Network Automation with a Single Source of Truth in a Heterogeneous Environment

Eueung Mulyana, Gulam Fakih

School of Electrical Engineering and Informatics
Bandung Institute of Technology
eueung@itb.ac.id

Abstract: The unprecedented growth of data and network traffic has steadily increased the complexity of management and operations in data centers. This poses many challenges since resources have to be managed flawlessly and efficiently while satisfying many operational constraints. In this article we propose a scheme for network automation in a heterogeneous environment with a single source of truth. We demonstrate the applicability of such a scheme to provision L2/L3 Virtual Private Network services involving several devices in a typical Spine-Leaf architecture. The devices are initiated in a Zero-Touch Provisioning scenario, form an underlay network and communicate with each other through eBGP protocol. The proposed solution utilizes open source tools and technologies such as Ansible for the automation process and NetBox as the only source of truth for configuration data. Our experiment with three types of devices in a typical network scenario shows that the proposed scheme has achieved the targeted automation goal by reducing the required provisioning time by 70%-92%. The results also indicate a clear benefit compared to manually provisioned systems.

Keywords: Data Center; Network Automation; Ansible; Source of Truth; Zero Touch Provisioning; ZTP; EVPN; VXLAN;

1. Introduction

As a major component of modern Internet infrastructure, data centers have to always be adapted to the actual requirements as defined by many factors such as pattern and amount of usage, service architecture, type of application, etc. From a networking perspective it means that the underlying virtual and physical infrastructure needs to be provisioned, monitored and maintained efficiently so that services can be established as soon as possible and run uninterrupted. As the scale and complexity increases, network provisioning can only hard be achieved with conventional methods heavily relied on manual human intervention. Even for a most skilled network engineer, manual provisioning is still prone to configuration errors and maintainability issues. If we can substitute some of these manual, often repetitive interventions with a standard method which can be executed remotely, the previously mentioned risks are significantly reduced while increasing the scalability and thus the efficiency. Such efforts are the primary role of network automation. In the past, network automation was hardly possible to be applied due to the proprietary nature of many networking devices and due to relatively small demands for it. This has changed today since the scale of the network has reached a level that without automation it can not be run as it is required, both technically and from a business perspective. Thus currently, some automation features are offered in almost every new networking device. The resulting paradigm shift is also strengthened by the rise of open-source, collaborative and open culture. Some recent collaboration efforts target an interoperability issue and reference implementations for networking hardware and software that consider automation as a first class feature. However, in a heterogeneous network environment, automation is and will still be challenging due to diverse implementations, feature incompatibilities, and other specific mechanisms of each device. In our work we address network automation in a heterogeneous environment with an explicit, single Source of Truth (SoT) of configuration data. We adopt a two-layer networking model providing L2/L3 Virtual Private Network (L2/L3VPN) services on top of a typical underlay network in data centers. Furthermore, all devices are operated with a Zero-Touch Provisioning (ZTP) scheme so that no other initial setup is necessary except powering up and connecting the devices to the network.

Received: July 16th, 2021. Accepted: March 24th, 2022
DOI: 10.15676/ijeei.2022.14.1.6

Network automation shall be done in a consistent way so that it requires an agreed source of valid information, which includes information regarding the actual state of the network (e.g. devices, cabling, IP address, etc.). Such a source of information is usually referred to as Source of Truth. For this purpose, we utilize an open-source platform called NetBox. ZTP aims to reduce manual interventions on devices, so that the provisioning process can be done faster while simultaneously reducing possibility of errors and misconations. The automation process is then carried out and glued together by Ansible, a framework originally used for server provisioning and now also widely used in the networking landscape. Ansible automation scripts read information as provided by the SoT and use it to configuration the corresponding devices.

2. Key Terminology

This section provides a short explanation and summary of some key terms used throughout the article.

- **Ansible** - An open-source IT automation tool that automates provisioning, configuration management, application deployment, orchestration, and many other many other manual processes. An automation tool such as Ansible enables Infrastructure as a Code.
- **eBGP** - External Border Gateway Protocol. A routing protocol used to exchange routing information between autonomous networks. Recently, it is also popular to replace traditional Interior Gateway Protocols such as IS-IS and OSPF, for routing within an IP fabric.
- **EVPN** - Ethernet Virtual Private Network. A VPN technology that supports bridged, routed, and hybrid network overlay services. EVPN is defined in RFC 7432 with extensions defined in several IETF draft standards.
- **L2/L3VPN** - Layer 2 / Layer 3 Virtual Private Network. L2VPN virtualizes the data link layer (L2) so as to make two sites as if they were operating in the same LAN. L3VPN virtualizes the network layer (L3) so as to route remote sites over the provider network.
- **NetBox** - An Infrastructure Resource Modeling (IRM) application designed to empower network automation. NetBox is intended to function as a domain-specific Source of Truth for network operations.
- **VXLAN** - Virtual Extensible Local Area Network. VXLAN is the network virtualization tunneling protocol (RFC 7348) used to build virtual networks over an IP-routed infrastructure.
- **Underlay Network** - A network that provides basic network connectivity between devices. An underlay network can be an IP fabric that provides the basic IP connectivity.
- **ZTP** - Zero Touch Provisioning. ZTP is a scheme to automatically configuration networking appliances when adding them to the network. ZTP features can be found in switches, wireless access points, routers, firewalls, etc.

3. Related Work

Modern data centers are required to support physical, virtualized and cloud environments. They must be flexible, resilient, scalable and multi-tenancy-compatible, while fulfilling the performance requirements in terms of bandwidth, latency and other measures, fully independent of the actual load of the network. All of these aspects suggest a better networking infrastructure, that is simpler but more agile in terms of operations and management, as well as of service implementation and provisioning. In a broader context, it is the main driver for the development of Software-Defined Network (SDN) paradigm which makes softwarization and programmability feasible to be applied to the real networking domain. However, networking is heterogeneous in nature and not all networks can or need to be SDN-ized. Still, the “traditional” non-SDN networks need to keep the pace and respond to dynamic network changes. The resulting demands have triggered an accelerated development of network programmability and automation features for almost any recent networking devices. This phenomena enriches alternatives for innovation in the networking domain. And it is no exception in the area of network virtualization or overlay network, on which our research focused. In the following we will highlight some of the works most relevant to our research.

Table 1. Summary of the approaches from the literature

Publication	Problem Addressed	Type/Approach
A. M. Mazin, R. A. Rahman, M. Kassim and A. R. Mahmud [1]	Performance analysis on network automation (typical enterprise network scenario: connection to the Internet, routing)	● Simulation (EVE-NG) ● Python-based (Paramiko) ● 36 Cisco devices
E. F. Naranjo and G. D. Salazar Ch [2]	VXLAN/EVPN in a cloud oriented architecture (Data Center, Service Provider)	● Simulation ● 2 Scenarios: Cisco (CSR1000v, Cisco 7200), Cumulus Linux
K. A. Noghani and A. Kassier [3]	EVPN for Data Center Interconnect using SDN	● Simulation/Emulation (Mininet) ● OpenDayLight Controller
K. A. Noghani, C. H. Benet, A. Kassler, A. Marotta, P. Jestin and V. V. Srivastava [4]	Automating EVPN deployment in SDN-based Data Center	● Testbed/Implementation ● OpenStack (orchestrator) ● OpenDayLight Controller
P. Mihaila, T. Balan, R. Curpen and F. Sandu [5]	Cases of network automation (VLANs)	● Simulation (GNS3) ● Python-based (Paramiko, Netmiko) ● Cisco IOS
S. Barguil, O. Gonzalez de Dios, V. Lopez Alvarez, R. Gagliano, I. Carretero and R. Vilalta [6]	Provisioning MPLS L3VPN using SDN (intent based)	● Testbed/Implementation ● Netconf/Restconf/YANG ● Cisco IOS-XR ● Cisco NSO
T. Singh, V. Jain and G. S. Babu [7]	VXLAN/EVPN for Data Center Network Transformation	● Comparative Study
W. Brockelsby and S. Dilda [8]	Tactical deployment of network automation and ZTP concept (baseline, VLANs)	● Implementation (Duke) ● 860 switches
Y. Demchenko et al. [9]	ZTP concept for network service provisioning for cloud-based applications	● Concept Study ● GEANT & European NRENs

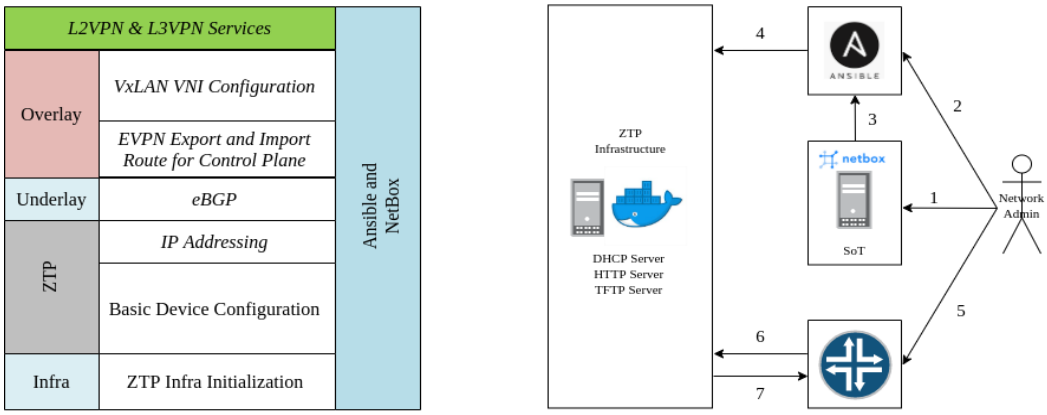
L2/L3VPN technologies are widely deployed both in data centers and particularly in transport networks to seamlessly interconnect sites over WAN, mostly using overlay/underlay paradigms [2][4]. One of the most anticipated and effective overlay technologies is Virtual eXtensible LAN (VXLAN), which can provide L2 and L3 connectivity over a L3 IP based underlay network. Due to limitations of the legacy L2VPN control plane, VXLAN is usually paired with Ethernet VPN (EVPN) as the control plane to provide flexibility and scalability for the overlay network. Using

VXLAN/EVPN technology, data center and cloud operators can deploy much larger networks than are otherwise possible with traditional L2VPN architectures [7].

The landscape of recent research in the field of L2/L3VPN is mostly influenced by the SDN paradigm, both in a full SDN or a hybrid approach. Some interesting views of management and provisioning methods for EVPN services with SDN infrastructure can be found e.g. in [3][4][6]. An overview of VXLAN/EVPN as an overlay technology can be found in e.g. [2][7]. [2] addresses VXLAN encapsulation on an underlay network to solve addressing and segmentation problems usually found in the legacy VPN services. Other relevant works are e.g. [1][8][9]. The authors in [9] presented the possibilities for automated network services provisioning using the ZTP concept. [8] explores tactical automation solutions to provision and deploy the access layer devices and showed some results for the case of Duke University's campus network. Lastly, the authors in [1] and [9] address network automation cases with Python programming languages. [1] presents performance analysis on network automation interaction with network devices. [9] gives a general overview of network programmability and automation using Python. Table 1 summarizes all the above mentioned approaches from the literature.

From the conducted surveys, there is currently no work similar to our proposal, especially in the following aspects. First, we introduce an integrated approach by combining ZTP and L2/L3VPN service provisioning. Thus, manual human-device interaction is very minimal. Second, we use a single, explicit component as the Source of Truth for configuration and for storing the state of devices in the network. This will increase visibility and consistency which are necessary for excellent, flawless network operations.

4. Automation Architecture



Figureure 1. (a) Network Automation Model (b) System Interaction and Flow for Zero-Touch Provisioning Automation

From a high level perspective, we propose and implement the network automation model as shown in Figure. 1(a). There are basically three components: data source (SoT), automation scripts and infrastructure. The SoT function is delivered by a containerized NetBox application. It will store relevant information such as device manufacturer, type, role, interfaces, IP addresses etc. The automation process is performed by Ansible scripts, which communicate with the SoT via NetBox API calls and configuration the infrastructure accordingly. The aforementioned infrastructure consists of networking devices and a group of servers that are responsible for initial device configuration. We will refer to these servers as the ZTP and Automation Infrastructure (ZAI). All devices, ZAI and the provisioning machine (i.e. a machine that runs the automation scripts) are connected via a common management network. The automation tasks can be grouped and categorized into four main parts:

- Part 1 (Infra) consists of tasks for initializing all required services i.e. DHCP, HTTP, and TFTP servers. The DHCP server assigns an IP address to every new connected device, while

the HTTP/TFTP servers provide relevant configuration files. For provisioning simplicity, all these servers are containerized.

- Part 2 (ZTP) consists of tasks for bootstrapping basic device configuration and IP address assignment. configuration is device-specific and dependent on the type and manufacturer of each device.
- Part 3 (Overlay/Underlay) is responsible for service-specific control plane configuration. In our case, it consists of tasks for initialization and configuration eBGP, EVPN and VXLAN instances.
- Part 4 (L2/L3VPN) consists of tasks for establishing the requested L2/L3VPN services.

Figure 1(b) depicts a generic system interaction and flow for the first three groups of the automation tasks (Part 1, 2 and 3). At the end of these processes, devices shall be connected to each other, forming a functional underlay network for L2/L3VPN services. The flow represents the following process respectively: (1) Network administrator or operational engineering team enters all relevant device and network parameters to the SoT; (2) Network administrator executes the automation scripts for the ZAI initialization; (3) Automation scripts read all relevant information from the SoT via NetBox API calls; (4) Automation scripts configuration the target servers remotely and initialize all required services; (5) Network administrator attaches a new device to the management network and power it on; (6) Device connects to the servers. This might include several communication sessions; and (7) Device is assigned an IP address and receives an appropriate configuration file.

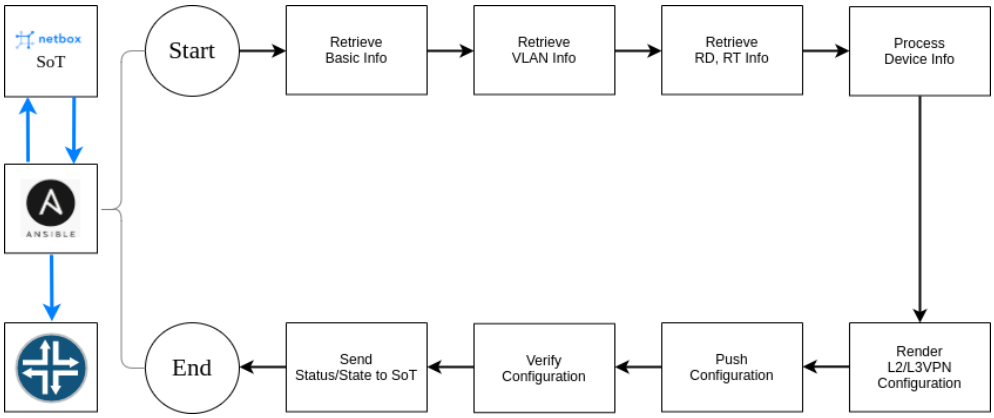


Figure 2. System Interaction and Flow for Overlay/Underlay Automation

Figure. 2 shows system interaction and flow for overlay/underlay automation. Basically the automation scripts read information from the SoT, covering all relevant device parameters, VLAN parameters, and Route Distinguisher (RD) / Route Target (RT) parameters. RD/RT parameters are used for VPN routing purposes. RD distinguishes between one set of routes (one VPN Routing and Forwarding - VRF) from another, while RT is used to share routes among different VRFs. All this information is then processed and used to render the actual configuration for L2/L3VPN services on every device. Once ready, the configuration is pushed to the corresponding device. The last steps are to verify the new configuration on each device and to store device state to the SoT.

Templating plays a very important role in the aforecited process, in particular for supporting a large variety of devices and thus a heterogeneous environment. Templates are specifically prepared for devices of different types and/or manufacturers. Automation scripts choose and render the right template based on the information given by the SoT. The resulting configuration file is put on the certain location served by either the HTTP or TFTP server. It will then be pulled and executed by the device once automation scripts command it to do so.

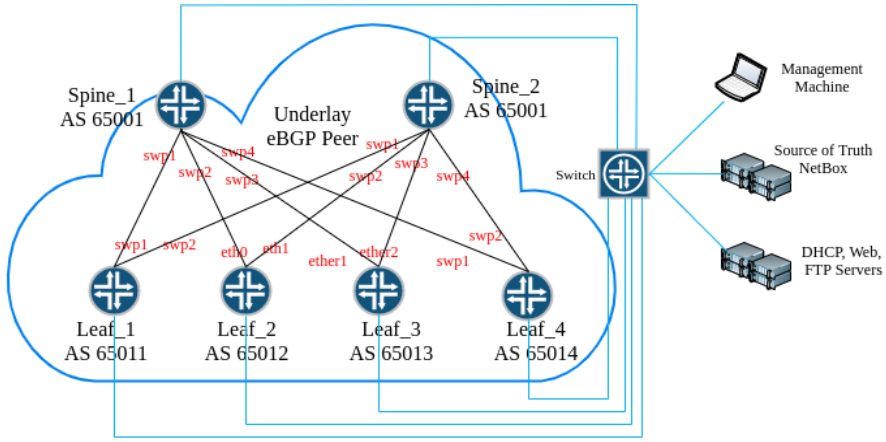


Figure 3. A Six-Node Network in a Spine-Leaf Topology and the Corresponding Infrastructure for ZTP and Service Automation

5. Evaluation and Result

Figure. 3 shows a six-node network in a Spine-Leaf architecture which is used in our proof-of-concept investigation. Such a network structure, also known as the Clos topology, is a standard architecture in modern data centers. The network consists of two Spine- and four Leaf-devices. To demonstrate the heterogeneous environment, we are using three different systems. Leaf_2 and Leaf_3 run Arista vEOS/4.24 and Cisco Nexus/9k operating systems, respectively. All other devices run Cumulus Linux/4.1. Those devices are connected via an out-of-band interface to the management network, where all the NetBox server, the ZAI and an automation/management machine reside. The experiment is conducted in an EVE-NG emulated environment with 14 vCPU and 26GB RAM running on a physical machine with Intel Xeon E5-2620v4 (16 CPUs @2.1GHz, 256GB RAM). Each of the six devices is assigned resources of 1 vCPU / 2GB RAM / 3GB storage. Two VLANs shall be built on the top of this network, accessible from each Leaf node and connected to each other with an L3VPN service.

Table 2 summarizes the result of functionality tests both on the underlay and the overlay L2/L3 network. The functionality is validated based on either the output of a certain command or the occurrence of certain log messages. All the results suggest that our network automation works as expected.

Table 3 gives the resulting average completion time of the automation process for four stages as mentioned in Section 3. Note that Part 2 and 3 are joined together in one process to eliminate the need for configuration and thus, to save execution time. The time required for establishing the ZTP and Automation Infrastructure is less than two minutes on average.

For the ZTP/Underlay/Overlay automation, the average completion times of three device types deviate significantly: Cisco NX is the slowest with around two and a half minutes, while Arista vEOS requires only around 15 seconds. For L2/L3VPN service provisioning, the times are not so different from each other i.e. around 12-15 seconds. Since the Ansible scripts run sequentially, the average completion time for the whole process can be obtained by summation of the times consumed by each device. The average completion time for ZTP/Underlay/Overlay is 277.33 seconds, while that for L2/L3VPN is 85.61 seconds.

Though from execution time perspective, the benefit of automation is very obvious, it can be enlightening to have some numerical Figures for comparison. Thus by conducting manual configuration experiments, we have the following results: 1494 seconds (Part 1), 982 seconds (Part 2,3) and 292 seconds (Part 4). Comparing these two experiments, the automation process can save provisioning time by 92%, 71% and 70% for each stage respectively. In the manual experiment there are surely many factors that may lead to different numerical results e.g. typing speed, personnel's familiarity with specific device commands, etc. Thus, the comparison Figures are only useful for

rough estimates. Completion time gap between an automated and manual process will also increase as the size of the network grows and change as the device heterogeneity profile varies.

Table 2. The Functional Test and Validation Result

Object	Item	Indicator	Result
Underlay Network	DHCP	DHCP Request (Log)	Pass
	Interface	net show interface (CLI)	Pass
	BGP IPv4 Family	net show bgp sum (CLI)	Pass
	BGP L2VPN EVPN	show bgp evpn summary (CLI)	Pass
	Route (Spine)	net show route (CLI, Spine)	Pass
	Route (Leaf)	show ip route (CLI, Leaf)	Pass
L2VPN	VNI	show bgp evpn vni 13 (CLI)	Pass
	MAC Address (between VTEPs)	net show bgp evpn route type macip (CLI)	Pass
	Ping (between hosts of different Leaf)	ping 192.168.13.200 source 192.168.13.20 repeat 50 (CLI)	Pass
	Multipath EVPN	ping 192.168.13.200 source 192.168.13.20 repeat 100 (CLI, 1 Spine down)	Pass
L3VPN	Ping (between hosts of different VLAN)	ping 192.168.13.20 source 192.168.100.30 repeat 100 (CLI, @VLAN 100)	Pass
	Traceroute (between hosts of different VLAN)	traceroute 192.168.13.20 source 192.168.100.30 (CLI, @VLAN 100)	Pass

Table 3. Average Completion Time for Each Component/Device

	Device	Average Time to Complete (seconds)
Infra (Part 1)	ZTP and Automation Infrastructure	112.3
ZTP/Underlay/Overlay (Part 2,3)	Cumulus Linux	26.6
	Arista vEOS	15.3
	Cisco NX	155.6
L2/L3VPN (Part 4)	Cumulus Linux	14.9
	Arista vEOS	13.2
	Cisco NX	12.8

Last but not least, we can process the numerical results to obtain the corresponding atomic normalized values, so that we can roughly estimate the benefits of network automation for different network scenarios. The comparison of manual and automated provisioning with respect to configuration completion time is shown in Table 4. For the underlay case, the time unit is sec/device, while that for the overlay case is sec/device/VLAN. Comparing the values for underlay network provisioning with the result in [1] shows that: (1) the manual provisioning time is quite similar i.e. 163 vs 161 seconds, but (2) the automated provisioning time is very different i.e. 46 vs 3.3 seconds. The differences occur since the scope of automation is also different i.e. we consider ZTP initialization and eBGP setup, while [1] only automates OSPF routing.

In our case, if a network consists of 8 Spine and 192 Leaf devices and 40 VLANs shall be provisioned on each Leaf, by applying network automation we have the following estimate of time savings Figure: 6.5 hours for underlay and 35.95 hours of overlay provisioning. Surely, we can further increase these Figures by applying parallelism i.e. by deploying several threads simultaneously or by adding more provisioning machines.

Table 4. Average Completion Time for Manual vs. Automated Provisioning

	Underlay	**Overlay (Service)**
	/Device (seconds)	/Device/VLAN (seconds)
Manual Provisioning	163	24
Network Automation	46	7.15
Time Savings	117	16.85

6. Conclusion

In this paper, we have presented a scheme for network automation in a heterogeneous environment with a single source of truth. We have introduced a flexible network automation model which leverages open source technologies such as Ansible and NetBox. Our approach basically combines a ZTP paradigm for minimal human interaction with automation best practices originated from Development and Operations (DevOps) culture. The applicability of such a scheme has been demonstrated in the case of provisioning L2/L3 VXLAN/EVPN services involving several devices in a typical Spine-Leaf architecture. Our experiment shows that the proposed scheme has reduced the required provisioning time by 70%-92%, which strongly indicates a clear benefit compared to manually provisioned systems.

7. Acknowledgment

The authors would like to gratefully acknowledge Institut Teknologi Bandung for providing financial support for this research via the P2MI ITB 2021 program.

8. References

[1]. A. M. Mazin, R. A. Rahman, M. Kassim and A. R. Mahmud, "Performance Analysis on Network Automation Interaction with Network Devices Using Python," 2021 IEEE 11th IEEE Symposium on Computer Applications & Industrial Electronics (ISCAIE), 2021, pp. 360-366, doi: 10.1109/ISCAIE51753.2021.9431823.

[2]. E. F. Naranjo and G. D. Salazar Ch, "Underlay and overlay networks: The approach to solve addressing and segmentation problems in the new networking era: VXLAN encapsulation with Cisco and open source networks," 2017 IEEE Second Ecuador Technical Chapters Meeting (ETCM), 2017, pp. 1-6, doi: 10.1109/ETCM.2017.8247505.

[3]. K. A. Noghani and A. Kassier, "SDN enhanced ethernet VPN for data center interconnect," 2017 IEEE 6th International Conference on Cloud Networking (CloudNet), 2017, pp. 1-6, doi: 10.1109/CloudNet.2017.8071535.
[4]. K. A. Noghani, C. H. Benet, A. Kassler, A. Marotta, P. Jestin and V. V. Srivastava, "Automating Ethernet VPN deployment in SDN-based Data Centers," 2017 Fourth International Conference on Software Defined Systems (SDS), 2017, pp. 61-66, doi: 10.1109/SDS.2017.7939142.
[5]. P. Mihaila, T. Balan, R. Curpen and F. Sandu. "Network Automation and Abstraction using Python Programming Methods" MACRo 2015, vol.2, no.1, 2017, pp.95-103.
[6]. S. Barguil, O. Gonzalez de Dios, V. Lopez Alvarez, R. Gagliano, I. Carretero and R. Vilalta, "Experimental validation of L3 VPN Network Model for improving VPN service design and provisioning," 2020 16th International Conference on Network and Service Management (CNSM), 2020, pp. 1-5, doi: 10.23919/CNSM50824.2020.9269043.
[7]. T. Singh, V. Jain and G. S. Babu, "VXLAN and EVPN for data center network transformation," 2017 8th International Conference on Computing, Communication and Networking Technologies (ICCCNT), 2017, pp. 1-6, doi: 10.1109/ICCCNT.2017.8203947.
[8]. W. Brockelsby and S. Dilda, "Tactical Network Automation with NetZTP and One Shot," 2019 IEEE 40th Sarnoff Symposium, 2019, pp. 1-3, doi: 10.1109/Sarnoff47838.2019.9067817.
[9]. Y. Demchenko et al., "Enabling Automated Network Services Provisioning for Cloud Based Applications Using Zero Touch Provisioning," 2015 IEEE/ACM 8th International Conference on Utility and Cloud Computing (UCC), 2015, pp. 458-464, doi: 10.1109/UCC.2015.82.

Eueung Mulyana is a lecturer and researcher at the School of Electrical Engineering and Informatics of Bandung Institute of Technology (ITB). He received his Doctoral degree from Technische Universitaet Hamburg-Harburg, Germany, in 2006. His expertise areas are Communication Networks, Connected Services and Cloud Computing. Email: eueung@itb.ac.id.

Gulam Fakih received a Master degree (MT) from the School of Electrical Engineering and Informatics of Bandung Institute of Technology (ITB), in 2021. Since 2013, he has been working as a network and DevOps Engineer at ITB. Email: gulam@itb.ac.id