Generated FortiManager Ansible Modules

Test Report

Version 1.0

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| Nov 19, 2019 | 1.0 | Created | Link Zheng |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table of Contents

[1. Introduction 5](#_Toc25674702)

[2. Requirements for Test 5](#_Toc25674703)

[3. Test Strategy 5](#_Toc25674704)

[3.1 Topology Diagram 5](#_Toc25674705)

[3.2 Function Testing 7](#_Toc25674706)

[3.2.1 Device Operations Test 8](#_Toc25674707)

[3.2.2 Device Group Operation Test 12](#_Toc25674708)

[3.2.3 Script Operation Test 16](#_Toc25674709)

[3.2.4 Firewall IPv4 Address Object Test 21](#_Toc25674710)

[3.2.5 Firewall IPv6 Address Object Test 25](#_Toc25674711)

[3.2.6 Firewall IPv4 Ippool Object Test 29](#_Toc25674712)

[3.2.7 Firewall IPv6 Ippool Object Test 33](#_Toc25674713)

[3.2.8 Firewall IPv4 VIP Object Test 37](#_Toc25674714)

[3.2.9 Firewall IPv6 VIP Object Test 41](#_Toc25674715)

[3.2.10 Firewall Policy Package Test 45](#_Toc25674716)

[3.2.11 Firewall IPv4 Policy Test 48](#_Toc25674717)

[3.2.12 Firewall IPv6 Policy Test 54](#_Toc25674718)

[3.2.13 Firewall Service Test 54](#_Toc25674719)

[3.2.14 Security Profile DNS Filter Test 58](#_Toc25674720)

[3.2.15 Security Profile Anti Virus Test 62](#_Toc25674721)

[3.2.16 Security Profile IPS Sensor Test 67](#_Toc25674722)

[3.2.17 Security Profile Web Filter Test 71](#_Toc25674723)

[3.2.18 Security Profile Application Control Test 76](#_Toc25674724)

[3.2.19 Security Profile Group Test 80](#_Toc25674725)

[3.2.20 Security Profile Web Proxy Test 84](#_Toc25674726)

[3.2.21 Security Profile Spam Filter Test 88](#_Toc25674727)

[3.2.22 Security Profile ssl-ssh Test 92](#_Toc25674728)

[3.2.23 Security Profile voip Test 97](#_Toc25674729)

[3.2.24 Security Profile WAF Test 102](#_Toc25674730)

[3.2.25 Security Profile Wanopt Test 106](#_Toc25674731)

[3.2.26 Device Profile Test 111](#_Toc25674732)

[3.2.27 Provisioning Template Test 115](#_Toc25674733)

[3.2.28 [template] 118](#_Toc25674734)

[3.3 User Interface Testing 119](#_Toc25674735)

[3.4 Performance Profiling 119](#_Toc25674736)

[3.5 Load Testing 119](#_Toc25674737)

[3.6 Stress Testing 119](#_Toc25674738)

[3.7 Security and Access Control Testing 119](#_Toc25674739)

[3.8 Failover and Recovery Testing 119](#_Toc25674740)

[4. Resources 119](#_Toc25674741)

[4.1 Workers 119](#_Toc25674742)

[4.2 System Resources 119](#_Toc25674743)

[5. Deliverables 119](#_Toc25674744)

[6. Appendix A: Sample Full Terraform Settings 119](#_Toc25674745)

# Introduction

The document gives the functional tests result of the major FortiManager components using generated

FortiManager ansible modules.

# Requirements for Test

The test requires one Linux host for issuing Ansible playbook and one FortiManager host and one FortiGate host.

# Test Strategy

## Topology Diagram

The following diagram shows the network topology.

**Note** this is how my tested looks like, there is no need to follow the exact topology as long as the connectivity between any of them is good.

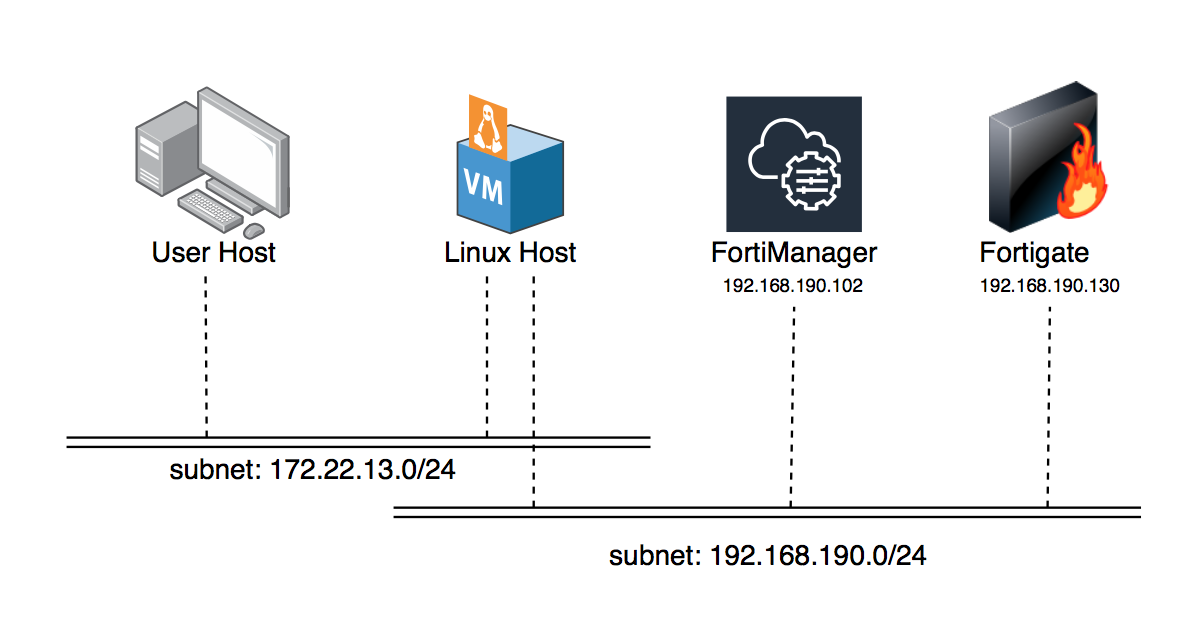


Figure The topology of devices under test

user and password for the devices:

FortiManager: APIUser/Fortinet1!

Fortigate: admin/admin

The Ansible inventory has the FortiManager address and credentials:

Table The ansible inventory of fortimanager

|  |
| --- |
| # cat hosts/hosts  [myfortimanagers]  fortimanager01 ansible\_host=192.168.190.102 ansible\_user=APIUser ansible\_password=Fortinet1!  [myfortimanagers:vars]  ansible\_network\_os=fortimanager |

## Function Testing

### Device Operations Test

|  |  |
| --- | --- |
| Test Objective: | Test The Basic operations with FortiManager device  Operations include: **Add**, **Install** , **Get** and **Delete** |
| Technique: | Playbooks for each operation:   * Add a Fortigate Device   - hosts: fortimanager01  connection: httpapi  vars:  adom: "root"  ansible\_httpapi\_use\_ssl: True  ansible\_httpapi\_validate\_certs: False  ansible\_httpapi\_port: 443  tasks:  - name: 'add a device'  fmgr\_dvm\_cmd\_add\_device:  method: "exec"  params:  - data:  adom: 'root'  device:  adm\_usr: 'admin'  adm\_pass:  - 'admin'  desc: 'The Fortigate with IP 192.168.190.130'  ip: '192.168.190.130'  os\_type: 'fos'  name: 'FGVM04TM19006963'  sn: 'FGVM04TM19006963'   * Install a Fortigate device   - hosts: fortimanager01  connection: httpapi  vars:  adom: "root"  ansible\_httpapi\_use\_ssl: True  ansible\_httpapi\_validate\_certs: False  ansible\_httpapi\_port: 443  tasks:  - name: 'install a device'  fmgr\_securityconsole\_install\_device:  method: "exec"  params:  - data:  adom: 'root'  dev\_rev\_comments: 'This is to install the FGT device'  scope:  - name: 'FGVM04TM19006963'  vdom: 'root'   * Get a Fortigate device   - hosts: fortimanager01  connection: httpapi  vars:  adom: "root"  ansible\_httpapi\_use\_ssl: True  ansible\_httpapi\_validate\_certs: False  ansible\_httpapi\_port: 443  tasks:  - name: 'add a device'  fmgr\_dvmdb\_device:  method: "get"   * Delete a fortigate device   - hosts: fortimanager01  connection: httpapi  vars:  adom: "root"  ansible\_httpapi\_use\_ssl: True  ansible\_httpapi\_validate\_certs: False  ansible\_httpapi\_port: 443  tasks:  - name: 'delete a device'  fmgr\_dvm\_cmd\_del\_device:  method: "exec"  params:  - data:  adom: 'root'  device: 'FGVM04TM19006963'  **the playbooks list:**   * device\_add.yml * device\_delete.yml * device\_get.yml * device\_install.yml |
| Validation: | * ansible-playbook -i hosts/ device\_add.yml -vvv   text in response:  "device": {  "adm\_usr": "admin",  "beta": -1,  "conn\_mode": 1,  "desc": "The Fortigate with IP 192.168.190.130",  "dev\_status": 1,  "flags": 2097152,  "ip": "192.168.190.130",  "maxvdom": 10,  "mr": -1,  "name": "FGVM04TM19006963",  "os\_type": 0,  "os\_ver": -1,  "patch": -1,  "platform\_id": 111,  "platform\_str": "FortiGate-VM",  "sn": "FGVM04TM19006963",  "source": 1,  "tab\_status": "<unknown>",  "vm.lic\_type": 4,  "vm\_lic\_expire": 0  }   * ansible-playbook -i hosts/ device\_install.yml -vvv   "response": [  0,  {  "task": 43  }  ]   * ansible-playbook -i hosts/ device\_get.yml –vvv   the full configuration for all devices could be seen in response.   * ansible-playbook -i hosts/ device\_delete.yml -vvv   "response": [  0,  {  "status": {  "code": 0,  "message": "OK"  },  "url": "/dvm/cmd/del/device"  }  ]   * run device\_delete.yml one more time:   ansible-playbook -i hosts/ device\_delete.yml -vvv  "response": [  -20002,  {  "status": {  "code": -20002,  "message": "Invalid argument"  },  "url": "/dvm/cmd/del/device"  }  ] |
| Completion Criteria: | Success |
| Special Considerations: |  |

### Device Group Operation Test

|  |  |
| --- | --- |
| Test Objective: | Test The Basic operations with FortiManager device Group  Operations include: **Add**, **Get**, **Delete** and **add/delete** device member |
| Technique: | Playbooks for each operation:   * Add a Device Group   - hosts: fortimanager01  connection: httpapi  vars:  adom: "root"  ansible\_httpapi\_use\_ssl: True  ansible\_httpapi\_validate\_certs: False  ansible\_httpapi\_port: 443  tasks:  - name: 'add a device group'  fmgr\_dvmdb\_adom\_adom\_group:  method: 'add'  params:  - data:  - name: 'demo\_group'  desc: 'The demo group added'  os\_type: 'fos'  type: 'normal'  url\_params:  adom: 'root'   * Get all device groups   - hosts: fortimanager01  connection: httpapi  vars:  adom: "root"  ansible\_httpapi\_use\_ssl: True  ansible\_httpapi\_validate\_certs: False  ansible\_httpapi\_port: 443  tasks:  - name: 'get the device group'  fmgr\_dvmdb\_adom\_adom\_group:  method: 'get'  url\_params:  adom: 'root'  params:  - sortings:  - name: 1   * Add a device as the member of a device group   - hosts: fortimanager01  connection: httpapi  vars:  adom: "root"  ansible\_httpapi\_use\_ssl: True  ansible\_httpapi\_validate\_certs: False  ansible\_httpapi\_port: 443  tasks:  - name: 'add a device as a member'  fmgr\_dvmdb\_adom\_adom\_group\_group:  method: 'add'  url\_params:  adom: 'root'  group: 'demo\_group'  params:  - data:  - name: 'FGVM04TM19006963'  vdom: 'root'   * Delete a device group’s device member   - hosts: fortimanager01  connection: httpapi  vars:  adom: "root"  ansible\_httpapi\_use\_ssl: True  ansible\_httpapi\_validate\_certs: False  ansible\_httpapi\_port: 443  tasks:  - name: 'delete a device member from a device group'  fmgr\_dvmdb\_adom\_adom\_group\_group:  method: 'delete'  url\_params:  adom: 'root'  group: 'demo\_group'  params:  - data:  - name: 'FGVM04TM19006963'  vdom: 'root'   * Delete a device group   - hosts: fortimanager01  connection: httpapi  vars:  adom: "root"  ansible\_httpapi\_use\_ssl: True  ansible\_httpapi\_validate\_certs: False  ansible\_httpapi\_port: 443  tasks:  - name: 'delete a device group'  fmgr\_dvmdb\_adom\_adom\_group\_group:  method: 'delete'  url\_params:  adom: 'root'  group: 'demo\_group'  **the playbooks list**:   * device\_group\_add\_device\_member.yml * device\_group\_add.yml * device\_group\_delete\_device\_member.yml * device\_group\_delete.yml * device\_group\_get.yml   **requirements**:  the device was added to the manager to the FortiManager. |
| Validation: | * ansible-playbook -i hosts/ device\_group\_add.yml -vvv   "response": [  0,  {  "status": {  "code": 0,  "message": "OK"  },  "url": "/dvmdb/adom/root/group"  }  ]   * ansible-playbook -i hosts/ device\_group\_get.yml -vvv   "response": [  0,  [  {  "desc": "",  "name": "All\_FortiGate",  "oid": 101,  "os\_type": "fos",  "type": "default"  },  {  "desc": "The demo group added",  "name": "demo\_group",  "oid": 136,  "os\_type": "fos",  "type": "normal"  }  ]  ]   * ansible-playbook -i hosts/ device\_group\_add\_device\_member.yml -vvv * ansible-playbook -i hosts/ device\_group\_delete\_device\_member.yml -vvv * ansible-playbook -i hosts/ device\_group\_delete.yml -vvv   "response": [  0,  {  "status": {  "code": 0,  "message": "OK"  },  "url": "/dvmdb/adom/root/group/demo\_group"  }  ] |
| Completion Criteria: | Partial success, the group member operations not successful. |
| Special Considerations: |  |

### Script Operation Test

|  |  |
| --- | --- |
| Test Objective: | Test The Basic operations with FortiManager scripts  Operations include: **Add**, **Set**, **Get**, **Eexcute** and **Delete**  **\*** The difference between **Add** and **Set**:  Operation Set is an incremental update for the script object and it can be invoked more than once with warnings or errors. Operation Add will report “Object already exists” if the script object indeed exists. |
| Technique: | Playbooks for each operation:   * Add a script   - hosts: fortimanager01  connection: httpapi  vars:  adom: "root"  ansible\_httpapi\_use\_ssl: True  ansible\_httpapi\_validate\_certs: False  ansible\_httpapi\_port: 443  tasks:  - name: 'add a script'  fmgr\_dvmdb\_script:  method: "add"  params:  - data:  - name: 'add\_firewall\_policy0'  desc: 'the demo script to add'  type: 'cli'  filter\_ostype: 'unknown'  content: |  config firewall policy  edit 1  set name foopolicy  next  end   * Update a script   - hosts: fortimanager01  connection: httpapi  vars:  adom: "root"  ansible\_httpapi\_use\_ssl: True  ansible\_httpapi\_validate\_certs: False  ansible\_httpapi\_port: 443  tasks:  - name: 'update a script'  fmgr\_dvmdb\_script:  method: "set"  params:  - data:  - name: 'add\_firewall\_policy0'  desc: 'the demo script to set'  type: 'cli'  filter\_ostype: 'unknown'  content: |  config firewall policy  edit 1  set name foopolicy  next  end   * Get all the scripts   - hosts: fortimanager01  connection: httpapi  vars:  adom: "root"  ansible\_httpapi\_use\_ssl: True  ansible\_httpapi\_validate\_certs: False  ansible\_httpapi\_port: 443  tasks:  - name: 'fetch all scripts'  fmgr\_dvmdb\_script:  method: 'get'  params:  - fields:  - - 'name'  - 'type'  - 'desc'  sortings:  - name: -1  - type: 1  - content: 1   * Execute a script   - hosts: fortimanager01  connection: httpapi  vars:  adom: "root"  ansible\_httpapi\_use\_ssl: True  ansible\_httpapi\_validate\_certs: False  ansible\_httpapi\_port: 443  tasks:  - name: 'exec a script'  #Not Able To work: invalid data content  fmgr\_dvmdb\_script\_execute:  method: "exec"  params:  - workflow:  adom: 'root'  package: 'labpolicy'  script: 'add\_firewall\_policy0'  scope:  - name: 'FGVM04TM19006963'  vdom: 'root'   * Delete a script   - hosts: fortimanager01  connection: httpapi  vars:  adom: "root"  ansible\_httpapi\_use\_ssl: True  ansible\_httpapi\_validate\_certs: False  ansible\_httpapi\_port: 443  tasks:  - name: 'delete a script'  fmgr\_dvmdb\_script\_script:  method: "delete"  url\_params:  script: 'add\_firewall\_policy0'  **the playbooks list:**   * script\_add.yml * script\_delete.yml * script\_execute.yml * script\_get.yml * script\_set.yml   **requirements**:  the device was added to the manager to the FortiManager |
| Validation: | * ansible-playbook -i hosts/ script\_add.yml –vvv   "response": [  0,  {  "status": {  "code": 0,  "message": "OK"  },  "url": "/dvmdb/script"  }  ]   * ansible-playbook -i hosts/ script\_set.yml -vvv   response": [  0,  {  "status": {  "code": 0,  "message": "OK"  },  "url": "/dvmdb/script"  }  ]   * ansible-playbook -i hosts/ script\_get.yml -vvv   "response": [  0,  [  {  "content": "config firewall policy\n edit 1\n set name foopolicy\n next\nend\n",  "desc": "the demo script to set",  "filter\_build": -1,  "filter\_device": 0,  "filter\_hostname": null,  "filter\_ostype": "unknown",  "filter\_osver": "unknown",  "filter\_platform": "",  "filter\_serial": null,  "modification\_time": "2019-11-19 08:57:48",  "name": "add\_firewall\_policy0",  "oid": 136,  "script\_schedule": null,  "target": "device\_database",  "type": "cli"  }  ]  ]   * ansible-playbook -i hosts/ script\_execute.yml -vvv   "response": [  -10,  {  "status": {  "code": -10,  "message": "The data is invalid for selected url"  },  "url": "/dvmdb/script/execute"  }  ]   * ansible-playbook -i hosts/ script\_delete.yml -vvv   "response": [  0,  {  "status": {  "code": 0,  "message": "OK"  },  "url": "/dvmdb/script/add\_firewall\_policy0"  }  ]   * run it one more time:   ansible-playbook -i hosts/ script\_delete.yml -vvv  {  "status": {  "code": -3,  "message": "Object does not exist"  },  "url": "/dvmdb/script/add\_firewall\_policy0"  } |
| Completion Criteria: | Partial success, the operation to execute a script not successful. |
| Special Considerations: |  |

### Firewall IPv4 Address Object Test

|  |  |
| --- | --- |
| Test Objective: | Test The Basic operations with FortiManager Address Objects  Operations include: **Add**, **Update(Set)**, **Get** and **Delete** |
| Technique: | Playbooks for each operation:   * Add a firewall ipv4 address object   - hosts: fortimanager01  connection: httpapi  vars:  adom: "root"  ansible\_httpapi\_use\_ssl: True  ansible\_httpapi\_validate\_certs: False  ansible\_httpapi\_port: 443  tasks:  - name: 'add an ipv4 address'  fmgr\_pm\_config\_adom\_adom\_obj\_firewall\_address:  method: 'add'  url\_params:  adom: 'root'  params:  - data:  - name: 'firewall\_ipv4\_address\_object0'  comment: 'An ipv4 address object created by Ansible'  type: 'ipmask'  subnet: '11.22.33.44/24'   * Update a firewall ipv4 address object   - hosts: fortimanager01  connection: httpapi  vars:  adom: "root"  ansible\_httpapi\_use\_ssl: True  ansible\_httpapi\_validate\_certs: False  ansible\_httpapi\_port: 443  tasks:  - name: 'update an ipv4 address object'  fmgr\_pm\_config\_adom\_adom\_obj\_firewall\_address:  method: 'update'  url\_params:  adom: 'root'  params:  - data:  - name: 'firewall\_ipv4\_address\_object0'  comment: 'An ipv4 address object updated by Ansible'  type: 'ipmask'  subnet: '11.22.33.45/24'   * Get the address object using filter   - hosts: fortimanager01  connection: httpapi  vars:  adom: "root"  ansible\_httpapi\_use\_ssl: True  ansible\_httpapi\_validate\_certs: False  ansible\_httpapi\_port: 443  tasks:  - name: 'get an ipv4 address'  fmgr\_pm\_config\_adom\_adom\_obj\_firewall\_address:  method: 'get'  url\_params:  adom: 'root'  params:  - filter:  - 'name'  - '=='  - 'firewall\_ipv4\_address\_object0'   * Delete a firewall ipv4 address object   - hosts: fortimanager01  connection: httpapi  vars:  adom: "root"  ansible\_httpapi\_use\_ssl: True  ansible\_httpapi\_validate\_certs: False  ansible\_httpapi\_port: 443  tasks:  - name: 'delete an ipv4 address'  fmgr\_pm\_config\_adom\_adom\_obj\_firewall\_address\_address:  method: 'delete'  url\_params:  adom: 'root'  address: 'firewall\_ipv4\_address\_object0'  **the playbooks list:**   * firewall\_address\_add.yml * firewall\_address\_delete.yml * firewall\_address\_get.yml * firewall\_address\_update.yml |
| Validation: | * ansible-playbook -i hosts/ firewall\_address\_add.yml –vvv   "response": [  0,  {  "status": {  "code": 0,  "message": "OK"  },  "url": "/pm/config/adom/root/obj/firewall/address"  }  ]   * ansible-playbook -i hosts/ firewall\_address\_update.yml –vvv   "ansible\_module\_results": {  "status": {  "code": 0,  "message": "OK"  },  "url": "/pm/config/adom/root/obj/firewall/address"  }   * ansible-playbook -i hosts/ firewall\_address\_get.yml –vvv   "results": [  {  "allow-routing": "disable",  "associated-interface": [  "any"  ],  "color": 0,  "comment": "An ipv4 address object updated by Ansible",  "dynamic\_mapping": null,  "list": null,  "name": "firewall\_ipv4\_address\_object0",  "subnet": [  "11.22.33.0",  "255.255.255.0"  ],  "tagging": null,  "type": "ipmask",  "uuid": "3b305a2c-0af9-51ea-810d-6d563981915b",  "visibility": "enable"  }  ]   * ansible-playbook -i hosts/ firewall\_address\_delete.yml –vvv   "response": [  0,  {  "status": {  "code": 0,  "message": "OK"  },  "url": "/pm/config/adom/root/obj/firewall/address/firewall\_ipv4\_address\_object0"  }  ] |
| Completion Criteria: | success. |
| Special Considerations: |  |

### Firewall IPv6 Address Object Test

|  |  |
| --- | --- |
| Test Objective: | Test The Basic operations with FortiManager IPv6 Address Objects  Operations include: **Add**, **Update(Set)**, **Get** and **Delete** |
| Technique: | Playbooks for each operation:   * Add a firewall ipv6 address object   - hosts: fortimanager01  connection: httpapi  vars:  adom: "root"  ansible\_httpapi\_use\_ssl: True  ansible\_httpapi\_validate\_certs: False  ansible\_httpapi\_port: 443  tasks:  - name: 'add an ipv6 address object'  fmgr\_pm\_config\_adom\_adom\_obj\_firewall\_address6:  method: 'add'  url\_params:  adom: 'root'  params:  - data:  - name: 'firewall\_ipv6\_address\_object0'  comment: 'An ipv6 address object created by Ansible'  type: 'ipprefix'  ip6: '2001:0db8:85a3:0000:0000:8a2e:0370:7334'   * Update a firewall ipv6 address object   - hosts: fortimanager01  connection: httpapi  vars:  adom: "root"  ansible\_httpapi\_use\_ssl: True  ansible\_httpapi\_validate\_certs: False  ansible\_httpapi\_port: 443  tasks:  - name: 'add an ipv6 address object'  fmgr\_pm\_config\_adom\_adom\_obj\_firewall\_address6:  method: 'update'  url\_params:  adom: 'root'  params:  - data:  - name: 'firewall\_ipv6\_address\_object0'  comment: 'An ipv6 address object updated by Ansible'  type: 'ipprefix'  ip6: '2001:0db8:85a3:0000:0000:8a2e:0370:7335'   * Get the ipv6 address object using filter   - hosts: fortimanager01  connection: httpapi  vars:  adom: "root"  ansible\_httpapi\_use\_ssl: True  ansible\_httpapi\_validate\_certs: False  ansible\_httpapi\_port: 443  tasks:  - name: 'get an ipv6 address object'  fmgr\_pm\_config\_adom\_adom\_obj\_firewall\_address6:  method: 'get'  url\_params:  adom: 'root'  params:  - filter:  - 'name'  - '=='  - 'firewall\_ipv6\_address\_object0'   * Delete a firewall ipv6 address object   - hosts: fortimanager01  connection: httpapi  vars:  adom: "root"  ansible\_httpapi\_use\_ssl: True  ansible\_httpapi\_validate\_certs: False  ansible\_httpapi\_port: 443  tasks:  - name: 'delete an ipv6 address object'  fmgr\_pm\_config\_adom\_adom\_obj\_firewall\_address6\_address6:  method: 'delete'  url\_params:  adom: 'root'  address6: 'firewall\_ipv6\_address\_object0'  **the playbooks list:**   * firewall\_address\_ipv6\_add.yml * firewall\_address\_ipv6\_delete.yml * firewall\_address\_ipv6\_get.yml * firewall\_address\_ipv6\_update.yml |
| Validation: | * ansible-playbook -i hosts/ firewall\_address\_ipv6\_add.yml –vvv   "response": [  0,  {  "status": {  "code": 0,  "message": "OK"  },  "url": "/pm/config/adom/root/obj/firewall/address6"  }  ]   * ansible-playbook -i hosts/ firewall\_address\_ipv6\_update.yml –vvv   "response": [  0,  {  "status": {  "code": 0,  "message": "OK"  },  "url": "/pm/config/adom/root/obj/firewall/address6"  }  ]   * ansible-playbook -i hosts/ firewall\_address\_ipv6\_get.yml –vvv   "response": [  0,  [  {  "cache-ttl": 0,  "color": 0,  "comment": "An ipv6 address object updated by Ansible",  "dynamic\_mapping": null,  "end-ip": "::",  "host": "::",  "host-type": "any",  "ip6": "2001:db8:85a3::8a2e:370:7335/128",  "list": null,  "name": "firewall\_ipv6\_address\_object0",  "start-ip": "::",  "subnet-segment": null,  "tagging": null,  "template": [],  "type": "ipprefix",  "uuid": "061e0e0a-0ba4-51ea-7e63-16a3ecef090e",  "visibility": "enable"  }  ]  ]   * ansible-playbook -i hosts/ firewall\_address\_ipv6\_delete.yml –vvv   "response": [  0,  {  "status": {  "code": 0,  "message": "OK"  },  "url": "/pm/config/adom/root/obj/firewall/address6/firewall\_ipv6\_address\_object0"  }  ]   * run it one more time to make sure it indeed is deleted.   ansible-playbook -i hosts/ firewall\_address\_ipv6\_delete.yml –vvv  "status": {  "code": -3,  "message": "Object does not exist"  } |
| Completion Criteria: | success. |
| Special Considerations: |  |

### Firewall IPv4 Ippool Object Test

|  |  |
| --- | --- |
| Test Objective: | Test The Basic operations with FortiManager IPv4 Ippool Objects  Operations include: **Add**, **Update(Set)**, **Get** and **Delete** |
| Technique: | Playbooks for each operation:   * Add a firewall ipv4 ippool object   - hosts: fortimanager01  connection: httpapi  vars:  adom: "root"  ansible\_httpapi\_use\_ssl: True  ansible\_httpapi\_validate\_certs: False  ansible\_httpapi\_port: 443  tasks:  - name: 'add an ipv4 ippool'  fmgr\_pm\_config\_adom\_adom\_obj\_firewall\_ippool:  method: 'add'  url\_params:  adom: 'root'  params:  - data:  - name: 'firewall\_ipv4\_ippool\_object0'  comments: 'An ipv4 ippool object created by Ansible'  type: 'one-to-one'  startip: '192.168.190.100'  endip: '192.168.190.250'  arp-reply: 'enable'   * Update a firewall ipv4 ippool object   - hosts: fortimanager01  connection: httpapi  vars:  adom: "root"  ansible\_httpapi\_use\_ssl: True  ansible\_httpapi\_validate\_certs: False  ansible\_httpapi\_port: 443  tasks:  - name: 'update an ipv4 ippool'  fmgr\_pm\_config\_adom\_adom\_obj\_firewall\_ippool:  method: 'update'  url\_params:  adom: 'root'  params:  - data:  - name: 'firewall\_ipv4\_ippool\_object0'  comments: 'An ipv4 ippool object updated by Ansible'  type: 'one-to-one'  startip: '192.168.190.110'  endip: '192.168.190.150'  arp-reply: 'disable'   * Get the ipv4 ippool object using filter   - hosts: fortimanager01  connection: httpapi  vars:  adom: "root"  ansible\_httpapi\_use\_ssl: True  ansible\_httpapi\_validate\_certs: False  ansible\_httpapi\_port: 443  tasks:  - name: 'get an ipv4 ippool'  fmgr\_pm\_config\_adom\_adom\_obj\_firewall\_ippool:  method: 'get'  url\_params:  adom: 'root'  params:  - filter:  - 'name'  - '=='  - 'firewall\_ipv4\_ippool\_object0'   * Delete a firewall ipv4 ippool object   - hosts: fortimanager01  connection: httpapi  vars:  adom: "root"  ansible\_httpapi\_use\_ssl: True  ansible\_httpapi\_validate\_certs: False  ansible\_httpapi\_port: 443  tasks:  - name: 'delete an ipv4 ippool'  fmgr\_pm\_config\_adom\_adom\_obj\_firewall\_ippool\_ippool:  method: 'delete'  url\_params:  adom: 'root'  ippool: 'firewall\_ipv4\_ippool\_object0'  bject0'  **the playbooks list:**   * firewall\_ippool\_add.yml * firewall\_ippool\_delete.yml * firewall\_ippool\_get.yml * firewall\_ippool\_update.yml |
| Validation: | * ansible-playbook -i hosts/ firewall\_ippool\_add.yml –vvv   "response": [  0,  {  "status": {  "code": 0,  "message": "OK"  },  "url": "/pm/config/adom/root/obj/firewall/ippool"  }  ]   * ansible-playbook -i hosts/ firewall\_ippool\_update.yml –vvv   "response": [  0,  {  "status": {  "code": 0,  "message": "OK"  },  "url": "/pm/config/adom/root/obj/firewall/ippool"  }  ]   * ansible-playbook -i hosts/ firewall\_ippool\_get.yml –vvv   "response": [  0,  [  {  "arp-intf": [],  "arp-reply": "disable",  "associated-interface": [],  "comments": "An ipv4 ippool object updated by Ansible",  "dynamic\_mapping": null,  "endip": "192.168.190.150",  "name": "firewall\_ipv4\_ippool\_object0",  "startip": "192.168.190.110",  "type": "one-to-one"  }  ]  ]   * ansible-playbook -i hosts/ firewall\_ippool\_delete.yml –vvv   "response": [  0,  {  "status": {  "code": 0,  "message": "OK"  },  "url": "/pm/config/adom/root/obj/firewall/ippool/firewall\_ipv4\_ippool\_object0"  }  ]   * run it one more time to make sure it indeed is deleted.   ansible-playbook -i hosts/ firewall\_ippool\_delete.yml –vvv  "status": {  "code": -3,  "message": "Object does not exist"  } |
| Completion Criteria: | success. |
| Special Considerations: |  |

### Firewall IPv6 Ippool Object Test

|  |  |
| --- | --- |
| Test Objective: | Test The Basic operations with FortiManager IPv6 Ippool Objects  Operations include: **Add**, **Update(Set)**, **Get** and **Delete** |
| Technique: | Playbooks for each operation:   * Add a firewall ipv6 ippool object   - hosts: fortimanager01  connection: httpapi  vars:  adom: "root"  ansible\_httpapi\_use\_ssl: True  ansible\_httpapi\_validate\_certs: False  ansible\_httpapi\_port: 443  tasks:  - name: 'add an ipv6 ippool'  fmgr\_pm\_config\_adom\_adom\_obj\_firewall\_ippool6:  method: 'add'  url\_params:  adom: 'root'  params:  - data:  - name: 'firewall\_ipv6\_ippool\_object0'  comments: 'An ipv6 ippool object created by Ansible'  startip: '2001:0db8:85a3:0000:0000:8a2e:0370:1000'  endip: '2001:0db8:85a3:0000:0000:8a2e:0370:3333'   * Update a firewall ipv6 ippool object   - hosts: fortimanager01  connection: httpapi  vars:  adom: "root"  ansible\_httpapi\_use\_ssl: True  ansible\_httpapi\_validate\_certs: False  ansible\_httpapi\_port: 443  tasks:  - name: 'update an ipv6 ippool'  fmgr\_pm\_config\_adom\_adom\_obj\_firewall\_ippool6:  method: 'update'  url\_params:  adom: 'root'  params:  - data:  - name: 'firewall\_ipv6\_ippool\_object0'  comments: 'An ipv6 ippool object updated by Ansible'  startip: '2001:0db8:85a3:0000:0000:8a2e:0370:2222'  endip: '2001:0db8:85a3:0000:0000:8a2e:0370:3333'   * Get the ipv6 ippool object using filter   - hosts: fortimanager01  connection: httpapi  vars:  adom: "root"  ansible\_httpapi\_use\_ssl: True  ansible\_httpapi\_validate\_certs: False  ansible\_httpapi\_port: 443  tasks:  - name: 'get an ipv6 ippool'  fmgr\_pm\_config\_adom\_adom\_obj\_firewall\_ippool6:  method: 'get'  url\_params:  adom: 'root'  params:  - filter:  - 'name'  - '=='  - 'firewall\_ipv6\_ippool\_object0'   * Delete a firewall ipv6 ippool object   - hosts: fortimanager01  connection: httpapi  vars:  adom: "root"  ansible\_httpapi\_use\_ssl: True  ansible\_httpapi\_validate\_certs: False  ansible\_httpapi\_port: 443  tasks:  - name: 'delete an ipv6 ippool'  fmgr\_pm\_config\_adom\_adom\_obj\_firewall\_ippool6\_ippool6:  method: 'delete'  url\_params:  adom: 'root'  ippool6: 'firewall\_ipv6\_ippool\_object0'  **the playbooks list:**   * firewall\_ippool\_ipv6\_add.yml * firewall\_ippool\_ipv6\_delete.yml * firewall\_ippool\_ipv6\_get.yml * firewall\_ippool\_ipv6\_update.yml |
| Validation: | * ansible-playbook -i hosts/ firewall\_ippool\_ipv6\_add.yml –vvv   "response": [  0,  {  "status": {  "code": 0,  "message": "OK"  },  "url": "/pm/config/adom/root/obj/firewall/ippool6"  }  ]   * ansible-playbook -i hosts/ firewall\_ippool\_ipv6\_update.yml –vvv   "response": [  0,  {  "status": {  "code": 0,  "message": "OK"  },  "url": "/pm/config/adom/root/obj/firewall/ippool6"  }  ]   * ansible-playbook -i hosts/ firewall\_ippool\_ipv6\_get.yml –vvv   "response": [  0,  [  {  "comments": "An ipv6 ippool object updated by Ansible",  "dynamic\_mapping": null,  "endip": "2001:db8:85a3::8a2e:370:3333",  "name": "firewall\_ipv6\_ippool\_object0",  "startip": "2001:db8:85a3::8a2e:370:2222"  }  ]  ]   * ansible-playbook -i hosts/ firewall\_ippool\_ipv6\_delete.yml –vvv   "response": [  0,  {  "status": {  "code": 0,  "message": "OK"  },  "url": "/pm/config/adom/root/obj/firewall/ippool6/firewall\_ipv6\_ippool\_object0"  }  ]   * run it one more time to make sure it indeed is deleted.   ansible-playbook -i hosts/ firewall\_ippool\_ipv6\_delete.yml –vvv  "status": {  "code": -3,  "message": "Object does not exist"  } |
| Completion Criteria: | success. |
| Special Considerations: |  |

### Firewall IPv4 VIP Object Test

|  |  |
| --- | --- |
| Test Objective: | Test The Basic operations with FortiManager Firewall VIP Objects  Operations include: **Add**, **Update(Set)**, **Get** and **Delete** |
| Technique: | Playbooks for each operation:   * Add a firewall vip object   - hosts: fortimanager01  connection: httpapi  vars:  adom: "root"  ansible\_httpapi\_use\_ssl: True  ansible\_httpapi\_validate\_certs: False  ansible\_httpapi\_port: 443  tasks:  - name: 'add firewall vip object'  fmgr\_pm\_config\_adom\_adom\_obj\_firewall\_vip:  method: 'add'  url\_params:  adom: 'root'  params:  - data:  - name: 'firewall\_vip\_object0'  comment: 'A firewall vip object created by Ansible'  arp-reply: 'enable'   * Update a firewall vip object   - hosts: fortimanager01  connection: httpapi  vars:  adom: "root"  ansible\_httpapi\_use\_ssl: True  ansible\_httpapi\_validate\_certs: False  ansible\_httpapi\_port: 443  tasks:  - name: 'update firewall vip object'  fmgr\_pm\_config\_adom\_adom\_obj\_firewall\_vip:  method: 'update'  url\_params:  adom: 'root'  params:  - data:  - name: 'firewall\_vip\_object0'  comment: 'A firewall vip object updated by Ansible'  arp-reply: 'disable'  type: 'static-nat'  color: 2  weblogic-server: 'disable'  extintf: 'port1'   * Get the firewall vip object using filter   - hosts: fortimanager01  connection: httpapi  vars:  adom: "root"  ansible\_httpapi\_use\_ssl: True  ansible\_httpapi\_validate\_certs: False  ansible\_httpapi\_port: 443  tasks:  - name: 'get firewall vip object'  fmgr\_pm\_config\_adom\_adom\_obj\_firewall\_vip:  method: 'get'  url\_params:  adom: 'root'  params:  - filter:  - 'name'  - '=='  - 'firewall\_vip\_object0'   * Delete a firewall vip object   - hosts: fortimanager01  connection: httpapi  vars:  adom: "root"  ansible\_httpapi\_use\_ssl: True  ansible\_httpapi\_validate\_certs: False  ansible\_httpapi\_port: 443  tasks:  - name: 'delete a firewall vip object'  fmgr\_pm\_config\_adom\_adom\_obj\_firewall\_vip\_vip:  method: 'delete'  url\_params:  adom: 'root'  vip: 'firewall\_vip\_object0'  **the playbooks list:**   * firewall\_vip\_add.yml * firewall\_vip\_delete.yml * firewall\_vip\_get.yml * firewall\_vip\_update.yml |
| Validation: | * ansible-playbook -i hosts/ firewall\_vip\_add.yml –vvv   "response": [  0,  {  "status": {  "code": 0,  "message": "OK"  },  "url": "/pm/config/adom/root/obj/firewall/vip"  }  ]   * ansible-playbook -i hosts/ firewall\_vip\_update.yml –vvv   "response": [  0,  {  "status": {  "code": 0,  "message": "OK"  },  "url": "/pm/config/adom/root/obj/firewall/vip"  }  ]   * ansible-playbook -i hosts/ firewall\_vip\_get.yml –vvv   "response": [  0,  [  {  "\_if\_no\_default": 0,  "arp-reply": "disable",  "color": 2,  "comment": "A firewall vip object updated by Ansible",  "dns-mapping-ttl": 0,  "dynamic\_mapping": null,  "extaddr": [],  "extintf": [  "port1"  ],  "extip": [  "0.0.0.0"  ],  "gratuitous-arp-interval": 0,  "id": 0,  "mappedip": [],  "name": "firewall\_vip\_object0",  "nat-source-vip": "disable",  "obj seq": 1,  "portforward": "disable",  "realservers": null,  "service": [],  "src-filter": [],  "srcintf-filter": [],  "ssl-cipher-suites": null,  "ssl-client-fallback": "enable",  "ssl-server-algorithm": "client",  "ssl-server-cipher-suites": null,  "ssl-server-max-version": "client",  "ssl-server-min-version": "client",  "type": "static-nat",  "uuid": "8580930c-0c49-51ea-3a21-376c9eff5f05"  }  ]  ]   * ansible-playbook -i hosts/ firewall\_vip\_delete.yml –vvv   "response": [  0,  {  "status": {  "code": 0,  "message": "OK"  },  "url": "/pm/config/adom/root/obj/firewall/vip/firewall\_vip\_object0"  }  ]   * run it one more time to make sure it indeed is deleted.   ansible-playbook -i hosts/ firewall\_vip\_delete.yml –vvv  "status": {  "code": -3,  "message": "Object does not exist"  } |
| Completion Criteria: | success. |
| Special Considerations: |  |

### Firewall IPv6 VIP Object Test

|  |  |
| --- | --- |
| Test Objective: | Test The Basic operations with FortiManager Firewall IPv6 VIP Objects  Operations include: **Add**, **Update(Set)**, **Get** and **Delete** |
| Technique: | Playbooks for each operation:   * Add a firewall IPv6 vip object   - hosts: fortimanager01  connection: httpapi  vars:  adom: "root"  ansible\_httpapi\_use\_ssl: True  ansible\_httpapi\_validate\_certs: False  ansible\_httpapi\_port: 443  tasks:  - name: 'add firewall IPv6 vip object'  fmgr\_pm\_config\_adom\_adom\_obj\_firewall\_vip6:  method: 'add'  url\_params:  adom: 'root'  params:  - data:  - name: 'firewall\_vip\_ipv6\_object0'  comment: 'A firewall IPv6 vip object created by Ansible'  arp-reply: 'enable'   * Update a firewall IPv6 vip object   - hosts: fortimanager01  connection: httpapi  vars:  adom: "root"  ansible\_httpapi\_use\_ssl: True  ansible\_httpapi\_validate\_certs: False  ansible\_httpapi\_port: 443  tasks:  - name: 'update firewall IPv6 vip object'  fmgr\_pm\_config\_adom\_adom\_obj\_firewall\_vip6:  method: 'update'  url\_params:  adom: 'root'  params:  - data:  - name: 'firewall\_vip\_ipv6\_object0'  comment: 'A firewall IPv6 vip object updated by Ansible'   * Get the firewall IPv6 vip object using filter   - hosts: fortimanager01  connection: httpapi  vars:  adom: "root"  ansible\_httpapi\_use\_ssl: True  ansible\_httpapi\_validate\_certs: False  ansible\_httpapi\_port: 443  tasks:  - name: 'get firewall IPv6 vip object'  fmgr\_pm\_config\_adom\_adom\_obj\_firewall\_vip6:  method: 'get'  url\_params:  adom: 'root'  params:  - filter:  - 'name'  - '=='  - 'firewall\_vip\_ipv6\_object0'   * Delete a firewall vip object   - hosts: fortimanager01  connection: httpapi  vars:  adom: "root"  ansible\_httpapi\_use\_ssl: True  ansible\_httpapi\_validate\_certs: False  ansible\_httpapi\_port: 443  tasks:  - name: 'delete firewall IPv6 vip object'  fmgr\_pm\_config\_adom\_adom\_obj\_firewall\_vip6\_vip6:  method: 'delete'  url\_params:  adom: 'root'  vip6: 'firewall\_vip\_ipv6\_object0'  **the playbooks list:**   * firewall\_vip\_ipv6\_add.yml * firewall\_vip\_ipv6\_delete.yml * firewall\_vip\_ipv6\_get.yml * firewall\_vip\_ipv6\_update.yml |
| Validation: | * ansible-playbook -i hosts/ firewall\_vip\_ipv6\_add.yml –vvv   "response": [  0,  {  "status": {  "code": 0,  "message": "OK"  },  "url": "/pm/config/adom/root/obj/firewall/vip6"  }  ]   * ansible-playbook -i hosts/ firewall\_vip\_ipv6\_update.yml –vvv   "response": [  0,  {  "status": {  "code": 0,  "message": "OK"  },  "url": "/pm/config/adom/root/obj/firewall/vip6"  }  ]   * ansible-playbook -i hosts/ firewall\_vip\_ipv6\_get.yml –vvv   "response": [  0,  [  {  "\_if\_no\_default": 0,  "arp-reply": "enable",  "color": 0,  "comment": "A firewall IPv6 vip object updated by Ansible",  "dynamic\_mapping": null,  "extip": [  "::"  ],  "extport": "0",  "http-cookie-age": 60,  "http-cookie-domain-from-host": "disable",  "http-cookie-generation": 0,  "http-cookie-share": "same-ip",  "http-ip-header": "disable",  "http-multiplex": "disable",  "https-cookie-secure": "disable",  "id": 0,  "ldb-method": "static",  "mappedip": [  "::"  ],  "mappedport": "0",  "max-embryonic-connections": 1000,  "name": "firewall\_vip\_ipv6\_object0",  "outlook-web-access": "disable",  "persistence": "none",  "portforward": "disable",  "protocol": "tcp",  "realservers": null,  "src-filter": [],  "ssl-cipher-suites": null,  "ssl-client-fallback": "enable",  "ssl-hsts": "disable",  "ssl-hsts-age": 5184000,  "ssl-hsts-include-subdomains": "disable",  "ssl-http-location-conversion": "disable",  "ssl-server-algorithm": "client",  "ssl-server-cipher-suites": null,  "ssl-server-max-version": "client",  "ssl-server-min-version": "client",  "ssl-server-session-state-type": "both",  "type": "static-nat",  "uuid": "3fc2aeb0-0c51-51ea-5207-b67a3d8e7274",  "weblogic-server": "disable",  "websphere-server": "disable"  }  ]  ]   * ansible-playbook -i hosts/ firewall\_vip\_ipv6\_delete.yml –vvv   "response": [  0,  {  "status": {  "code": 0,  "message": "OK"  },  "url": "/pm/config/adom/root/obj/firewall/vip6/firewall\_vip\_ipv6\_object0"  }  ]   * run it one more time to make sure it indeed is deleted.   ansible-playbook -i hosts/ firewall\_vip\_ipv6\_delete.yml –vvv  "status": {  "code": -3,  "message": "Object does not exist"  } |
| Completion Criteria: | success. |
| Special Considerations: |  |

### Firewall Policy Package Test

|  |  |
| --- | --- |
| Test Objective: | Test The Basic operations with FortiManager Policy Packages  Operations include: **Add**, **Get** and **Delete** |
| Technique: | Playbooks for each operation:   * Add a firewall package   - hosts: fortimanager01  connection: httpapi  vars:  adom: "root"  ansible\_httpapi\_use\_ssl: True  ansible\_httpapi\_validate\_certs: False  ansible\_httpapi\_port: 443  tasks:  - name: 'add a firewall package'  fmgr\_pm\_pkg\_adom\_adom:  method: 'add'  url\_params:  adom: 'root'  params:  - data:  - name: 'testpackage0'  'obj ver': 1  type: 'pkg'   * Get all the firewall packages (no filters can be applied)   - hosts: fortimanager01  connection: httpapi  vars:  adom: "root"  ansible\_httpapi\_use\_ssl: True  ansible\_httpapi\_validate\_certs: False  ansible\_httpapi\_port: 443  tasks:  - name: 'get a firewall package'  fmgr\_pm\_pkg\_adom\_adom:  method: 'get'  url\_params:  adom: 'root'   * Delete a firewall package   - hosts: fortimanager01  connection: httpapi  vars:  adom: "root"  ansible\_httpapi\_use\_ssl: True  ansible\_httpapi\_validate\_certs: False  ansible\_httpapi\_port: 443  tasks:  - name: 'delete a firewall package'  fmgr\_pm\_pkg\_adom\_adom\_pkg\_path:  method: 'delete'  url\_params:  adom: 'root'  pkg\_path: 'testpackage0'  **the playbooks list:**   * firewall\_package\_add.yml * firewall\_package\_delete.yml * firewall\_package\_get.yml |
| Validation: | * ansible-playbook -i hosts/ firewall\_package\_add.yml –vvv   "response": [  0,  {  "status": {  "code": 0,  "message": "OK"  },  "url": "/pm/pkg/adom/root"  }  ]   * ansible-playbook -i hosts/ firewall\_package\_get.yml –vvv   One item in response:  {  "name": "testpackage0",  "obj ver": 4,  "oid": 1342,  "package settings": {  "central-nat": "disable",  "fwpolicy-implicit-log": "disable",  "fwpolicy6-implicit-log": "disable",  "inspection-mode": "enable",  "ngfw-mode": "disable"  },  "type": "pkg"  }   * ansible-playbook -i hosts/ firewall\_package\_delete.yml –vvv   "response": [  0,  {  "status": {  "code": 0,  "message": "OK"  },  "url": "/pm/pkg/adom/root/testpackage0"  }  ]   * run it one more time to make sure it indeed is deleted.   ansible-playbook -i hosts/ firewall\_package\_delete.yml –vvv  "status": {  "code": -3,  "message": "Object does not exist"  } |
| Completion Criteria: | success. |
| Special Considerations: |  |

### Firewall IPv4 Policy Test

|  |  |
| --- | --- |
| Test Objective: | Test The Basic operations with FortiManager IPv4 Policy  Operations include: **Add**,**Update**(**Set**), **Get** and **Delete** |
| Technique: | Playbooks for each operation:   * Add a firewall IPv4 policy   - hosts: fortimanager01  connection: httpapi  vars:  adom: "root"  package: 'testpackage0'  ansible\_httpapi\_use\_ssl: True  ansible\_httpapi\_validate\_certs: False  ansible\_httpapi\_port: 443  tasks:  - name: 'prerequisite: add a firewall package'  fmgr\_pm\_pkg\_adom\_adom:  method: 'add'  url\_params:  adom: "{{ adom }}"  params:  - data:  - name: "{{ package }}"  'obj ver': 1  type: 'pkg'  - name: 'Add an IPv4 firewall policy'  fmgr\_pm\_config\_adom\_adom\_pkg\_pkg\_firewall\_policy:  method: 'add'  url\_params:  adom: "{{ adom }}"  pkg: "{{ package }}"  params:  - data:  - name: 'firewall\_ipv4\_policy0'  srcintf: 'port1'  srcaddr: 'all'  dstintf: 'port1'  dstaddr: 'all'  action: 'accept'  service: 'ALL'  schedule: 'always'  comments: 'An IPv4 policy created by ansible'   * Update a firewall IPv4 policy   - hosts: fortimanager01  connection: httpapi  vars:  adom: "root"  package: 'testpackage0'  ansible\_httpapi\_use\_ssl: True  ansible\_httpapi\_validate\_certs: False  ansible\_httpapi\_port: 443  tasks:  - name: 'Update an IPv4 firewall policy'  fmgr\_pm\_config\_adom\_adom\_pkg\_pkg\_firewall\_policy:  method: 'update'  url\_params:  adom: "{{ adom }}"  pkg: "{{ package }}"  params:  - data:  # Identifying a policy is via policy\_id, not name  # To get the policy of the policy we just created just now,  # invoke firewall\_policy\_get.yml  - policyid: < PolicyID >  name: 'firewall\_ipv4\_policy0'  srcintf: 'port1'  srcaddr: 'all'  dstintf: 'port1'  dstaddr: 'all'  action: 'accept'  service: 'ALL'  schedule: 'always'  comments: 'An IPv4 policy updated by ansible'   * Get a firewall IPv4 policy using filter   - hosts: fortimanager01  connection: httpapi  vars:  adom: "root"  package: 'testpackage0'  ansible\_httpapi\_use\_ssl: True  ansible\_httpapi\_validate\_certs: False  ansible\_httpapi\_port: 443  tasks:  - name: 'Get an IPv4 firewall policy'  fmgr\_pm\_config\_adom\_adom\_pkg\_pkg\_firewall\_policy:  method: 'get'  url\_params:  adom: "{{ adom }}"  pkg: "{{ package }}"  params:  - filter:  - 'name'  - '=='  - 'firewall\_ipv4\_policy0'   * Delete a firewall package   - hosts: fortimanager01  connection: httpapi  vars:  adom: "root"  package: 'testpackage0'  ansible\_httpapi\_use\_ssl: True  ansible\_httpapi\_validate\_certs: False  ansible\_httpapi\_port: 443  tasks:  - name: 'Delete an IPv4 firewall policy'  fmgr\_pm\_config\_adom\_adom\_pkg\_pkg\_firewall\_policy\_policy:  method: 'delete'  url\_params:  adom: "{{ adom }}"  pkg: "{{ package }}"  # looks like the policy is actually the policy ID.  policy: <PolicyID>  - name: 'purge the package container'  fmgr\_pm\_pkg\_adom\_adom\_pkg\_path:  method: 'delete'  url\_params:  adom: '{{ adom }}'  pkg\_path: '{{ package }}'  **the playbooks list:**   * firewall\_policy\_add.yml * firewall\_policy\_delete.yml * firewall\_policy\_get.yml * firewall\_policy\_update.yml   **Prerequisites**:  \* for all the policy API, a package name is required as a parameter in url. |
| Validation: | * ansible-playbook -i hosts/ firewall\_policy\_add.yml –vvv   the response returns the policyID which we created.  "response": [  0,  {  "policyid": 5  }  ]   * ansible-playbook -i hosts/ firewall\_policy\_update.yml–vvv   "response": [  0,  {  "policyid": 5  }  ]   * ansible-playbook -i hosts/ firewall\_policy\_get.yml –vvv   "response": [  0,  [  {  "\_byte": 0,  "\_first\_hit": 0,  "\_first\_session": 0,  "\_global-vpn": [],  "\_hitcount": 0,  "\_last\_hit": 0,  "\_last\_session": 0,  "\_pkts": 0,  "\_sesscount": 0,  "action": "accept",  "app-group": [],  "auto-asic-offload": "enable",  "block-notification": "disable",  "captive-portal-exempt": "disable",  "capture-packet": "disable",  "comments": "An IPv4 policy updated by ansible",  "custom-log-fields": [],  "delay-tcp-npu-session": "disable",  "devices": [],  "diffserv-forward": "disable",  "diffserv-reverse": "disable",  "disclaimer": "disable",  "dnsfilter-profile": [],  "dscp-match": "disable",  "dscp-negate": "disable",  "dscp-value": "000000",  "dsri": "disable",  "dstaddr": [  "all"  ],  "dstaddr-negate": "disable",  "dstintf": [  "port1"  ],  "fsso-agent-for-ntlm": [],  "groups": [],  "internet-service": "disable",  "internet-service-src": "disable",  "internet-service-src-custom": [],  "internet-service-src-id": [],  "internet-service-src-negate": "disable",  "learning-mode": "disable",  "logtraffic": "utm",  "logtraffic-start": "disable",  "match-vip": "disable",  "name": "firewall\_ipv4\_policy0",  "nat": "disable",  "natip": [  "0.0.0.0",  "0.0.0.0"  ],  "np-acceleration": "enable",  "obj seq": 1,  "per-ip-shaper": [],  "permit-any-host": "disable",  "policyid": 5,  "profile-protocol-options": [  "default"  ],  "profile-type": "single",  "radius-mac-auth-bypass": "disable",  "replacemsg-override-group": [],  "rtp-nat": "disable",  "scan-botnet-connections": "disable",  "schedule": [  "always"  ],  "schedule-timeout": "disable",  "service": [  "ALL"  ],  "service-negate": "disable",  "session-ttl": 0,  "srcaddr": [  "all"  ],  "srcaddr-negate": "disable",  "srcintf": [  "port1"  ],  "ssl-mirror": "disable",  "ssl-mirror-intf": [],  "status": "enable",  "tcp-mss-receiver": 0,  "tcp-mss-sender": 0,  "tcp-session-without-syn": "disable",  "timeout-send-rst": "disable",  "traffic-shaper": [],  "traffic-shaper-reverse": [],  "users": [],  "utm-status": "disable",  "uuid": "4178e834-0c81-51ea-2b58-bc217e0a3aab",  "vlan-cos-fwd": 255,  "vlan-cos-rev": 255,  "vpn\_dst\_node": null,  "vpn\_src\_node": null,  "waf-profile": [],  "wanopt": "disable",  "wccp": "disable",  "webcache": "disable",  "webcache-https": "disable"  }  ]  ]   * ansible-playbook -i hosts/ firewall\_policy\_delete.yml –vvv   "response": [  0,  {  "status": {  "code": 0,  "message": "OK"  },  "url": "/pm/config/adom/root/pkg/testpackage0/firewall/policy/7"  }  ] |
| Completion Criteria: | success. The **Update** and **Get** operations require the **policy ID** which is generated as an integer when the policy is created. |
| Special Considerations: |  |

### Firewall IPv6 Policy Test

The IPv6 Ansible module: **fmgr\_pm\_config\_adom\_adom\_pkg\_pkg\_firewall\_policy6** and f**mgr\_pm\_config\_adom\_adom\_pkg\_pkg\_firewall\_policy6\_policy6**.

The IPv6 policies don’t display on the UI.

### Firewall Service Test

|  |  |
| --- | --- |
| Test Objective: | Test The Basic operations with FortiManager Firewall Service  Operations include: **Add**, **Update**(**Set**), **Get** and **Delete** |
| Technique: | Playbooks for each operation:   * Add a firewall service   - hosts: fortimanager01  connection: httpapi  vars:  adom: "root"  ansible\_httpapi\_use\_ssl: True  ansible\_httpapi\_validate\_certs: False  ansible\_httpapi\_port: 443  tasks:  - name: 'Add a firewall service'  fmgr\_pm\_config\_adom\_adom\_obj\_firewall\_service\_custom:  method: 'add'  url\_params:  adom: "{{ adom }}"  params:  - data:  - name: 'firewall\_demo\_service0'  comment: 'An Firewall Service created by Ansible'  protocol: 'TCP/UDP/SCTP'  tcp-portrange: '55-87'   * Update a firewall service   - hosts: fortimanager01  connection: httpapi  vars:  adom: "root"  ansible\_httpapi\_use\_ssl: True  ansible\_httpapi\_validate\_certs: False  ansible\_httpapi\_port: 443  tasks:  - name: 'Update a firewall service'  fmgr\_pm\_config\_adom\_adom\_obj\_firewall\_service\_custom:  method: 'update'  url\_params:  adom: "{{ adom }}"  params:  - data:  - name: 'firewall\_demo\_service0'  comment: 'An Firewall Service updated by Ansible'  protocol: 'TCP/UDP/SCTP'  tcp-portrange: '66-88'   * Get a firewall service using filter   - hosts: fortimanager01  connection: httpapi  vars:  adom: "root"  ansible\_httpapi\_use\_ssl: True  ansible\_httpapi\_validate\_certs: False  ansible\_httpapi\_port: 443  tasks:  - name: 'get a firewall service'  fmgr\_pm\_config\_adom\_adom\_obj\_firewall\_service\_custom:  method: 'get'  url\_params:  adom: "{{ adom }}"  params:  - filter:  - 'name'  - '=='  - 'firewall\_demo\_service0'   * Delete a firewall service   - hosts: fortimanager01  connection: httpapi  vars:  adom: "root"  ansible\_httpapi\_use\_ssl: True  ansible\_httpapi\_validate\_certs: False  ansible\_httpapi\_port: 443  tasks:  - name: 'delete a firewall service'  fmgr\_pm\_config\_adom\_adom\_obj\_firewall\_service\_custom\_custom:  method: 'delete'  url\_params:  adom: "{{ adom }}"  custom: 'firewall\_demo\_service0'  **the playbooks list:**   * firewall\_service\_add.yml * firewall\_service\_delete.yml * firewall\_service\_get.yml * firewall\_service\_update.yml |
| Validation: | * ansible-playbook -i hosts/ firewall\_service\_add.yml –vvv   "response": [  0,  {  "status": {  "code": 0,  "message": "OK"  },  "url": "/pm/config/adom/root/obj/firewall/service/custom"  }  ]   * ansible-playbook -i hosts/ firewall\_service\_update.yml–vvv   "response": [  0,  {  "status": {  "code": 0,  "message": "OK"  },  "url": "/pm/config/adom/root/obj/firewall/service/custom"  }  ]   * ansible-playbook -i hosts/ firewall\_service\_get.yml –vvv   "response": [  0,  [  {  "app-category": [],  "app-service-type": "disable",  "application": [],  "category": [],  "check-reset-range": "default",  "color": 0,  "comment": "An Firewall Service updated by Ansible",  "helper": "auto",  "iprange": "0.0.0.0",  "name": "firewall\_demo\_service0",  "obj seq": 89,  "protocol": "TCP/UDP/SCTP",  "proxy": "disable",  "sctp-portrange": [],  "session-ttl": 0,  "tcp-halfclose-timer": 0,  "tcp-halfopen-timer": 0,  "tcp-portrange": [  "66-88"  ],  "tcp-timewait-timer": 0,  "udp-idle-timer": 0,  "udp-portrange": [],  "visibility": "enable"  }  ]  ]   * ansible-playbook -i hosts/ firewall\_service\_delete.yml –vvv   "response": [  0,  {  "status": {  "code": 0,  "message": "OK"  },  "url": "/pm/config/adom/root/obj/firewall/service/custom/firewall\_demo\_service0"  }  ]   * run it one more time to make sure it indeed is deleted.   ansible-playbook -i hosts/ firewall\_service\_delete.yml –vvv  "status": {  "code": -3,  "message": "Object does not exist"  } |
| Completion Criteria: | success. |
| Special Considerations: |  |

### Security Profile DNS Filter Test

|  |  |
| --- | --- |
| Test Objective: | Test The Basic operations with FortiManager Security Profile DNS Filter  Operations include: **Add**, **Update**(**Set**), **Get** and **Delete** |
| Technique: | Playbooks for each operation:   * Add a security profile DNS Filter   - hosts: fortimanager01  connection: httpapi  vars:  adom: "root"  ansible\_httpapi\_use\_ssl: True  ansible\_httpapi\_validate\_certs: False  ansible\_httpapi\_port: 443  tasks:  - name: 'add a seurity profile dns filter entry'  fmgr\_pm\_config\_adom\_adom\_obj\_dnsfilter\_profile:  method: 'add'  url\_params:  adom: 'root'  params:  - data:  - name: 'secprof\_dns\_filter0'  comment: 'A seurity profile dns filter created by Ansible'  block-action: 'block'  safe-search: 'enable'   * Update a security profile DNS Filter   - hosts: fortimanager01  connection: httpapi  vars:  adom: "root"  ansible\_httpapi\_use\_ssl: True  ansible\_httpapi\_validate\_certs: False  ansible\_httpapi\_port: 443  tasks:  - name: 'update a seurity profile dns filter entry'  fmgr\_pm\_config\_adom\_adom\_obj\_dnsfilter\_profile:  method: 'update'  url\_params:  adom: 'root'  params:  - data:  - name: 'secprof\_dns\_filter0'  comment: 'A seurity profile dns filter updated by Ansible'  block-action: 'block'  safe-search: 'disable'   * Get a security profile DNS Filter using filter   - hosts: fortimanager01  connection: httpapi  vars:  adom: "root"  ansible\_httpapi\_use\_ssl: True  ansible\_httpapi\_validate\_certs: False  ansible\_httpapi\_port: 443  tasks:  - name: 'get a security profile dns filter entry'  fmgr\_pm\_config\_adom\_adom\_obj\_dnsfilter\_profile:  method: 'get'  url\_params:  adom: 'root'  params:  - filter:  - 'name'  - '=='  - 'secprof\_dns\_filter0'   * Delete a security profile DNS Filter   - hosts: fortimanager01  connection: httpapi  vars:  adom: "root"  ansible\_httpapi\_use\_ssl: True  ansible\_httpapi\_validate\_certs: False  ansible\_httpapi\_port: 443  tasks:  - name: 'delete a security profile dns filter entry'  fmgr\_pm\_config\_adom\_adom\_obj\_dnsfilter\_profile\_profile:  method: 'delete'  url\_params:  adom: 'root'  profile: 'secprof\_dns\_filter0'  **the playbooks list:**   * secprof\_dnsfilter\_add.yml * secprof\_dnsfilter\_delete.yml * secprof\_dnsfilter\_get.yml * secprof\_dnsfilter\_update.yml |
| Validation: | * ansible-playbook -i hosts/ secprof\_dnsfilter\_add.yml –vvv   "response": [  0,  {  "status": {  "code": 0,  "message": "OK"  },  "url": "/pm/config/adom/root/obj/dnsfilter/profile"  }  ]   * ansible-playbook -i hosts/ secprof\_dnsfilter\_update.yml –vvv   "response": [  0,  {  "status": {  "code": 0,  "message": "OK"  },  "url": "/pm/config/adom/root/obj/dnsfilter/profile"  }  ]   * ansible-playbook -i hosts/ secprof\_dnsfilter\_get.yml –vvv   "response": [  0,  [  {  "block-action": "block",  "block-botnet": "disable",  "comment": "A security profile dns filter updated by Ansible",  "domain-filter": {  "domain-filter-table": 0  },  "external-ip-blocklist": [],  "ftgd-dns": {  "filters": null,  "options": null  },  "log-all-domain": "disable",  "name": "secprof\_dns\_filter0",  "redirect-portal": "0.0.0.0",  "safe-search": "disable",  "sdns-domain-log": "enable",  "sdns-ftgd-err-log": "enable",  "youtube-restrict": "strict"  }  ]  ]   * ansible-playbook -i hosts/ secprof\_dnsfilter\_delete.yml –vvv   "response": [  0,  {  "status": {  "code": 0,  "message": "OK"  },  "url": "/pm/config/adom/root/obj/dnsfilter/profile/secprof\_dns\_filter0"  }  ]   * run it one more time to make sure it indeed is deleted.   ansible-playbook -i hosts/ secprof\_dnsfilter\_delete.yml –vvv  "status": {  "code": -3,  "message": "Object does not exist"  } |
| Completion Criteria: | success. |
| Special Considerations: |  |

### Security Profile Anti Virus Test

|  |  |
| --- | --- |
| Test Objective: | Test The Basic operations with FortiManager Security Profile Anti-Virus  Operations include: **Add**,**Update**(**Set**), **Get** and **Delete** |
| Technique: | Playbooks for each operation:   * Add an antivirus security profile   - hosts: fortimanager01  connection: httpapi  vars:  adom: "root"  ansible\_httpapi\_use\_ssl: True  ansible\_httpapi\_validate\_certs: False  ansible\_httpapi\_port: 443  tasks:  - name: 'add a security profile anti-virus entry'  fmgr\_pm\_config\_adom\_adom\_obj\_antivirus\_profile:  method: 'add'  url\_params:  adom: 'root'  params:  - data:  - name: 'secprof\_anti\_virus0'  comment: 'A seurity profile anti virus entry created by Ansible'  inspection-mode: 'proxy'   * Update an antivirus security profile   - hosts: fortimanager01  connection: httpapi  vars:  adom: "root"  ansible\_httpapi\_use\_ssl: True  ansible\_httpapi\_validate\_certs: False  ansible\_httpapi\_port: 443  tasks:  - name: 'Update a security profile anti-virus entry'  fmgr\_pm\_config\_adom\_adom\_obj\_antivirus\_profile:  method: 'update'  url\_params:  adom: 'root'  params:  - data:  - name: 'secprof\_anti\_virus0'  comment: 'A seurity profile anti virus entry updated by Ansible'  inspection-mode: 'flow-based'   * Get an antivirus security profile using filter   - hosts: fortimanager01  connection: httpapi  vars:  adom: "root"  ansible\_httpapi\_use\_ssl: True  ansible\_httpapi\_validate\_certs: False  ansible\_httpapi\_port: 443  tasks:  - name: 'get a security profile anti-virus entry'  fmgr\_pm\_config\_adom\_adom\_obj\_antivirus\_profile:  method: 'get'  url\_params:  adom: 'root'  params:  - filter:  - 'name'  - '=='  - 'secprof\_anti\_virus0'   * Delete an antivirus security profile   - hosts: fortimanager01  connection: httpapi  vars:  adom: "root"  ansible\_httpapi\_use\_ssl: True  ansible\_httpapi\_validate\_certs: False  ansible\_httpapi\_port: 443  tasks:  - name: 'delete a security profile anti-virus entry'  fmgr\_pm\_config\_adom\_adom\_obj\_antivirus\_profile\_profile:  method: 'delete'  url\_params:  adom: 'root'  profile: 'secprof\_anti\_virus0'  **the playbooks list:**   * secprof\_antivirus\_add.yml * secprof\_antivirus\_delete.yml * secprof\_antivirus\_get.yml * secprof\_antivirus\_update.yml |
| Validation: | * ansible-playbook -i hosts/ secprof\_antivirus\_add.yml –vvv   "response": [  0,  {  "status": {  "code": 0,  "message": "OK"  },  "url": "/pm/config/adom/root/obj/antivirus/profile"  }  ]   * ansible-playbook -i hosts/ secprof\_antivirus\_update.yml –vvv   "response": [  0,  {  "status": {  "code": 0,  "message": "OK"  },  "url": "/pm/config/adom/root/obj/antivirus/profile"  }  ]   * ansible-playbook -i hosts/ secprof\_antivirus\_get.yml –vvv   "response": [  0,  [  {  "analytics-db": "disable",  "av-block-log": "enable",  "av-virus-log": "enable",  "comment": "A seurity profile anti virus entry updated by Ansible",  "content-disarm": {  "cover-page": "enable",  "detect-only": "disable",  "office-embed": "enable",  "office-hylink": "enable",  "office-linked": "enable",  "office-macro": "enable",  "original-file-destination": "discard",  "pdf-act-form": "enable",  "pdf-act-gotor": "enable",  "pdf-act-java": "enable",  "pdf-act-launch": "enable",  "pdf-act-movie": "enable",  "pdf-act-sound": "enable",  "pdf-embedfile": "enable",  "pdf-hyperlink": "enable",  "pdf-javacode": "enable"  },  "extended-log": "disable",  "ftgd-analytics": "disable",  "ftp": {  "archive-block": null,  "archive-log": null,  "emulator": "enable",  "options": null,  "outbreak-prevention": "disabled"  },  "http": {  "archive-block": null,  "archive-log": null,  "content-disarm": "disable",  "emulator": "enable",  "options": null,  "outbreak-prevention": "disabled"  },  "imap": {  "archive-block": null,  "archive-log": null,  "content-disarm": "disable",  "emulator": "enable",  "executables": "default",  "options": null,  "outbreak-prevention": "disabled"  },  "inspection-mode": "flow-based",  "mapi": null,  "mobile-malware-db": "enable",  "nac-quar": {  "expiry": "5m",  "infected": "none",  "log": "disable"  },  "name": "secprof\_anti\_virus0",  "nntp": null,  "pop3": {  "archive-block": null,  "archive-log": null,  "content-disarm": "disable",  "emulator": "enable",  "executables": "default",  "options": null,  "outbreak-prevention": "disabled"  },  "replacemsg-group": [],  "scan-mode": "full",  "smb": {  "archive-block": null,  "archive-log": null,  "emulator": "enable",  "options": null,  "outbreak-prevention": "disabled"  },  "smtp": {  "archive-block": null,  "archive-log": null,  "content-disarm": "disable",  "emulator": "enable",  "executables": "default",  "options": null,  "outbreak-prevention": "disabled"  }  }  ]  ]   * ansible-playbook -i hosts/ secprof\_antivirus\_delete.yml –vvv   "response": [  0,  {  "status": {  "code": 0,  "message": "OK"  },  "url": "/pm/config/adom/root/obj/antivirus/profile/secprof\_anti\_virus0"  }  ]   * run it one more time to make sure it indeed is deleted.   ansible-playbook -i hosts/ secprof\_antivirus\_delete.yml –vvv  "status": {  "code": -3,  "message": "Object does not exist"  } |
| Completion Criteria: | success. |
| Special Considerations: |  |

### Security Profile IPS Sensor Test

|  |  |
| --- | --- |
| Test Objective: | Test The Basic operations with FortiManager Policy Packages  Operations include: **Add**, **Update**(**Set**), **Get** and **Delete** |
| Technique: | Playbooks for each operation:   * Add an IPS sensor security profile   - hosts: fortimanager01  connection: httpapi  vars:  adom: "root"  ansible\_httpapi\_use\_ssl: True  ansible\_httpapi\_validate\_certs: False  ansible\_httpapi\_port: 443  tasks:  - name: 'add a seurity profile IPS sensor entry'  fmgr\_pm\_config\_adom\_adom\_obj\_ips\_sensor:  method: 'add'  url\_params:  adom: 'root'  params:  - data:  - name: 'secprof\_ips\_sensor0'  comment: 'A seurity profile ips sensor entry created by Ansible'   * Update an IPS sensor security profile   - hosts: fortimanager01  connection: httpapi  vars:  adom: "root"  ansible\_httpapi\_use\_ssl: True  ansible\_httpapi\_validate\_certs: False  ansible\_httpapi\_port: 443  tasks:  - name: 'update a seurity profile IPS sensor entry'  fmgr\_pm\_config\_adom\_adom\_obj\_ips\_sensor:  method: 'update'  url\_params:  adom: 'root'  params:  - data:  - name: 'secprof\_ips\_sensor0'  comment: 'A seurity profile ips sensor entry updated by Ansible'   * Get an IPS sensor security profile using filter   - hosts: fortimanager01  connection: httpapi  vars:  adom: "root"  ansible\_httpapi\_use\_ssl: True  ansible\_httpapi\_validate\_certs: False  ansible\_httpapi\_port: 443  tasks:  - name: 'get a seurity profile IPS sensor entry'  fmgr\_pm\_config\_adom\_adom\_obj\_ips\_sensor:  method: 'get'  url\_params:  adom: 'root'  params:  - filter:  - 'name'  - '=='  - 'secprof\_ips\_sensor0'   * Delete an IPS sensor security profile   - hosts: fortimanager01  connection: httpapi  vars:  adom: "root"  ansible\_httpapi\_use\_ssl: True  ansible\_httpapi\_validate\_certs: False  ansible\_httpapi\_port: 443  tasks:  - name: 'delete a seurity profile IPS sensor entry'  fmgr\_pm\_config\_adom\_adom\_obj\_ips\_sensor\_sensor:  method: 'delete'  url\_params:  adom: 'root'  sensor: 'secprof\_ips\_sensor0'  **the playbooks list:**   * secprof\_ips\_sensor\_add.yml * secprof\_ips\_sensor\_delete.yml * secprof\_ips\_sensor\_get.yml * secprof\_ips\_sensor\_update.yml |
| Validation: | * ansible-playbook -i hosts/ secprof\_ips\_sensor\_add.yml –vvv   "response": [  0,  {  "status": {  "code": 0,  "message": "OK"  },  "url": "/pm/config/adom/root/obj/ips/sensor"  }  ]   * ansible-playbook -i hosts/ secprof\_ips\_sensor\_update.yml –vvv   "response": [  0,  {  "status": {  "code": 0,  "message": "OK"  },  "url": "/pm/config/adom/root/obj/ips/sensor"  }  ]   * ansible-playbook -i hosts/ secprof\_ips\_sensor\_get.yml –vvv   "response": [  0,  [  {  "block-malicious-url": "disable",  "comment": "A seurity profile ips sensor entry updated by Ansible",  "entries": null,  "extended-log": "disable",  "name": "secprof\_ips\_sensor0",  "replacemsg-group": []  }  ]  ]   * ansible-playbook -i hosts/ firewall\_package\_delete.yml –vvv   "response": [  0,  {  "status": {  "code": 0,  "message": "OK"  },  "url": "/pm/config/adom/root/obj/ips/sensor/secprof\_ips\_sensor0"  }  ]   * run it one more time to make sure it indeed is deleted.   ansible-playbook -i hosts/ firewall\_package\_delete.yml –vvv  "status": {  "code": -3,  "message": "Object does not exist"  } |
| Completion Criteria: | success. |
| Special Considerations: |  |

### Security Profile Web Filter Test

|  |  |
| --- | --- |
| Test Objective: | Test The Basic operations with FortiManager Security Profile Web Filter  Operations include: **Add**, **Update**(**Set**), **Get** and **Delete** |
| Technique: | Playbooks for each operation:   * Add a web filter security profile   - hosts: fortimanager01  connection: httpapi  vars:  adom: "root"  ansible\_httpapi\_use\_ssl: True  ansible\_httpapi\_validate\_certs: False  ansible\_httpapi\_port: 443  tasks:  - name: 'add a seurity profile webfilter entry'  fmgr\_pm\_config\_adom\_adom\_obj\_webfilter\_profile:  method: 'add'  url\_params:  adom: 'root'  params:  - data:  - name: 'secprof\_web\_filter0'  comment: 'A seurity profile web filter entry created by Ansible'  options:  - 'jscript'  - 'cookiefilter'   * Update a web filter security profile   - hosts: fortimanager01  connection: httpapi  vars:  adom: "root"  ansible\_httpapi\_use\_ssl: True  ansible\_httpapi\_validate\_certs: False  ansible\_httpapi\_port: 443  tasks:  - name: 'update a seurity profile webfilter entry'  fmgr\_pm\_config\_adom\_adom\_obj\_webfilter\_profile:  method: 'update'  url\_params:  adom: 'root'  params:  - data:  - name: 'secprof\_web\_filter0'  comment: 'A seurity profile web filter entry updated by Ansible'  options:  - 'jscript'  - 'cookiefilter'  inspection-mode: 'flow-based'   * Get a web filter security profile using filter   - hosts: fortimanager01  connection: httpapi  vars:  adom: "root"  ansible\_httpapi\_use\_ssl: True  ansible\_httpapi\_validate\_certs: False  ansible\_httpapi\_port: 443  tasks:  - name: 'get a seurity profile webfilter entry'  fmgr\_pm\_config\_adom\_adom\_obj\_webfilter\_profile:  method: 'get'  url\_params:  adom: 'root'  params:  - filter:  - 'name'  - '=='  - 'secprof\_web\_filter0'   * Delete a web filter security profile   - hosts: fortimanager01  connection: httpapi  vars:  adom: "root"  ansible\_httpapi\_use\_ssl: True  ansible\_httpapi\_validate\_certs: False  ansible\_httpapi\_port: 443  tasks:  - name: 'delete a seurity profile webfilter entry'  fmgr\_pm\_config\_adom\_adom\_obj\_webfilter\_profile\_profile:  method: 'delete'  url\_params:  adom: 'root'  profile: 'secprof\_web\_filter0'  **the playbooks list:**   * secprof\_webfilter\_add.yml * secprof\_webfilter\_delete.yml * secprof\_webfilter\_get.yml * secprof\_webfilter\_update.yml |
| Validation: | * ansible-playbook -i hosts/ secprof\_webfilter\_add.yml –vvv   "response": [  0,  {  "status": {  "code": 0,  "message": "OK"  },  "url": "/pm/config/adom/root/obj/webfilter/profile"  }  ]   * ansible-playbook -i hosts/ secprof\_webfilter\_update.yml –vvv   "response": [  0,  {  "status": {  "code": 0,  "message": "OK"  },  "url": "/pm/config/adom/root/obj/webfilter/profile"  }  ]   * ansible-playbook -i hosts/ secprof\_webfilter\_get.yml –vvv   "response": [  0,  [  {  "comment": "A seurity profile web filter entry updated by Ansible",  "extended-log": "disable",  "ftgd-wf": {  "exempt-quota": [  "17"  ],  "filters": null,  "max-quota-timeout": 300,  "options": "ftgd-disable",  "ovrd": [],  "quota": null,  "rate-crl-urls": "enable",  "rate-css-urls": "enable",  "rate-image-urls": "enable",  "rate-javascript-urls": "enable"  },  "https-replacemsg": "enable",  "inspection-mode": "flow-based",  "log-all-url": "disable",  "name": "secprof\_web\_filter0",  "options": [  "jscript",  "cookiefilter"  ],  "override": {  "ovrd-cookie": "deny",  "ovrd-dur": "15m",  "ovrd-dur-mode": "constant",  "ovrd-scope": "user",  "ovrd-user-group": [],  "profile": [],  "profile-attribute": "Login-LAT-Service",  "profile-type": "list"  },  "post-action": "normal",  "replacemsg-group": [],  "url-extraction": null,  "web": {  "blacklist": "disable",  "bword-table": [],  "bword-threshold": 10,  "content-header-list": [],  "keyword-match": [],  "log-search": "disable",  "safe-search": null,  "urlfilter-table": [],  "whitelist": null,  "youtube-restrict": "none"  },  "web-content-log": "enable",  "web-extended-all-action-log": "disable",  "web-filter-activex-log": "enable",  "web-filter-applet-log": "enable",  "web-filter-command-block-log": "enable",  "web-filter-cookie-log": "enable",  "web-filter-cookie-removal-log": "enable",  "web-filter-js-log": "enable",  "web-filter-jscript-log": "enable",  "web-filter-referer-log": "enable",  "web-filter-unknown-log": "enable",  "web-filter-vbs-log": "enable",  "web-ftgd-err-log": "enable",  "web-ftgd-quota-usage": "enable",  "web-invalid-domain-log": "enable",  "web-url-log": "enable",  "wisp": "disable",  "wisp-algorithm": "auto-learning",  "youtube-channel-filter": null,  "youtube-channel-status": "disable"  }  ]  ]   * ansible-playbook -i hosts/ secprof\_webfilter\_delete.yml –vvv   "response": [  0,  {  "status": {  "code": 0,  "message": "OK"  },  "url": "/pm/config/adom/root/obj/webfilter/profile/secprof\_web\_filter0"  }  ]   * run it one more time to make sure it indeed is deleted.   ansible-playbook -i hosts/ secprof\_webfilter\_delete.yml –vvv  "status": {  "code": -3,  "message": "Object does not exist"  } |
| Completion Criteria: | success. |
| Special Considerations: |  |

### Security Profile Application Control Test

|  |  |
| --- | --- |
| Test Objective: | Test The Basic operations with FortiManager Security Profile Application Control  Operations include: **Add**, **Update**(**Set**), **Get** and **Delete** |
| Technique: | Playbooks for each operation:   * Add an application control security profile   - hosts: fortimanager01  connection: httpapi  vars:  adom: "root"  ansible\_httpapi\_use\_ssl: True  ansible\_httpapi\_validate\_certs: False  ansible\_httpapi\_port: 443  tasks:  - name: 'add a seurity profile application control entry'  fmgr\_pm\_config\_adom\_adom\_obj\_application\_list:  method: 'add'  url\_params:  adom: 'root'  params:  - data:  - name: 'secprof\_application\_control0'  comment: 'A seurity profile application control entry created by Ansible'  options:  - 'allow-icmp'  - 'allow-http'   * Update an application control security profile   - hosts: fortimanager01  connection: httpapi  vars:  adom: "root"  ansible\_httpapi\_use\_ssl: True  ansible\_httpapi\_validate\_certs: False  ansible\_httpapi\_port: 443  tasks:  - name: 'update a seurity profile application control entry'  fmgr\_pm\_config\_adom\_adom\_obj\_application\_list:  method: 'update'  url\_params:  adom: 'root'  params:  - data:  - name: 'secprof\_application\_control0'  comment: 'A seurity profile application control entry updated by Ansible'  options:  - 'allow-icmp'  - 'allow-http'  deep-app-inspection: 'disable'  extended-log: 'enable'   * Get an application control security profile using filter   - hosts: fortimanager01  connection: httpapi  vars:  adom: "root"  ansible\_httpapi\_use\_ssl: True  ansible\_httpapi\_validate\_certs: False  ansible\_httpapi\_port: 443  tasks:  - name: 'get a seurity profile application control entry'  fmgr\_pm\_config\_adom\_adom\_obj\_application\_list:  method: 'get'  url\_params:  adom: 'root'  params:  - filter:  - 'name'  - '=='  - 'secprof\_application\_control0'   * Delete an application control security profile   - hosts: fortimanager01  connection: httpapi  vars:  adom: "root"  ansible\_httpapi\_use\_ssl: True  ansible\_httpapi\_validate\_certs: False  ansible\_httpapi\_port: 443  tasks:  - name: 'delete a seurity profile application control entry'  fmgr\_pm\_config\_adom\_adom\_obj\_application\_list\_list:  method: 'delete'  url\_params:  adom: 'root'  list: 'secprof\_application\_control0'  **the playbooks list:**   * secprof\_appctrl\_add.yml * secprof\_appctrl\_delete.yml * secprof\_appctrl\_get.yml * secprof\_appctrl\_update.yml |
| Validation: | * ansible-playbook -i hosts/ secprof\_appctrl\_add.yml –vvv   "response": [  0,  {  "status": {  "code": 0,  "message": "OK"  },  "url": "/pm/config/adom/root/obj/application/list"  }  ]   * ansible-playbook -i hosts/ secprof\_appctrl\_update.yml –vvv   "response": [  0,  {  "status": {  "code": 0,  "message": "OK"  },  "url": "/pm/config/adom/root/obj/application/list"  }  ]   * ansible-playbook -i hosts/ secprof\_appctrl\_get.yml –vvv   "response": [  0,  [  {  "app-replacemsg": "enable",  "comment": "A seurity profile application control entry updated by Ansible",  "deep-app-inspection": "disable",  "entries": null,  "extended-log": "enable",  "name": "secprof\_application\_control0",  "options": [  "allow-icmp",  "allow-http"  ],  "other-application-action": "pass",  "other-application-log": "disable",  "replacemsg-group": [],  "unknown-application-action": "pass",  "unknown-application-log": "disable"  }  ]  ]   * ansible-playbook -i hosts/ secprof\_appctrl\_delete.yml –vvv   "response": [  0,  {  "status": {  "code": 0,  "message": "OK"  },  "url": "/pm/config/adom/root/obj/application/list/secprof\_application\_control0"  }  ]   * run it one more time to make sure it indeed is deleted.   ansible-playbook -i hosts/ secprof\_appctrl\_delete.yml –vvv  "status": {  "code": -3,  "message": "Object does not exist"  } |
| Completion Criteria: | success. |
| Special Considerations: |  |

### Security Profile Group Test

|  |  |
| --- | --- |
| Test Objective: | Test The Basic operations with FortiManager Security Profile Group  Operations include: **Add**, **Get** and **Delete**  **\*** from api url: **/pm/config/adom/{adom}/obj/firewall/profile-group** it looks more like a firewall object. However it’s categorized as security profile here: <https://github.com/ansible/ansible/tree/devel/lib/ansible/modules/network/fortimanager> |
| Technique: | Playbooks for each operation:   * Add a firewall profile group   - hosts: fortimanager01  connection: httpapi  vars:  adom: "root"  ansible\_httpapi\_use\_ssl: True  ansible\_httpapi\_validate\_certs: False  ansible\_httpapi\_port: 443  tasks:  - name: 'add a seurity profile group'  fmgr\_pm\_config\_adom\_adom\_obj\_firewall\_profile\_group:  method: 'add'  url\_params:  adom: 'root'  params:  - data:  - name: 'secprof\_group0'   * Get a firewall profile group using filter   - hosts: fortimanager01  connection: httpapi  vars:  adom: "root"  ansible\_httpapi\_use\_ssl: True  ansible\_httpapi\_validate\_certs: False  ansible\_httpapi\_port: 443  tasks:  - name: 'get a seurity profile group'  fmgr\_pm\_config\_adom\_adom\_obj\_firewall\_profile\_group:  method: 'get'  url\_params:  adom: 'root'  params:  - filter:  - 'name'  - '=='  - 'secprof\_group0'   * Delete a firewall profile group   - hosts: fortimanager01  connection: httpapi  vars:  adom: "root"  ansible\_httpapi\_use\_ssl: True  ansible\_httpapi\_validate\_certs: False  ansible\_httpapi\_port: 443  tasks:  - name: 'delete a seurity profile group'  fmgr\_pm\_config\_adom\_adom\_obj\_firewall\_profile\_group\_profile\_group:  method: 'delete'  url\_params:  adom: 'root'  profile-group: 'secprof\_group0'  **the playbooks list:**   * secprof\_group\_add.yml * secprof\_group\_delete.yml * secprof\_group\_get.yml |
| Validation: | * ansible-playbook -i hosts/ secprof\_group\_add.yml –vvv   "response": [  0,  {  "status": {  "code": 0,  "message": "OK"  },  "url": "/pm/config/adom/root/obj/firewall/profile-group"  }  ]   * ansible-playbook -i hosts/ secprof\_group\_get.yml –vvv   "response": [  0,  [  {  "application-list": [],  "av-profile": [],  "dlp-sensor": [],  "dnsfilter-profile": [],  "icap-profile": [],  "ips-sensor": [],  "name": "secprof\_group0",  "profile-protocol-options": [  "default"  ],  "spamfilter-profile": [],  "ssh-filter-profile": [],  "ssl-ssh-profile": [  "certificate-inspection"  ],  "voip-profile": [],  "waf-profile": [],  "webfilter-profile": []  }  ]  ]   * ansible-playbook -i hosts/ secprof\_group\_delete.yml –vvv   "response": [  0,  {  "status": {  "code": 0,  "message": "OK"  },  "url": "/pm/config/adom/root/obj/firewall/profile-group/secprof\_group0"  }  ]   * run it one more time to make sure it indeed is deleted.   ansible-playbook -i hosts/ secprof\_group\_delete.yml –vvv  "status": {  "code": -3,  "message": "Object does not exist"  } |
| Completion Criteria: | success. |
| Special Considerations: |  |

### Security Profile Web Proxy Test

|  |  |
| --- | --- |
| Test Objective: | Test The Basic operations with FortiManager Security Profile Web Proxy  Operations include: **Add**, **Update**(**Set**), **Get** and **Delete** |
| Technique: | Playbooks for each operation:   * Add a web proxy profile   - hosts: fortimanager01  connection: httpapi  vars:  adom: "root"  ansible\_httpapi\_use\_ssl: True  ansible\_httpapi\_validate\_certs: False  ansible\_httpapi\_port: 443  tasks:  - name: 'add a seurity profile web-proxy entry'  fmgr\_pm\_config\_adom\_adom\_obj\_web\_proxy\_profile:  method: 'add'  url\_params:  adom: 'root'  params:  - data:  - name: 'secprof\_webproxy0'  strip-encoding: 'enable'   * Update a web proxy profile   - hosts: fortimanager01  connection: httpapi  vars:  adom: "root"  ansible\_httpapi\_use\_ssl: True  ansible\_httpapi\_validate\_certs: False  ansible\_httpapi\_port: 443  tasks:  - name: 'update a seurity profile web-proxy entry'  fmgr\_pm\_config\_adom\_adom\_obj\_web\_proxy\_profile:  method: 'update'  url\_params:  adom: 'root'  params:  - data:  - name: 'secprof\_webproxy0'  strip-encoding: 'disable'   * Get a web proxy profile using filter   - hosts: fortimanager01  connection: httpapi  vars:  adom: "root"  ansible\_httpapi\_use\_ssl: True  ansible\_httpapi\_validate\_certs: False  ansible\_httpapi\_port: 443  tasks:  - name: 'get a seurity profile web-proxy entry'  fmgr\_pm\_config\_adom\_adom\_obj\_web\_proxy\_profile:  method: 'get'  url\_params:  adom: 'root'  params:  - filter:  - 'name'  - '=='  - 'secprof\_webproxy0'   * Delete a web proxy profile   - hosts: fortimanager01  connection: httpapi  vars:  adom: "root"  ansible\_httpapi\_use\_ssl: True  ansible\_httpapi\_validate\_certs: False  ansible\_httpapi\_port: 443  tasks:  - name: 'delete a seurity profile web-proxy entry'  fmgr\_pm\_config\_adom\_adom\_obj\_web\_proxy\_profile\_profile:  method: 'delete'  url\_params:  adom: 'root'  profile: 'secprof\_webproxy0'  **the playbooks list:**   * secprof\_webproxy\_add.yml * secprof\_webproxy\_delete.yml * secprof\_webproxy\_get.yml * secprof\_webproxy\_update.yml |
| Validation: | * ansible-playbook -i hosts/ secprof\_webproxy\_add.yml –vvv   "response": [  0,  {  "status": {  "code": 0,  "message": "OK"  },  "url": "/pm/config/adom/root/obj/web-proxy/profile"  }  ]   * ansible-playbook -i hosts/ secprof\_webproxy\_update.yml –vvv   "response": [  0,  {  "status": {  "code": 0,  "message": "OK"  },  "url": "/pm/config/adom/root/obj/web-proxy/profile"  }  ]   * ansible-playbook -i hosts/ secprof\_webproxy\_get.yml –vvv   "response": [  0,  [  {  "header-client-ip": "pass",  "header-front-end-https": "pass",  "header-via-request": "pass",  "header-via-response": "pass",  "header-x-authenticated-groups": "pass",  "header-x-authenticated-user": "pass",  "header-x-forwarded-for": "pass",  "headers": null,  "log-header-change": "disable",  "name": "secprof\_webproxy0",  "strip-encoding": "disable"  }  ]  ]   * ansible-playbook -i hosts/ secprof\_webproxy\_delete.yml –vvv   "response": [  0,  {  "status": {  "code": 0,  "message": "OK"  },  "url": "/pm/config/adom/root/obj/web-proxy/profile/secprof\_webproxy0"  }  ]   * run it one more time to make sure it indeed is deleted.   ansible-playbook -i hosts secprof\_webproxy\_delete.yml –vvv  "status": {  "code": -3,  "message": "Object does not exist"  } |
| Completion Criteria: | success. |
| Special Considerations: |  |

### Security Profile Spam Filter Test

|  |  |
| --- | --- |
| Test Objective: | Test The Basic operations with FortiManager Security Profile Spam Filter  Operations include: **Add**, **Update**(**Set**), **Get** and **Delete** |
| Technique: | Playbooks for each operation:   * Add a spam filter security profile   - hosts: fortimanager01  connection: httpapi  vars:  adom: "root"  ansible\_httpapi\_use\_ssl: True  ansible\_httpapi\_validate\_certs: False  ansible\_httpapi\_port: 443  tasks:  - name: 'add a seurity profile spam filter entry'  fmgr\_pm\_config\_adom\_adom\_obj\_spamfilter\_profile:  method: 'add'  url\_params:  adom: 'root'  params:  - data:  - name: 'secprof\_spam\_filter0'  comment: 'A seurity profile spam filter entry created by Ansible'  options:  - 'bannedword'  - 'spamemailbwl'  - 'spamfsphish'  flow-based: 'enable'   * Update a spam filter security profile   - hosts: fortimanager01  connection: httpapi  vars:  adom: "root"  ansible\_httpapi\_use\_ssl: True  ansible\_httpapi\_validate\_certs: False  ansible\_httpapi\_port: 443  tasks:  - name: 'update a seurity profile spam filter entry'  fmgr\_pm\_config\_adom\_adom\_obj\_spamfilter\_profile:  method: 'update'  url\_params:  adom: 'root'  params:  - data:  - name: 'secprof\_spam\_filter0'  comment: 'A seurity profile spam filter entry updated by Ansible'  options:  - 'bannedword'  - 'spamemailbwl'  - 'spamfsphish'  flow-based: 'disable'   * Get a spam filter security profile using filter   - hosts: fortimanager01  connection: httpapi  vars:  adom: "root"  ansible\_httpapi\_use\_ssl: True  ansible\_httpapi\_validate\_certs: False  ansible\_httpapi\_port: 443  tasks:  - name: 'get a seurity profile spam filter entry'  fmgr\_pm\_config\_adom\_adom\_obj\_spamfilter\_profile:  method: 'get'  url\_params:  adom: 'root'  params:  - filter:  - 'name'  - '=='  - 'secprof\_spam\_filter0'   * Delete a spam filter security profile   - hosts: fortimanager01  connection: httpapi  vars:  adom: "root"  ansible\_httpapi\_use\_ssl: True  ansible\_httpapi\_validate\_certs: False  ansible\_httpapi\_port: 443  tasks:  - name: 'delete a seurity profile spam filter entry'  fmgr\_pm\_config\_adom\_adom\_obj\_spamfilter\_profile\_profile:  method: 'delete'  url\_params:  adom: 'root'  profile: 'secprof\_spam\_filter0'  **the playbooks list:**   * secprof\_spamfilter\_add.yml * secprof\_spamfilter\_delete.yml * secprof\_spamfilter\_get.yml * secprof\_spamfilter\_update.yml |
| Validation: | * ansible-playbook -i hosts/ secprof\_spamfilter\_add.yml –vvv   "response": [  0,  {  "status": {  "code": 0,  "message": "OK"  },  "url": "/pm/config/adom/root/obj/spamfilter/profile"  }  ]   * ansible-playbook -i hosts/ secprof\_spamfilter\_update.yml –vvv   "response": [  0,  {  "status": {  "code": 0,  "message": "OK"  },  "url": "/pm/config/adom/root/obj/spamfilter/profile"  }  ]   * ansible-playbook -i hosts/ secprof\_spamfilter\_get.yml –vvv   "response": [  0,  [  {  "comment": "A seurity profile spam filter entry updated by Ansible",  "external": "disable",  "flow-based": "disable",  "gmail": {  "log": "disable"  },  "imap": {  "log": "disable"  },  "mapi": {  "action": "discard",  "log": "disable"  },  "msn-hotmail": {  "log": "disable"  },  "name": "secprof\_spam\_filter0",  "options": [  "bannedword",  "spamfsphish"  ],  "pop3": {  "log": "disable"  },  "replacemsg-group": [],  "smtp": {  "log": "disable"  },  "spam-bwl-table": [],  "spam-bword-table": [],  "spam-bword-threshold": 10,  "spam-filtering": "disable",  "spam-iptrust-table": [],  "spam-log": "enable",  "spam-log-fortiguard-response": "disable",  "spam-mheader-table": [],  "spam-rbl-table": []  }  ]  ]   * ansible-playbook -i hosts/ secprof\_spamfilter\_delete.yml –vvv   "response": [  0,  {  "status": {  "code": 0,  "message": "OK"  },  "url": "/pm/config/adom/root/obj/spamfilter/profile/secprof\_spam\_filter0"  }  ]   * run it one more time to make sure it indeed is deleted.   ansible-playbook -i hosts/ secprof\_spamfilter\_delete.yml –vvv  "status": {  "code": -3,  "message": "Object does not exist"  } |
| Completion Criteria: | success. |
| Special Considerations: |  |

### Security Profile ssl-ssh Test

|  |  |
| --- | --- |
| Test Objective: | Test The Basic operations with FortiManager Security Profile ssl and ssh  Operations include: **Add**, **Update**(**Set**), **Get** and **Delete** |
| Technique: | Playbooks for each operation:   * Add a ssl-ssh security profile   - hosts: fortimanager01  connection: httpapi  vars:  adom: "root"  ansible\_httpapi\_use\_ssl: True  ansible\_httpapi\_validate\_certs: False  ansible\_httpapi\_port: 443  tasks:  - name: 'add a seurity profile ssl\_ssh entry'  fmgr\_pm\_config\_adom\_adom\_obj\_firewall\_ssl\_ssh\_profile:  method: 'add'  url\_params:  adom: 'root'  params:  - data:  - name: 'secprof\_ssl\_ssh0'  comment: 'A ssl&ssh security profile created by Ansible'  mapi-over-https: 'enable'  whitelist: 'disable'   * Update a ssl-ssh security profile   - hosts: fortimanager01  connection: httpapi  vars:  adom: "root"  ansible\_httpapi\_use\_ssl: True  ansible\_httpapi\_validate\_certs: False  ansible\_httpapi\_port: 443  tasks:  - name: 'update a seurity profile ssl\_ssh entry'  fmgr\_pm\_config\_adom\_adom\_obj\_firewall\_ssl\_ssh\_profile:  method: 'update'  url\_params:  adom: 'root'  params:  - data:  - name: 'secprof\_ssl\_ssh0'  comment: 'A ssl&ssh security profile updated by Ansible'  mapi-over-https: 'disable'  whitelist: 'enable'   * Get a ssl-ssh security profile using filter   - hosts: fortimanager01  connection: httpapi  vars:  adom: "root"  ansible\_httpapi\_use\_ssl: True  ansible\_httpapi\_validate\_certs: False  ansible\_httpapi\_port: 443  tasks:  - name: 'get a seurity profile ssl\_ssh entry'  fmgr\_pm\_config\_adom\_adom\_obj\_firewall\_ssl\_ssh\_profile:  method: 'get'  url\_params:  adom: 'root'  params:  - filter:  - 'name'  - '=='  - 'secprof\_ssl\_ssh0'   * Delete a ssl-ssh security profile   - hosts: fortimanager01  connection: httpapi  vars:  adom: "root"  ansible\_httpapi\_use\_ssl: True  ansible\_httpapi\_validate\_certs: False  ansible\_httpapi\_port: 443  tasks:  - name: 'delete a seurity profile ssl\_ssh entry'  fmgr\_pm\_config\_adom\_adom\_obj\_firewall\_ssl\_ssh\_profile\_ssl\_ssh\_profile:  method: 'delete'  url\_params:  adom: 'root'  ssl-ssh-profile: 'secprof\_ssl\_ssh0'  **the playbooks list:**   * secprof\_ssl\_ssh\_add.yml * secprof\_ssl\_ssh\_delete.yml * secprof\_ssl\_ssh\_get.yml * secprof\_ssl\_ssh\_update.yml |
| Validation: | * ansible-playbook -i hosts/ secprof\_ssl\_ssh\_add.yml –vvv   "response": [  0,  {  "status": {  "code": 0,  "message": "OK"  },  "url": "/pm/config/adom/root/obj/firewall/ssl-ssh-profile"  }  ]   * ansible-playbook -i hosts/ secprof\_ssl\_ssh\_update.yml –vvv   "response": [  0,  {  "status": {  "code": 0,  "message": "OK"  },  "url": "/pm/config/adom/root/obj/firewall/ssl-ssh-profile"  }  ]   * ansible-playbook -i hosts/ secprof\_ssl\_ssh\_get.yml –vvv   "response": [  0,  [  {  "caname": [  "Fortinet\_CA\_SSL"  ],  "comment": "A ssl&ssh security profile updated by Ansible",  "ftps": {  "allow-invalid-server-cert": "disable",  "client-cert-request": "bypass",  "ports": [],  "status": "deep-inspection",  "unsupported-ssl": "bypass",  "untrusted-cert": "allow"  },  "https": {  "allow-invalid-server-cert": "disable",  "client-cert-request": "bypass",  "ports": [],  "status": "deep-inspection",  "unsupported-ssl": "bypass",  "untrusted-cert": "allow"  },  "imaps": {  "allow-invalid-server-cert": "disable",  "client-cert-request": "inspect",  "ports": [],  "status": "deep-inspection",  "unsupported-ssl": "bypass",  "untrusted-cert": "allow"  },  "mapi-over-https": "disable",  "name": "secprof\_ssl\_ssh0",  "pop3s": {  "allow-invalid-server-cert": "disable",  "client-cert-request": "inspect",  "ports": [],  "status": "deep-inspection",  "unsupported-ssl": "bypass",  "untrusted-cert": "allow"  },  "rpc-over-https": "disable",  "server-cert-mode": "re-sign",  "smtps": {  "allow-invalid-server-cert": "disable",  "client-cert-request": "inspect",  "ports": [],  "status": "deep-inspection",  "unsupported-ssl": "bypass",  "untrusted-cert": "allow"  },  "ssh": {  "inspect-all": "disable",  "ports": [],  "ssh-algorithm": "compatible",  "ssh-policy-check": "disable",  "ssh-tun-policy-check": "disable",  "status": "deep-inspection",  "unsupported-version": "bypass"  },  "ssl": {  "inspect-all": "disable"  },  "ssl-anomalies-log": "enable",  "ssl-exempt": null,  "ssl-exemptions-log": "disable",  "ssl-server": null,  "untrusted-caname": [  "Fortinet\_CA\_Untrusted"  ],  "use-ssl-server": "disable",  "whitelist": "enable"  }  ]  ]   * ansible-playbook -i hosts/ secprof\_ssl\_ssh\_delete.yml –vvv   "response": [  0,  {  "status": {  "code": 0,  "message": "OK"  },  "url": "/pm/config/adom/root/obj/firewall/ssl-ssh-profile/secprof\_ssl\_ssh0"  }  ]   * run it one more time to make sure it indeed is deleted.   ansible-playbook -i hosts/ secprof\_ssl\_ssh\_delete.yml –vvv  "status": {  "code": -3,  "message": "Object does not exist"  } |
| Completion Criteria: | success. |
| Special Considerations: |  |

### Security Profile voip Test

|  |  |
| --- | --- |
| Test Objective: | Test The Basic operations with FortiManager Security Profile VOIP  Operations include: **Add**, **Update**(**Set**), **Get** and **Delete** |
| Technique: | Playbooks for each operation:   * Add a voip security profile   - hosts: fortimanager01  connection: httpapi  vars:  adom: "root"  ansible\_httpapi\_use\_ssl: True  ansible\_httpapi\_validate\_certs: False  ansible\_httpapi\_port: 443  tasks:  - name: 'add a seurity profile voip entry'  fmgr\_pm\_config\_adom\_adom\_obj\_voip\_profile:  method: 'add'  url\_params:  adom: 'root'  params:  - data:  - name: 'secprof\_voip0'  comment: 'A voip security profile created by Ansible'   * Update a voip security profile   - hosts: fortimanager01  connection: httpapi  vars:  adom: "root"  ansible\_httpapi\_use\_ssl: True  ansible\_httpapi\_validate\_certs: False  ansible\_httpapi\_port: 443  tasks:  - name: 'update a seurity profile voip entry'  fmgr\_pm\_config\_adom\_adom\_obj\_voip\_profile:  method: 'update'  url\_params:  adom: 'root'  params:  - data:  - name: 'secprof\_voip0'  comment: 'A voip security profile updated by Ansible'   * Get a voip security profile using filter   - hosts: fortimanager01  connection: httpapi  vars:  adom: "root"  ansible\_httpapi\_use\_ssl: True  ansible\_httpapi\_validate\_certs: False  ansible\_httpapi\_port: 443  tasks:  - name: 'get a seurity profile voip entry'  fmgr\_pm\_config\_adom\_adom\_obj\_voip\_profile:  method: 'get'  url\_params:  adom: 'root'  params:  - filter:  - 'name'  - '=='  - 'secprof\_voip0'   * Delete a voip security profile   - hosts: fortimanager01  connection: httpapi  vars:  adom: "root"  ansible\_httpapi\_use\_ssl: True  ansible\_httpapi\_validate\_certs: False  ansible\_httpapi\_port: 443  tasks:  - name: 'delete a seurity profile voip entry'  fmgr\_pm\_config\_adom\_adom\_obj\_voip\_profile\_profile:  method: 'delete'  url\_params:  adom: 'root'  profile: 'secprof\_voip0'  **the playbooks list:**   * secprof\_voip\_add.yml * secprof\_voip\_delete.yml * secprof\_voip\_get.yml * secprof\_voip\_update.yml |
| Validation: | * ansible-playbook -i hosts/ secprof\_voip\_add.yml –vvv   "response": [  0,  {  "status": {  "code": 0,  "message": "OK"  },  "url": "/pm/config/adom/root/obj/voip/profile"  }  ]   * ansible-playbook -i hosts/ secprof\_voip\_update.yml –vvv   "response": [  0,  {  "status": {  "code": 0,  "message": "OK"  },  "url": "/pm/config/adom/root/obj/voip/profile"  }  ]   * ansible-playbook -i hosts/ secprof\_voip\_get.yml –vvv   "response": [  0,  [  {  "comment": "A voip security profile updated by Ansible",  "name": "secprof\_voip0",  "sccp": {  "block-mcast": "disable",  "log-call-summary": "disable",  "log-violations": "disable",  "max-calls": 0,  "status": "enable",  "verify-header": "disable"  },  "sip": {  "ack-rate": 0,  "block-ack": "disable",  "block-bye": "disable",  "block-cancel": "disable",  "block-geo-red-options": "disable",  "block-info": "disable",  "block-invite": "disable",  "block-long-lines": "enable",  "block-message": "disable",  "block-notify": "disable",  "block-options": "disable",  "block-prack": "disable",  "block-publish": "disable",  "block-refer": "disable",  "block-register": "disable",  "block-subscribe": "disable",  "block-unknown": "enable",  "block-update": "disable",  "bye-rate": 0,  "call-keepalive": 0,  "cancel-rate": 0,  "contact-fixup": "enable",  "hnt-restrict-source-ip": "disable",  "hosted-nat-traversal": "disable",  "info-rate": 0,  "invite-rate": 0,  "ips-rtp": "enable",  "log-call-summary": "enable",  "log-violations": "disable",  "malformed-header-allow": "pass",  "malformed-header-call-id": "pass",  "malformed-header-contact": "pass",  "malformed-header-content-length": "pass",  "malformed-header-content-type": "pass",  "malformed-header-cseq": "pass",  "malformed-header-expires": "pass",  "malformed-header-from": "pass",  "malformed-header-max-forwards": "pass",  "malformed-header-p-asserted-identity": "pass",  "malformed-header-rack": "pass",  "malformed-header-record-route": "pass",  "malformed-header-route": "pass",  "malformed-header-rseq": "pass",  "malformed-header-sdp-a": "pass",  "malformed-header-sdp-b": "pass",  "malformed-header-sdp-c": "pass",  "malformed-header-sdp-i": "pass",  "malformed-header-sdp-k": "pass",  "malformed-header-sdp-m": "pass",  "malformed-header-sdp-o": "pass",  "malformed-header-sdp-r": "pass",  "malformed-header-sdp-s": "pass",  "malformed-header-sdp-t": "pass",  "malformed-header-sdp-v": "pass",  "malformed-header-sdp-z": "pass",  "malformed-header-to": "pass",  "malformed-header-via": "pass",  "malformed-request-line": "pass",  "max-body-length": 0,  "max-dialogs": 0,  "max-idle-dialogs": 0,  "max-line-length": 998,  "message-rate": 0,  "nat-trace": "enable",  "no-sdp-fixup": "disable",  "notify-rate": 0,  "open-contact-pinhole": "enable",  "open-record-route-pinhole": "enable",  "open-register-pinhole": "enable",  "open-via-pinhole": "disable",  "options-rate": 0,  "prack-rate": 0,  "preserve-override": "disable",  "provisional-invite-expiry-time": 210,  "publish-rate": 0,  "refer-rate": 0,  "register-contact-trace": "disable",  "register-rate": 0,  "rfc2543-branch": "disable",  "rtp": "enable",  "ssl-mode": "off",  "ssl-pfs": "allow",  "status": "enable",  "strict-register": "enable",  "subscribe-rate": 0,  "unknown-header": "pass",  "update-rate": 0  }  }  ]  ]   * ansible-playbook -i hosts/ secprof\_voip\_delete.yml –vvv   "response": [  0,  {  "status": {  "code": 0,  "message": "OK"  },  "url": "/pm/config/adom/root/obj/voip/profile/secprof\_voip0"  }  ]   * run it one more time to make sure it indeed is deleted.   ansible-playbook -i hosts/ secprof\_voip\_delete.yml –vvv  "status": {  "code": -3,  "message": "Object does not exist"  } |
| Completion Criteria: | success. |
| Special Considerations: |  |

### Security Profile WAF Test

|  |  |
| --- | --- |
| Test Objective: | Test The Basic operations with FortiManager Security Profile Web Application Firewall(WAF)  Operations include: **Add**, **Update**(**Set**), **Get** and **Delete** |
| Technique: | Playbooks for each operation:   * Add a waf security profile   - hosts: fortimanager01  connection: httpapi  vars:  adom: "root"  ansible\_httpapi\_use\_ssl: True  ansible\_httpapi\_validate\_certs: False  ansible\_httpapi\_port: 443  tasks:  - name: 'add a seurity profile WAF entry'  fmgr\_pm\_config\_adom\_adom\_obj\_waf\_profile:  method: 'add'  url\_params:  adom: 'root'  params:  - data:  - name: 'secprof\_waf0'  comment: 'A seurity profile waf entry created by Ansible'  external: 'disable'  extended-log: 'enable'   * Update a waf security profile   - hosts: fortimanager01  connection: httpapi  vars:  adom: "root"  ansible\_httpapi\_use\_ssl: True  ansible\_httpapi\_validate\_certs: False  ansible\_httpapi\_port: 443  tasks:  - name: 'Update a seurity profile WAF entry'  fmgr\_pm\_config\_adom\_adom\_obj\_waf\_profile:  method: 'update'  url\_params:  adom: 'root'  params:  - data:  - name: 'secprof\_waf0'  comment: 'A seurity profile waf entry updated by Ansible'  external: 'enable'  extended-log: 'disable'   * Get a waf security profile using filter   - hosts: fortimanager01  connection: httpapi  vars:  adom: "root"  ansible\_httpapi\_use\_ssl: True  ansible\_httpapi\_validate\_certs: False  ansible\_httpapi\_port: 443  tasks:  - name: 'Get a seurity profile WAF entry'  fmgr\_pm\_config\_adom\_adom\_obj\_waf\_profile:  method: 'get'  url\_params:  adom: 'root'  params:  - filter:  - 'name'  - '=='  - 'secprof\_waf0'   * Delete a waf security profile   - hosts: fortimanager01  connection: httpapi  vars:  adom: "root"  ansible\_httpapi\_use\_ssl: True  ansible\_httpapi\_validate\_certs: False  ansible\_httpapi\_port: 443  tasks:  - name: 'Delete a seurity profile WAF entry'  fmgr\_pm\_config\_adom\_adom\_obj\_waf\_profile\_profile:  method: 'delete'  url\_params:  adom: 'root'  profile: 'secprof\_waf0'  **the playbooks list:**   * secprof\_waf\_add.yml * secprof\_waf\_delete.yml * secprof\_waf\_get.yml * secprof\_waf\_update.yml |
| Validation: | * ansible-playbook -i hosts/ secprof\_waf\_add.yml –vvv   "response": [  0,  {  "status": {  "code": 0,  "message": "OK"  },  "url": "/pm/config/adom/root/obj/waf/profile"  }  ]   * ansible-playbook -i hosts/ secprof\_waf\_update.yml –vvv   "response": [  0,  {  "status": {  "code": 0,  "message": "OK"  },  "url": "/pm/config/adom/root/obj/waf/profile"  }  ]   * ansible-playbook -i hosts/ secprof\_waf\_get.yml –vvv   "response": [  0,  [  {  "address-list": null,  "comment": "A seurity profile waf entry updated by Ansible",  "constraint": null,  "extended-log": "disable",  "external": "enable",  "method": null,  "name": "secprof\_web\_filter0",  "signature": null,  "url-access": null  }  ]  ]   * ansible-playbook -i hosts/ secprof\_waf\_delete.yml –vvv   "response": [  0,  {  "status": {  "code": 0,  "message": "OK"  },  "url": "/pm/config/adom/root/obj/waf/profile/secprof\_web\_filter0"  }  ]   * run it one more time to make sure it indeed is deleted.   ansible-playbook -i hosts/ secprof\_waf\_delete.yml –vvv  "status": {  "code": -3,  "message": "Object does not exist"  } |
| Completion Criteria: | success. |
| Special Considerations: |  |

### Security Profile Wanopt Test

|  |  |
| --- | --- |
| Test Objective: | Test The Basic operations with FortiManager Security Profile WANOPT  Operations include: **Add**, **Update**(**Set**), **Get** and **Delete** |
| Technique: | Playbooks for each operation:   * Add a wanopt security profile   - hosts: fortimanager01  connection: httpapi  vars:  adom: "root"  ansible\_httpapi\_use\_ssl: True  ansible\_httpapi\_validate\_certs: False  ansible\_httpapi\_port: 443  tasks:  - name: 'add a seurity profile wanopt entry'  fmgr\_pm\_config\_adom\_adom\_obj\_wanopt\_profile:  method: 'add'  url\_params:  adom: 'root'  params:  - data:  - name: 'secprof\_wanopt0'  comments: 'A seurity profile wanopt entry created by Ansible'  transparent: 'disable'   * Update a wanopt security profile   - hosts: fortimanager01  connection: httpapi  vars:  adom: "root"  ansible\_httpapi\_use\_ssl: True  ansible\_httpapi\_validate\_certs: False  ansible\_httpapi\_port: 443  tasks:  - name: 'Update a seurity profile wanopt entry'  fmgr\_pm\_config\_adom\_adom\_obj\_wanopt\_profile:  method: 'update'  url\_params:  adom: 'root'  params:  - data:  - name: 'secprof\_wanopt0'  comments: 'A seurity profile wanopt entry updated by Ansible'  transparent: 'enable'   * Get a wanopt security profile using filter   - hosts: fortimanager01  connection: httpapi  vars:  adom: "root"  ansible\_httpapi\_use\_ssl: True  ansible\_httpapi\_validate\_certs: False  ansible\_httpapi\_port: 443  tasks:  - name: 'get a seurity profile wanopt entry'  fmgr\_pm\_config\_adom\_adom\_obj\_wanopt\_profile:  method: 'get'  url\_params:  adom: 'root'  params:  - filter:  - 'name'  - '=='  - 'secprof\_wanopt0'   * Delete a wanopt security profile   - hosts: fortimanager01  connection: httpapi  vars:  adom: "root"  ansible\_httpapi\_use\_ssl: True  ansible\_httpapi\_validate\_certs: False  ansible\_httpapi\_port: 443  tasks:  - name: 'delete a seurity profile wanopt entry'  fmgr\_pm\_config\_adom\_adom\_obj\_wanopt\_profile\_profile:  method: 'delete'  url\_params:  adom: 'root'  profile: 'secprof\_wanopt0'  **the playbooks list:**   * secprof\_wanopt\_add.yml * secprof\_wanopt\_delete.yml * secprof\_wanopt\_get.yml * secprof\_wanopt\_update.yml |
| Validation: | * ansible-playbook -i hosts/ secprof\_wanopt\_add.yml –vvv   "response": [  0,  {  "status": {  "code": 0,  "message": "OK"  },  "url": "/pm/config/adom/root/obj/wanopt/profile"  }  ]   * ansible-playbook -i hosts/ secprof\_wanopt\_update.yml –vvv   "response": [  0,  {  "status": {  "code": 0,  "message": "OK"  },  "url": "/pm/config/adom/root/obj/wanopt/profile"  }  ]   * ansible-playbook -i hosts/ secprof\_wanopt\_get.yml –vvv   "response": [  0,  [  {  "auth-group": [],  "cifs": {  "byte-caching": "enable",  "log-traffic": "enable",  "port": [  445  ],  "prefer-chunking": "fix",  "secure-tunnel": "disable",  "status": "disable",  "tunnel-sharing": "private"  },  "comments": "A seurity profile wanopt entry updated by Ansible",  "ftp": {  "byte-caching": "enable",  "log-traffic": "enable",  "port": [  21  ],  "prefer-chunking": "fix",  "secure-tunnel": "disable",  "status": "disable",  "tunnel-sharing": "private"  },  "http": {  "byte-caching": "enable",  "log-traffic": "enable",  "port": [  80  ],  "prefer-chunking": "fix",  "secure-tunnel": "disable",  "ssl": "disable",  "status": "disable",  "tunnel-non-http": "disable",  "tunnel-sharing": "private",  "unknown-http-version": "tunnel"  },  "mapi": {  "byte-caching": "enable",  "log-traffic": "enable",  "port": [  135  ],  "secure-tunnel": "disable",  "status": "disable",  "tunnel-sharing": "private"  },  "name": "secprof\_wanopt0",  "tcp": {  "byte-caching": "disable",  "byte-caching-opt": "mem-only",  "log-traffic": "enable",  "port": "1-65535",  "secure-tunnel": "disable",  "ssl": "disable",  "status": "disable",  "tunnel-sharing": "private"  },  "transparent": "enable"  }  ]  ]   * ansible-playbook -i hosts/ secprof\_wanopt\_delete.yml –vvv   "response": [  0,  {  "status": {  "code": 0,  "message": "OK"  },  "url": "/pm/config/adom/root/obj/wanopt/profile/secprof\_wanopt0"  }  ]   * run it one more time to make sure it indeed is deleted.   ansible-playbook -i hosts/ secprof\_wanopt\_delete.yml –vvv  "status": {  "code": -3,  "message": "Object does not exist"  } |
| Completion Criteria: | success. |
| Special Considerations: |  |

### Device Profile Test

|  |  |
| --- | --- |
| Test Objective: | Test The Basic operations with FortiManager Device Profile  Operations include: **Add**, **Update**(**Set**), **Get** and **Delete** |
| Technique: | Playbooks for each operation:   * Add a device profile   - hosts: fortimanager01  connection: httpapi  vars:  adom: "root"  ansible\_httpapi\_use\_ssl: True  ansible\_httpapi\_validate\_certs: False  ansible\_httpapi\_port: 443  tasks:  - name: 'add a device profile entry'  fmgr\_pm\_devprof\_adom\_adom:  method: 'add'  url\_params:  adom: 'root'  params:  - data:  - name: 'device\_profile0'  description: 'a device profile created by Ansible'  type: 'devprof'   * Get all device profiles (no filter supported)   - hosts: fortimanager01  connection: httpapi  vars:  adom: "root"  ansible\_httpapi\_use\_ssl: True  ansible\_httpapi\_validate\_certs: False  ansible\_httpapi\_port: 443  tasks:  - name: 'get a device profile entry'  fmgr\_pm\_devprof\_adom\_adom:  method: 'get'  url\_params:  adom: 'root'   * Delete a device profile   - hosts: fortimanager01  connection: httpapi  vars:  adom: "root"  ansible\_httpapi\_use\_ssl: True  ansible\_httpapi\_validate\_certs: False  ansible\_httpapi\_port: 443  tasks:  - name: 'delete a device profile entry'  fmgr\_pm\_devprof\_adom\_adom\_pkg\_path:  method: 'delete'  url\_params:  adom: 'root'  pkg\_path: 'device\_profile0'  **the playbooks list:**   * devprof\_add.yml * devprof\_delete.yml * devprof\_get.yml |
| Validation: | * ansible-playbook -i hosts/ devprof\_add.yml –vvv   "response": [  0,  {  "status": {  "code": 0,  "message": "OK"  },  "url": "/pm/devprof/adom/root"  }  ]   * ansible-playbook -i hosts/ devprof\_get.yml –vvv   "response": [  0,  [  {  "description": "",  "enabled options": [  "dns",  "ntp",  "email",  "admin",  "snmp",  "repmsg",  "ftgd",  "log"  ],  "name": "default",  "oid": 1306,  "type": "devprof"  },  {  "description": "",  "enabled options": [  "dns",  "ntp",  "email",  "admin",  "snmp",  "repmsg",  "ftgd",  "log"  ],  "name": "device\_profile0",  "oid": 1349,  "type": "devprof"  }  ]  ]   * ansible-playbook -i hosts/ devprof\_delete.yml –vvv   "response": [  0,  {  "status": {  "code": 0,  "message": "OK"  },  "url": "/pm/devprof/adom/root/device\_profile0"  }  ]   * run it one more time to make sure it indeed is deleted.   ansible-playbook -i hosts/ devprof\_delete.yml –vvv  "status": {  "code": -3,  "message": "Object does not exist"  } |
| Completion Criteria: | success. |
| Special Considerations: |  |

### Provisioning Template Test

|  |  |
| --- | --- |
| Test Objective: | Test The Basic operations with FortiManager Device Profile Template:  Tests include:   * DNS setting * Alert Main setting * NTP setting * Admin Setting * SNMP v1/v2 setting * SNMP v3 setting   Operations include: **Set**, **Get** and **Delete** |
| Technique: | Playbooks for each operation:   * Add and set the device profile template   - hosts: fortimanager01  connection: httpapi  vars:  adom: "root"  devprof\_name: 'device\_profile1'  ansible\_httpapi\_use\_ssl: True  ansible\_httpapi\_validate\_certs: False  ansible\_httpapi\_port: 443  tasks:  - name: 'create a device profile entry'  fmgr\_pm\_devprof\_adom\_adom:  method: 'add'  url\_params:  adom: 'root'  params:  - data:  - name: '{{ devprof\_name }}'  type: 'devprof'  - name: 'modify the dns setting'  fmgr\_pm\_config\_adom\_adom\_devprof\_devprof\_system\_dns:  method: 'set'  url\_params:  adom: '{{ adom }}'  devprof: '{{ devprof\_name }}'  params:  - data:  domain: 'foo.bar.baz'  primary: '4.4.4.4'  secondary: '8.8.8.8'  - name: 'modify the alert mail'  fmgr\_pm\_config\_adom\_adom\_devprof\_devprof\_system\_email\_server:  method: 'set'  url\_params:  adom: '{{ adom }}'  devprof: '{{ devprof\_name }}'  params:  - data:  username: 'USERNAME'  password:  - 'PASSWORD'  authenticate: 'enable'  server: 'smtp.foo.bar.baz'  - name: 'modify the ntp setting'  fmgr\_pm\_config\_adom\_adom\_devprof\_devprof\_system\_ntp:  method: 'set'  url\_params:  adom: '{{ adom }}'  devprof: '{{ devprof\_name }}'  params:  - data:  ntpsync: 'enable'  ntpserver:  - authentication: 'disable'  server: 'ntp.foo.bar.baz'  type: 'custom'  - name: 'modify the admin setting'  fmgr\_pm\_config\_adom\_adom\_devprof\_devprof\_system\_global:  method: 'set'  url\_params:  adom: '{{ adom }}'  devprof: '{{ devprof\_name }}'  params:  - data:  gui-theme: 'melongene'  language: 'spanish'  switch-controller: 'enable'  admin-scp: 'enable'  admin-https-redirect: 'disable'  - name: 'modify the snmp v1/v2 community entry'  fmgr\_pm\_config\_adom\_adom\_devprof\_devprof\_system\_snmp\_community:  method: 'set'  url\_params:  adom: '{{ adom }}'  devprof: '{{ devprof\_name }}'  params:  - data:  - name: 'community\_name0'  status: 'enable'  events:  - 'cpu-high'  - 'mem-low'  - name: 'modify the snmp v3 user entry'  fmgr\_pm\_config\_adom\_adom\_devprof\_devprof\_system\_snmp\_user:  method: 'set'  url\_params:  adom: '{{ adom }}'  devprof: '{{ devprof\_name }}'  params:  - data:  - name: 'user\_name0'  status: 'enable'  events:  - 'intf-ip'  - 'log-full'   * Delete the profile and all the template for it.   **\* The same step with 3.2.26.**  **the playbooks list:**   * devprof\_template\_setup.yml |
| Validation: | * ansible-playbook -i hosts/ devprof\_template\_setup.yml –vvv   **\* validate from UI** |
| Completion Criteria: | success. |
| Special Considerations: |  |

### [template]

|  |  |
| --- | --- |
| Test Objective: | Test The Basic operations with FortiManager Security Profile IPS  Operations include: **Add**, **Update**(**Set**), **Get** and **Delete** |
| Technique: | Playbooks for each operation:   * Add a se * Update a firewall package * Get a firewall packages using filter * Delete a firewall package   **the playbooks list:**   * firewall\_package\_add.yml * firewall\_package\_delete.yml * firewall\_package\_get.yml |
| Validation: | * ansible-playbook -i hosts/ firewall\_package\_add.yml –vvv * ansible-playbook -i hosts/ firewall\_policy\_add.yml –vvv * ansible-playbook -i hosts/ firewall\_package\_get.yml –vvv * ansible-playbook -i hosts/ firewall\_package\_delete.yml –vvv * run it one more time to make sure it indeed is deleted.   ansible-playbook -i hosts/ firewall\_package\_delete.yml –vvv  "status": {  "code": -3,  "message": "Object does not exist"  } |
| Completion Criteria: | success. |
| Special Considerations: |  |

## User Interface Testing

## Performance Profiling

## Load Testing

## Stress Testing

## Security and Access Control Testing

## Failover and Recovery Testing

# Resources

## Workers

N/A

## System Resources

N/A

# Deliverables

N/A

# Appendix A: Sample Full Terraform Settings