Student Profile

Name	SHIVAM SINGH
ID	00053724
Major	INFORMATION TECHNOLOGY - NETWORKING
Report	
Project	The Application of DevOps in Campus Area Networks

Assessment

1. **Provisional Project Title** *(max 50 words)[Word count 45]*

Student Answer:

Improving the manageability, availability and scalability of campus area networks through the application of software engineering techniques to traditional network environments to improve the overall efficiency and growth of COSTAATT's operational capabilities.

The term CAN or Campus Area Network is not limited to just learning institutions but to computer networks that span a limited geographic area. Several buildings within close proximity that need to be interconnected to one another would be considered a CAN.

Marker Feedback:

Okay, the second sentence may not really be needed, still not very clear, redefined

2. **Outline of the Project Environment** *(max 200 words)[Word count 232]*

Who is the client? What do they do? What is their problem (briefly, as you will elaborate on this in the next question)? Why does it need to be solved?

Student Answer:

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

The problem that needs to be addressed is that they need to improve uptime and network availability due to the rise in the reliance and implementation of cloud based services in the organization. Many services such as file sharing, active directory and email services which was originally on premise is now being hosted in cloud based platforms. This means that high availability fault tolerant networks have become a necessity.

The traditional way of managing the network has led to longer turnaround times with regards to network troubleshooting, and scalability when making changes in the network to accommodate new devices or applications. This causes a halt in productivity as well as growth in the organization when they attempt to integrate new services.

Marker Feedback:

Very good

3. **Problem to be Solved and Indication of Solution** *(max 200 words)[Word count 305]*

Give more detail about the problem. How do you propose to solve it? Why do you propose to solve in that way? What constraints are there on your solution to the problem? Why?

Student Answer:

The problem is the traditional method of deploying and managing networks is no longer scalable for COSTAATT. We first need to understand how we traditionally manage the network and how this is an issue today. Below breaks down each of the primary issues and the DevOps approach that will be taken to rectify these issues.

1) Documentation and Production Network as the single source of truth:

Problem: We rely on documentation such as the classic Visio diagrams for topology requirements, Spreadsheets/Word Docs for IP Addressing Scheme and VLAN provisioning as well as emails communicating network changes. Manual modification to documentation is necessary to track changes and this is not done frequently. It is also difficult to maintain these documents consistency across multiple teams.

Solution: A version control system like Git or GitHub with infrastructure as code (Python) so the changes become self documenting and are at a centralized point of control.

Problem: 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.

Solution: Keep the single source of truth separate by having a separate system from the production network like a server, database or version control system that captures information such as network configuration and related information to the network. We will explore NetBox which is a software designed to keep network information in a centralized location.

2) Configuration Management:

Problem: We are 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.

Solution: We can use automation tools and frameworks in order to simplify the configuration process and adopt an Infrastructure as Code solution. In this scenario we will utilize *Ansible*, a python based automation framework originally developed for server administration but has now been adapted to manage network devices.

3) <u>Testing and Testing Frameworks:</u>

Problem: 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.

Solution: Virtualized networks can be used for testing in order to anticipate network behaviour. It may not give us an exact replication of our production environment but it at least helps us determine the approximate effects of making a change to our network. GNS3 lets us work with virtualized network operating systems in order to test potential network implementations.

Problem: Traditionally we relied on protocols and CLI tools such as ping, traceroute and SNMP as a means to test our networks efficiency and reliability. We are now utilizing Python to manage our network so we require a way to not only test our network but test the code we deploy to our network to ensure we are making non destructive changes to our environment.

Solution: We have testing frameworks available to validate our code before pushing changes to production. The following tools will be explored.

pytest https://docs.pytest.org/en/7.2.x/

pyATS https://developer.cisco.com/docs/pyats/

A constraint of this project would be along the lines of getting the rest of the operations team at COSTAATT to adopt these changes. Not everyone will have the willingness to learn a new tool, either because they think this is time consuming or too difficult. There are always those with the usual "this is how we always did it in the past" and "it isn't broken so don't fix it" mentality. Aside from the technical implementation, there has to be a cultural shift in how we approach our problems in operations. As a COSTAATT student in networking I would always hear my fellow students in the same discipline say "why am I learning OOP, I don't need this", we want to move away from this mindset and instead of saying why do I need this, say how can I use this to make my life easier or more productive. I think some workshops with live demonstrations will help ease the operations team into embracing this solution and they will see the benefits it can bring the organization.

<u>Marker Feedback:</u>

Good

4. **Aim and Objectives** *(max 200 words)[Word count 253]*

What is your single project aim? What are the most important associated project objectives? Why have you picked this particular aim and these objectives?

Student Answer:

My single project aim is to implement a solution that enhances the management of COSTAATT's network which will ultimately allow for efficient troubleshooting and contribute to the overall growth of the organization. The solution contributes to growth because automating repeatable or time consuming tasks will allow IT operations to concentrate on projects geared towards optimizing infrastructure. The success of this project rides on the delivery of the Infrastructure as Code solution, use of the version control system and the testing environment. At the core of everything it is graceful execution of the code that will drive this solution forward.

I picked this particular aim and objectives due to observations that are made in my career thus far. In my first job as a service desk admin I struggled with management of the organizations network due to outdated documentation and a lack of centralized documentation. The year was 2019 and I was relying on Visio documents, Spreadsheets and Word documents where the changes that were last made were in 2016. Not to mention the documentation was spread across several different folders in a file server. This did not inspire confidence and the only way for me to be sure when troubleshooting was I had to validate the configurations of each device and this was time consuming. This network is relatively small compared against COSTAATT's infrastructure.

In my current job I troubleshoot MPLS services for business customers and its very easy to identify when a customer is struggling to troubleshoot their internal network. The customer will share internal configurations and tests that have nothing to do with Cable and Wireless network and then expect us to provide some kind of support but the reality is if they are not fully managed by us it is up to them to fix the issue. Seeing customers in that desperate state raises concerns about how networks are being treated as an afterthought and the only time the operations team takes interest is when something goes wrong. I want the network to be recognized as an almost organic

We also have customers that have many switches in their network, I am talking about maybe 15 switches in a single building and we would have to make a change like add a VLAN to all the ports on each switch. As you can imagine, remoting into each switch individually and adding a VLAN could be a very time consuming task. These objectives are aimed at addressing these universal challenges in network operations.

Marker Feedback:

Okay

5. Resources *(max 200 words)* [Word count 253]

What facilities/resources/technologies will you use or require? Are there constraints on their availability? If funds are required to acquire them, have these been allocated? Will they be available in time? Please do not state obvious or trivial ones, such as word processing or statistical packages or library resources. Provide reasons for all your decisions/choices.

Student Answer:

Most of the key tools that are utilized in this project are open source so cost is not a major issue.

The tool that will make all of this possible is a network simulation tool called ***GNS3***. The reason I chose this tool is because I am familiar with it and it has been around for a while so it has plenty of community support. The alternatives were EVE-NG and Cisco VIRL. The problem with EVE-NG is the free version has a limitation of nodes (network devices) that could be implemented. Cisco VIRL was originally developed to simulate networks with Cisco devices and if you want to implement a device from another vendor like Juniper this has to be achieved in a hacky way. GNS3 has multi-vendor support and no limit to the amount of nodes that could be used in the free version. Your computers hardware just has to scale to what is being implemented. I anticipate approximately 60 devices will be utilized across 5 campuses.

Network Operating Systems for ***Cisco, Juniper*** and possibly other vendors like ***Arista*** will be used in GNS3. Most vendors now allow you to test a virtualized trial version of their Network OS in platforms like GNS3 so licensing should not be an issue.

The ***Ansible*** framework will also be a necessary component and because it is built on ***Python*** this will also be an integral part of the solution. There are many online courses available with regards to utilizing Ansible in network solutions so that was the deciding factor for this tool.

When you implement ***Linux*** based appliances in GNS3 for server-based functions, this implementation is handled by ***Docker***. Ansible will sit in an ***Ubuntu*** docker container.

Rest API's will also most likely be utilized in order to push configurations to the network devices or pull information from the devices. Rest API's are usually supported by most hardware vendors today.

Hardware:

I will most likely have to use an Ubuntu host machine for this project. If you use GNS3 on Windows you have to use a GNS3VM that is running on VMware. The reason for this is because Windows does not have native support to efficiently emulate the network OS and the GNS3VM (which is really an Ubuntu VM) handles this. The problem with this is that there is nested virtualization taking place and you won't utilize the full capabilities of your systems hardware. If you run GNS3 directly on Ubuntu then you will be able to make full utilization of your systems hardware. I intend on running this on a *AMD Ryzen 5 6-Core Processor, 32 Gigs DDR4 Memory and a 1000 GB hard drive.*

Marker Feedback:

Good

6. Project Evaluation *(max 300 words)* [Word count 268]

How will you be able to judge whether you have undertaken you project in a suitable way? How will you be able to measure this? How will you be able to judge whether the solution that you have investigated is appropriate or will be successful? Don't forget that you will not have time to implement anything yet. Provide reasons for all your decisions/choices.

Student Answer:

The delivery of following components will help me determine the success in 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.

Marker Feedback:

Great

7. Project Plan *(max 300 words)* [Word count 300]

What are you going to do when? Develop a plain list of clearly staged activities and approximate durations in weeks, no longer than 24 weeks in total. Do ***not*** *attach documents such as Gantt charts or spreadsheets. What risks to the success of the project have you identified? What steps can you take to minimize them? What backup plans do you have if identified things go wrong? Provide reasons for all your decisions/choices.*

Student Answer:

I do not have a formal project plan with regards to the stages at this moment, but I anticipate that this project will be completed within 3 months. It will take some weeks to familiarize myself with the technology and then decide how everything will be put together. There is also 2 sides to this project, the Developer side and the Operational side (see tasks for each below).

There are many things that can go wrong with regards to compatibility between the different components in the project, specifically with Python Version 2.7 and Python 3 and the implementation of Ansible. The great thing about Ansible is that network devices just have to support SSH in order to establish connectivity. If I run into issues with Ansible, there are alternative Python libraries that can be imported such as Netmiko and Nornir that support network functions. I am exploring these libraries as an option if Ansible does not meet all my requirements.

There may also be challenges with a multi-vendor environment in that each vendor has their own syntax. I am counting on the REST API to provide me with a level of abstraction but if push comes to shove I may end up substituting devices based on compatibility.

There is also scalability challenges with my own hardware since I never built a network of this size in GNS3, in this case I may have to increase the amount of RAM in my system and maybe even upgrade to an SSD.

Another major challenge is how I interconnect all the campus branches. Configuring an MPLS network to interconnect each branch in itself is a challenge since this is considered an advanced implementation of networking. If I fail to configure an MPLS network I may be forced to use a switch or cloud appliance which would be disappointing.

Developer:

Infrastructure as code: Ansible, Python, Netmiko and Nornir libraries

Version Control System: Git or Github

CI/CD pipeline: Jenkins

Documentation: Utilized through NetBox and VCS

Operations:

Provisioning of IP addresses: Public and Private

Assignment of both static and dynamic addresses

Routing Protocols: OSPF/BGP/MPLS

VLAN assignments

Network topology/design

Device vendors implemented

Cabling types utilized

Administrative Configs: management - SSH, naming conventions, circuit numbers

Port assignments

Access Control Lists

Marker Feedback:

Understandable, please update when you are clear

Final Assessment Mark and Overall Feedback:

You have a clear idea of what you are trying to achieve, being something new I understand you do not have all the answer just yet, update as it becomes clearer. Great project idea. Please proceed