A Web Driven SDN Orchestrator For The Provisioning of ACI Fabric and Lab Infrastructure

Matthew Gaynor

922830

A dissertation submitted in partial fulfilment
of the requirements for the degree of
Bachelor of Science
of the
University of Portsmouth.

School of Computing
Engineering Project

2023

Declaration

No portion of the work contained in this document has been submitted in support of an application for a degree or qualification of this or any other university or other institution of learning. All verbatim extracts have been distinguished by quotation marks, and all sources of information have been specifically acknowledged.

Date: 2023

Abstract

The purpose of this project is to provide a solution to CX Labs UK within Cisco Systems LTD that will streamline the operation of DMZ lab-space. The application shall allow lab staff to prepare network infrastructure for an incoming project with little to no CLI interaction and it shall be web based. ACI, VMware ESXi and vCenter shall be used to manage the networking and virtual machines respectively. The lab-space will be represented by a small testbed containing the minimum quantity of networking equipment and compute resources required to accurately simulate the environment.

Acknowledgements

I would like to thank Cisco, my manager and good friend, Dave Smith, for his close help in allowing me to use equipment necessary for testing and developing the solution.

Consent to Share

I consent / do not consent for this project to be archived by the University Library and potentially used as an example project for future students.

Abbreviations

Solution Validation Services	SVS
Customer Experience	CX
Demilitarised Zone	DMZ
Wide Area Network	WAN
Virtual Private Network	VPN
Network Operating System	NOS
Fabric Extender	FEX
Top of Rack	ToR
Distrubited Port Group	DPG
Distributed Virtual Switch	DVS
Network Time Protocol	NTP
Domain Name System	DNS
Remote Authentication Dial-In User Service	RADIUS

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 The Client

The client is CX Labs UK within Cisco Systems. CX Labs provides lab space for use by business units internal to the company. Most of the space is used for the testing of customer networks by SVS (Solution Validation Services). SVS provide bespoke testing services to customers wishing to use Cisco's expertise to test a range of situations, from regression and firmware testing to full upgrade and migration plans.

In order to match the customers environment as close as possible, a scaled down version of the customers network is usually recreated in the lab space managed by CX Labs. CX Labs hold many devices that cover most of the Cisco portfolio, which allows for the recreation of most networks. Most of this lab space is hosted within the internal Cisco corporate network, which requires any users to be employees of Cisco in order to access testbeds. More and more customers however are requesting remote access to their testbeds. To facilitate this, a fully isolated DMZ environment is provided, which allows direct WAN connectivity to a testbed, allowing for a VPN tunnel to be established and hence remote access granted to a testbed from any location to any permitted person.

1.2 The Problem

Currently the solution for the DMZ network infrastructure consists of 4 Nexus 9K devices, with FEXs for RJ45 connectivity and 2960S ToR switches. Whilst this solution is functional and works, the major downside is that there is no automation or configuration management solution deployed. This means over time, configuration drift occurs as manual changes are made, but not made the same to all switches. Unused VLANs are also not removed from the switches which leads to bloat and a larger configuration than is required. The same situation also occurs on vCenter with the configuration of the DPGs within the DVS where unused DPGs are never removed or are labelled incorrectly.

The manual configuration of the required routing and VPN termination as well as other lab requirements such as a NTP, DNS and RADIUS all take a lot of time to configure. This time could be better utilised, such as preparing the physical rack space for the racking and stacking of new equipment.

Chapter 2

Literature Review

The aim of this literature review is to research and analyse existing solutions, documentation and research on the automation of networking infrastructure. To ensure this review is of maximum usefulness, it will also involve analysing best practises and standards when developing software and automation solutions. This will allow for the optimisation of the planning and implementation stages of the project that will subsequently follow.

Sources for this review will be from relevant books, online websites, professional publications and Request for Comments. Multiple services will be used to identify and obtain sources that can be used for the purposes of this review.

- Google Scholar
- IEEE Explore
- ResearchGate
- University of Portsmouth Library

Research was conducted that was related to the following set of topics:

- What is software defined networking?
- What types of software defined networking exist?

- What are the advantages and disadvantages?

2.1 Definition of Software Defined Networks

All industry experts and academics define software defined networking similarly, that is providing automation and intelligence to networks via the means of software and APIs. Kreutz et al. (2015) state that “SDN was originally coined to represent the ideas and work around OpenFlow at Stanford University”.
SDN is often defined using four pillars, Kreutz et al. (2015) define these as the following:

1. “The control and data planes are decoupled. Control functionality is removed from network devices that will become simple (packet) forwarding elements”

2. “Forwarding decisions are flow based, instead of destination based. A flow is broadly defined by a set of packet field values acting as a match (filter) criterion and a set of actions (instructions)”

3. “Control logic is moved to an external entity, theso-called SDN controller”

4. “The network is programmable through software applications running on top of the NOS that in-teracts with the underlying data plane devices.”

2.2 Overview of Software Defined Networks

Since this project is centred around software defined networking and the automation of these networks, it is critical to ensure that the principles and their method of operations are understood.
An official survey paper from the IEEE that analysed the state of SDN provides a good explanation as to why the need for network programmability arose in the first place. Conventionally, “Computer networks are typically built from a large number of network devices such as routers, switches and numerous types of middleboxes” Nunes et al., 2014. Nunes et al. (2014)

go on to state that due to the large amount of manual configuration required to achieve the desired traffic flow, “network management and performance tuning is quite challenging and thus error-prone”. This leads to one of the solutions, Software Defined Networking. “Software Defined Networking (SDN) is a new networking paradigm which the forwarding hardware is decoupled from control decisions. It promises to dramatically simplify network management and enable innovation and evolution”. Software defined networking “is designed to use standardized application programming interfaces (APIs) to quickly allow network programmers to define and reconfigure the way data or resources are handled within a network.” Kirkpatrick, 2013

2.3 Types of Software Defined Networks

Software Defined Networking is a blanket term for any solution that tackles conventional networking with a programmatic view. There are “the two main delivery models: Imperative and declarative” CDW (2015). CDW (2015) go on to state that Imperative SDN is where “A centralized controller (typically a clustered set of controllers) functions as the network’s ‘brain”’ and that declarative SDN is where “the intelligence is distributed out to the network fabric. While policy is centralized, policy enforcement isn’t”. CDW (2015) give “a protocol such as OpenFlow explicitly telling network switches precisely what to do and how to do it” to be an example of imperative SDN.

References

CDW. (2015). The future of networking arrives. *Commun. ACM*, 16–19. https://webobjects.cdw.com/webobjects/media/pdf/CDWCA/White-Paper-The-Future-of-Networking-Arrives-MKT2862CA.pdf.

Kirkpatrick, K. (2013). Software-defined networking. *Commun. ACM*, *56*(9), 16–19. https://doi.org/10.1145/2500468.2500473

Kreutz, D., Ramos, F. M. V., Veríssimo, P. E., Rothenberg, C. E., Azodolmolky, S., & Uhlig, S. (2015). Software-defined networking: A comprehensive survey. *Proceedings of the IEEE*, *103*(1), 14–76. https://doi.org/10.1109/JPROC.2014.2371999

Nunes, B. A. A., Mendonca, M., Nguyen, X.-N., Obraczka, K., & Turletti, T. (2014). A survey of software-defined networking: Past, present, and future of programmable networks. *IEEE Communications Surveys & Tutorials*, *16*(3), 1617–1634. https://doi.org/10.1109/SURV.2014.012214.00180