Sign-in

Our story

Robotics Media

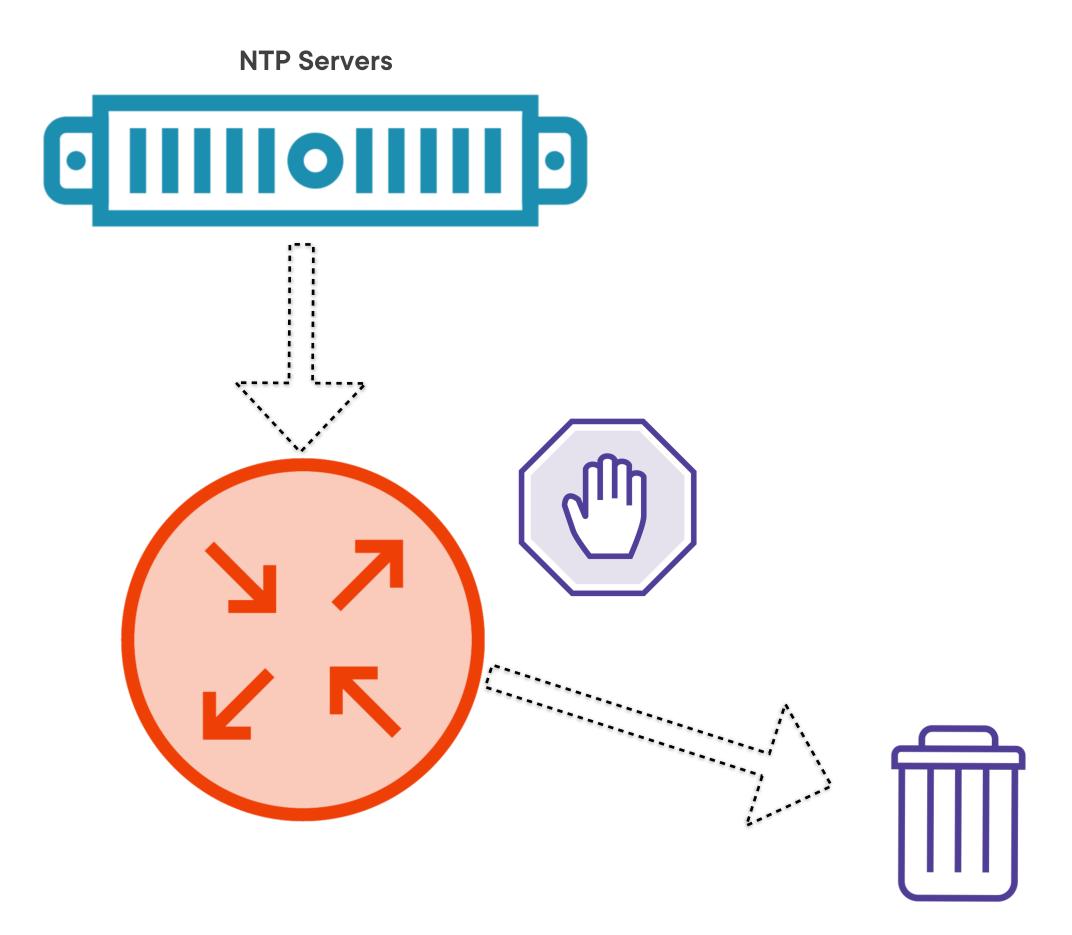
Support

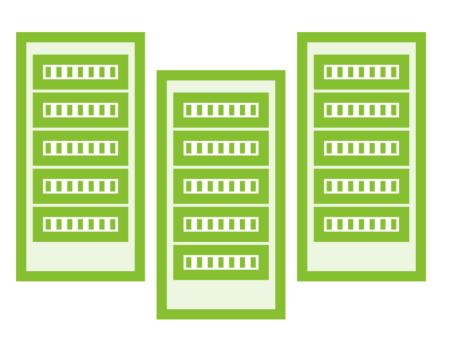
ROBOTICS ALSO A STORY OF LIFE

Life evolves, robotics are no different.









Globomantics Servers

```
- name: configure interface settings
  cisco.ios.ios_config:
    lines:
    - description test interface
    - ip address 172.31.2.1 255.255.255.0
    parents: interface Ethernet3
```

Configuration Management with Config Modules

- Device native commands are supported
- Relies heavily on network expertise

```
- name: render a Jinja2 template
  cisco.ios.ios_config:
    backup: yes
    src: snmp_template.j2
```

Configuration Management with Jinja Templating

- Uses variables, looping, conditionals, etc.
- Implementation and maintenance might be challenging

```
- name: Make sure VLAN configuration is updated.
  cisco.ios.ios_vlans:
    config: "{{ vlans }}"
    state: merged
```

Configuration Management with Resource Modules

- Uses structured data
- Resource Module: Specific network function mapped to a single Ansible module
- There is a one-to-one mapping between facts and resource modules
- Named according to the platform OS and the resource involved, ie: junos_interfaces, ios_interfaces, etc.

Network Facts

- Most network devices accept native device commands only
- Resource modules and corresponding facts bridge between structured data and native device configuration
- Enhanced facts modules can gather device configuration as structured data

Demo

Gathering Facts from Network Devices

```
#YAML vars file
interface_ip_addresses:
- ipv4:
  - address: 10.10.4.2 255.255.255.252
  name: GigabitEthernet1
- ipv4:
  - address: 10.10.6.3 255.255.255.0
  name: GigabitEthernet2
#Task definition
- name: Make sure VLAN configuration is updated.
  cisco.ios.ios_vlans:
    config: "{{ vlans }}"
    state: merged
```

Network Resource Modules

- Each module specializes in configuring a separate network function
- 3 possible parameters: config, running_config, state

```
#Task definition
- name: Replace module attributes of given access-groups
cisco.ios.ios_acl_interfaces:
    config:
        - name: GigabitEthernet0/1
        access_groups:
        - afi: ipv4
        acls:
        - name: 100
            direction: out
        - name: 110
            direction: in
        state: replaced
```

Config Parameter

- Requires structured data
- Dictionary or a list of dictionaries

```
#acl_to_parse.cfg
ip access list extended outbound_acl
    15 permit ip host 192.0.2.15 any
ip access list extended inbound_acl
    10 permit ip 10.1.1.0 0.0.0.255 20.1.1.0 0.0.0.255
    20 permit tcp 10.2.2.0 0.0.0.255 20.2.2.0 0.0.0.255 eq www

#Task definition
- name: Replace module attributes of given access-groups
    cisco.ios.ios_acl_interfaces:
    running_config: "{{ lookup('file', 'acl_to_parse.cfg') }}"
    state: parced
```

Running_config Parameter

- Accepts native device commands
- Only used when parsing device config into structured data

State Parameter

Determines the action to be taken by the module

Action states:

- Merged
- Replaced
- Overridden
- Deleted

Non-action states:

- Gathered
- Rendered
- Parsed

Demo

Retrieving Network Configuration as Structured Data

Demo

Configuring Globomantics Router with Network Resource Modules

Summary

Network resource modules use structured data

Network facts modules gather device config as structured data

Each resource module has a corresponding fact

Resource modules take action based on the state parameter

Non-action states do not alter the device configuration.

Action states are used to update device configuration