Ansible Tutorial

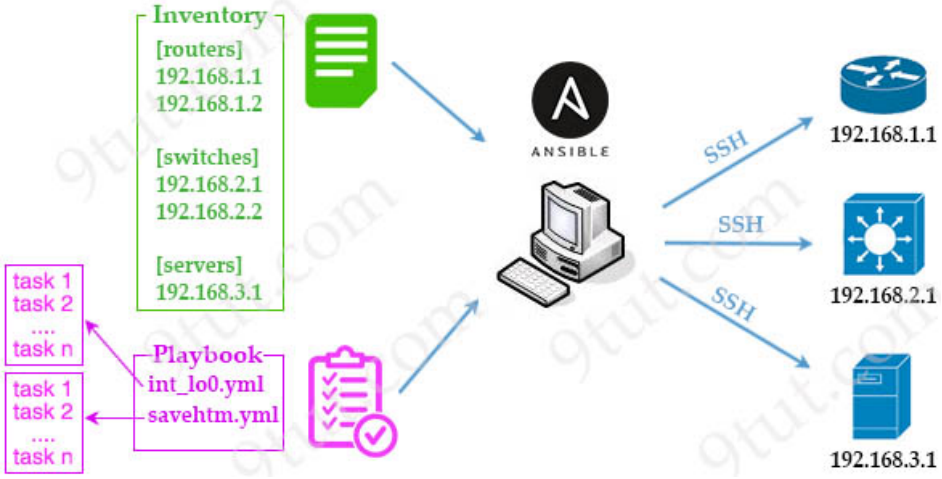
November 23rd, 2020 Go to comments

There are several automation tools available to make configuration management easier: Ansible, Chef, Puppet… The goal of these tools is to reduce the complexity and time to configure and maintain networks (especially big ones with hundreds of devices). In this tutorial we will learn some basic knowledge of Ansible in the scope of CCNA level.

Ansible uses an agentless architecture to manage network devices. Agentless means that the managed device does not need any code (agent) to be installed on it. Therefore Ansible uses SSH (NETCONF over SSH in particular) to "push" changes and extract information to managed devices.

Once Ansible is installed, it creates several text files:

+ **Playbooks**: These files provide actions and logic about what Ansible should do. Ansible playbooks are files that contain tasks to configure hosts. Ansible playbooks are written in YAML format.
+ **Inventory**: a file contains a list of the hosts (usually their IP addresses, ports) which you want to configure or manage. Hosts in an inventory can be divided into smaller groups for easier management and configuration. Each group can run different tasks. An example of a task is to ping all hosts in group [routers].
+ **Templates**: Using Jinja2 language, the templates represent a device's configuration but with variables.
+ **Variables**: Using YAML, a file can list variables that Ansible will substitute into templates.



Templates and variables are optional so they are not discussed here to keep this tutorial simple. An inventory and playbook are enough to run our first Ansible program! (in fact, only a playbook is enough to run). For example if we have an inventory named "hosts" (without file extension) and a playbook named "int_lo0.yml" (to configure loopback 0 interface for each host) in "playbooks" directory then we can run them via this command:

```
$ ansible-playbook -i hosts playbooks/int_lo0.yml
```

Another example of the "hosts" inventory and "command_ios.yml" playbook is shown below:

"hosts" Inventory

```
[ios_devices]
R1 ansible_host=192.168.1.10
R2 ansible_host=192.168.1.11

[ios_devices:vars]
username=9tut
password=mySecretPassword!
```

"command_ios.yml" Playbook

```
---
- name: IOS Show Commands
hosts: "ios_devices"
gather_facts: false
connection: local

vars:
  cli:
    host: "{{ ansible_host }}"
    username: "{{ username }}"
    password: "{{ password }}"
    transport: cli

tasks:
```

```
- name: ios show commands
  ios_command:
    commands:
      - show version | i IOS
      - show run | i hostname
    provider: "{{ cli }}"

  register: output

  - name: show output of IOS
    debug:
      var: output
```

The above playbook would display “show version” and “show run” output when we run it with command:

```
$ansible-playbook -i hosts command_ios.yml
```

And the result is shown below:

```
PLAY [IOS Show Commands] *******
TASK [ios show commands] ********************************************************
ok: [ios-r1]
ok: [ios-r2]

TASK [show output of IOS]
*************************
ok: [ios-r1] => {
    "output": {
        "changed": false,
        "stdout": [
             "Cisco IOS Software, IOSv Software (VIOS-ADVENTERPRISEK9-M), Version 15.6(3)M2, RELEASE SOFTWARE (fc2)
ROM: Bootstrap program is IOSv\nCisco IOSv (revision 1.0) with with 460033K/62464K bytes o£ memory.",
              "hostname iosv-1"
        ],
        "stdoutlines": [
              [
                   "Cisco IOS Software, IOSv Software (VIOS-ADVENTERPRISEK9-M), Version 15.6(3)M2, RELEASE SOPTWARE (fc2)",
                   "ROM: Bootstrap program is IOSv", "Cisco IOSv (revision 1.0) with 460033K/62464K bytes of memory."
              ],
              [
                   "hostname iosv-1"
              ]
        ],
        "warnings": []
    }
}
ok: [ios-r2] => {
    "output": {
         "changed": false,
         "stdout": [
             "Cisco IOS Software, IOSv Software (VIOS-ADVENTERPRISEK9-M), Version 15.6(3)M2, RELEASE SOFTWARE (fc2)
ROM: Bootstrap program is IOSv\nCisco IOSv (revision 1.0) with 460033K/62464K bytes of memory.",
             "hostname iosv-2"
         ],
         "stdout_lines": [
             [
                  "Cisco IOS Software, IOSv Software (VIOS-ADVENTERPRISEK9-M), Version 15.6(3)M2, RELEASE SOFTWARE (fc2)",
"ROM: Bootstrap program is IOSv”, "Cisco IOSv (revision 1.0) with with 460033K/62464K bytes of memory."
             ],
             [
                  "hostname iosv-2"
             ]
         ],
         "warnings": []
     }
}

PLAY RECAP ********************************************************************
ios-r1 : ok=2 changed=0 unreachable=0 failed=0
ios-r2 : ok=2 changed=0 unreachable=0 failed=0
```

In summary, please remember the following important facts about Ansible:
+ Use “push” model (push configuration from a centralized server to end devices)
+ Use SSH (TCP port 22) for remote communication
+ Use YAML for device configuration
+ Files needed for operation: Playbook, Inventory…
+ Ansible requires a Linux-based system to run. Though it can run under the Windows Subsystem for Linux but it should not be used for production systems

We also made a comparison list of Ansible, Puppet and Chef automation tool here:

Criteria	Ansible	Puppet	Chef
Configuration Language	YAML, Python	+ Puppet DSL + Embedded Ruby	Ruby DSL, JSON
Architecture	Agentless (Client only)	Both (Agentless & Agent Based)	Agent Based (Client-Server)
Deployment Method	Push Model	Pull Model	Pull Model
Files created before Operation	Playbook	Manifest	Recipe
Availability	Ansible Primary Instance	Puppet Master	Chef Master
How to manage devices	Any device (can become controller)	Puppet Master	Chef Master
Installation	easiest	medium	hard
Transport Mechanism	SSH/NETCONF	REST	REST
Port used	TCP port 22	TCP port 8140	TCP port 10002
Initial Release	2012	2005	2009

Comments (6) Comments

1. sc
   January 12th, 2021

   I want to setup a GNS3 lab to run Ansible. Anyone have a good resource on getting an Ansible IOS/installing it on GNS3?

2. Afrikan_CCNP
   January 13th, 2021

   @SC
   learn how to use eve-ng is better
   https://www.eve-ng.net/

3. Russian_CCNP
   April 4th, 2021

   Doesn't agree with "better" (without any explanation / description): GNS3 has a lot of functionalities now (docker, VM integration… etc…) and it is totally free while, with eve-ng to get some features, you have to go to eve-ng pro which is around $120,-

4. MariaMashaBabko
   April 8th, 2021

   use Pnetlab, the pirate version of EVE-NG PRO, is incredible!

5. biggz3307
   August 15th, 2021

   where are you all finding images to run on eve-ng?

6. Akuma
   January 14th, 2022

   sc: if using windows 10, you can install linux WSL and run Ansible, I did once, config a test network on gns3 and connect to the cloud

Add a Comment

Name

Submit Comment

Subscribe to comments feed

Premium Member Zone

Welcome Gurjeet singh!

- Welcome Premium Member
- CCNA – New Questions Part 5
- CCNA – New Questions Part 6
- CCNA – New Questions Part 7
- CCNA – New Questions Part 8
- CCNA – New Questions Part 9
- Composite Quizzes

- Logout

CCNA 200-301

- Basic Questions
- Topology Architecture Questions
- Cloud & Virtualization Questions
- CDP & LLDP Questions
- Switch Questions
- VLAN & Trunking Questions
- VLAN & Trunking Questions 2
- STP & VTP Questions
- EtherChannel Questions
- TCP & UDP Questions
- IP Address & Subnetting Questions
- IP Routing Questions
- IP Routing Questions 2
- OSPF Questions
- OSPF Questions 2
- EIGRP Questions
- NAT Questions
- NTP Questions
- Syslog Questions
- HSRP Questions
- Access-list Questions
- AAA Questions
- Security Questions
- Security Questions 2
- DAI Questions
- IPv6 Questions
- DNS Questions
- QoS Questions
- Port Security Questions
- Wireless Questions
- Wireless Questions 2
- SDN Questions
- DNA Center Questions
- Drag Drop Questions
- Drag Drop Questions 2
- Drag Drop Questions 3
- VPN Questions
- DHCP Questions
- Automation Questions
- Miscellaneous Questions
- CCNA FAQs & Tips
- Share your new CCNA Experience

CCNA Self-Study

- Practice CCNA GNS3 Labs
- CCNA Knowledge
- CCNA Lab Challenges
- Puppet Tutorial
- Chef Tutorial
- Ansible Tutorial
- JSON Tutorial
- Layer 2 Threats and Security Features
- AAA TACACS+ and RADIUS Tutorial

- STP Root Port Election Tutorial
- GRE Tunnel Tutorial
- Basic MPLS Tutorial
- TCP and UDP Tutorial
- Border Gateway Protocol BGP Tutorial
- Point to Point Protocol (PPP) Tutorial
- WAN Tutorial
- DHCP Tutorial
- Simple Network Management Protocol SNMP Tutorial
- Syslog Tutorial
- Gateway Load Balancing Protocol GLBP Tutorial
- EtherChannel Tutorial
- Hot Standby Router Protocol HSRP Tutorial
- InterVLAN Routing Tutorial
- Cisco Command Line Interface CLI
- Cisco Router Boot Sequence Tutorial
- OSI Model Tutorial
- Subnetting Tutorial – Subnetting Made Easy
- Frame Relay Tutorial
- Wireless Tutorial
- Virtual Local Area Network VLAN Tutorial
- VLAN Trunking Protocol VTP Tutorial
- IPv6 Tutorial
- Rapid Spanning Tree Protocol RSTP Tutorial
- Spanning Tree Protocol STP Tutorial
- Network Address Translation NAT Tutorial
- Access List Tutorial
- RIP Tutorial
- EIGRP Tutorial
- OSPF Tutorial

Network Resources

- Free Router Simulators
  - CCNA Website
  - ENCOR Website
  - ENSDWI Website
  - ENARSI Website
  - DevNet Website
  - CCIE R&S Website
  - Security Website
  - Wireless Website
  - Design Website
  - Data Center Website
  - Service Provider Website
  - Collaboration Website

Top