vCPE/SD-WAN Network Orchestration

An Integration-Based, Open Source Approach

vCPE/SD-WAN Network Orchestration

Table of Contents

Cloudify

Introduction

According to an Analysys Mason report, "Enterprise vCPE represents a lucrative opportunity for CSPs as it offers a ROI of 156% and NPV of USD1.1 billion over a five-year migration period to first movers." Gartner notes that by 2018, over 40% of WAN edge infrastructure refresh projects will be SD-WAN or vCPE based. It's no surprise then, that the demand for vCPE and SD-WAN solutions would be quite high. The motivation for CSPs to build out these solutions is to keep costs very low for broadband network links for OTT (over the top) application traffic, while utilizing the expensive MPLS links only for QOS (quality of service) sensitive traffic like voice and video.

There are many startups that try to provide a turnkey solution with their own CPE devices and management tools. However, the reality is that many customers already have an install base of CPE devices, like Cisco ISRs, generic off the shelf universal CPEs, or white-labeled CPE boxes.

For those CSPs looking to continue using their existing devices, the only missing piece is generic, open orchestration. This approach integrates existing CPE devices, establishes secure overlay network connections, and chains virtual network functions (VNFs) like vRouters and vFirewalls, creating an end-to-end flexible, dynamic service.

This service can securely connect a remote branch to its corporate headquarters, the internet, and cloud-based provisioned applications, or another branch. The service must be dynamic and customers should be able to manipulate its capabilities in real time based on ad-hoc business requirements.

This white paper details how to take an integration-based approach to vCPE/SD-WAN using a generic, open source, TOSCA-based solution, and dynamically connect and swap existing components, open and proprietary, all while providing the required vCPE and SD-WAN orchestration functionality to automatically provision, configure, and chain VNFs. With Cloudify as the orchestrator, this approach brings back the power, and control, over the network back to the service providers and breaks them free from vendor lock-in.

Read Our Other NFV White Papers

From Virtual Appliance to Cloud Native VNF

NFV and What it Means to You

Automatic Orchestration Upon Registration

The initial assumption is that the provider/vendor has the means to ship a new device to a remote location or utilize an existing device at the remote location, connect it to the network, get basic IP connectivity, and register the device. Once the device is registered, Cloudify automatically kicks in and provisions day 0, day 1, and day 2 configurations.

Looking at Figure 1 below, we can see a generic uCPE device at the remote branch office connected with a number of link options: DSL, MPLS and LTE. This is a generic device and could be supplied by any uCPE vendor. The service provider or the enterprise can decide which access line types to use. Typically the broadband lines are used for over the top (OTT) traffic, while the MPLS lines are used for performance sensitive data such as corporate voice and video.

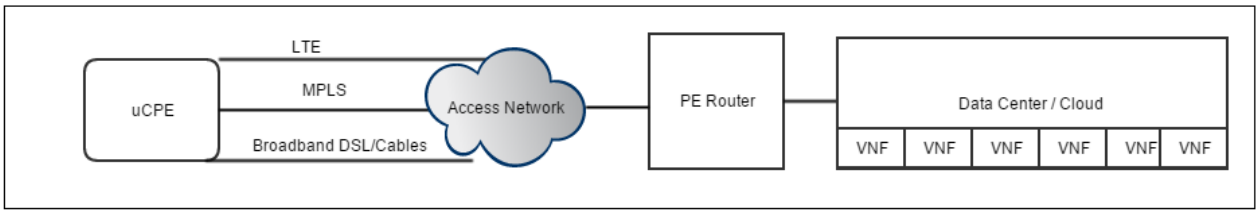

Figure 1

Overlay Network Connections

Overlay connections are needed in order to connect branch offices to the data center, whether it is the HQ, ISP data center, or cloud. Site-to-site overlay connections could be defined as well.

Devices have different configuration methodologies for overlay connections. The most common are:

- Netconf/YANG - Netconf uses the XML format to configure network devices
- CLI/SSH - Proprietary CLI commands per device

Cloudify supports both methods and utilizes different plugins for each device type. Moreover, due to its modular architecture, it is easy to add new plugin types and interfaces to Cloudify without changing the core engine.

Dynamic VNF On-Boarding and Service Chaining

Additional functionality is introduced through the provisioning of VNFs in the data center and making them part of the already defined overlay connection. Let's take, for example, a secure internet connection. By introducing a

virtual firewall (vFW), such as Fortigate or PaloAlto Networks, we can create a secure internet connection. Figure 2 shows a schematic diagram of such a connection.

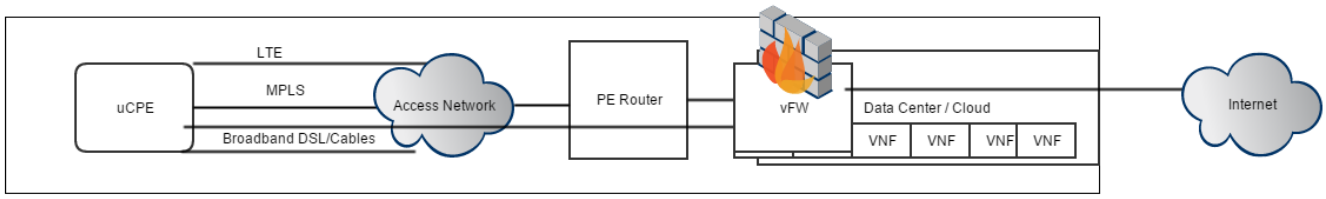

Figure 2

Additional VNFs such as UTM URL filtering for content filtering, DPI for deep packet analysis, and WOC WAN optimization controller, among others, can be added as well and chained together. All of these VNFs can be added or removed from the service chain dynamically during runtime.

Complex Topologies Made Easy With TOSCA

Topologies which involve many branches, data centers, HQ and cloud applications can become complex to describe and model. Taking a look at Figure 3 below, we can see a common topology that gets more complex as branches and applications are added. The topology describes not only the infrastructure connections but the application's topologies as well.

TOSCA, Topology and Orchestration for Cloud Applications, in the context of NFV, is a language that enables data modeling - similar to YANG for configuration data. TOSCA's primary use is its focus on topology and workflows, processes, and policies. Therefore, when you want to manage the full lifecycle of orchestration in an NFV use case, you usually have to start with provisioning VIM resources, next deploy software components and VNFs, then wire them in and push configurations to these devices to make everything work, and chain them to other services as needed.

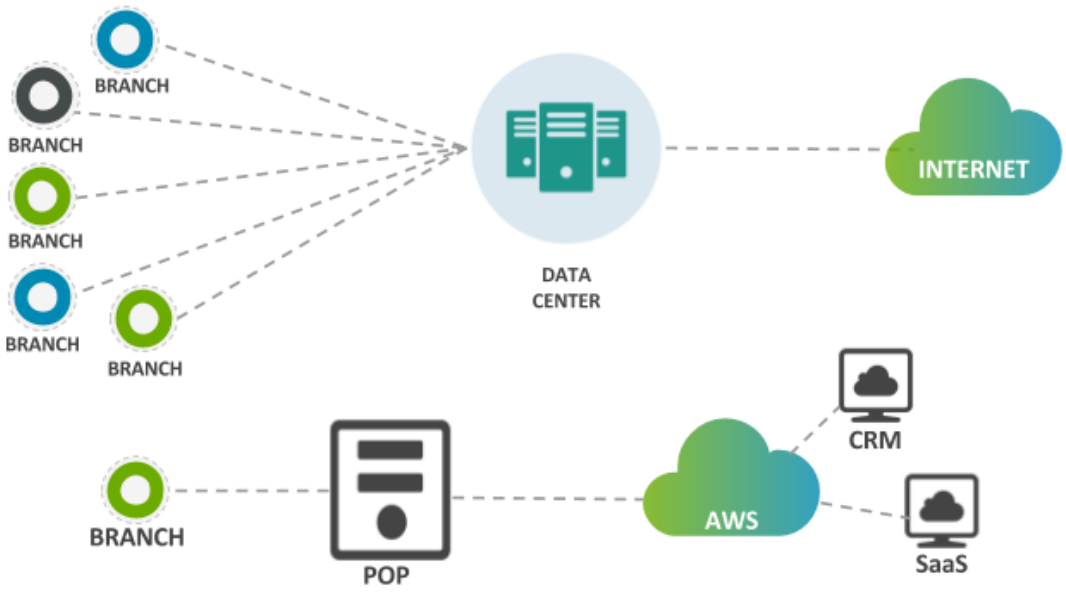

Figure 3

The service Lifecycle Management (LCM) automation is crucial and needs to be defined in a simple way. The service includes the overlay connections and VNFs participating in the service chain. Each VNF has its own lifecycle events and is part of the VNF onboarding process.

TOSCA utilizes information and data modeling for describing complex topologies. Many vendors have already moved over to TOSCA, including AT&T, Deutsche Telekom, and Orange according to SDxCentral.

The provisioning and management of an integration-based approach to vCPE/SD-WAN requires orchestration, and TOSCA is an ideal candidate to define the service topology data model as well as the VNF lifecycle events.

Cloudify, the most extensive implementation of TOSCA, can manage Physical Network Functions (PNFs) as well as VNFs, so, if the customer is already using physical devices, those can be utilized and integrated as well.

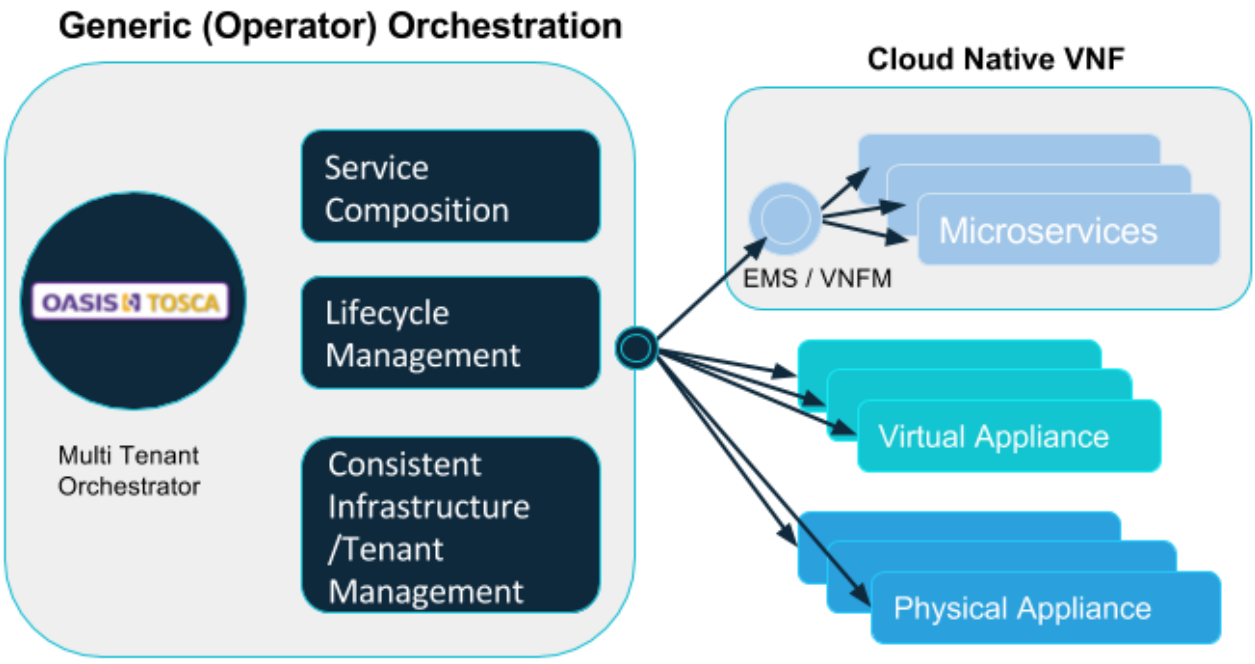

Figure 4

How it Works

When a new CPE device is shipped to a remote office and connected to the network, the CPE device goes through the ZTP process and Cloudify is then notified that a new device was registered. Cloudify adds this device to the dynamic TOSCA graph representation in memory as well as pushes Day 0 and Day 1 configuration to the remote device. In this manner, a dynamic graph with multiple remote branch offices, HQ, and cloud applications is dynamically built.

Day 2 policy-based operations can be applied to the vCPE/SD-WAN TOSCA graph as well. For example, updating an existing VNF or changing the security rules of a vFW.

More complex changes, such as adding a DPI device for traffic analysis, are achieved by dynamically adding a new VNF to the service chain through the blueprint deployment update.

Under the Hood

Behind the scenes, Cloudify implements a TOSCA parser and a workflow execution engine as well as a Telco-grade, feature-based orchestrator with capabilities including high availability and multi-tenancy.

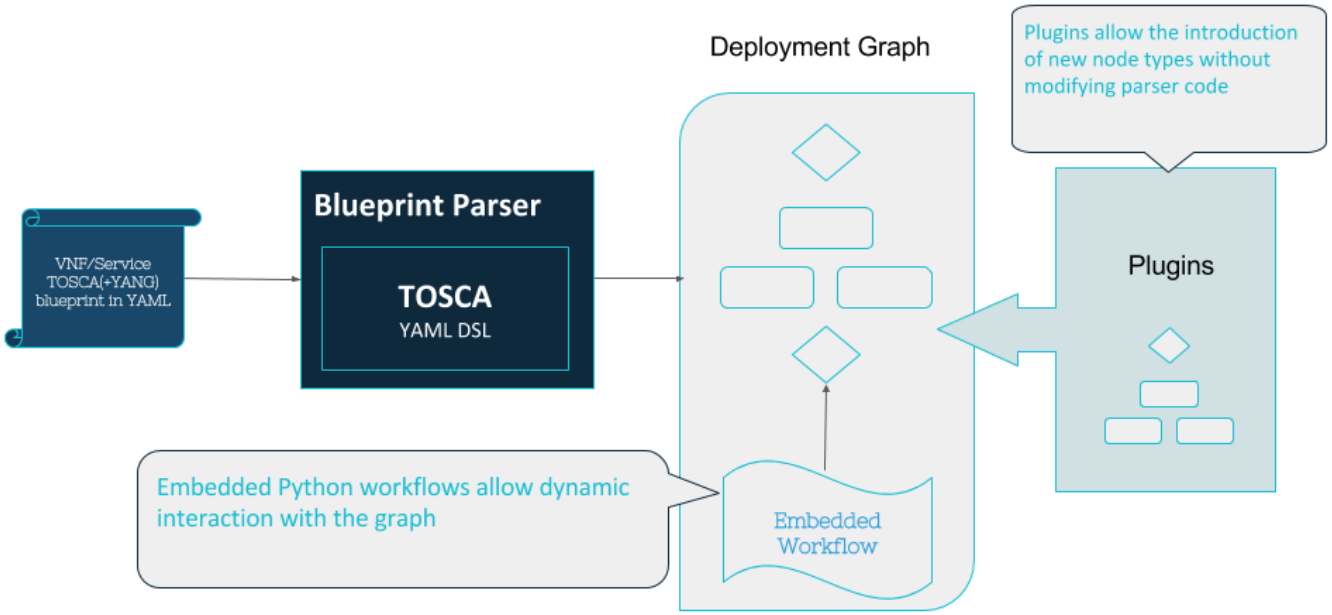

Figure 5

Cloudify also builds the topology graph dynamically. This is achieved through a new feature introduced in version 3.4 called “deployment update” where users can take a currently running blueprint and update it without downtime. Cloudify calculates the diff between both blueprints, for example, if new nodes are added to the TOSCA graph, and updates it on the fly.

Cloudify comes with the components as they appear in Figures 6 and 7.

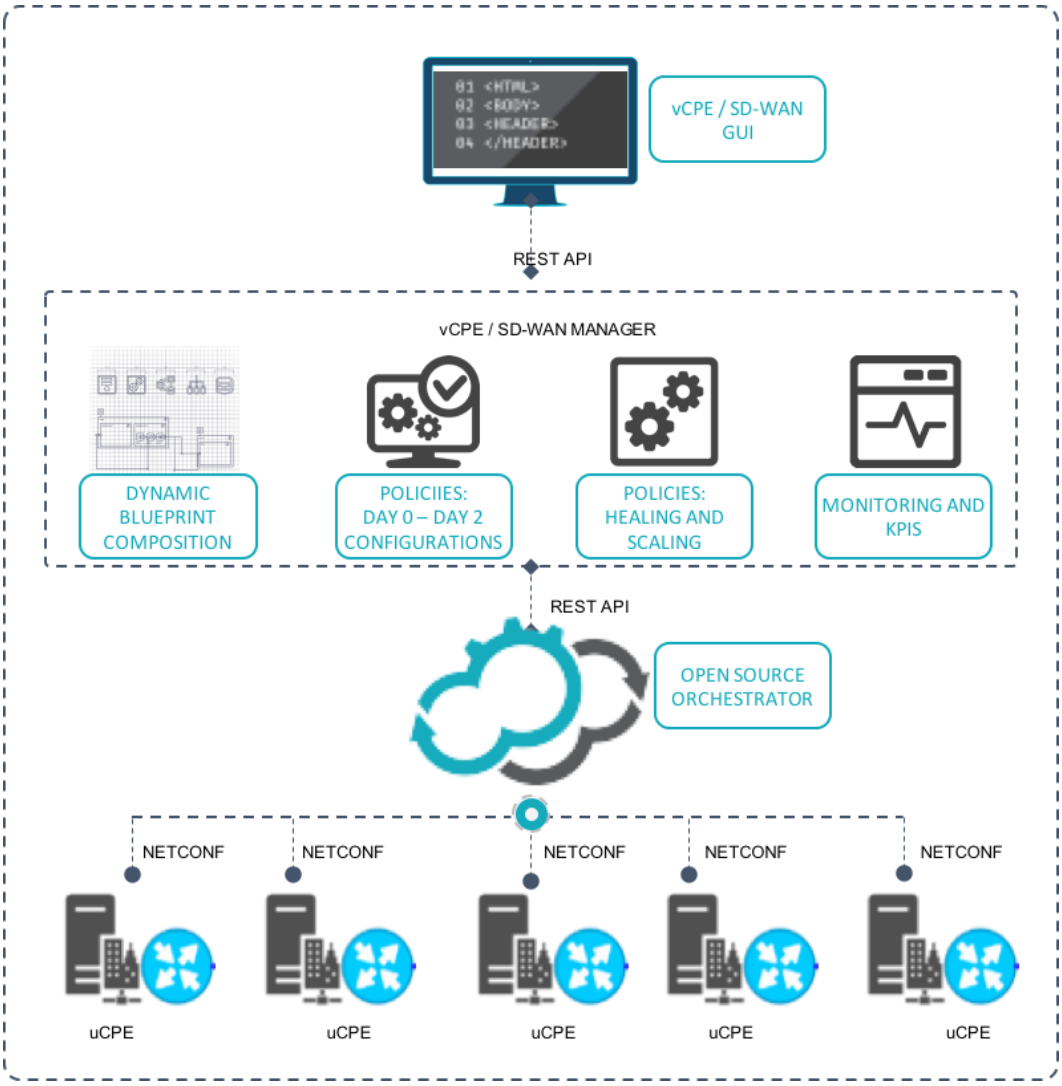

Figure 6

The Architecture

The vCPE/SD-WAN solution utilizes Cloudify. At the bottom of Figure 6 are the uCPEs at the branches. Above it is the Cloudify Orchestrator which provisions and monitors the resources, including creation of the overlay networks, provisioning the VNFs, and creating the service chains per customer.

The component above Cloudify holds the vCPE/SD-WAN domain logic:

- It creates and manages the dynamic TOSCA graph, where each TOSCA node represents a uCPE or an application component, as well as relationships between them, which are easily expressed in TOSCA.
- It holds all the necessary Day0, Day 1, and Day 2 configurations which are required to provision the overlay networks and VNFs participating in the service chain.

- It holds the policies required to manage the vCPE / SD-WAN domain:
  - Healing - What you do when a node fails, how to heal it. It is not enough only to bring up the node, but also to ensure it is correctly placed in the graph with all of its connections.
  - Scaling - When there is a lot of traffic, additional VNFs from the same type could be added, e.g. additional vFWs. Of course, this is done with the help of a load balancer in front to direct the traffic to the vFWs as needed.
- Monitoring & KPIs - Cloudify, with it's remote agents and monitoring capabilities, collects KPIs, both at the infrastructure and application levels, from the different nodes, e.g. node or link utilization, etc. All KPIs are kept in an open source time series database as shown in Figure 7 below. Combining the TOSCA model, which represents the deployed real time graph, and monitoring KPIs provides the basis for many policies that utilize the KPIs time series DB and execute Day 2 workflows. Cloudify presents these KPIs in a graphical user interface.
- Cloudify's Netconf plugin is used to communicate with the uCPE devices.

The GUI component is a portal for controlling the vCPE/SD-WAN Manager that presents the user with a graphical map of all uCPE devices shown, and presents a catalog of available VNFs the user can provision. The UI can be used to apply operations like configuring an overlay network for a new uCPE that shipped and was just connected to the network. Users can also drill down to each branch location to view location-based statistics as well as provision VNFs and add them to the service chain.

Figure 7 below shows, in detail, Cloudify's architecture and its basic components. Included in Cloudify are the GUI and NBI REST API, TOSCA composer and workflow engine, the policy engine, monitoring KPIs DB, and the structure of the provisioned VM (in the separate box below).

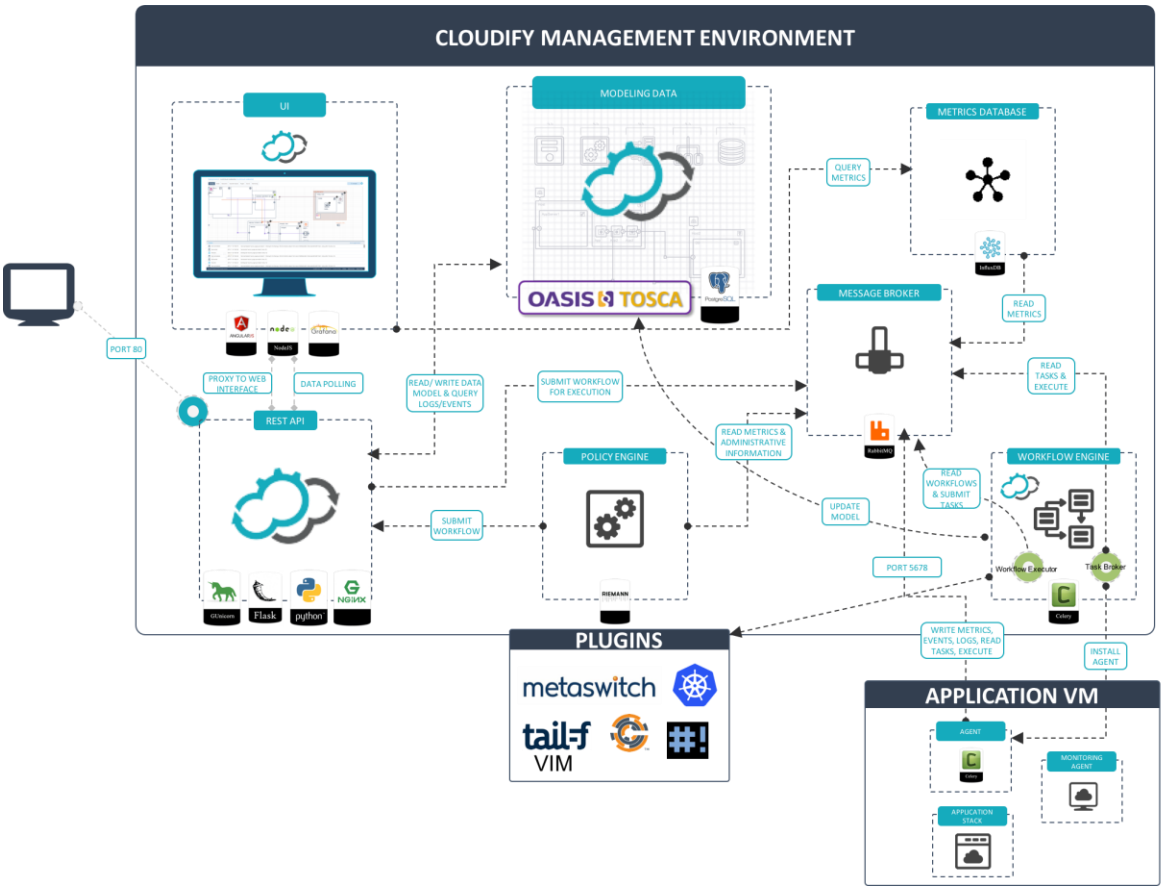

Figure 7

Conclusion

With generic, open source cloud orchestration, it is possible to break free of vendor lock-in via a vCPE/SD-WAN engine that is dynamically configured to simplify the setup and deployment of any new remote branch connection automatically upon registration, including Day 1 and Day 2 post-deployment operations. Cloudify, together with TOSCA, ensures even the most complex topologies can be orchestrated as well as making it easy to deploy, service chain, and swap VNFs on the fly.

Cloudify

www.getcloudify.org