Introduction

This chapter provides an overview of Cisco Virtual Topology System (VTS). It has the following sections:

- Understanding Cisco VTS, page 1
- Cisco VTS Architecture Overview, page 2
- Virtual Topology Forwarder, page 3
- Virtual Topology System High Availability, page 3

Understanding Cisco VTS

The Cisco Virtual Topology System (VTS) is a standards-based, open, overlay management and provisioning system for data center networks. It automates DC overlay fabric provisioning for both physical and virtual workloads.

Cisco VTS provides a network virtualization architecture and software-defined networking (SDN) framework that meets the requirements of multitenant data centers for cloud services. It enables a policy-based approach for overlay provisioning.

Cisco VTS automates complex network overlay provisioning and management tasks through integration with cloud orchestration systems such as OpenStack and VMware vCenter and abstracts out the complexity involved in managing heterogeneous network environments. The solution can be managed from the embedded Cisco VTS GUI or entirely by a set of northbound Representational State Transfer (REST) APIs that can be consumed by orchestration and cloud management systems.

Cisco VTS provides:

- Fabric automation
- Programmability
- Open, scalable, standards based solution
- Cisco Nexus 2000, 3000, 5000, 7000, and 9000 Series Switches. For more information, see Supported Platforms in *Cisco VTS Installation Guide*.
- Software forwarder (Virtual Topology Forwarder [VTF])

VTS performs the role of an overlay orchestrator in data-center networks. In this role, it manages configuration on the data center leaf and spine devices. The configuration of the devices is dependent on the type of overlay

service that the Cisco VTS user intends to create. The Cisco VTS user in this context could either be manual users interfacing via GUI or APIs, or could be virtual machine managers like OpenStack or vSphere. Since the device configuration is derived from overlay service instances, Cisco VTS holds the 'desired' device configuration in its database.

Whenever, there is a change to the overlay service instances, it generates desired device configuration and applies them to the relevant set of devices. This is the prime functionality of an orchestrator. Changing any of the device configuration outside of Cisco VTS (For example, using CLI or other programmatic interfaces to the device), can result in service disruption. Hence Cisco VTS always reconciles its view of the device configuration and pushes that to the devices. Cisco VTS holds the master database of all device configuration in the fabric.

However, there are some practical use cases where Cisco VTS accommodates out-of-band device configuration.

- Day0 underlay configuration—Cisco VTS is an overlay manager, but overlays cannot be established without an underlay. Underlay configuration on each leaf/spine device is unique. Typically, such underlay configuration is laid out even before Cisco VTS can manage the overlays. Assuming all the devices in the fabric are physically connected, the data center administrator establishes the underlay configuration by manually connecting to the devices and configuring them OR using an underlay manager to perform this. When Cisco VTS performs a scan of the fabric inventory and discovers the topology, it is expected that all the underlay configuration has been fully established. At this point, Cisco VTS treats all the pre-existing device configuration to be Day0 configuration. Day0 configuration is synced up from the devices and stored in VTS database as a baseline. All overlay service configuration is built on top of this day-0 device configuration.
- Day*N* underlay configuration—While Cisco VTS manages overlay specific device configuration, there is always the need for the fabric operators to customize device underlay configuration. Typical operations include physical link management, applying link specific features, managing underlay routing protocols and setting up the security. Recognizing this need, Cisco VTS supports the concept of 'device' templates. These are essentially device configuration parameters exposed to the VTS user using GUI/APIs. VTS users can customize device configuration using device templates and use that to create the consolidated device configuration.
- Day*N* overlay configuration—While the overlay specific configuration pushed by Cisco VTS is sufficient to establish overlays, every deployment may require some customization around this configuration. Since VTS holds the master device configuration, it is essential that any customization flows through Cisco VTS. To address this, Cisco VTS supports the concept of a 'service' template. Service templates allow the Cisco VTS user to extend the service specific device configuration via GUI/API.

Note Service templates always 'augment' the configuration. They cannot modify or remove configuration that is constructed by the VTS service layer.

We recommend that you do not modify device configuration outside of VTS. Doing so, can result in misconfiguration of devices and will result in service outage. If there is a real need to do so, you may follow one of the three models of device configuration to achieve the desired customization.

Cisco VTS Architecture Overview

Cisco VTS architecture has two main components: the Policy Plane and the Control Plane. These perform core functions such as SDN control, resource allocation, and core management function.

- **Policy Plane**: The policy plane enables Cisco VTS to implement a declarative policy model designed to capture user intent and render it into specific device-level constructs. The solution exposes a set of modular policy constructs that can be flexibly organized into user-defined services for use cases across service provider and cloud environments. These policy constructs are exposed through a set of REST APIs that can be consumed by orchestrators and applications to express user intent, or instantiated through the Cisco VTS GUI. Policy models are exposed as system policies or service policies.

  System policies allow administrators to logically group devices into pods within or across data centers to define Admin Domains with common system parameters (for example, BGP-EVPN control plane with distributed Layer 2 and 3 gateways).

  The inventory module maintains a database of the available physical entities (for example, data center interconnect [DCI] routers and top-of-rack leaf, spine, and border-leaf switches) and virtual entities (for example, VTFs) in the Virtual Topology System domain. The database also includes interconnections between these entities and details about all services instantiated within a Virtual Topology System domain.

  The resource management module manages all available resource pools in the Virtual Topology System domain, including VLANs, VXLAN Network Identifiers (VNIs), IP addresses, and multicast groups.

- **Control Plane**: The control plane module serves as the SDN control subsystem that programs the various data planes including the VTFs residing on the x86 servers, hardware leafs, DCI gateways. The Control plane hosts Service Routing (SR) module, which provides routing services to Cisco VTS. The Service Routing (SR) module is responsible for calculating L2 and L3 tables and routes to provide connectivity between the different VMs for a given tenant and service chaining. The main components of this module are the VTSR and VTF. VTSR is the controller and Virtual topology forwarder (VTF) runs on each compute server hosting the tenant VMs.

Virtual Topology Forwarder

Virtual Topology Forwarder (VTF) runs on each compute server in the DC and provides connectivity to all tenant VMs hosted on the compute server. VTF supports both intra and inter DC/WAN connectivity. VTF allows Cisco VTS to terminate VXLAN tunnels on host servers by using the VTF as a Software VXLAN Tunnel Endpoint (VTEP). Cisco VTS also supports hybrid overlays by stitching together physical and virtual endpoints into a single VXLAN segment.

VTF has 2 major components—Cisco's VPP (Vector Packet Processing) and VPFA. VPFA is a Cisco agent running on each VMM compute resource. VPFA is FIB agent which receives L2/L3 table forwarding information from VTSR needed to provide the connectivity to local tenant VMs hosted on its compute, and programs them in the VPP.

VTF is deployed as a virtual machine or in vhost mode, to deliver a high-performance software data plane on a host server.

Virtual Topology System High Availability

The Virtual Topology System solution is designed to support redundancy, with two solution instances running on separate hosts in an active-standby configuration.

During initial setup, each instance is configured with both an underlay IP address and a virtual IP address. Virtual Router Redundancy Protocol (VRRP) is used between the instances to determine which instance is active.

The active-instance data is synchronized with the standby instance after each transaction to help ensure consistency of the control-plane information to accelerate failover after a failure. BGP peering is established from both Virtual Topology System instances for the distribution of tenant-specific routes. During the switchover, nonstop forwarding (NSF) and graceful restart help ensure that services are not disrupted.

See the *Installing VTS in High Availability Mode* section of the *Cisco VTS Installation Guide* for the detailed procedure about setting up high availability.