International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 05 Issue: 06 | June - 2021 ISSN: 2582-3930

Enhancing network compliance with state awareness using ansible network automation

Sriranjan M[1], Shyam Sunder[2], Dr Radhika KR[3]

[1]Computer Network Engineering, Department of ISE, BMS College of Engineering
[2]Network Specialist, Intel Corporation.
[3]Dr Radhika KR, Department of ISE, BMS College of Engineering

---

Abstract -In the current era of computer networks, management of a network and its devices play a vital role for any Network infrastructure. Network management is one such field where there is a rise in integration of automation to its activities to facilitate continuous delivery. Automating redundant network-management tasks not only overcomes manual error but also provides greater magnitude of efficiency. This project aims to provide solution to such redundant activities of network infrastructure management such as device configuration state auditing, fixing non-compliant devices and leveraging ad-hoc methods for upgrades and security policies. This is achieved by using one of the powerful open source tools, Ansible. A centralized server where ansible is deployed in the network provides the platform to achieve automation of the tasks mentioned. Here the playbook scripts are scheduled to perform the tasks ensuring high reliability. Outcome of this automation is auto-fixing of non-compliant configuration, daily report of couple-devices configuration auditing and ad-hoc platform for revising security policies and upgrade management. Thus, this ensures and enhances network compliance and reliability for continuous delivery

Key Words: Network automation, Ansible, Configuration auditing.

1.INTRODUCTION

This Computer network infrastructure grow and scale up on daily basis. All the nodes of the network are to be managed with automation. Automation of network management ensures maximum efficiency of the operational tasks. Ansible is one such powerful open-source configuration management tool that aids the automation efforts in enhancing the network compliance and provide better reliability. In this project, the primary focus is on enhancing network compliance and reliability by deploying various ansible playbooks. Ansible play books which are scripts written in YAML, control and manage the network nodes. These read, write, and update over any network node with its respective modules. The advantage here is ansible does these operations over ssh and in a command cli mode thus making it platform independent. A single ansible playbook will be running over many devices at a single instance thus taking least amount of time for a configuration fix. Costs are minimized greatly in terms of human resource, time efforts and power.

A simulated network environment built over GNS3 that depicts a real network infrastructure on which ansible is deployed to demonstrate enhancement of network state compliance. A centralized ansible server inside the simulated network where ansible in installed is responsible for managing the device configuration on all devices, auto-fixing of non-compliant configuration using standard config templates. Also, it has the capability to generate the audit report for all the devices to the concerned personal for extended verification. Scheduling these playbooks that manage the network node configuration ensures higher uptime.

2. CHALLENGES

Any IT infrastructure in today's times has a large network of devices. Keeping such large network infrastructure up and managing each node of the network is a bigger challenge. Configuration discrepancy, faulty values, non-compliance, and wrong updates amongst the nodes cause serious problem on overall communication of the network. Managing all such nodes manually is not only tedious task but also has its own risks. Human error can cause a critical impact in managing such nodes. Such problems may put down minor to major operations of the infrastructure which can cause huge losses. These challenges motivate the need of automation of such tasks

3. STATE OF THE ART

Here [1] highlights various techniques used in ansible automation along with Jenkins with pipelining and how to trigger configuration changes. It also provides effective management of Agile based CICD projects.

[2] Demonstrates framework of ansible and its capabilities. Advanced level of scripting fundamentals including service management are presented in this paper that aids for scripting of this project for its use case.

[3] Presents configuring and monitoring any device via automation, irrespective of vendors can be implemented not only on SDN devices, but also on other networking solutions. Python is used as the tool to automate and also there is more focus on the legacy device management along with automation.

4. TABLE OF WORK SO FAR

Work Done	Sacrosanct Knowledge
Dirk Schulz in [4] on integrating network management as a subsystem of an automation system	The designed model is interoperable integrating AutomationML
Gaurav Kumar in [5] on DevOps and its automation	Automating builds, developments, and testing.
Stephen Sun in [6] on python filtering	Effective python filters used to fetch required data
Rohit Kativar in [7] on auto-configuration mechanism for SDN switches	Providing appropriate switch configuration whenever a new SDN switch is detected on the network
Nishanth Kumar Singh in [7] demonstrates automated application deployment with ansible as the orchestration engine	Automating right from environment provisioning to application deployment

5. PROPOSED STUDY

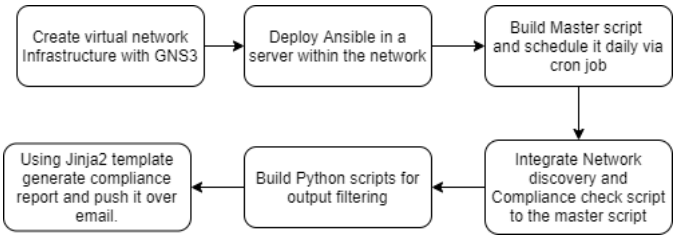

Fig -1: Block diagram of the proposed work

To demonstrate the network management automation using ansible, it is necessary to have a network infrastructure to run the ansible scripts. To aid that, GNS3 is used to create a virtual infrastructure that mimics the real one along with its configuration. Multiple devices have different OS vendors and configurations

Virtual network comprises of router and switches with cross vendor platforms with its respective image. In this network there is a centralized Linux server where ansible is installed. All the playbooks are stored in this server and will be executed against any bunch of devices of the network as required.

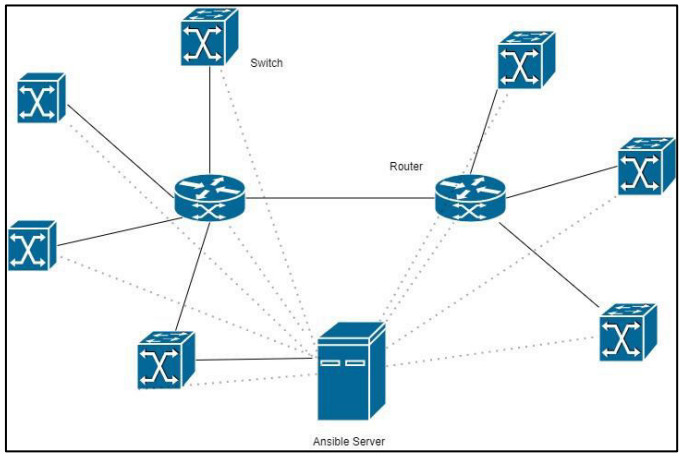

Fig -2: Virtual network infrastructure.

The dotted line indicates that the ansible server connects to any device over ssh while executing any script.

The proposed solution which aims to enhance the network compliance has the following components mentioned below.A centralized Ansible server where ansible is deployed with all the required modules. This server is used to run the ansible playbooks over the bunch of devices mentioned in ansible inventory. The server is a linux platform where all the ansible jobs could be scheduled to run daily using the system cron job.Inventory is the file where all the network nodes are grouped and its stored in a list. This is called while running any playbook.

List of scripts that are scheduled on daily basis:

- Network discovery and access check script:
  This script ensures the ansible server is able to ping all the devices and has the access to beacome password of the device. Any devices that failed to reach or access it will be removed from the inventory and reported.
- Switch Configuration Audit script:
  This script has the following tasks mentioned. First it fetches the device configuration for certain features like spanning tree, ACL etc., Then it is checked against the standard configuration template for any difference. If difference is found it means that the device is non-compliant to the standard configuration hence the auto healing task is called where the configuration is written on the device based on the standard template of that component. The activity report is sent over mail.
- Couple-devices audit script.:
  There are certain couple switches in the network where one switch acts as backbone to another thus auditing configuration features of such couple device is necessary, thus this script performs such audit and provides an html report of the same.
- A master script that calls all of these individual scripts. This script is scheduled to run on daily basis.

```
master.yml
---
- gather_facts: false
  hosts: localhost
  name: "====================== Starting Daily auditing ============
  tasks:

    #Calling Network discovery scripts
    - name: "Invoke Network discovery and access check script"
      command: "ansible-playbook network_discovery.yml -i inventory"
      delegate_to: localhost
      ignore_errors: yes


    - name: "Invoke Switch compliance audit and auto fixscript"
      command: "ansible-playbook switch_compliance.yml -i inventory -DC
      delegate_to: localhost
      ignore_errors: yes

    - name: "Invoke switch couple audit script"
      command: "ansible-playbook couple_audit.yml -i inventory "
      delegate_to: localhost
      ignore_errors: yes

    - name: email notification of scheduled tasks
      mail:
        host: mail.gns3.com
        port: 25
        subject: Scheduled job on {{ansible.date}}
        to: sriranjan97@gmail.com
        from: ansible@gns3.com
        body: 'Finished executing {{ansible.date}} task.Please check em
```

Fig -3: Master script.yml

- A cronjob.sh where the master script is called every day at 4 AM.

Apart from the scripts listed above, there is an ad-hoc OS upgrade script where the OS version of all routers can be upgraded with the execution of this single playbook.

Here is the Architecture of the proposed system. This shows the structure details of ansible server and its functions

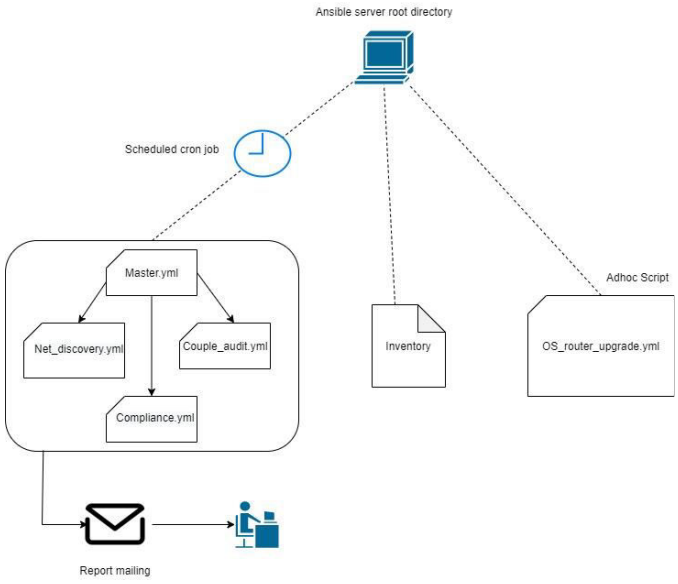

Fig -4: Architecture of ansible server and its functions

6. RESULTS

Once the scheduled job is finished it gives set of csv reports based on the jobs. The switch compliance report is as follows

Device	Ping	Compliance	Auto-fix	Error	After auto-fix
192.168.7.1	Success	Compliant	NA	NA	Compliant
192.168.7.3	Success	Non-Compliant	TRUE	NA	Compliant
192.168.7.5	Success	Compliant	NA	NA	Compliant
192.168.7.7	Success	Compliant	NA	NA	Compliant
192.168.7.9	Success	Compliant	NA	NA	Compliant
192.168.2.2	Success	Non-Compliant	TRUE	NA	Compliant
192.168.2.4	Success	Compliant	NA	NA	Compliant
192.168.2.6	Success	Compliant	NA	NA	Compliant
192.168.2.8	Success	Non-Compliant	TRUE	NA	Compliant
192.168.3.55	Failed	NA	NA	Unable to reach	NA

Fig -4: Switch compliance report

As shown in the report above one can observe it has the list of devices along with the details of compliance audit performed on it via switch compliance ansible playbook. This playbook internally has an email task defined to generate the csv report of the same and mail it to the concerned person.

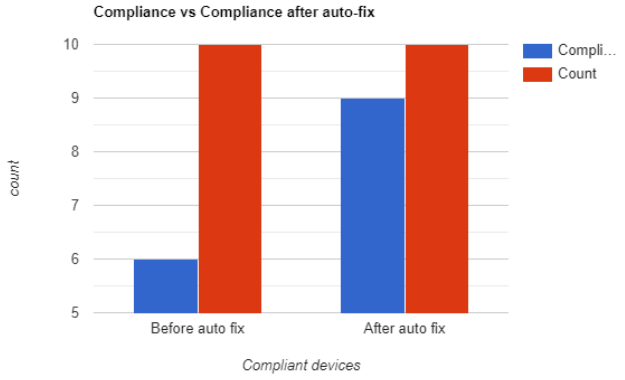

Fig -5: Bar-graph of device compliance and count before and after auto_fix.

The bar graph depicts number of devices that were fixed when the master script executed and fixed all the non-compliant devices.

7. COMPARISION

In [3] a typical use case of creating simultaneous VLan switches is achieved by using python where in one has to pass device name and its parameters every time to do a configuration change. However, in this proposed study, since ansible is used, it uses the concept of inventory where all the list of target hosts are defined along with its parameters thus giving ansible an upper hand in efficiency in triggering a configuration change.

8. CONCLUSION

The reports generated on daily basis provides better insights on device configuration behavior and the auto fix(self-heal) feature will fix the device configuration to the standard configuration of that device if there is any non-compliance found. The Ad-hoc OS upgrade script updates the OS versions of the routers mentioned in the inventory with a single execution than a manual person sshing for each device to make changes. Overall, the manual intervention is avoided here thus by enhancing the network compliance using ansible automation.

9. FUTURE WORK

The scope of the project can be further enhanced by bringing authentication and authorization layers with automations, Since it has limited GUI capabilities a future work on building such expert systems for wider community use can also be implemented.

REFERENCES

1. Arachchi, S A I B & Perera, Indika. (2018). Continuous Integration and Continuous Delivery Pipeline Automation for Agile Software Project Management. 10.1109/MERCon.2018.8421965.
2. P. Masek, M. Stusek, J. Krejci, K. Zeman, J. Pokorny and M. Kudlacek, "Unleashing Full Potential of Ansible Framework: University Labs Administration," 2018 22nd Conference of Open Innovations Association (FRUCT), 2018, pp. 144-150, doi: 10.23919/FRUCT.2018.8468270.
3. Mihăilă, Paul & Balan, Titus & Curpen, Radu & Sandu, Florin. (2017). Network Automation and Abstraction using Python Programming Methods. MACRo 2015. 2. 10.1515/macro-2017-0011.
4. D. Schulz, "Intent-based automation networks: Toward a common reference model for the self-orchestration of industrial intranets," IECON 2016 - 42nd Annual Conference of the IEEE Industrial Electronics Society, 2016, pp. 4657-4664, doi: 10.1109/IECON.2016.7792959.
5. Gaurav kumar & Mukesh kumar, “Approach of automation between development and operation by using DevOps”, International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395 -0056 Volume: 03 Issue: 12
6. S. Sun, "Stably extracting text contents from email messages with Python," *2009 Second International Conference on the Applications of Digital Information and Web Technologies*, 2009, pp. 199-203, doi: 10.1109/ICADIWT.2009.5273961.
7. *Rohit Katiyar, Prakash Pawar, Abhay Gupta, and Kotaro Kataoka. 2015. Auto-Configuration of SDN Switches in SDN/Non-SDN Hybrid Network. In Proceedings of the Asian Internet Engineering Conference (AINTEC '15). Association for Computing Machinery, New York, NY, USA, 48–53. DOI:https://doi.org/10.1145/2837030.2837037.*
8. Nishant Kumar Singh, S. Thakur, H. Chaurasiya and H. Nagdev, "Automated provisioning of application in IAAS cloud using Ansible configuration management," *2015 1st International Conference on Next Generation Computing Technologies (NGCT)*, 2015, pp. 81-85, doi: 10.1109/NGCT.2015.7375087.
9. D. F. Macedo, D. Guedes, L. F. M. Vieira, M. A. M. Vieira and M. Nogueira, "Programmable Networks—From Software-Defined Radio to Software-Defined Networking," in IEEE Communications Surveys & Tutorials, vol. 17, no. 2, pp. 1102-1125
10. P. Chaignon, K. Lazri, J. Francois and O. Festor, "Understanding disruptive monitoring capabilities of programmable networks," 2017 IEEE Conference on Network Softwarization (NetSoft), Bologna, 2017, pp. 1-6.
11. P. Goransson et al., "Software Defined Networks: A Comprehensive Approach", *Morgan Kaufmann Publishing*, pp. 978-0124166752, May 2014, ISBN -13.
12. Tischer R., Gooley J.: Programming and Automating Cisco Networks, Cisco Press, September 9th 2016
13. Cisco” DevNet” Open Source Dev Center - https://developer.cisco.com/site/opensource/
14. Netmikohttps://pynet.twb-ech.com/blog/automation/netmiko.html
15. Paramiko, http://www.paramiko.org/ 8. Network Test Automation Forum, NTAF White Paper. [online] Available: http:// www.ntaforum.org