Harnessing Software Engineering Principles in Network Operations

The Application of DevOps in Campus Area Networks

Shivam Singh

January 22, 2025

A Proposal Presented to
The Department of the Information Science and Technology
College of Science Technology and Applied Arts of Trinidad and Tobago

In (Partial) Fulfillment
of the Requirements for the Degree
Bachelor of Science

Complementary content available at
`https://github.com/Shivam-S-Singh/COSTAATT_Final_Project`

Contents

1 Executive Summary

1.1 Overview

This document serves as a proposal to adopt a DevOps approach to managing the network infrastructure for the College of Science, Technology and Applied Arts of Trinidad and Tobago.

Many of the software used is open source or freely available.

1.2 What this proposal does not cover ?

This solution does not include the implementation of the following elements which are common components in Campus Area Networks.

- Wireless Access
- Data Center Routing and Switching
- Voice and Collaboration

The reason these topics are not covered is because this proposal is not an all inclusive solution and is intended to provide a sense of direction for the organization to transition to utilizing the methodologies associated with Developer Operations. As such, I limited the proof of concept to network connectivity and security.

2 Introduction

2.1 Client Background & Relevance

COSTAATT is a public tertiary institution in Trinidad and Tobago that offers programs in the areas of Information Technology, Business and Nursing, just to name a few.

The solution being proposed can be applied to all types of organizations with sizeable networks as it is aimed towards addressing the management of a large-scale topology with limited human resources. The reason COSTAATT is an ideal candidate for this solution is because of their vast amount of networks and communication equipment that span multiple campuses across Trinidad and Tobago.

2.2 Problem

Critical services such as file sharing, active directory and email which was originally on premise is now being hosted on cloud based platforms or being migrated to decentralized infrastructure such as remote data centres. High availability, fault tolerant networks have become a requirement in order to ensure consistent access to these resources.

The traditional methods of managing the network has led to longer turnaround times with regards to troubleshooting issues, and scalability when making configuration changes to accommodate new devices or applications.

The following topics below are areas that present challenges within ***Network Operations*** that can be solved by adopting a ***Developer*** approach to the problem.

2.2.1 Documentation

During the planning phase in network design we typically use Visio, spreadsheets and word documents to map our infrastructure. Where Visio is used to visualize our topology and spreadsheets or word documents would be used for IP address or VLAN assignment information. Text files holding configuration information would be stored in a centralized location and would sometimes be altered and then ran on the device where changes are required. This has traditionally been the standard for documenting our network but it comes with its drawbacks, specifically with regards to maintaining the documentation after they are modified.

Manual updates are required whenever a change is made on the network, e.g., a network device is added or a new route or VLAN has been provisioned on a device. This would mean an engineer would have to amend the topology in the Visio diagram and make changes in the related spreadsheets or word documents. This can become a time consuming task that is usually not a priority for the engineer.The task of updating the documentation sometimes gets pushed back to the point where it never gets completed. When an incident occurs or configuration changes are requested you are stuck with unreliable documentation which can delay resolving the issue or carrying out changes.

There is a lack of accountability when changes are made. For instance, lets say an ACL entry was configured to block Facebook for a specific part of the network, why was this change made and who was the engineer that made the change. This change could have been implemented as a request from a manager who finds that their staff is spending too much time on social media or from the IT team in response to slow speeds due to over utilization of an internet circuit. When a change is made, we should have the necessary details when we look back on this change. Who made the change, when was it executed and why was it executed. This context and details is instrumental in effectively managing our network.

Collaboration on documentation can also have its limitations

2.2.2 Production Network as the Single Source of Truth

The Production network as the single source of truth means that whatever is in production and currently running is the state of the network. Often times when we deploy a network and we have to make changes we make these changes directly in production. This can be risky as changes that are thought to be non-detrimental can have a major negative impact on the network and its users. Configuration Drift.

2.2.3 Configuration Management

We are presently using CLI-based configuration changes to manually make changes on network devices. This is prone to human error since we can make mistakes with regards to syntax and maintaining consistency when configurations have to be made across a fleet of devices. Configuration changes can become a repeatable task where we have redundant switches or routers in place. This can become a very time consuming activity especially with limited staff. (Speak on vendor specific syntax and its complexity when managing a multi vendor environment)

2.2.4 Testing on Production

Testing is important in DevOps but this has been a huge limitation for network engineers since traditionally for testing we would have to utilize real equipment. Having a non production lab environment can be costly and most organizations cannot afford this.

2.2.5 Change Management

2.3 Aim

Implement a DevOps solution that enhances the management of COSTAATT's network which will improve system uptime and increased productivity for the organization. The solution contributes to productivity because automating repeatable or time consuming tasks will allow IT operations to concentrate on projects geared towards optimizing performance.

2.4 Evaluation

The delivery of the following components will determine the success of this project.

Build a network topology that is up to standards with regards to security and guidelines of Campus Area Networks. I will be relying on the NSA's Network Infrastructure Security Guide as well as Cisco's guide on implementing Campus Area Networks.

The devices must be communicating successfully with one another. We can use simple ping test in order to determine this.

The Ansible framework must be able to communicate with all devices that support SSH and I should be able to push configuration changes to all devices successfully. Ansible has built in features that indicate if you have successfully pushed configurations to a device.

Hypothetical scenarios will be introduced into the environment such as device or link failures and we will utilize the tools that we implemented to quickly recover. Failure to recover within an acceptable time frame will determine if the solution is fault tolerant. We want to recover within minutes. Examples of mass configuration changes will be carried out to test the efficiency of using Ansible.

3 Objectives

3.1 Building the Network

3.2 Establishing the Configuration Work Flow

3.3 Configuring our Control Node

4 Resources

4.1 Host System

4.2 Design

4.2.1 Topology and Physical Rack

4.2.2 IPAM

4.3 Network Simulation

4.3.1 Simulation Software

4.3.2 Network Operating Systems

4.3.3 IP Services

4.3.4 Docker

4.4 Infrastructure as Code

4.4.1 Automation

4.4.2 Version Control System

4.4.3 CI & CD

4.4.4 Automated Testing

5 Activities

6 Measurement

7 Budget