

Performance Evaluation of COINS Framework for Wireless Network Automation

Identificador

ER-01

Año

2021

Fuente

IEEEExplore

Tipo de publicación

Conference document

Palabras clave relacionadas con NetOps

Continuous integration, continuous delivery, devops, wireless network, network automation, automated operation

Resumen

With the evolution of mobile communications towards fifth-generation (5G) and beyond, all layers of the wireless networks are increasingly virtualized and software-controlled using automated tools. DevOps tools enabling smooth and fast testing through continuous integration (CI) and rapid deployment through continuous delivery (CD) of the software in cloud environments are permeating also communication infrastructures, eventually leading to the so-called “zero-touch network automation”. Such automation frameworks have been actively used in wireless testing infrastructures, such as FIRE and GENI. For instance, the continuous integration in wireless (COINS) technology development framework has been used in the LOG-a-TEC testbed facility of the FIRE federation. In this paper, we propose a task-based operational structure of COINS for a thorough evaluation of the time for operation (TFO) performance of each task. Moreover, we analyse the impact of different experiments and the number of nodes participating in the experiment on the TFO performance of the COINS framework. The results; i) demonstrate that the device tasks within COINS framework are the bottleneck occupying up to 92.6% of the total TFO, and ii) reveal that the TFO performances of the receive details task on the device side and the gather results task on the server side are significantly diminished with the increased number of nodes participating in the same type of experiment.

Referencia

Boskov, I., Yetgin, H., Fortuna, C., & Mohorcic, M. (2021). Performance evaluation of COINS Framework for wireless network Automation.

<https://doi.org/10.1109/eucnc/6gsummit51104.2021.9482454>

Datos

PI1. ¿Qué situaciones motivan la migración de las operaciones tradicionales a NetOps?

Las redes 5G y posteriores requieren asignación justa del tráfico de red y gestión de recursos eficiente para evitar congestiones en la red. Para ello, se utilizan bancos de pruebas con el fin de probar innovaciones tecnológicas antes de implementarlas en producción, sin embargo, los investigadores invierten grandes cantidades de tiempo y

esfuerzos en configurar el entorno de experimentación. Las herramientas DevOps permiten pruebas rápidas y fluidas a través de CI/CD, lo que conduce a la automatización de red

PI2. ¿Cuáles son los requisitos para desarrollar e implementar una solución NetOps?

PI3. ¿Qué avances y resultados se reportan en el desarrollo o la implementación de una solución relacionada con NetOps?

La introducción de prácticas DevOps como CI/CD para permitir una automatización de extremo a extremo se ha llevado a las redes inalámbricas en los últimos años, por ejemplo el framework COINS permite la implementación y ejecución automatizada de experimentos con soporte de CI/CD en bancos de pruebas inalámbricos.

COINS permite al experimentador centrarse en el desarrollo del experimento, selección y configuración de la topología y el análisis de datos, mientras que el resto de tareas se automatizan en el siguiente flujo:

1. Un experimentador realiza cambios en el repositorio de GitHub.
2. GitHub activa el controlador de experimentos, encargado de inicializar el entorno de experimentación y aprovisionamiento del firmware, sistemas operativos, extracción del código e instalación de dependencias.
3. Los detalles del experimento y el sistema operativo es distribuido entre los nodos.
4. El experimento se ejecuta en el nodo de destino y los resultados son mandados de vuelta al servidor para ser entregados en un repositorio accesible para el experimentador.

COINS aprovecha herramientas DevOps como Docker, Ansible y GitHub en la preparación de la imagen del experimento y su ejecución, lo que facilita la configuración del experimento

PI4. ¿Cuáles son las posibles limitaciones de NetOps y sus posibles alternativas?

Las redes están cada vez más virtualizadas y controladas por software, lo que permite la adopción de DevOps en infraestructura de comunicación al implementar CI/CD en infraestructura de pruebas inalámbricas, dando paso a tendencias como la denominada "zero-touch network automation", una tendencia en desarrollo que podría competir con la adopción de la automatización en la gestión de redes virtualizadas.

Los conceptos de DevOps están permeando cada vez más en bancos de pruebas inalámbricos e integrados así como en redes centrales comerciales, por ejemplo, la tendencia AIOps es una versión de automatización DevOps asistida con inteligencia artificial, dicha tendencia es aplicada también para la gestión de redes, volviéndola en una tendencia que comparte principios y objetivos con NetDevOps pero que difiere en sus métodos, prácticas y herramientas.

5GTANGO: An Approach for Testing NFV Deployments

Identificador

ER-02

Año

2018

Fuente

IEEE xplora

Tipo de publicación

Conference document

Palabras clave relacionadas con NetOps

Resumen

Programmability, control and flexibility can be considered as some of the indirect enablers for the success of 5G technologies. A key driver towards this are mechanisms or methodologies to drive shorter time to market for suppliers and operators of virtual network functions (VNFs) and network services. 5GTANGO includes a DevOps approach that can be utilized for the validation and verification (V&V) of VNFs and network services. In this paper, we elaborate further on the approaches made in the areas of testing, catalogues and package management as a means to enable that full DevOps V&V workflow. Finally, we explore the deployment requirement of the V&V via one of our pilot use cases.

Referencia

Twamley, P., Muller, M., Bok, P., Xilouris, G. K., Sakkas, C., Kourtis, M. A., Peuster, M., Schneider, S., Stavrianos, P., & Kyriazis, D. (2018). 5GTANGO: An Approach for Testing NFV Deployments. <https://doi.org/10.1109/eucnc.2018.8442844>

Datos

PI1. ¿Qué situaciones motivan la migración de las operaciones tradicionales a NetOps?

Los operadores de redes buscan un desarrollo más rápido y ágil para recuperar mercado frente a los proveedores de servicios "over-the-top", esto implica dejar los modelos de negocio tradicionales aplicados a una red de telecomunicaciones, para permitir el desarrollo de soluciones, se está adoptando el uso de NFV y SDN que a su vez, permiten la adopción de DevOps en el segmento de las telecomunicaciones.

PI2. ¿Cuáles son los requisitos para desarrollar e implementar una solución NetOps?

PI3. ¿Qué avances y resultados se reportan en el desarrollo o la implementación de una solución relacionada con NetOps?

La adopción de SDN y VFN permiten el uso de DevOps en las telecomunicaciones, esto se ve reflejado en proyectos como OSM y ONAP en los cuales DevOps respalda las implementaciones de nuevos servicios.

5GTANGO propone un framework de verificación y validación de VNF en todo el ciclo de vida de DevOps con el fin de evitar que una configuración incompleta o

inconsistente de nuevas VFN provoquen una avería de la infraestructura.

5GTANGO permite que tanto desarrolladores como operadores de red colaboren en el mismo flujo de trabajo, mientras que los desarrolladores crean y prueban un nuevo VNF, los operadores de red contarán con un mecanismo que permite el análisis automatizado de la información y resultados de las pruebas en dichos VNF. Todo esto, sin dejar de tener la flexibilidad de transferir el trabajo entre diferentes implementaciones al contar con un catálogo unificado de VNF y sus pruebas mediante paquetes firmados.

PI4. ¿Cuáles son las posibles limitaciones de NetOps y sus posibles alternativas?

Empowering Dynamic Task-Based Applications with Agile Virtual Infrastructure Programmability

Identificador

ER-03

Año

2018

Fuente

IEEE xplore

Tipo de publicación

Conference document

Palabras clave relacionadas con NetOps

Network topology, DevOps, programmable infraestructure

Resumen

The IaaS (Infrastructure-as-a-Service) offered by Clouds provides applications with the capability of customizing VMs and configuring their network. Compared to traditional service-based IaaS applications such as persistent web services, most task-based applications have a relatively short duration but are triggered on demand. A typical way to support such kinds of application is to provision a shared and fixed virtual infrastructure based on pre-estimated size in advance, and then perform all the processing tasks. However, due to unpredictable workloads, this solution can lead to either cost inefficiency caused by over-provisioning, or failure to deliver the performance required by applications. CloudsStorm is a dynamic control framework proposed to provide applications with agile programmability and flexibility in controlling the virtual infrastructure. With its front end, applications can design their networked infrastructure and program that infrastructure with our interpreted infrastructure code language. With the back-end engine, the infrastructure code can be executed to provision the networked infrastructure, deploy and execute the application to obtain results, and release resources. Moreover, we adopt multi-threading to support parallel operation. Finally, we conduct experiments in an assumed scenario to demonstrate functionalities of CloudsStorm. The evaluation results prove CloudsStorm is efficient for task-based applications that need to exploit Clouds but reduce the monetary cost.

Referencia

Zhou, H., Hu, Y., Su, J., Chi, M., De Laat, C., & Zhao, Z. (2018). Empowering Dynamic Task-Based Applications with Agile Virtual Infrastructure Programmability.
<https://doi.org/10.1109/cloud.2018.00068>

Datos

PI1. ¿Qué situaciones motivan la migración de las operaciones tradicionales a NetOps?

IaaS permite una gran flexibilidad en la configuración de máquinas virtuales y redes, sin embargo, también requiere que los desarrolladores tengan un conocimiento profundo de la planificación y configuración de la infraestructura, en particular cuando se utilizan diferentes centros de datos o nubes, estos problemas se vuelven urgentes cuando se habla del despliegue de aplicaciones basadas en tareas, que se ejecutan de manera distribuida y bajo demanda como en las aplicaciones científicas, por ejemplo, en predicciones de terremotos y secuencias de genoma.

Cuando se utiliza IaaS, los desarrolladores tienden a realizar tareas manuales como la configuración de los entornos de ejecución y las redes.

PI2. ¿Cuáles son los requisitos para desarrollar e implementar una solución NetOps?

PI3. ¿Qué avances y resultados se reportan en el desarrollo o la implementación de una solución relacionada con NetOps?

En los últimos años se han desarrollado aplicaciones para la implementación y aprovisionamiento de infraestructura que respaldan el enfoque DevOps como Puppet, Ansible, Chef, y concretamente en la nube.

CloudStorm es framework de control dinámico basado en un lenguaje interpretado de infraestructura para proporcionar aplicaciones con programabilidad ágil y flexibilidad en la infraestructura de red, aprovisionar la infraestructura, ejecutar la aplicación, extraer los resultados y liberar los recursos. CloudStorm trabaja en tres niveles: la topología general, las subtopologías creadas en cada nube y las máquinas virtuales para aprovisionar la aplicación, en donde, además, cuenta con soporte VNF para aprovisionar la infraestructura de red requerida.

PI4. ¿Cuáles son las posibles limitaciones de NetOps y sus posibles alternativas?

Towards Autonomic Policy-based Network Service Deployment with SLA and Monitoring

Identificador

ER-04

Año

2019

Fuente

IEEE xplora

Tipo de publicación

Conference document

Palabras clave relacionadas con NetOps

Network Function Virtualization, Software Defined Networking, DevOps

Resumen

Unlike previous generations, 5G will need to be faster, more efficient, reliable, flexible, agile, and, at the same time, cost less. For this to be possible, 5G has to engage with the best-of-breed of the emerging technologies, where NFV is definitely in the top list. In this context, this paper describes the SONATA Service Platform, an open source MANO framework extended in the scope of the 5GTANGO H2020 project. In particular, this demo presents some features that go beyond the state-of-the-art including Monitoring, Policy, and SLA Management

Referencia

G. Xilouris et al., "Towards Autonomic Policy-based Network Service Deployment with SLA and Monitoring," 2018 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN), Verona, Italy, 2018, pp. 1-2, doi: 10.1109/NFV-SDN.2018.8725695.

Datos

PI1. ¿Qué situaciones motivan la migración de las operaciones tradicionales a NetOps?

PI2. ¿Cuáles son los requisitos para desarrollar e implementar una solución NetOps?

PI3. ¿Qué avances y resultados se reportan en el desarrollo o la implementación de una solución relacionada con NetOps?

La versión 4.0 de SONOTA es un orquestador de funciones y servicios de red que permite el monitoreo, creación de políticas y gestión de SLA y abstracción de la infraestructura:

- a. Un desarrollador utiliza la API para incorporar un paquete de función o servicio de red recién creado.
- b. El operador define las características del servicio, en particular el SLA centrado en el cliente.
- c. Un desarrollador define políticas de monitoreo en tiempo de ejecución.
- d. El proveedor de servicios asigna políticas de servicio para cada función o servicio de red ofertado.
- e. El servicio es solicitado, se crea y es utilizado por el cliente, mientras, se recolectan métricas de monitoreo.
- f. En tiempo de ejecución, se puede realizar una ampliación del servicio o

determinar si ocurrió una violación del acuerdo de nivel de servicio.
Este flujo de trabajo corresponde con los principios de DevOps de colaboración entre desarrolladores y operadores, la automatización del ciclo de vida y el monitoreo y mejora continua.

PI4. ¿Cuáles son las posibles limitaciones de NetOps y sus posibles alternativas?

NetDevOps: A New Era Towards Networking & DevOps

Identificador

ER-05

Año

2020

Fuente

IEEE xplora

Tipo de publicación

Conference document

Palabras clave relacionadas con NetOps

Networking, NetDevOps, DevOps, programmability, Automation, Network configuration

Resumen

Today legacy networks are managed box by box and are CLI driven. Small changes often have large impacts, undesired behavior and very complex. The static network infrastructure deteriorates the application experience and cannot adapt to a change in application behavior. To avoid this chaos network should be made programmable using Software Defined Network (SDN) and Network Function Virtualization (NFV) while operation should be much more automated and simplified. New feature should be release and deliver by pressing a single button. Thus, this will increase revenue, cost reduction and cost avoidance. In this new paradigm, automated system to perform configuration changes carryout testing at each stage to ensure that the network changes do not result in downtime and finally push those configurations towards prospective devices. To summarize based on facts, network changes is directly correlated with increasing frequency of outages, with each outage can take up to many hours to resolve and majority of the network engineer are still manually verifying a network after changes are committed, which means that, we need a better way to make and verify network change, for which we enter in to the new era of Network configuration, programmability and automation. This new era of NetDevOps will embrace the best practices of adopting DevOps culture in the whole world of networking

Referencia

Shah, J. A., & Dubaria, D. (2019). NetDevOps: A New Era Towards Networking & DevOps.

<https://doi.org/10.1109/uemcon47517.2019.8992969>

Datos

PI1. ¿Qué situaciones motivan la migración de las operaciones tradicionales a NetOps?

Un estudio de Cisco demuestra que el 74% de los operadores de red informan que los cambios en la red han impactado significativamente su negocio, el 97% de los encuestados admite que los cortes en la red se deben a errores humanos, lo que lleva a la programabilidad y automatización de redes que ahorra tiempo, reduce el trabajo manual y los errores humanos, además de impulsar la innovación y personalización de servicios

PI2. ¿Cuáles son los requisitos para desarrollar e implementar una solución NetOps?

La infraestructura como código impulsa la adopción de la administración y aprovisionamiento de centros de datos informáticos a través de archivos de definición de dispositivos y sistemas.

PI3. ¿Qué avances y resultados se reportan en el desarrollo o la implementación de una solución relacionada con NetOps?

La infraestructura como código permite el almacenamiento de la configuración de red en repositorios Git, utilizando formatos legibles por máquina como JSON, YALM o XML, con este repositorio, los operadores y desarrolladores pueden desarrollar, probar e implementar un entorno de producción en un mismo código fuente unificado.

Siete casos de uso para la automatización y programabilidad de redes:

- a. aprovisionamiento de dispositivos
- b. Recopilación de datos
- c. Verificación de cumplimiento
- d. Migraciones
- e. Informes
- f. Solución de problemas
- g. Gestión de la configuración

Algunas herramientas utilizadas en NetDevOps son Git y Github; vagrant para controlar hipervisores e imágenes de máquinas virtuales; NSO NetSim el primer paso para la configuración en un proceso de NetDevOps; chef permite escribir infraestructura como código; Ansible permite automatizar todo el ciclo de vida del software; APIS y SDK como JDK, Cisco ACI, Cisco Nexus y Cisco USC y Cisco Tetration; CI/CD pipelines para implementar un enfoque de desarrollo y mejora continua en los cambios de red.

PI4. ¿Cuáles son las posibles limitaciones de NetOps y sus posibles alternativas?

Continuous Integration of Applications for ONOS

Identificador

ER-06

Año

2019

Fuente

IEEE xplora

Tipo de publicación

Conference document

Palabras clave relacionadas con NetOps

Network topology, SDN, Docker, Continuous integration.

Resumen

Micro-services dominate tomorrow's virtualized network infrastructures and are going to abolish monolithic software. Paradigms like fog computing and the Internet of Things require virtualized packages of software to be distributed all over the network to form a service chain that enhances the overall Quality of Service. Therefore, SDN controllers need to be extendable, flexible, updatable, and manageable, which necessitates automation in the controller's application development and deployment. Such novel agile service management up-rose due to the increasing demand of micro-service structures in data centers and nowadays falls into the scope of DevOps activities. In our paper, we build a sample application for ONOS and deploy it within a virtualized environment to different infrastructures. Additionally, we provide an evaluation of different Continuous Integration (CI) tools for building and deploying our ONOS controller application. Our investigation shows the applicability of current CI solutions regarding an automated build, test, and deploy chain for SDN controller applications.

Referencia

Grossmann, M., & Ioannidis, C. (2019b). Continuous Integration of Applications for ONOS.
<https://doi.org/10.1109/netsoft.2019.8806696>

Datos

PI1. ¿Qué situaciones motivan la migración de las operaciones tradicionales a NetOps?

Paradigmas como la computación en la niebla y el internet de las cosas requieren que se distribuyan paquetes virtualizados de aplicaciones por toda la red para generar una cadena de servicios que mejore la calidad del servicio en general, por lo tanto tecnologías como los controladores SDN deben ser ampliables, flexibles, actualizables y fáciles de usar, llevando a la automatización del desarrollo e implementación aplicaciones de controlador SDN.

PI2. ¿Cuáles son los requisitos para desarrollar e implementar una solución NetOps?

PI3. ¿Qué avances y resultados se reportan en el desarrollo o la implementación de una solución relacionada con NetOps?

Con la ayuda de tecnologías como Docker, mininet y herramientas CI, se puede crear un flujo de trabajo para el desarrollo de aplicaciones ONOS, por ejemplo, el desarrollo de una aplicación que funciona como un sistema de nombres de dominio (DNS), donde ONOS es un controlador de SDN que respalda la tendencia de una red totalmente programable y adaptable.

Para un enfoque CI en el desarrollo de aplicaciones de controlador SDN, se crea una red virtual de prueba que conecta varios dispositivos virtualizados por contenedores Docker, en donde se aprovisiona ONOS con la aplicación personalizada y en el segundo contenedor, una red de prueba con mininet que contiene vswitchs con OpenFlow para comunicarse con el controlador ONOS. La red virtual de mininet orquesta y programa la red a través de un script de python y un archivo de docker-compose, estos archivos de configuración son almacenados en un repositorio de GitHub y son implementados a través de herramientas de CI como Circle-CI, Gitlab-CI y Travis-CI.

Archivo de Docker-compose que organiza la construcción de aplicaciones personalizables en ONOS y su despliegue en un entorno de pruebas con Mininet

```
version: '3'
services:
  networks:
    controller:
      driver: bridge
      ipam:
        driver: default
        config:
          - subnet: 192.168.16.0/24
  onos:
    image: chrisioa/myonos:0.0.1
    environment:
      ONOS_APPS: openflow,pathpainter, project
      .ioannidis.onosApp
    networks:
      controller:
        ipv4_address: 192.168.16.99
  [...]

  mininet:
    image: unibaktr/mininet
    privileged: true
    volumes:
      - ./rundata/mininet/data:/data
    command: --mac --switch=ovsk,protocols=
      OpenFlow13 --controller=remote,ip
      =192.168.16.99,port=6653 --custom /
      data/topology.py --topo mytopo
  [...]
```

PI4. ¿Cuáles son las posibles limitaciones de NetOps y sus posibles alternativas?

Building an interactive Software Defined Network from the MPSI for MPLS Service provisioning with Gitlab and Ansible

Identificador

ER-07

Año

2022

Fuente

IEEE xplore

Tipo de publicación

Conference Document

Palabras clave relacionadas con NetOps

Ansible, Gitlab, SDN automation, MPSI,

Resumen

B5G (5G and beyond) networks are bringing numerous innovation possibilities that are challenging the statu-quo on the way Internet Service Providers' IP backbone networks are managed. Traditional CLI, per box, per supplier and per technology service provisioning methods are not adapted to new business requirements driven by a post severe-pandemic digital adoption and transformation. As a matter of fact, an effective IP service provisioning should be RFT (right-first time) at no time cost. We address this point (of and ever-increasing demands and requirements on backbones) by combining the programmability capability brought by Software Designed Networks and the automation from DevOps tools into an Interactive MPLS Provisioning Management System for SDN(IM-PMS-SDN). The system aims to bring new features on the Management plan Southbound Interface (MPSI) so MPLS service provisioning can be built in an interactive way between Gitlab, Ansible and the underlying Network Infrastructure; The end goal, presented in the experimental results, is to reduce to few minutes an end-to-end B2B service provisioning from prerequisite identification to acceptance tests.

Referencia

Seck, A., Bassene, C. S., Zabolo, S. E., & Ouya, S. (2022b). Building an interactive Software Defined Network from the MPSI for MPLS Service provisioning with Gitlab and Ansible. 2022 International Conference on Electrical, Computer, Communications and Mechatronics Engineering (ICECCME). <https://doi.org/10.1109/iceccme55909.2022.9988003>

Datos

PII. ¿Qué situaciones motivan la migración de las operaciones tradicionales a NetOps?

Los métodos tradicionales de prestación de servicios por CLI, caja, proveedor y tecnología no se adaptan a los nuevos requisitos comerciales impulsados por la adopción y transformación digital después de una pandemia grave, como el aprovisionamiento de nuevos servicios mediante RTF (right-first-time por sus siglas en inglés) sin coste de tiempo sobre redes troncales, para ello, se espera que la programabilidad de redes diseñadas por software y la automatización de herramientas DevOps permita solucionar estos problemas.

El tráfico empresarial creció un 100% entre 2019 y 2022 en los Estados Unidos y se

espera 224 millones de gigabytes en 2023, especialmente por el crecimiento del trabajo remoto, lo que aumenta los casos de uso y las necesidades comerciales que los ISP deben resolver, por ejemplo MPLS sigue siendo la principal tecnología de red troncal IP, sin embargo, para realizar cambios en la red y aprovisionamiento de nuevos servicios en estas redes, se pasa por un arduo proceso de gestión de cambios que puede llevar incluso días en completarse, lo que se ve reflejado directamente en la calidad de experiencia (QoE) de los clientes finales.

Los ISP están adoptando con mayor frecuencia enfoques de redes programables SDN con controladores como Open Day Light (ODL) y ONOS (open network operating system) para controlar el hardware subyacente.

La integración de DevOps en telco ha sido estudiada por Shah et al, en donde se demuestra la necesidad de herramientas de DevOps en la red, por ejemplo, el 69% de los encuestados utiliza comprobaciones manuales después de un cambio para verificar si la red funciona correctamente como se cita en Seck, Bassene, et. al.

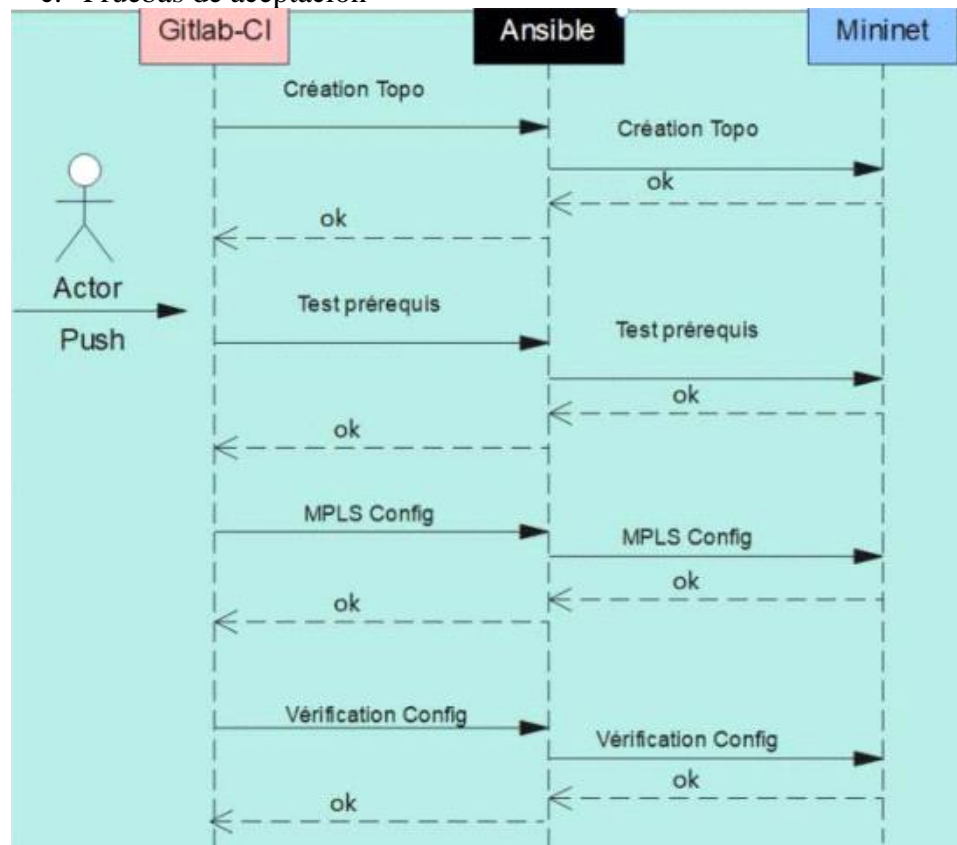
PI2. ¿Cuáles son los requisitos para desarrollar e implementar una solución NetOps?

PI3. ¿Qué avances y resultados se reportan en el desarrollo o la implementación de una solución relacionada con NetOps?

Actualmente se explora la posibilidad de implementar el aprovisionamiento de etiquetas MPLS en la interfaz de programación SDN Open Day Light (ODL) y combinar las canalizaciones CI de Gitlab, así como Ansible y Python para realizar pruebas de validación y el despliegue de topologías de red como código y el aprovisionamiento automático de dispositivos con etiquetas para MPLS.

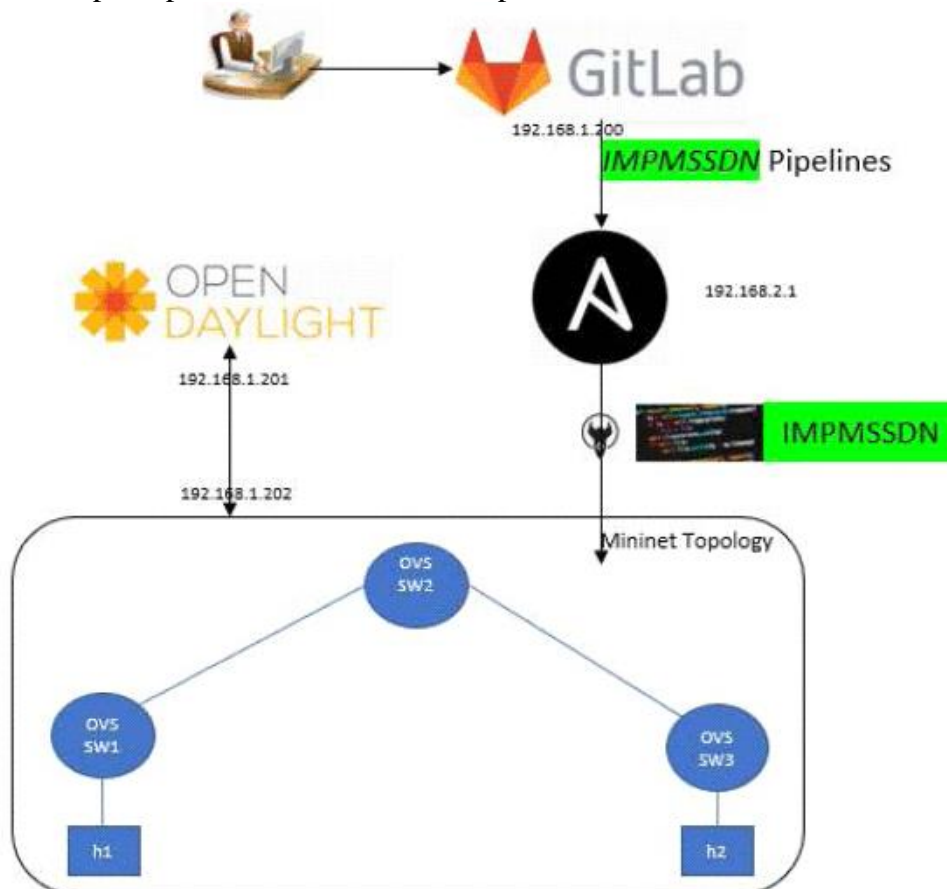
La propuesta de Seck, Bassene, et al. Consiste en tres tareas automatizadas con una canalización de CI:

- Pruebas de humno
- Aprovisionamiento automático de etiquetas MPLS
- Pruebas de aceptación



Para completar estas tareas, se realizó:

- Construcción del entorno CI/CD, este banco de pruebas se implementa en un servidor Gitlab local, con canalizaciones escritas en YAML, dichas canalizaciones son ejecutadas en contenedores Docker.
- Implementar IaaS en una máquina virtual con Mininet: el banco de pruebas se compone de tres open virtual switches y dos host que simulan dos sitios de una sucursal de clientes que necesitan MPLS para aprovisionar servicios de capa superior.
- Pruebas de microservicios de smoke ping: Un microservicio que valida el despliegue de la topología en la capa 2 y 3 antes de implementar un servicio de capa superior.
- Aprovisionamiento de etiquetas MPLS: Los OVS (open virtual Switch) se comunican por un puerto troncal mediante etiquetas; el aprovisionamiento se realiza mediante un playbook de Ansible que llama código Python encargado de las especificaciones de Open Flow.
- Interfaz para la validación de pruebas: el código python envía el comando tcpdump a las interfaces de OVS para verificar el funcionamiento de MPLS



PI4. ¿Cuáles son las posibles limitaciones de NetOps y sus posibles alternativas?

Open Disaggregated Optical Network Control with Network Management as Code

Identificador

ER-08

Año

2023

Fuente

IEEE xplora

Tipo de publicación

Conference document

Palabras clave relacionadas con NetOps

Network management as code, automation, network topology, optical fiber networks

Resumen

Network Management as Code is presented to manage the infrastructure and deployment of control functions in a declarative manner by adopting GitOps in open disaggregated optical networks. We introduce an Automation Engine that analyzes topology changes to trigger the Continuous Delivery procedure of SDN applications on a micro-service architecture.

Referencia

Errea, J., Quang, H. T., Verchere, D., Thieu, H. T., Mazzini, A., Abnaou, L., Pesic, J., Curtol, M., Imadi, A. E., Ksentini, A., & Zeglache, D. (2023). Open Disaggregated Optical Network Control with Network Management as Code. <https://doi.org/10.1364/ofc.2023.m3z.8>

Datos

PI1. ¿Qué situaciones motivan la migración de las operaciones tradicionales a NetOps?

PI2. ¿Cuáles son los requisitos para desarrollar e implementar una solución NetOps?

PI3. ¿Qué avances y resultados se reportan en el desarrollo o la implementación de una solución relacionada con NetOps?

Network Management as code es una solución de modelo operativo para configurar y gestionar la infraestructura y el despliegue de aplicaciones y funciones de control de forma declarativa para redes ópticas abiertas desagregadas impulsadas por open optical software defined network (OO-SDN), para ello, se utiliza el concepto GitOps para aprovechar la IaC controlada por Git y otras herramientas utilizadas ampliamente en DevOps como Terraform, Argo CD y Kubernetes para implementar CD y con ello mantener un estado consistente en la infraestructura ante cualquier cambio en las aplicaciones o configuración de la red, unificando la implementación de software y operaciones de infraestructura.

PI4. ¿Cuáles son las posibles limitaciones de NetOps y sus posibles alternativas?

Reproducible Functional Tests for Multi-scale Network Services

Identificador

ER-09

Año

2020

Fuente

IEEE xplora

Tipo de publicación

Conference document

Palabras clave relacionadas con NetOps

Network function virtualization, network services, functional networks, virtual network function, network configuration,

Resumen

Network functions virtualization (NFV) is developed to take advantage of virtualization technologies to separate network functions from the underlying hardware appliances. This approach brings a new level of flexibility in network services deployment for rapid composition, migration and scaling based on changing traffic load and current demand, leading to a higher QoS. Moreover, it allows introducing DevOps approach in the development process resulting in reduced life-cycle and shorter time-to-market. However, in order to realize the anticipated benefits of NFV, developers and network operators need mechanisms to adequately test Virtualized Network Functions (VNF) against different deployment scenarios and in different scales. To this end, we introduce a new library for light-weight automated functional testing of VNFs. It helps the developers to write functional tests in Python and run them on different platforms. Using the library, test developers choose which infrastructure to use, which network services and test VNFs to launch, how to interconnect them, how to trigger the test process and inspect, verify and validate the output against the expected values and conditions. Test developers can use network service packages to deploy services or can compose them in a test code using different parameters to simulate various deployment scenarios. Finally, the library can be used to set automated testing in CI/CD environments or can be used locally during the development phase. In this paper, we describe the architecture of the library, the basic workflow and give an example of testing multiple flavors of a network service using the same code

Referencia

Nuriddinov, A., Tavernier, W., Colle, D., Pickavet, M., Peustery, M., & Schneidery, S. (2019). Reproducible Functional Tests for Multi-scale Network Services. <https://doi.org/10.1109/nfv-sdn47374.2019.9039968>

Datos

PI1. ¿Qué situaciones motivan la migración de las operaciones tradicionales a NetOps?

La virtualización de funciones de red (NFV por sus siglas en inglés) se desarrollan para separar las funciones de red de los dispositivos de hardware, aumentando la flexibilidad en la implementación, migración y escalabilidad de servicios de red en función de cargas de tráfico dinámicas para ofrecer una mayor calidad de servicio (QoS por sus siglas en inglés) a través de nuevas formas de diseñar, implementar y gestionar servicios de red, sin embargo, los desarrolladores y operadores de red deben probar las

NFV antes de desplegarlas en producción, para ello, DevOps implementa pruebas automatizadas, haciendo que el modelo de funciones de red monolíticas, tradicionales y de un solo proveedor, así como los métodos de verificación deban evolucionar para cubrir un alcance más amplio de configuraciones y escenarios reales.

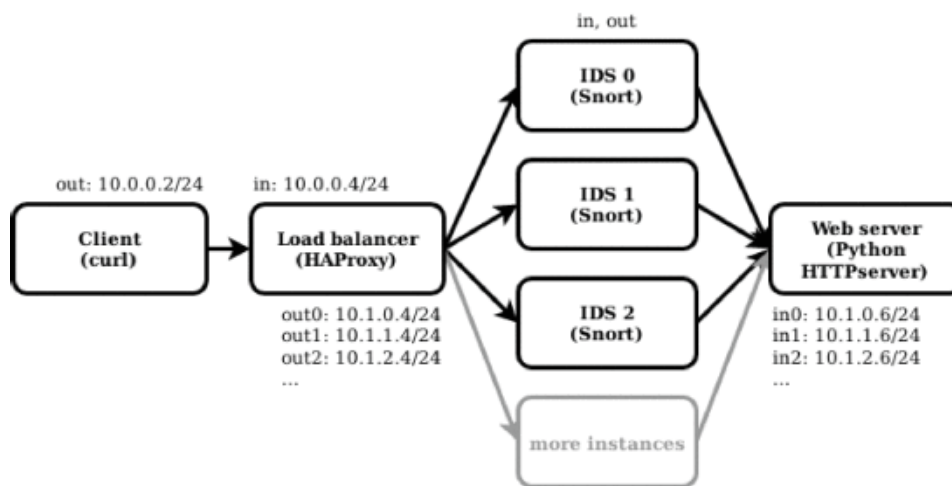
Se pronosticó que en 2021 habría el triple de tráfico de red por mes en comparación con 2019, además, el 75% del tráfico de red provendría de dispositivos móviles e inteligentes, así como la agregación de nuevas tecnologías como realidad virtual, aumentada y IoT que introducen nuevos requisitos en las redes de comunicación, aumentando la necesidad de contar con redes flexibles, escalables, ágiles y programables para implementar diferentes servicios con distintos requisitos.

PI2. ¿Cuáles son los requisitos para desarrollar e implementar una solución NetOps?

PI3. ¿Qué avances y resultados se reportan en el desarrollo o la implementación de una solución relacionada con NetOps?

Nuriddinov, Tavernier, Colle, et. Al. Proponen una biblioteca para pruebas funcionales de VNF, con el fin de automatizar las pruebas mediante canalizaciones de CI/CD al presentar una interfaz unificada de infraestructura. Mediante dicha interfaz, el desarrollador puede definir la infraestructura a utilizar para la prueba o puede dejar que la biblioteca realice la asignación mediante variables de entorno. Este conjunto de funciones permite componer, iniciar y conectar un VNF de pruebas personalizados o seleccionar alguno previamente instalado en la biblioteca como curl, netcat, netrw, socat y scapy al servicio a probar, estimular la prueba mediante ejecución de comandos y scripts y recuperar los resultados de las fuentes.

Un estudio de caso propone el uso de la biblioteca de automatización de pruebas de VNF para probar un servicio web compuesto por un balanceador de carga, varias instancias de IDS con capacidad de escalabilidad y un servidor web. El código de la prueba y la arquitectura del servicio se presentan a continuación.



```

1 from tangotest.vim import vim_from_env
2
3 def deploy_service(vim, scale):
4     client_net = {'out': '10.0.0.2/24'}
5     lb_net = {'in': '10.0.0.4/24'}
6     server_net = {}
7     snort_net = ['input', 'output']
8     for i in range(scale):
9         lb_net['out%d' % i] = '10.1.%d.4/24' % i
10        server_net['in%d' % i] = '10.1.%d.6/24' % i
11
12    vim.add_instance_from_image('client', 'client', client_net)
13    vim.add_instance_from_image('server', 'server', server_net)
14    vim.add_instance_from_image('lb', 'lb', lb_net)
15    vim.add_link('client', 'out', 'lb', 'in')
16
17    for i in range(scale):
18        name = 'snort%d' % i
19        vim.add_instance_from_image(name, 'snort', snort_net)
20        vim.add_link('lb', 'out%d' % i, name, 'in')
21        vim.add_link(name, 'out', 'server', 'in%d' % i)
22        vim.lb.execute('/add-server 10.1.%d.6' % i)
23
24 def test_service(scale, load_factor=2):
25     with vim_from_env() as vim:
26         deploy_service(vim, scale)
27         ip = vim.lb.get_ip('in')
28
29         bad_cmd = 'curl -s %s/restricted/' % ip
30         good_cmd = 'curl -s %s' % ip
31
32         for _ in range(load_factor * scale):
33             vim.client.execute(bad_cmd)
34
35         for _ in range(scale):
36             code, output = vim.client.execute(good_cmd)
37             assert code == 0
38             assert output.strip() == 'Home Page'
39
40         for i in range(scale):
41             inst = vim.instances['snort%d' % i]
42             records = inst.get_file('/logfile').splitlines()
43             alerts = [i for i in records if 'alert' in i]
44             assert len(alerts) == load_factor

```

PI4. ¿Cuáles son las posibles limitaciones de NetOps y sus posibles alternativas?

Introducing Development Features for Virtualized Network Services

Identificador

ER-10

Año

2018

Fuente

IEEE xplora

Tipo de publicación

Journal

Palabras clave relacionadas con NetOps

Network function virtualization, virtual network function, network infrastructure, software defined network, softwarizing network, devops,

Resumen

Network virtualization and softwarizing network functions are trends aiming at higher network efficiency, cost reduction and agility. They are driven by the evolution in Software Defined Networking (SDN) and Network Function Virtualization (NFV). This shows that software will play an increasingly important role within telecommunication services, which were previously dominated by hardware appliances. Service providers can benefit from this, as it enables faster introduction of new telecom services, combined with an agile set of possibilities to optimize and fine-tune their operations. However, the provided telecom services can only evolve if the adequate software tools are available. In this article, we explain how the development, deployment and maintenance of such an SDN/NFV-based telecom service puts specific requirements on the platform providing it. A Software Development Kit (SDK) is introduced, allowing service providers to adequately design, test and evaluate services before they are deployed in production and also update them during their lifetime. This continuous cycle between development and operations, a concept known as DevOps, is a well known strategy in software development. To extend its context further to SDN/NFV-based services, the functionalities provided by traditional cloud platforms are not yet sufficient. By giving an overview of the currently available tools and their limitations, the gaps in DevOps for SDN/NFV services are highlighted. The benefit of such an SDK is illustrated by a secure content delivery network service (enhanced with deep packet inspection and elastic routing capabilities). With this use-case, the dynamics between developing and deploying a service are further illustrated.

Referencia

Van Rossem, S., Tavernier, W., Colle, D., Pickavet, M., & Demeester, P. (2018b). Introducing development features for virtualized network services. IEEE Communications Magazine, 56(8), 184–192. <https://doi.org/10.1109/mcom.2018.1600104>

Datos

PI1. ¿Qué situaciones motivan la migración de las operaciones tradicionales a NetOps?

PI2. ¿Cuáles son los requisitos para desarrollar e implementar una solución NetOps?

La virtualización de redes y function network softwarizing, impulsadas por redes

definidas por software (SDN por sus siglas en inglés) y la virtualización de funciones de red (NFV por sus siglas en inglés) permiten que los servicios de telecomunicaciones introduzcan nuevos servicios al combinarlos con un conjunto ágil de desarrollo, implementación y mantenimiento antes de llevarlo a producción y durante su vida útil, habilitando una dinámica entre el desarrollo y la implementación de un servicio SDN/NFV.

SDN y NFV cambian fundamentalmente la forma en cómo se implementan servicios de red: NFV abstrae la funcionalidad de hardware de red y la transforma en software como el filtrado y reenvío de paquetes, traducción de direcciones de red (NAT), gestión de calidad de servicios (QoS), entre otros, mientras que SDN permite un control más flexible y más sencillo de la red entre NFV, al implementar un plano de control que instruye a los dispositivos de reenvío de paquetes utilizando protocolos como openflow y NETCONF. La infraestructura ahora está virtualizada, lo que permite formas más específicas de consumir recursos informáticos de almacenamiento y acceso a la red mientras se obtiene una gestión de la red centralizada y programable.

Existen tres áreas para el desarrollo de servicios SDN/NFV: lenguajes de programación y herramientas de software en general como Python, Go, C, C++, HTML, etc; tecnologías, técnicas y estándares basados en SDN/NFV como OpenFlow, OpenDaylight, ONOS, Snort, pfsense, TOSCA, ETSI-NFV Azure CLI, Amazon AWS CLI, Mininet, etc; y las plataformas de servicios como gestión y orquestación (MANO por sus siglas en inglés), OSM, SONOTA, ONAP, Docker, Docker swarm, NetFlow, Sflow, Nagios, Jenkins, Travis, etc;

PI3. ¿Qué avances y resultados se reportan en el desarrollo o la implementación de una solución relacionada con NetOps?

Van Rossem, Tavernier, Colle, et. Al. Proponen un kit de desarrollo de software (SDK) para servicios SDN/NFV, tomando en cuenta tres actores: el desarrollador del servicio, el operador o proveedor del servicio y los proveedores de infraestructura, promoviendo una integración de las diferentes tecnologías empleadas en dichos servicios para cerrar la brecha entre los desarrolladores y los operadores de servicios.

Van Rossem, et. Al. Proponen una arquitectura de una red de entrega de contenido utilizando servicios SDN/NFV susceptibles de ser gestionados con CI/CD, al mismo tiempo que las características de monitoreo continuo y automatización permiten que esta arquitectura pueda escalar dinámicamente en función del acuerdo de nivel de servicio (SLA) donde se propone utilizar funciones de router, inspección profunda de paquetes (DPI) y chace mediante NFV en redes perimetrales.

Al utilizar el SDK propuesto, el desarrollador puede desarrollar pruebas teniendo en cuenta los requisitos de los operadores mediante pruebas y monitoreo continuo, reempaquetando el paquete con nuevas actualizaciones en un flujo CI/CD.

PI4. ¿Cuáles son las posibles limitaciones de NetOps y sus posibles alternativas?

Una serie de desafíos puede frenar la adopción de herramientas como es el caso de este SDK al ecosistema de las telecomunicaciones como:

- Es necesario cultivar una cultura DevOps entre operadores y desarrolladores de SDN/NFV

- La plataforma de gestión y orquestación y las herramientas del operador deben admitir los mismos formatos de paquetes de servicios.

- Se debe considerar la autenticación y autorización de los operadores que interactúan con estas herramientas y se deben detectar vulnerabilidades de funciones de control de servicios antes de su despliegue, por ejemplo, escalamiento

infinito o acceso no autorizado a recursos.

Infrastructure-as-Code in Open-Networking: Git, Ansible, and Cumulus-Linux Case Study

Identificador

ER-11

Año

2023

Fuente

IEEE xplora

Tipo de publicación

Conference document

Palabras clave relacionadas con NetOps

NetDevOps, network automation server, ansible, git, vxlan

Resumen

One of the necessities for modern connectivity infrastructures is decreasing devices setup-times and giving configuration visibility to troubleshoot complex scenarios more easily. This proof-of-concept and case study scenario demonstrates programmability in Open-Networking through an emulation of Cumulus Linux as an Open NOS (Network Operating System) with Ansible acting as a NAS (Network Automation Server), and Git as a VCS (Version Control System) for networking scripts, giving a global view of the technological evolution in data networks, focusing on the importance of IaC (Infrastructure-as-a-Code) and NetDevOps.

Referencia

Salazar-Chacón, G., & Parra, D. M. (2023). Infrastructure-as-Code in Open-Networking: Git, Ansible, and Cumulus-Linux Case Study. <https://doi.org/10.1109/ccwc57344.2023.10099084>

Datos

PI1. ¿Qué situaciones motivan la migración de las operaciones tradicionales a NetOps?

Existe una necesidad de reducir los tiempos de configuración de los dispositivos de red y aumentar la visibilidad de dicha configuración, para ello, se aprovecha la infraestructura como código (IaC por sus siglas en inglés) y de tendencias como NetDevOps.

Los efectos postpandemia han demostrado la necesidad de más infraestructura de telecomunicaciones para facilitar la transformación digital, sin embargo, la infraestructura de red de muchas empresas es obsoleta y se mantiene bloqueada por software y hardware patentado, limitando la innovación de entornos informáticos abiertos.

El tradicional ciclo de vida de las redes PPDIOO propuesto por Cisco ahora exige más agilidad en todas las fases para alinearse con metodologías como Scrum y Agile que afectan directamente las fases de preparación, planificación y diseño (PPD), mientras que las fases de implementación, operación y optimización requieren de nuevas tecnologías para usar la programabilidad y automatización, dando paso a una nueva cultura

empresarial llamada networking development operations o NetDevOps.

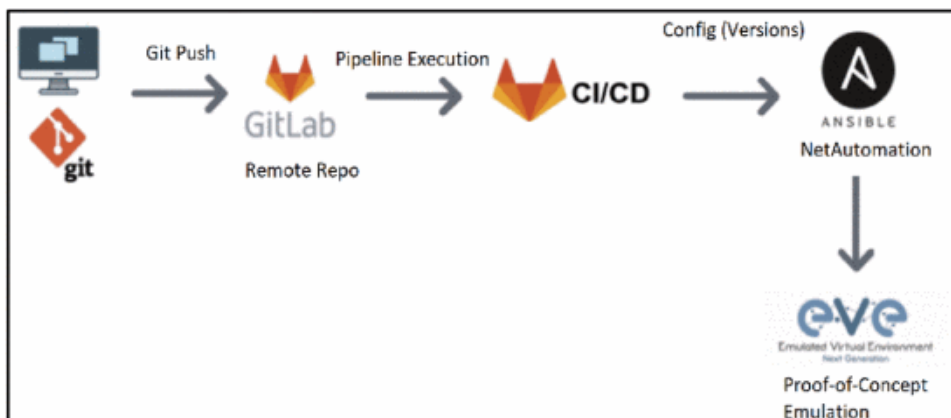
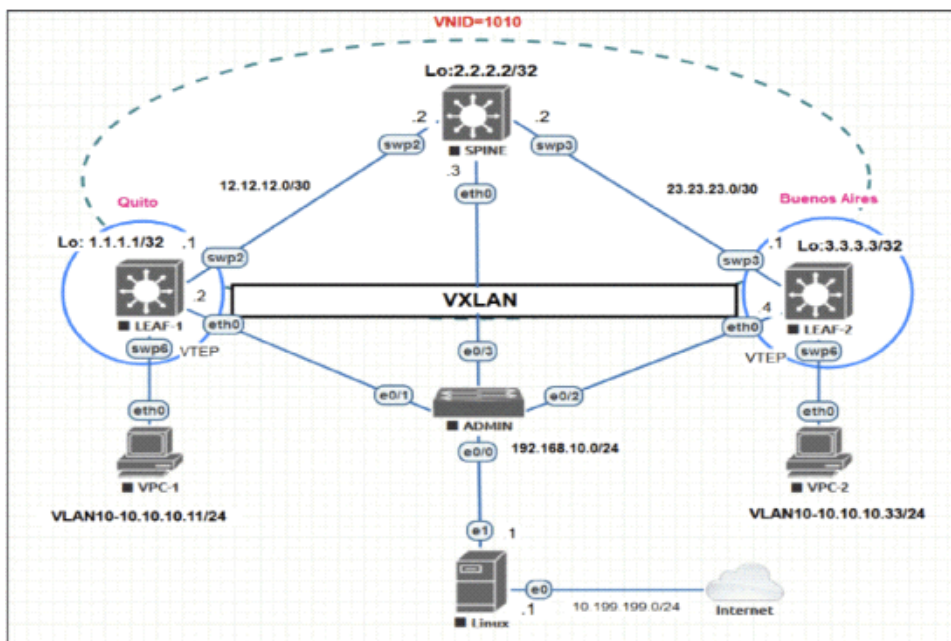
PI2. ¿Cuáles son los requisitos para desarrollar e implementar una solución NetOps?

NetDevOps implica la automatización de los cambios en la configuración de red, infraestructura como código, integración continua, monitoreo automatizado y encamina las operaciones hacia una red basada en intenciones (IBN por sus siglas en inglés)

PI3. ¿Qué avances y resultados se reportan en el desarrollo o la implementación de una solución relacionada con NetOps?

Salazar-Chacón, Parra, et. Al. Proponen un caso de estudio para la viabilidad de la configuración de Cumulus-Linux como VXLAN en un escenario IaaS como parte de un entorno NetDevOps.

El escenario es una topología VXLAN Spine-Leaf con tres dispositivos Cumulus-Linux L2/L3, un nodo de automatización de red Ansible, un repositorio GitLab remoto con una pipeline de CI/CD que ejecuta las rutinas del servidor Ansible para los cambios de configuración registrados en el repositorio. En dicho repositorio es posible realizar varias ramas para realizar pruebas en un entorno emulado (desarrollo continuo) antes de pasar a producción.



PI4. ¿Cuáles son las posibles limitaciones de NetOps y sus posibles alternativas?

Debido a la diversidad cambiante de configuraciones en los servidores del entorno de TI de las organizaciones, no es posible mantener guías estáticas de Ansible, por lo que se propone adoptar plantillas Jinja2 con Ansible para tener una mayor flexibilidad en la

configuración.

A NEAT way to test-driven network management

Identificador

ER-12

Año

2022

Fuente

IEEE xplore

Tipo de publicación

Conference document

Palabras clave relacionadas con NetOps

Network management, network topology, network operations, network configuration, network infrastructure, CI/CD, NetDevOps

Resumen

The increasing softwarization of network infrastructures introduces an important challenge for network configuration. The growth of the network configuration space as a result of new device types and the expanding inter-dependence of network service components increases the network configuration complexity. Compounding this issue, new service deployment architectures lack mechanisms to validate the impact of service configuration on network resilience. Network operators need to adopt new mechanisms to validate and verify network configuration changes, taking inspiration from popular Continuous Integration/Continuous Development (CI/CD) mechanisms. This paper introduces Network Emulation-based Automated Testing (NEAT), an automated testing framework for network configuration. NEAT allows network managers to define network topologies and tests through YAML files, to then be later executed by NEAT within realistic network topologies. Furthermore, network managers can control the fidelity of their network tests and bound the execution time of testing suites, as well as exploit parallelization of modern servers to speedup test execution.

Referencia

Fantom, W., Alcock, P., Simms, B., Rotsos, C., & Race, N. (2022). A NEAT way to test-driven network management. NOMS 2022-2022 IEEE/IFIP Network Operations and Management Symposium. <https://doi.org/10.1109/noms54207.2022.9789909>

Datos

PI1. ¿Qué situaciones motivan la migración de las operaciones tradicionales a NetOps?

La llamada softwarizing de la infraestructura de red introduce un nuevo desafío para la configuración de la red, así como el crecimiento del espacio de configuración como consecuencia de los nuevos dispositivos de red y la interdependencia de los componentes de los servicios. Por ello, los operadores de red deben adoptar nuevos mecanismos de verificación y validación en los cambios de la configuración inspirándose en mecanismos como CI/CD para habilitar pruebas automatizadas en la configuración de redes.

Tradicionalmente, los operadores de red utilizan laborations de pruebas con dispositivos

reales mientras inyectan verificadores basados en tiempo y declaraciones de verificación, sin embargo, estos enfoques centrados en la persona carecen de respaldo para adoptar el dinamismo necesario en nuevos servicios de red.

PI2. ¿Cuáles son los requisitos para desarrollar e implementar una solución NetOps?

PI3. ¿Qué avances y resultados se reportan en el desarrollo o la implementación de una solución relacionada con NetOps?

NEAT es un sistema de automatización de pruebas de red con el objetivo de promover DevOps de red mediante CI/CD, para este propósito, NEAT implementa emulación de red mediante Mininet en un contenedor Docker y con soporte para máquinas virtuales NEAT expone una API REST para Mininet para brindar información de la topología emulada, dicha topología de red es definida en un archivo YAML. En dicho archivo, las pruebas son definidas mediante bloques, valores mutables y expresiones a evaluar.

En primer lugar, los conjuntos de pruebas deben permitir a los usuarios controlar el nivel de realismo y admitir topologías de red emuladas de complejidad variable. En segundo lugar, los operadores y desarrolladores deben ser capaces de definir conjuntos de pruebas repetibles y los datos producidos a partir de estas pruebas deben evaluarse fácilmente mediante políticas de prueba (por ejemplo, salida binaria). Por último, todos estos objetivos deben ser capaces de ser ejecutados localmente por los desarrolladores, pero también permitir la integración con sistemas de CI/CD populares para soportar grandes pruebas con altas demandas de recursos que no pueden satisfacerse localmente. Todos estos objetivos deben limitar al mínimo la necesidad de codificación.

```
1  topologies:
2    - name: clickos-loadbalancer
3      topology: cdn-topo
4      assets: ["$(pwd)/click-images"]
5      libvirt: true
6      post_start_script: ./install_routers.sh
7  blocks:
8    - name: h1 connect h4
9      variant: ping
10     topologies: [clickos-loadbalancer]
11     mutables: { sender: h1, target: h4, count: 5}
12     expressions:
13       - Sent == Received
```

PI4. ¿Cuáles son las posibles limitaciones de NetOps y sus posibles alternativas?

Un gran desafío para la implementación de network softwarazing es la falta de mecanismos para validar de forma integral la configuración de red, este problema crece a medida que los dispositivos heredados necesitan coexistir con nuevas tecnologías de redes en el futuro.

Automated VPN configuration using DevOps

Identificador

ER-13

Año

2022

Fuente

scienceDirect

Tipo de publicación

Journal

Palabras clave relacionadas con NetOps

Automation, vpn, configuration, devops, agile,

Resumen

Enterprise networks are becoming increasingly sophisticated and large in scale due to the critical need for interconnectivity. For the interconnection of sites, VPN technology is essential. Indeed, this technology allows a partially or completely meshed connection between the various sites in a secure way. IPsec is one of the most widely deployed VPN technologies due to its many advantages, including data confidentiality, integrity and authentication. However, implementing this technology requires considerable technical expertise given the diversity of gateway manufacturers that a company may have, advanced engineering given the set of technical parameters that a VPN tunnel may have for its proper functioning, and caution when setting up a large-scale network given that a simple error may prevent the creation of tunnels. Taking these limitations into account, the automation of IT infrastructures has become indispensable, known as DevOps, which promotes continuous communication, collaboration, integration, visibility and transparency between the teams responsible for application development (Dev) and those responsible for IT operations (Ops). With infrastructure automation, networks are becoming easier to manage, diagnose and configure. This paper proposes a new architecture that automates the deployment of VPN tunnels via a web-based graphical interface. This architecture is adapted with a variety of equipment manufacturers and delivers configurations generated via an SSH channel in an automatic way.

Referencia

Firdaouss, L., Ayoub, B., Manal, B., & Ikrame, Y. (2022). Automated VPN configuration using DevOps. *Procedia Computer Science*, 198, 632–637. <https://doi.org/10.1016/j.procs.2021.12.298>

Datos

PI1. ¿Qué situaciones motivan la migración de las operaciones tradicionales a NetOps?

Las redes empresariales necesitan una escala mayor de sus recursos de red debido a la necesidad crítica de conectividad por VPN como consecuencia del aumento del teletrabajo en temporada de Covid-19, sin embargo, la implementación de esta tecnología requiere de profundos conocimientos técnicos para el correcto funcionamiento del tunel, considerando que se extrema la precaución de la configuración pues, un error podría impedir la creación de túneles en una gestión a gran escala. Por si fuera poco, la enorme cantidad de fabricantes dificulta aún más la configuración, dando como resultado que para desplegar una VPN entre 10 sitios con malla completa, requiera al

menos de 3 horas de configuración.

PI2. ¿Cuáles son los requisitos para desarrollar e implementar una solución NetOps?

Para solucionar este problema, se utiliza la automatización de infraestructura de TI, una práctica definida por DevOps, donde además de ofrecer ahorros de tiempo y evitar errores de configuración, permite la comunicación continua, colaboración y la integración de la visibilidad entre equipos de desarrollo y operaciones.

Para realizar una arquitectura de despliegue automático de VPN y siguiendo el trabajo propuesto, es necesario un controlador SDN, dispositivos de enrutamiento sin importar el fabricante, que posean la capacidad de crear túneles VPN y una aplicación web que permita gestionar y almacenar los datos de configuración de los túneles VPN.

PI3. ¿Qué avances y resultados se reportan en el desarrollo o la implementación de una solución relacionada con NetOps?

Se propone un modelo SDN híbrido que permite conectar equipos de enrutamiento de diferentes proveedores a través de VPN automáticamente. Para esto se piensa en un modelo de tres capas, donde la primera capa corresponde con el aplicativo web para que el usuario final configure la VPN; la segunda capa es el plano de control, que define el funcionamiento del modelo y finalmente, la capa 3 es el plano de datos, la cual aloja los equipos físicos de la infraestructura. La segunda capa a su vez, se compone de 9 fases:

Envío de información sobre la arquitectura para el despliegue del tunel VPN.

Generar la configuración de las conexiones, el usuario proporciona los detalles como dirección IP, credenciales de acceso SSH.

Generar parámetros IKE e Ipsec: El usuario proporciona la configuración de seguridad y cifrado para el tunel,

Configuración de políticas del tunel: el usuario elige el tipo de tráfico, puertos, y las ACL.

Configurar rutas estáticas

Detectar el SO del router: mediante SNMP, se recolecta el OID del dispositivo para determinar los parámetros que se deben configurar

Guardar los atributos recolectados en una base de datos

Verificar la disponibilidad de SSH

Preparar y entregar las configuraciones VPN

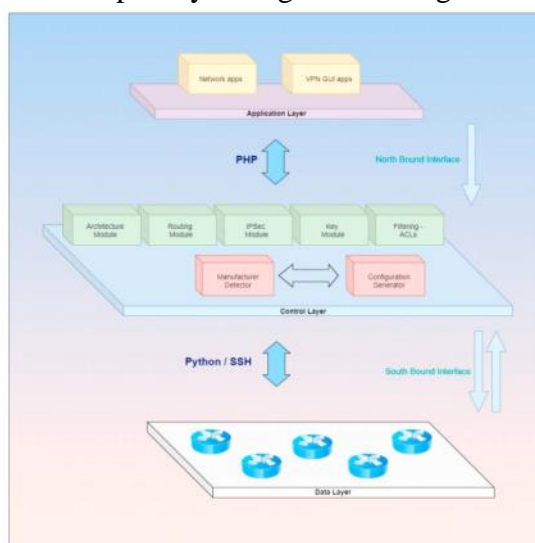


Figura 2 Arquitectura SDN híbrida para la gestión de túneles VPN

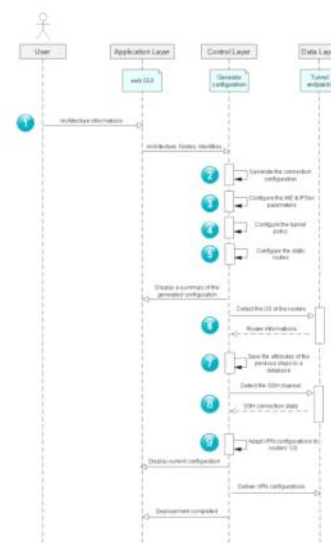


Figura 3 Diagrama de secuencia UML para la gestión de túneles VPN

IKEI informations

Encryption Algorithm *
DES

Hashing Algorithm *
MD5

Key Length *
Diffie-Hellman group 1

Lifetime (s) *
8640

Password *

Tunnel Policy development

Traffic type *
FTP

or specify manually the port number *

☒ TCP ☐ UDP

Previous Next

Summary

General Informations I

SITE	IP ADDRESS	PUBLIC INTERFACE	USERNAME	PASSWORD
site I	1110	fa/0	user1	***** show
site II	2220	fa/0	user2	***** show

General Informations II

SITE	HOSTNAME	PUBLIC IP	PRIVATE IP
site I	1AM	1111	192.168.21/24
site II	ORANGE	2221	192.168.22/24

IKE I

ENCRYPTION ALGORITHM	HASHING ALGORITHM	KEY LENGTH	LIFE TIME	PASSWORD
DES	MD5	group 1	8640	***** show

IKE II

PROTOCOLE	ENCRYPTION ALGORITHM	HASHING ALGORITHM	TUNNEL MODE
IPSEC	DES	MD5	tunnel

Tunnel Policy Developpement

TRAFFIC TYPE	PORT NUMBER	PROTOCOLE
FTP	-	-

Send

PI4. ¿Cuáles son las posibles limitaciones de NetOps y sus posibles alternativas?

Implementation of an Agile SDLC CI/CD pipeline for managing a SDN VXLAN-EVPN fabric

Identificador

ER-14

Año

2023

Fuente

IEEE xplora

Tipo de publicación

Conference document

Palabras clave relacionadas con NetOps

DevOps, software defined networking, network management, CI/CD, network automation

Resumen

In this paper, a DevOps oriented approach to managing a Software Defined Networking (SDN) based massively scalable data center fabric, using automation and software development principles is proposed and implemented. An Agile based methodology for the software development lifecycle (SDLC) of the coded infrastructure is implemented allowing for a flexible workflow compared to the traditional methods like Waterfall. Finally, it is demonstrated how the SDN based fabric can in the same way be extended into the top-of-rack access layer using the same workflow for non-SDN compliant devices, providing a unified method for automated network management and deployment.

Referencia

Georgiev, S., & Nikolova, K. (2023). Implementation of an Agile SDLC CI/CD pipeline for managing a SDN VXLAN-EVPN fabric. <https://doi.org/10.1109/telecom59629.2023.10409668>

Datos

PI1. ¿Qué situaciones motivan la migración de las operaciones tradicionales a NetOps?

Se puede aprovechar DevOps para gestionar el tejido de red de tecnologías como VXLAN-EVPN para centros de datos masivos y escalables basadas en redes definidas por software (SDN) al implementar principios de desarrollo de software agile y automatización.

Redes definidas por software (SDN) es un nuevo paradigma en la gestión de redes tradicionales, SDN desacopla los dispositivos del plano de control, interactuando con los dispositivos y aplicaciones de gestión de terceros mediante API para implementar un método unificado de administración e implementación de redes automatizadas.

PI2. ¿Cuáles son los requisitos para desarrollar e implementar una solución NetOps?

Cuando se aplica gestión de infraestructura y automatización, existen herramientas de

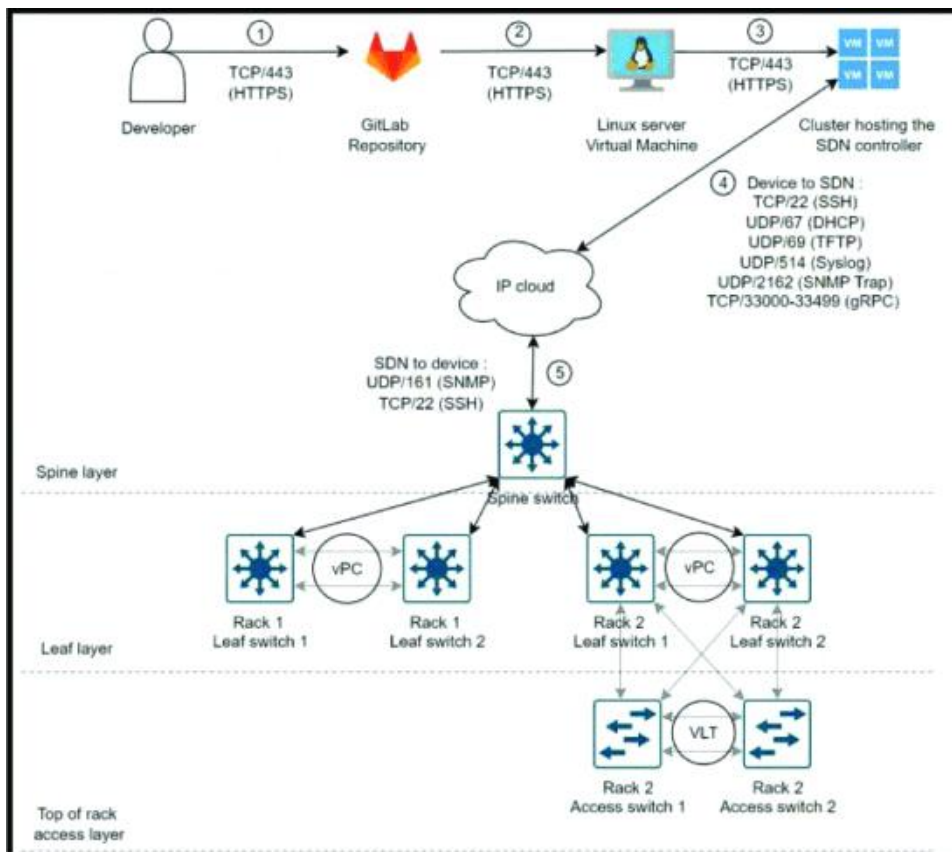
Network DevOps (NetDevOps) que pueden clasificarse como:

- Automatización de infraestructura mediante IaC dentro de CI/CD
- Gestión de configuración con herramientas como Ansible o Nornir
- Orquestación de implementación para entrega continua de infraestructura codificada
- Pruebas de validación de red como herramientas como Genie y PyATS
- Monitoreo y agregación de registros

PI3. ¿Qué avances y resultados se reportan en el desarrollo o la implementación de una solución relacionada con NetOps?

Para implementar un VXLAN basado en SDN con principios de desarrollo de software ágiles, se empleó el controlador SDN de Cisco Systems, Data Center Network Management (DCNM), IaC mediante Ansible y una canalización de CI/CD mediante un servidor Git y GitLab.


Un banco de pruebas para una implementación de red SDN VXLAN BGP-EVPN administrado con ansible y CI/CD utiliza una topología de red spine/leaf, con una capa superior no administrada por SDN, sino por infraestructura como código (IaC). La red física utiliza virtual port channel (VPC) y Virtual Link Trunking (VLT); un cluster virtualizado para alojar recursos del controlador SDN y el mismo controlador; Un servidor Linux para las tareas de automatización (GitLab).



```
main ▾ dcnm-templates / templates.yaml

templates.yaml 363 bytes

1 ---
2 - hosts: dcnm controllers
```

The screenshot shows a code editor interface for a file named `templates.yaml` located in the `dcnm-templates` repository. The file is 363 bytes in size. The code is written in YAML and defines an Ansible task to create a template policy. The task is named "Create template policy using Ansible" and uses the `cisco.dcnm.dcnm_template` module. The module's state is set to `merged` and its configuration includes a sub-task named `ansible_template` with a description "Test Ansible Template", a tag "ansible", and a content field that uses a Jinja2 lookup to read the content of `./templates/content.txt`.

```
1 ---
2 - hosts: dcnm_controllers
3   gather_facts: false
4   tasks:
5     - name: Create template policy using Ansible
6       cisco.dcnm.dcnm_template:
7         state: merged
8         config:
9           - name: ansible_template
10             description: "Test Ansible Template"
11             tags: "ansible"
12             content: "{{ lookup('file', './templates/content.txt') }}"
```

PI4. ¿Cuáles son las posibles limitaciones de NetOps y sus posibles alternativas?

Low Impact Tenant Code Updates on Multi-Tenant Programmable Switches

Identificador

ER-15

Año

2023

Fuente

IEEE xplore

Tipo de publicación

Conference document

Palabras clave relacionadas con NetOps

Programmable switches, continuous delivery, network infrastructure, network programmability,

Resumen

Software-defined Networking (SDN) and Programming Protocol-Independent Packet Processors (P4) introduced data processing within the network data plane. To offer multiple tenants to deploy individual code in programmable switches and network devices, code updates must ensure proper tenant isolation and minimal negative cross-tenant impact during code updates. Thus, this paper presents a code deployment pipeline, primarily implements an orchestrator allowing network device discovery, code verification, compilation and low impact deployment on multi-tenant programmable switches. Critical time windows and their durations for the code update are evaluated using hardware switches (Intel Tofino). Performance impacts of code updates on tenants using these switches are evaluated and discussed based on bandwidth tests considering different code deployment options. The architecture proposes a framework to enable seamless updates with minimal to no service interruptions, e.g., using gradual code updates of redundant links between programmable switches in data centers, internet service providers, and mobile networks used by various customers as tenants. Besides P4, common Kubernetes and continuous delivery solutions were used for the presented implementation that is offered as Open Source for further adaption and development.

Referencia

Geier, T., & Rieger, S. (2023). Low Impact Tenant Code Updates on Multi-Tenant Programmable Switches. <https://doi.org/10.23919/cnsm59352.2023.10327866>

Datos

PI1. ¿Qué situaciones motivan la migración de las operaciones tradicionales a NetOps?

Las redes definidas por software (SDN) y los procesadores independientes de protocolo de programación (P4) introdujeron el procesamiento del plano de datos en la red, permitiendo la posibilidad de que varios inquilinos (Tenant) puedan implementar código en switches programables y dispositivos de red, permitiendo la ingeniería de tráfico detallada, la detección y mitigación de ataques en la red o el procesamiento de datos, por ejemplo, para aprendizaje automático. Sin embargo, las implementaciones de

código no pueden garantizar un aislamiento adecuado de inquilinos y crean un impacto negativo en las operaciones de red entre cada actualización de código.

PI2. ¿Cuáles son los requisitos para desarrollar e implementar una solución NetOps?

PI3. ¿Qué avances y resultados se reportan en el desarrollo o la implementación de una solución relacionada con NetOps?

Una arquitectura y canalización de orquestación (OMuProCu o Orchestrator for Multi-tenant PDP Code Updates) de funciones de red en contenedores (CNF), permite implementar funciones de red empaquetadas en contenedores de Docker, orquestados con kubernetes, pudiendo actualizar el código mediante un sistema operativo de red (NOS) que gestiona una canalización de integración y entrega continua a través de software de versionado de código, permitiendo gestionar el ciclo de vida de las CNF aprovisionadas en los switches programables desde la implementación, actualizaciones y hasta el desmantelamiento.

PI4. ¿Cuáles son las posibles limitaciones de NetOps y sus posibles alternativas?

NetInfra - A Framework for Expressing Network Infrastructure as Code

Identificador

ER-16

Año

2022

Fuente

ACM

Tipo de publicación

Journal

Palabras clave relacionadas con NetOps

DNS, DHCP, configuration management, infrastructure as code

Resumen

NetInfra is a framework designed to manage both DHCP and DNS services. It does this through single source of truth formatted to be consumed by configuration management software which reduces duplication of efforts and unifies the configuration of these interrelated services across otherwise unrelated software systems. In this paper, we review a production deployment of managing DHCP and DNS services via the NetInfra framework and discuss the strengths and weaknesses of the framework itself.

Referencia

McGhee, E. J., Krobatsch, T., & Milton, S. (2022). NetInfra - a framework for expressing network infrastructure as code. Practice and Experience in Advanced Research Computing.

<https://doi.org/10.1145/3491418.3530762>

Datos

PI1. ¿Qué situaciones motivan la migración de las operaciones tradicionales a NetOps?

DNS y DHCP son servicios que forman parte del funcionamiento de la mayoría de centros informáticos, cada servicio cuenta con sus propios archivos y opciones de configuración, haciendo que la gestión de estos servicios se realice de forma independiente, sin embargo, al añadir nuevos sistemas o realizar cambios, la falta de paridad en la configuración de los servicios puede llevar a una desviación de la configuración que da como resultado posponer el despliegue y tiempo de inactividad mientras se solucionan los errores de configuración.

Configurar servicios DHCP y DNS de forma manual en los archivos de configuración de cada servicio puede ser difícil a medida que se añaden más hosts al sistema, por lo tanto, se necesita encontrar nuevas formas de gestionar la sobrecarga y reducir los errores humanos, para ello se utiliza la automatización y software de gestión de configuración como la mejor práctica.

PI2. ¿Cuáles son los requisitos para desarrollar e implementar una solución NetOps?

La automatización es una de las características más importantes de NetOps, para cumplir con este propósito, se utiliza software de versionado de código para proporcionar una única fuente de verdad para la configuración de todo el sistema, y,

para desplegar dicho código, se utiliza software de gestión de configuración como Ansible o Puppet.

PI3. ¿Qué avances y resultados se reportan en el desarrollo o la implementación de una solución relacionada con NetOps?

NetInfra es un framework que proporciona una fuente de verdad en un formato único YAML de Ansible, para definir configuraciones DHCP y DNS, esto incluye los servidores de servicios, las subredes que ofrecen DHCP, las zonas controladas por DNS, y los hosts individuales en redes IPv4. Adicionalmente, NetInfra puede ser modificado para soportar otros servicios de infraestructura que soporten software de gestión de configuración

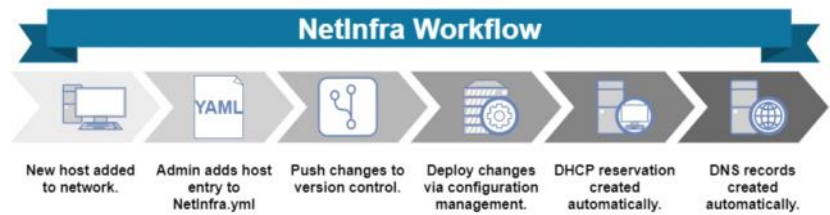


Figura 1: Flujo de trabajo de NetInfra

```
dhcp_servers:
  - name: dhcp1
    ip: 192.168.1.131
    role: primary
  - name: dhcp2
    ip: 192.168.1.132
    role: secondary
next_server: 192.168.1.133

dns_servers:
  - name: dns1.example.com
    ip: 192.168.1.129
    type: primary
  - name: dns2.example.com
    ip: 192.168.1.130
    type: secondary
external_dns_servers:
  - 1.1.1.1
  - 8.8.8.8

subnets:
  - id: 100
    cidr: 192.168.1.0/24
    dns_servers: "192.168.1.129, 192.168.1.130"
    subnet_mask: 255.255.255.0
    routers: 192.168.1.1
    dhcp_options:
      - interface_mtu: 1500
    pools:
      - "192.168.1.64/28"
      - "192.168.1.79-192.168.1.127"
  - id: 101
    cidr: 10.10.0.0/16
    dns_servers: "192.168.1.129, 192.168.1.130"
    subnet_mask: 255.255.0.0
    routers: 10.10.0.1
    required_client_classes:
      - "custom-example"
```

Figure 2: Server Declarations

Figure 3: Subnet Declarations

PI4. ¿Cuáles son las posibles limitaciones de NetOps y sus posibles alternativas?

Wide Area Network Autoscaling for Cloud Applications

Identificador

ER-17

Año

2021

Fuente

ACM

Tipo de publicación

Journal

Palabras clave relacionadas con NetOps

Autoscaling, Wide area networks, kubernetes, cloud computing

Resumen

Modern cloud orchestrators like Kubernetes provide a versatile and robust way to host applications at scale. One of their key features is autoscaling, which automatically adjusts cloud resources (compute, memory, storage) in order to adapt to the demands of applications. However, the scope of cloud autoscaling is limited to the datacenter hosting the cloud and it doesn't apply uniformly to the allocation of network resources. In I/O-constrained or data-in-motion use cases this can lead to severe performance degradation for the application. For example, when the load on a cloud service increases and the Wide Area Network (WAN) connecting the datacenter to the Internet becomes saturated, the application flows experience an increase in delay and loss. In many cases this is dealt with overprovisioning network capacity, which introduces additional costs and inefficiencies.

On the other hand, thanks to the concept of "Network as Code", the WAN exposes a set of APIs that can be used to dynamically allocate and de-allocate capacity on-demand. In this paper we propose extending the concept of cloud autoscaling into the network to address this limitation. This way, applications running in the cloud can communicate their networking requirements, like bandwidth or traffic profile, to a Software-Defined Networking (SDN) controller or Network as a Service (NaaS) platform. Moreover, we aim to define the concepts of vertical and horizontal autoscaling applied to networking. We present a prototype that automatically allocates bandwidth to the underlay network, according to the requirements of the applications hosted in Kubernetes. Finally, we discuss open research challenges

Referencia

Serracanta, B., Paillisse, J., Cabellos, A., Claiborne, A., Rodriguez-Natal, A., Ward, D., & Maino, F. (2021). Wide Area Network Autoscaling for Cloud Applications.
<https://doi.org/10.1145/3472727.3472797>

Datos

PI1. ¿Qué situaciones motivan la migración de las operaciones tradicionales a NetOps?

Recientemente varios escenarios de redes comerciales han adoptado el enfoque SDN, dando lugar a una nueva generación de soluciones de red que permiten la programación de la gestión de red y la gestión dinámica de la misma. Estas soluciones junto a orquestadores de recursos virtuales en entornos cloud como OpenStack y Kubernetes, permiten superar las limitaciones actuales de escalabilidad dinámica ofrecida por los

clústers de infraestructura en la nube, superando los problemas de calidad de experiencia (QoE por sus siglas en inglés) ocasionados por la congestión del tráfico en redes sin capacidad de auto escalado y los costes adicionales relacionados con un sobre aprovisionamiento de recursos de red.

PI2. ¿Cuáles son los requisitos para desarrollar e implementar una solución NetOps?

PI3. ¿Qué avances y resultados se reportan en el desarrollo o la implementación de una solución relacionada con NetOps?

La colaboración entre ingenieros de redes y desarrolladores de software se incrementa al permitir que los desarrolladores puedan añadir etiquetas sobre los requisitos de red de sus aplicaciones a las cargas de trabajo, que luego, pueden ser utilizadas por los operadores de red mediante API de recursos virtuales de red como SDN y orquestadores como Kubernetes para satisfacer dichos requisitos de infraestructura, permitiendo escalado vertical, horizontal y dinámico de los recursos de red.

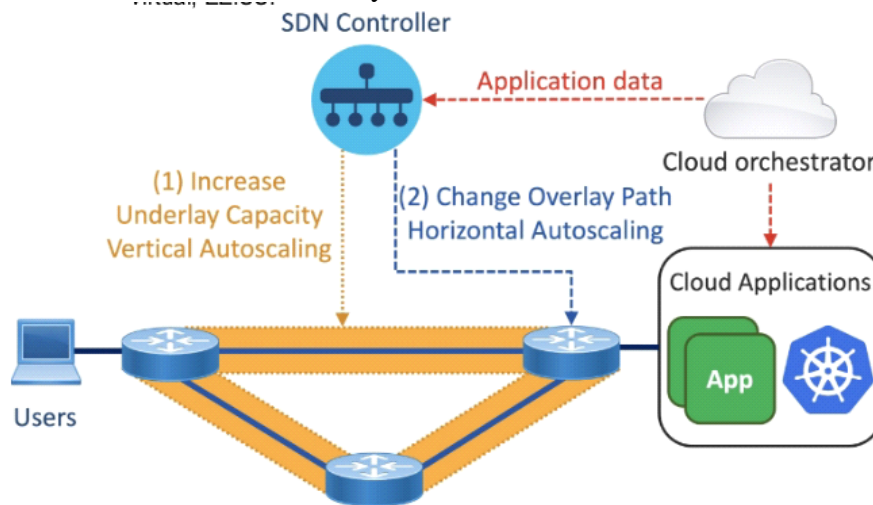


Figura 1: Autoescalado vertical y horizontal de la red

CN-WAN es una API de código abierto para comunicar recursos de red con los recursos del centro de datos de una nube, esta interfaz identifica que aplicaciones de la nube requieren un autoescalado de red, comunica eventos de autoescalado de aplicaciones en la nube a la red y cuantifica el factor de escalado requerido, permitiendo automatización y colaboración entre equipos de desarrollo (DevOps) y operadores de red (NetOps): Por un lado, los DevOps hacen uso de herramientas en la nube para añadir información contextual a sus aplicaciones por ejemplo, definir el perfil del tráfico de red necesario para el correcto funcionamiento de la aplicación, mientras que NetOps puede automatizar la asignación justa de recursos de red en función del conocimiento y la gestión sobre la red WAN.

Autoescalado vertical de red: definido como la modificación de las propiedades de una conexión existente como túneles MPLS, VXLAN y LISP. Ante el aumento de instancias de una aplicación, se puede aumentar el ancho de banda de la conexión de forma proporcional.

Un experimento utiliza una nube pública controlada por Kubernetes y un servidor HTTP Echo, mediante el autoescalado vertical de Kubernetes y la definición de objetivos de uso de CPU máximo al 40%, se crean solicitudes HTTP en 700, 1900, 2900, 3500 y 4500 conexiones a los 30, 90, 150 y 210 segundos. Iniciando con una conexión de 50Mbps y con un consumo de 1Mbps por réplica, se aumenta el tráfico hasta llegar a las 150 réplicas autoescaladas, en donde se puede observar que la red aumentó su ancho de banda en intervalos de 50Mbps a medida que el tráfico y el consumo de CPU dispararon nuevas réplicas con un tiempo de reacción de decenas de segundos.

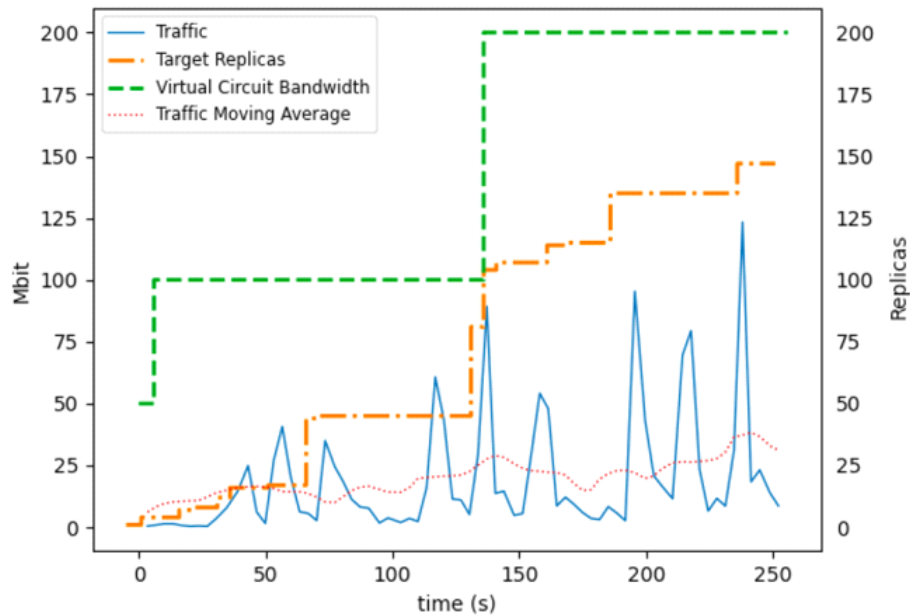


Figure 3: Vertical autoscaling performance

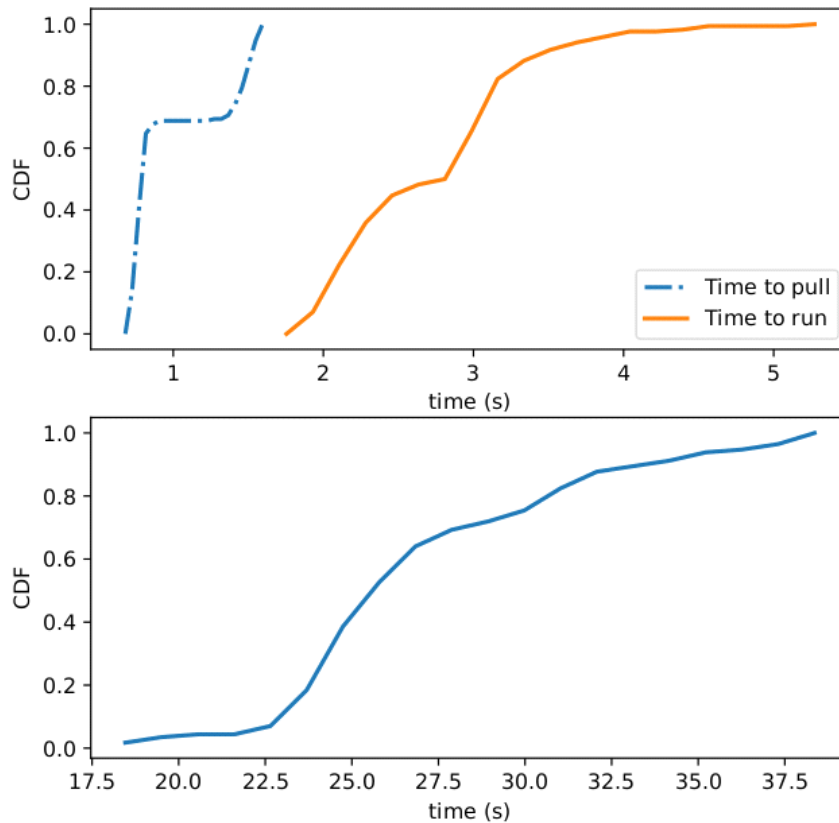
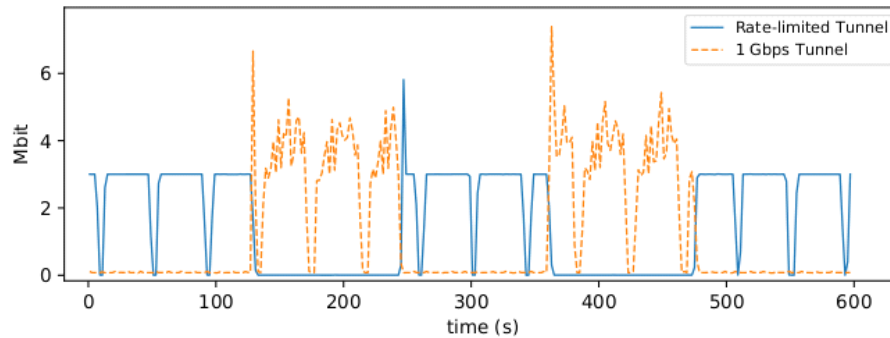


Figure 4: Kubernetes controller (top) and overlay controller (bottom) response time

Autoescalado horizontal de red: cambio dentro de las rutas de red debido a los requisitos de la aplicación o el cambio de variables que pueden afectar el funcionamiento de una ruta de red específica, por ejemplo, ante los cambios en la fluctuación, el ancho de banda y el retardo, la red SDN puede optar por cambiar la ruta según los requisitos de latencia descritos en las etiquetas de la aplicación. Se modificó el escenario de experimentación de escalado vertical para admitir la experimentación de escalado horizontal, para esto, se configuró un servicio de streaming de video en lugar del servidor web y en el otro extremo de la red SD-WAN,

se conectó un endpoint, pudiendo cambiar las rutas entre dos opciones, una con un ancho de banda de 1Gbps y otra de 3Mbps. Cuando el operador CN-WAN lee la dirección IP y el número de puerto del contenedor kubernetes y las etiquetas del perfil de red necesario definido por el desarrollador, el operador CN-WAN se comunica con el adaptador SD-WAN para modificar las rutas en función de las etiquetas de tráfico de red y las políticas SD-WAN definidas por el operador de red. Para cambiar entre rutas, se modificaron las etiquetas del contenedor de streaming de video varias veces para observar el cambio de rutas de red, obteniendo un retraso de inferior a 23 segundos en el 80% de los casos en una muestra de 200 intentos.



PI4. ¿Cuáles son las posibles limitaciones de NetOps y sus posibles alternativas?

Kathará: A container-based framework for implementing network function virtualization and software defined networks

Identificador

ER-18

Año

2018

Fuente

IEEE explore

Tipo de publicación

Conference document

Palabras clave relacionadas con NetOps

containers, software, routing protocols, virtual machine, virtual networks functions, network topology, dynamic scaling,

Resumen

Network Function Virtualization (NFV) and Software-Defined Networking (SDN) are deeply changing the networking field by introducing software at any level, aiming at decoupling the logic from the hardware. Together, they bring several benefits, mostly in terms of scalability and flexibility. Up to now, SDN has been used to support NFV from the routing and the architectural point of view. In this paper we present Kathará, a framework based on containers, that allows network operators to deploy Virtual Network Functions (VNFs) through the adoption of emerging data-plane programmable capabilities, such as P4-compliant switches. It also supports the coexistence of SDN and traditional routing protocols in order to set up arbitrarily complex networks. As a side effect, thanks to Kathará, we demonstrate that implementing NFV by means of specific-purpose equipment is feasible and it provides a gain in performance while preserving the benefits of NFV. We measure the resource consumption of Kathará and we show that it performs better than frameworks that implement virtual networks using virtual machines by several orders of magnitude.

Referencia

Bonofiglio, G., Iovinella, V., Lospoto, G., & Di Battista, G. (2018). Kathará: A container-based framework for implementing network function virtualization and software defined networks. <https://doi.org/10.1109/noms.2018.8406267>

Datos

PI1. ¿Qué situaciones motivan la migración de las operaciones tradicionales a NetOps?

Las redes definidas por software (SDN) y la virtualización de funciones de red (NFV) ha demostrado un cambio considerable en el campo de las redes al introducir software en cualquier nivel, promoviendo entornos de red escalables y flexibles.

SDN es el ejemplo claro de cómo se puede gestionar el enrutamiento escribiendo software ejecutable en equipos de propósito general,

Las funciones de red se han softwarizado en funciones de red virtuales (VNF), sumándose a los contenedores y las capacidades de programabilidad del plano de datos

da como resultado que emerjan nuevas formas de implementar servicios y funciones de red,

La arquitectura VNF está ganando fuerza porque permite a los profesionales crear versiones basadas en software de funciones de red como firewalls, contadores de paquetes, etc. produciendo el desacoplamiento del hardware, la implementación flexible, escalamiento dinámico y en tiempo real dado que los servicios son instanciables.

PI2. ¿Cuáles son los requisitos para desarrollar e implementar una solución NetOps?

PI3. ¿Qué avances y resultados se reportan en el desarrollo o la implementación de una solución relacionada con NetOps?

Kathará es un framework que combina la flexibilidad de los contenedores, portabilidad y programabilidad del plano de datos, SDN y VNF para crear redes virtuales complejas que sean suficientemente eficientes como para usarlas en producción, permitiendo además, la heterogeneidad de infraestructura al admitir la transición entre nodos de red virtuales a físicos y viceversa.

Kathará es una herramienta que proporciona agilidad y rapidez en el aprovisionamiento de recursos de red como ejecución de protocolos y servicios de red estándar (OSPF, BGP, etc), protocolos SDN que permiten la programabilidad del plano de control (OpenFlow, Ryu, etc) y conmutadores de red programables como modelos compatibles con el lenguaje P4.

Al implementar Kathará con conmutadores compatibles con P4, es posible crear un sistema de gestión que permite la compilación de código P4 remota y la entrega continua del servicio.

PI4. ¿Cuáles son las posibles limitaciones de NetOps y sus posibles alternativas?

Towards Production Deployment of a SDX Control Framework

Identificador

ER-19

Año

2022

Fuente

IEEE explore

Tipo de publicación

Conference document

Palabras clave relacionadas con NetOps

pipelines, software defined networking, software reliability

Resumen

Developing a distributed controller system for production software defined networks (SDN) requires substantial engineering effort to satisfy the stringent performance, adaptability, availability and security requirements that are well beyond the basic function development. The softwarization nature of SDN provides the opportunity to leverage the recent advancements in software engineering and system automation. In this paper we present our recent work to develop and deploy a wide-area SDN control framework prototype towards production deployment and operation. We primarily focus on three critical areas (1) enhancement of the software functions and the substrate virtualization configurations to fully support advanced connection services and fault tolerance in both data plane and control plane (2) a high-fidelity testing pipeline consisting of unit tests emulation and a testbed and (3) a continuous integration and continuous deployment (CI/CD) pipeline. Our experience proved that the presented environment and process greatly increased the software quality and development efficiency which would ultimately lead to a reliable and continuous deployment and automation of the targeted production SDN network.

Referencia

Cevik, M., Stealey, M., Wang, C., Bezerra, J., Ibarra, J., Chergarova, V., Morgan, H., & Xin, Y. (2022). Towards Production Deployment of a SDX Control Framework.
<https://doi.org/10.1109/iccn54977.2022.9868884>

Datos

PI1. ¿Qué situaciones motivan la migración de las operaciones tradicionales a NetOps?

Las redes SDN proporcionan un nuevo paradigma del aprovisionamiento de las redes basadas en la programabilidad y automatización, permitiendo interfaces flexibles para que los servicios interactúen directamente con la red.

En este nuevo territorio, la ingeniería de software y la ingeniería de redes deben considerarse de manera integral para garantizar el funcionamiento de la red.

Muchos proveedores de infraestructura han adoptado la metodología DevOps en casos como integración y despliegue continuos e infraestructura como código, esta

metodología y las herramientas utilizadas, han demostrado ser valiosas para mejorar la eficacia y la disponibilidad de un entorno de red en producción.

PI2. ¿Cuáles son los requisitos para desarrollar e implementar una solución NetOps?

Independientemente del tamaño de los equipos, existen dos tipos herramientas de alto valor para todo el ciclo de vida del desarrollo de software: el control de versiones de código y la integración y despliegue continuos.

PI3. ¿Qué avances y resultados se reportan en el desarrollo o la implementación de una solución relacionada con NetOps?

PI4. ¿Cuáles son las posibles limitaciones de NetOps y sus posibles alternativas?

actualmente es un desafío convertir una red tradicional en una red SDN que sea capaz de automatizar los servicios de red de extremo a extremo en las capas inferiores (1 y 2) y el aprovisionamiento con garantía de QoS debido a la complejidad del aprovisionamiento y la heterogeneidad de infraestructura, sumado a que las mejores prácticas de ingeniería todavía se basan en comunicación de persona a persona y configuración manual de la red

PipeConf: An Integrated Architecture for the Automated Configuration of Network Assets

Identificador

ER-20

Año

2022

Fuente

IEEE explore

Tipo de publicación

journal

Palabras clave relacionadas con NetOps

pipelines, software quality, software reliability, software defined networking, software development, continuous integration, network environment, wide area network, continuous delivery, network services, production networks,

Resumen

The manual management of network assets is susceptible to configuration errors, lack of standardization, a large amount of repetitive work, and little or no traceability of changes over time. The Infrastructure as Code (IaC) approach makes it possible to automate the process of configuring resources such as operating systems, network services, containers, and applications, by treating them as software and allowing standardization and configuration rollback. This paper proposes an integrated architecture based on different software tools that use the IaC approach to automate the configuration of network assets, considering different models and manufacturers. A quantitative analysis shows the architecture's efficiency in response time and scalability. The architecture has achieved a proportional gain of 83% in the average response time to manage 128 assets, with no significant increase in processing and memory usage.

Referencia

Pires, A. D. S., Matos, F. M., Santos, A. L. D., Pessoa, D. E. R., & Maciel, P. D. (2022). PipeConf: an integrated architecture for the automated configuration of network assets. *IEEE Transactions on Network and Service Management/IEEE eTransactions on Network and Service Management*, 19(3), 3657–3669. <https://doi.org/10.1109/tnsm.2022.3195382>

Datos

PII. ¿Qué situaciones motivan la migración de las operaciones tradicionales a NetOps?

El aprovisionamiento y la gestión bien planificada de activos de red como firewalls, routers y access point son aspectos fundamentales para la conectividad de una empresa para alcanzar ventajas comerciales competitivas, sin embargo, gestionar las configuraciones de estos activos es un desafío para los administradores de red, pues, estas tareas suelen realizarse a manualmente, introduciendo el riesgo de cometer errores de configuración, la nula estandarización y trazabilidad de los cambios y la gran cantidad de trabajo repetitivo son problemas presentes en la configuración manual. Ante esta situación la infraestructura como código (IaC por sus siglas en inglés) y la

"softwarazing" de la infraestructura de red permite la automatización de la gestión de dispositivos de red, tomando prácticas de desarrollo de software (DevOps).

La tendencia de software de red es impulsada por SDN y NFV en relación con aspectos como la gestión de recursos virtualizados y el encadenamiento de servicios, esta tendencia indica un cambio de paradigma en la administración de redes hacia infraestructuras de red completamente "softwarazing".

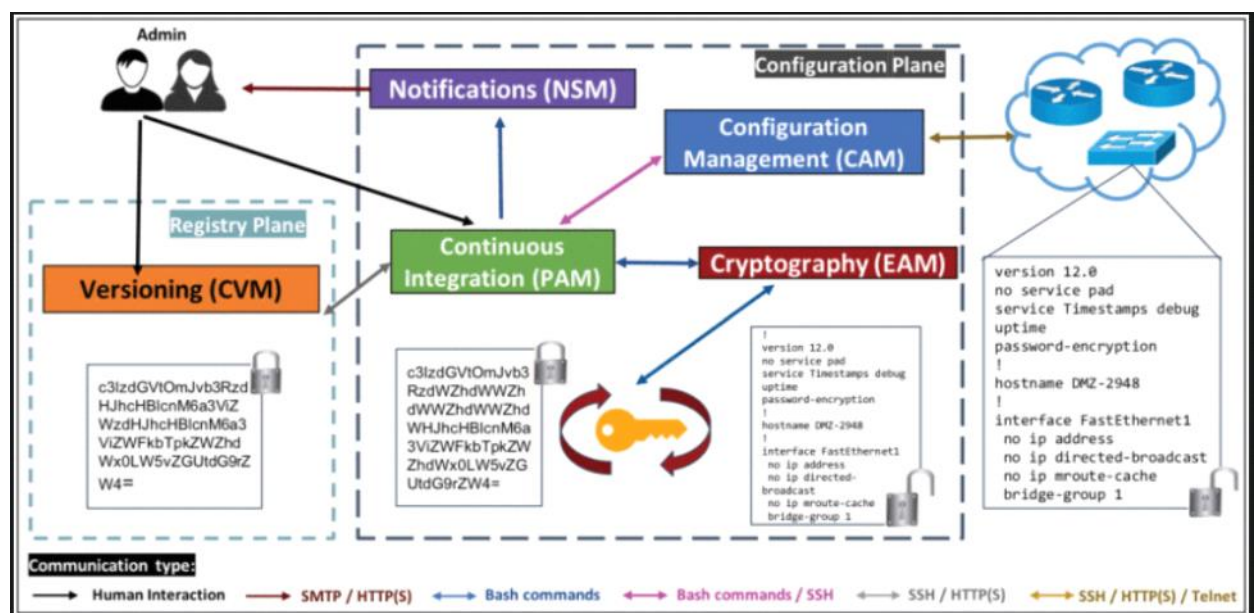
Una arquitectura que automatiza el proceso de gestión de configuración a través de IaC como PipeConf permite varios casos de uso como: laboratorios de informática para impartir cursos de redes, demostración de la gestión de redes con IaC y gestión de activos de red en entornos empresariales de producción.

PI2. ¿Cuáles son los requisitos para desarrollar e implementar una solución NetOps?

PI3. ¿Qué avances y resultados se reportan en el desarrollo o la implementación de una solución relacionada con NetOps?

PipeConf es una arquitectura de software que permite gestionar infraestructura de red como código, una solución automatizada para la gestión de configuración, control de versionado de software, soporte para múltiples fabricantes y uso de políticas de gestión.

La arquitectura propuesta para PipeConf incluye el uso de gog y Postgresql como CVM (control version module), Jenkins para el PAM (Pipeline application module), Sops para el CM (Notification module), Salt SProxy y Napalm para el CAM (Configuration application module), Prometheus, Grafana y Node Exporter como monitor de métricas.



PI4. ¿Cuáles son las posibles limitaciones de NetOps y sus posibles alternativas?