
    body: {
    {
       'zoneName': '{{ zone_name }}',
       'owner': '{{owner_name}}',
       'rrclass': 'IN',
       'rrtype': 'SRV',
       'ttl': '1200',
       'data': '{{data}}',
       'description': '',
       'is_external_rr': 0,
       'status': 1,
       'organization_name': '{{ organization_name }}'
    }
    status_code: 204
    body_format: json
    validate_certs: no
```

To create a Zone RR SRV in the TCPWave IPAM, you are required to execute the following command:

ansible-playbook create_zone_rr_srv.yaml -v

Command-Line Output



Create Zone RR NAPTR Record Type

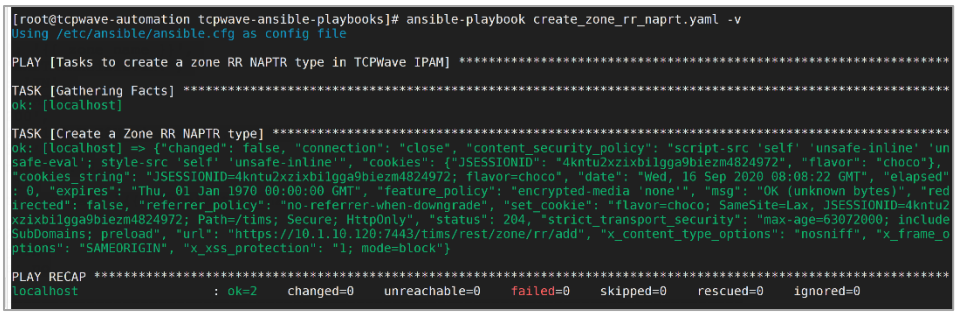
```
---
- name: Tasks to create a Zone RR NAPTR type in TCPWave IPAM
  hosts: localhost
```

```
vars:
  user_cert_file: /opt/ansible/keys/ansible.crt
  user_cert_key_file: /opt/ansible/keys/ansible.key
  ipam_ip: 10.1.10.120
  organization_name: Dunkin
  zone_name: test.in
  owner_name: NAPTR.test.in.
  data: 100 100 U E2U+si example.com.


tasks:
  - name: Create a Zone RR NAPTR type
    uri:
      url: 'https://{{ ipam_ip }}:7443/tims/rest/zone/rr/add'
      client_cert: '{{ user_cert_file }}'
      client_key: '{{ user_cert_key_file }}'
      method: POST
      headers:
        Content-Type: application/json
      body: {
      {
          'zoneName': '{{ zone_name }}',
          'owner': '{{owner_name}}',
          'rrclass': 'IN',
          'rrtype': 'NAPTR',
          'ttl': '1200',
          'data': '{{data}}',
          'description': '',
          'is_external_rr': 0,
          'status': 1,
          'organization_name': '{{ organization_name }}'
      }
      status_code: 204
      body_format: json
      validate_certs: no
```

To create a Zone RR NAPTR in the TCPWave IPAM, you are required to execute the following command:

```
ansible-playbook create_zone_rr_naptr.yaml -v
```

Command-Line Output



Create Next Available Subnet

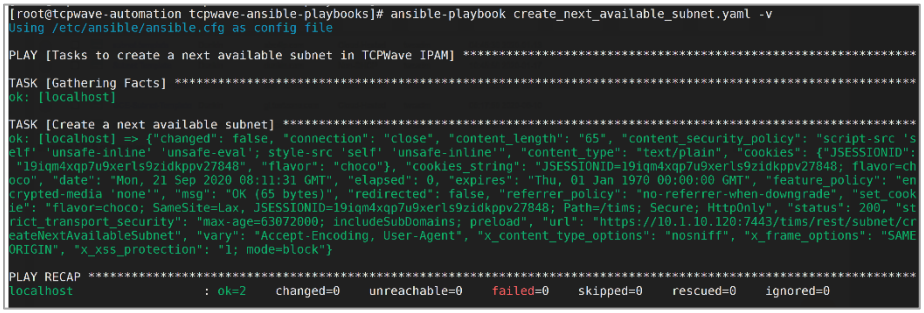
- name: Tasks to create a next available subnet in TCPWave IPAM
hosts: localhost
vars:
user_cert_file: /opt/ansible/keys/ansible.crt
user_cert_key_file: /opt/ansible/keys/ansible.key
ipam_ip: 10.1.10.120
organization_name: Dunkin
name: Ansible Subnet
subnet_template: subnet-template
ip: 10.10.0.0/16

tasks:
- name: Create a next available subnet
uri:
url: 'https://{{ ipam_ip }}:7443/tims/rest/subnet/createNextAvailableSubnet'
client_cert: '{{ user_cert_file }}'
client_key: '{{ user_cert_key_file }}'
method: POST
headers:
Content-Type: application/json
body: {
'mask_length':24,
'name':'{{ name }}',
'subnetTemplateName':'{{ subnet_template }}',
'network_address':'{{ ip }}',
'organization_name':'{{ organization_name }}'
}
status_code: 200
body_format: json
validate_certs: no

To create the next available Subnet in the TCPWave IPAM, you are required to execute the following command:

```
ansible-playbook create_next_available_subnet.yaml -v
```

Command-Line Output



Create a DHCP Scope

```
---
- name: Tasks to create a DHCP Scopes in TCPWave IPAM
hosts: localhost
vars:
user_cert_file: /opt/ansible/keys/ansible.crt
user_cert_key_file: /opt/ansible/keys/ansible.key
ipam_ip: 10.1.10.120
organization_name: Dunkin
server_name: CS-DHCP-Primary
template: DHCP-Std-Option-Template
ip: 1.0.0.0
start_ip : 1.0.0.60
end_ip : 1.0.0.70

 tasks:
- name: Create a DHCP Scopes
uri:
url: 'https://{{ ipam_ip }}:7443/tims/rest/scope/create'
client_cert: '{{ user_cert_file }}'
client_key: '{{ user_cert_key_file }}'
method: POST
headers:
Content-Type: application/json
body: {
            'ttl': '1200',
            'scope': {
```

```
        'class_code': '3G Phone',
        'description': '',
        'organization_name': '{{ organization_name }}',
        'primary_dhcp_server': '{{ server_name }}',
        'allocation_type': 'dynamic',
        'template_name': '{{ template }}',
        'addressRanges': [
        {
        'startIP': '{{ start_ip }}',
        'endIP': '{{ end_ip }}'
        }
        ]
        },
        'subnetAddress': '{{ ip }}'
}
status_code: 200
body_format: json
validate_certs: no
```

To create DHCP Scope in the TCPWave IPAM, you are required to execute the following command:

```
ansible-playbook create_dhcp_scopes.yaml -v
```

Command-Line Output