# Cisco Software-Defined Wide-Area Networks

## Designing, Deploying and Securing Your Next Generation WAN with Cisco SD-WAN

**JASON GOOLEY,** CCIE® x2 (RS & SP) NO. 38759
**DANA YANCH,** CCDE® NO. 20130071, CCIE (RS, DC) NO. 25567
**DUSTIN SCHUEMANN,** CCIE® (RS) NO. 59235
**JOHN CURRAN**

ciscopress.com

*Foreword by* **KHALID RAZA,** Founder/CTO Viptela

# Cisco Software-Defined Wide-Area Networks

## Designing, Deploying, and Securing Your Next Generation WAN with Cisco SD-WAN

Jason Gooley CCIE No. 38759
Dana Yanch, CCDE No. 20130071, CCIE No. 25567
Dustin Schuemann, CCIE No. 59235
John Curran

**Cisco Press**

221 River St.

Hoboken, NJ 07030 USA

# Cisco Software-Defined Wide-Area Networks

Jason Gooley
Dana Yanch
Dustin Schuemann
John Curran

Published by:
Cisco Press

Library of Congress Control Number: 2020937215

ISBN-13: 978-0-13-653317-7

ISBN-10: 0-13-653317-5

## Warning and Disclaimer

This book is designed to provide information about Cisco Software-Defined Wide-Area Networks. Every effort has been made to make this book as complete and as accurate as possible, but no warranty or fitness is implied.

The information is provided on an “as is” basis. The authors, Cisco Press, and Cisco Systems, Inc. shall have neither liability nor responsibility to any person or entity with respect to any loss or damages arising from the information contained in this book or from the use of the discs or programs that may accompany it.

The opinions expressed in this book belong to the author and are not necessarily those of Cisco Systems, Inc.

Microsoft and/or its respective suppliers make no representations about the suitability of the information contained in the documents and related graphics published as part of the services for any purpose. All such documents and related graphics are provided “as is” without warranty of any kind. Microsoft and/or its respective suppliers hereby disclaim all warranties and conditions with regard to this information, including all warranties and conditions of merchantability, whether express, implied or statutory, fitness for a particular purpose, title and non-infringement. In no event shall Microsoft and/or its respective suppliers be liable for any special, indirect or consequential damages or any damages whatsoever resulting from loss of use, data or profits, whether in an action of contract, negligence or other tortious action, arising out of or in connection with the use or performance of information available from the services. The documents and related graphics contained herein could include technical inaccuracies or typographical errors. Changes are periodically added to the information herein. Microsoft and/or its respective suppliers may make improvements and/or changes in the product(s) and/or the program(s) described herein at any time. Partial screen shots may be viewed in full within the software version specified.

Microsoft® and Window® are registered trademarks of the Microsoft Corporation in the U.S.A. and other countries. Screenshots and icons reprinted with permission from the Microsoft Corporation. This book is not sponsored or endorsed by or affiliated with the Microsoft Corporation.

## Trademark Acknowledgments

## Special Sales

For information about buying this title in bulk quantities, or for special sales opportunities (which may include electronic versions; custom cover designs; and content particular to your business, training goals, marketing focus, or branding interests), please contact our corporate sales department at corpsales@pearsoned.com or (800) 382-3419.

For government sales inquiries, please contact governmentsales@pearsoned.com.

For questions about sales outside the U.S., please contact intlcs@pearson.com.

## Feedback Information

At Cisco Press, our goal is to create in-depth technical books of the highest quality and value. Each book is crafted with care and precision, undergoing rigorous development that involves the unique expertise of members from the professional technical community.

Readers' feedback is a natural continuation of this process. If you have any comments regarding how we could improve the quality of this book, or otherwise alter it to better suit your needs, you can contact us through email at feedback@ciscopress.com. Please make sure to include the book title and ISBN in your message.

We greatly appreciate your assistance.

**Editor-in-Chief:** Mark Taub

**Alliances Manager, Cisco Press:**
Makarand Chitale

**Director, Product Management:** Brett Bartow

**Managing Editor:** Sandra Schroeder

**Development Editor:** Christopher Cleveland

**Technical Editors:** Phil Davis, Aaron Rohyans

**Project Editor:** Lori Lyons

**Copy Editor:** Bart Reed

**Editorial Assistant:** Cindy Teeters

**Cover Designer:** Chuti Prasertsith

**Production Manager:** Aswini Kumar / codeMantra

**Composition:** codeMantra

**Indexer:** Tim Wright

**Proofreader:** Donna Mulder

Americas Headquarters
Cisco Systems, Inc.
San Jose, CA

Asia Pacific Headquarters
Cisco Systems (USA) Pte. Ltd.
Singapore

Europe Headquarters
Cisco Systems International BV
Amsterdam, The Netherlands

Cisco has more than 200 offices worldwide. Addresses, phone numbers, and fax numbers are listed on the Cisco Website at **www.cisco.com/go/offices.**

## About the Authors

**Jason Gooley, CCIE No. 38759 (RS and SP),** is a very enthusiastic and spontaneous person who has more than 25 years of experience in the industry. Currently, Jason works as a Technical Evangelist for the Worldwide Enterprise Networking Sales team at Cisco Systems. Jason is very passionate about helping others in the industry succeed. In addition to being a Cisco Press author, Jason is a distinguished speaker at Cisco Live, contributes to the development of the Cisco CCIE and DevNet exams, provides training for Learning@Cisco, is an active CCIE mentor, is a committee member for the Cisco Continuing Education Program (CE), and is a program committee member of the Chicago Network Operators Group (CHI-NOG), www.chinog.org. Jason also hosts a show called MetalDevOps. Jason can be found at www.MetalDevOps.com, @MetalDevOps, and @Jason_Gooley on all social media platforms.

**Dana Yanch, CCIE No. 25567 (RS,DC) CCDE No. 20130071,** at the time of writing content for this book was a Global Technical Solutions Architect at Cisco focused on designing and deploying SD-WAN solutions for large enterprises around the world. Prior to spending the last six years working with Viptela and other SD-WAN technologies, Dana had a focus on fabric-based data center technologies. Dana has presented at several Cisco Live Events worldwide and has a passion for public speaking and mentorship. Dana can now be found at Aviatrix, the multi-cloud networking platform, designing cloud connectivity architectures every single day. Dana can be found at www.danayanch.com or @DanaYanch on Twitter.

**Dustin Schuemann, CCIE No. 59235 (R&S),** is a Technical Solutions Architect at Cisco Systems. Within the Demo CoE organization, Dustin is a subject matter expert on all things SD-WAN, including development of SD-WAN demo offerings and CPOC labs for some of Cisco's largest customers. He has been a distinguished speaker at Cisco Live multiple times, where he has presented on multiple topics around Cisco SD-WAN. Dustin has more than 17 years of experience in the network engineering field, and before Cisco he was a network architect for multiple firms within the manufacturing and financial industries. He is very passionate about giving back to the IT community and helping to mentor other network engineers. Dustin currently resides in Raleigh, North Carolina. Dustin can be followed on Twitter as @dschuemann.

**John Curran** is a Technical Solutions Architect with Cisco's Global Virtual Engineering team, where he assists customers and partners with the design of their next-generation networks. John is a subject matter expert in routing and SD-WAN and is excited to spend time teaching and training on these topics. John presents regularly at Cisco Live events around the world and has been repeatedly recognized as a Distinguished Speaker. In his prior role at Cisco, John worked as a Network Consulting Engineer for Cisco's Advanced Services team, supporting government and education customers. John holds a Bachelor of Science degree in Computer Engineering Technology from the University of Cincinnati.

## Contents at a Glance

Introduction xix

Chapter 1 Introduction to Cisco Software-Defined Wide Area Networking (SD-WAN) 1

Chapter 2 Cisco SD-WAN Components 25

Chapter 3 Control Plane and Data Plane Operations 43

Chapter 4 Onboarding and Provisioning 91

Chapter 5 Introduction to Cisco SD-WAN Policies 109

Chapter 6 Centralized Control Policies 133

Chapter 7 Centralized Data Policies 227

Chapter 8 Application-Aware Routing Policies 285

Chapter 9 Localized Policies 319

Chapter 10 Cisco SD-WAN Security 349

Chapter 11 Cisco SD-WAN Cloud onRamp 393

Chapter 12 Cisco SD-WAN Design and Migration 459

Chapter 13 Provisioning Cisco SD-WAN Controllers in a Private Cloud 493

Appendix A: Answers to Chapter Review Questions 527

Appendix B: Example 7-17 539

Glossary of Key Terms 553

Index 557

## Reader Services

**Register your copy** at www.ciscopress.com/title/9780136533177 for convenient access to downloads, updates, and corrections as they become available. To start the registration process, go to www.ciscopress.com/register and log in or create an account.* Enter the product ISBN 9780136533177 and click Submit. When the process is complete, you will find any available bonus content under Registered Products.

*Be sure to check the box that you would like to hear from us to receive exclusive discounts on future editions of this product.

Chapter 4

# Onboarding and Provisioning

This chapter covers the following topics:

- **Configuration Templates:** In this section, we discuss various template types, including CLI, device, and feature templates. Design and scaling techniques with templates are discussed in this section as well.
- **Developing and Deploying Templates:** This section provides step-by-step instructions on how to build and deploy device and feature templates.
- **Onboarding Devices:** Onboarding of devices with manual bootstrapping as well as automatic provisioning with techniques such as Plug and Play (PNP) and Zero Touch Provisioning (ZTP) are discussed in this section.

Current methods for managing configurations on network devices pose a lot of challenges. These challenges include version control, human error, and scaling considerations when deploying to a large number of devices. Traditionally, network engineers will make individual changes to various network devices via the command line interface (CLI). As networks grow, these configurations are often shared or piecemealed with other network devices (such as QoS or Routing Protocol configurations). Using QoS as an example, many questions must be answered before deployment and, depending on the device, different options exist on how to modify the configuration. Is it MLS queuing or MQC queueing? What hardware platform is it? How many queues has the service provider provided? What DSCP values are you using? This creates a lot of complexity when managing configuration options. In a perfect world, all of our devices and configurations would be standardized across locations. But this isn't realistic due to a multitude of reasons (such as different providers, hardware upgrade cycles, business needs, and so on). As the network grows, the disparity among network devices and network functions makes operations and troubleshooting even more difficult. To compound this issue, network configurations tend to persist as devices are upgraded and replaced. As such, the original intent of the network configuration gets lost as the IT staff turns over or other factors change. In most cases, network device configurations are rarely revisited for cleanup. One last issue to confront is version control. Network administrators tend to make configuration changes on the fly wherein the previous config gets lost. Version control is important, especially when there are outages. Having a working configuration to roll back to can save a lot of headache. Oftentimes, outages don't always get noticed immediately after the change. With no version control or change management in place, rolling back to the last-known configuration becomes difficult, as the previous configuration was not tracked.

With the Cisco SD-WAN solution, configuration management is maintained via a robust templating engine that supports automatic rollback. Templates are all built around your intent wherein the network administrator doesn't need to be concerned with what type of device this configuration is being applied to or the specific configuration options available in that version of the operating system. Templates are built in a modular fashion where they can be reused across differing device types. This allows the network administrator to quickly roll out configurations or changes at a wide scale while also ensuring the syntax is correct and is supported on any platform. Another feature of the Cisco SD-WAN solution is automatic rollback. If a configuration option gets applied to a device and it cuts off the device's ability to be managed by vManage, then the device will automatically roll back and allow the network engineer to correct the issue.

With today's networks getting larger and larger, the need to reduce provisioning and onboarding of devices needs to decrease. For example, to onboard a network device at a branch, the following approach is usually taken:

1. The device ships from manufacturing to IT staff.
2. The network administrator applies configuration to the device.
3. The IT staff drives or ships the device to the location and physically installs it.

This process is expensive for the organization, especially when looking at performing these actions across thousands of devices. If there are issues, and the device won't connect to the network, this adds further delay and operational expense as IT teams work to troubleshoot. In some cases, these installs are performed remotely. While this reduces the cost of traveling, it can create a lot of frustration if and when an issue arises. With a lack of local IT support at the site, network administrators often end up having to rely on non-IT staff to perform operations on the device. This puts a lot of faith in the remote staff and potentially takes them away from their day job. The Cisco SD-WAN solution simplifies onboarding and provisioning of a device. WAN Edge devices support mechanisms such as PNP and ZTP to automatically bring the device online and into the fabric. These onboarding solutions work by allowing the network administrator to pre-configure the device within the vManage controller. Once the vManage controller sees the device, it will automatically apply the specified configuration. WAN Edges can be shipped directly to the remote location and don't require initial configuration from IT staff. Once physically installed at the remote site, the device will automatically locate the vManage controller and begin the provisioning process. By reducing the time to bring up remote sites, the IT staff can bring more devices online more quickly and with fewer errors. This time-savings reduces operational cost, which allows IT staff to focus on bringing additional capabilities to the business.

## Configuration Templates

Configurations can be applied one of two ways in the Cisco SD-WAN solution. The network administrator can either apply configuration manually via the CLI (that is, by using SSH to connect to the device or by connecting via the console port) or by using the vManage GUI. Using the vManage GUI is the preferred mechanism, as it is less error-prone and has support for automatic recovery. Configurations provisioned on vManage can be applied to both WAN Edges and vSmart controllers. When vManage is responsible for applying the configuration, it is the single source of truth, and changes can only be applied via vManage.

**Note** To apply centralized policy to the vSmart controller, vSmart needs to be under the control of the vManage controller. When a component is managed by vManage, the administrator will be unable to make changes locally to the device.

Key Topic

When applying configuration to WAN Edge devices or controllers using the vManage GUI, a network administrator will apply a device template to a single device or multiple devices. These **device templates** (see Figure 4-1) can either be CLI based or feature template based. When a CLI template is being built, the whole configuration must be in the template (not just specific configuration snippets) as opposed to feature templates. Feature templates can be thought of as building blocks wherein each block is a specific technology feature. Feature templates define what specific feature or technology you want enabled or configured, such as routing protocols, interface parameters, and Overlay Management Protocol (OMP). Feature templates can be reused between multiple device templates, and it is this flexibility that brings greater scale to the solution (and why feature templates are the recommended way of configuring devices). Feature templates can be device type agnostic as well. The network administrator needs only to be concerned with the intent of the configuration. When vManage applies the configuration to a specific device, be it a Cisco IOS-based device or a Viptela OS device, vManage will apply the correct device syntax.

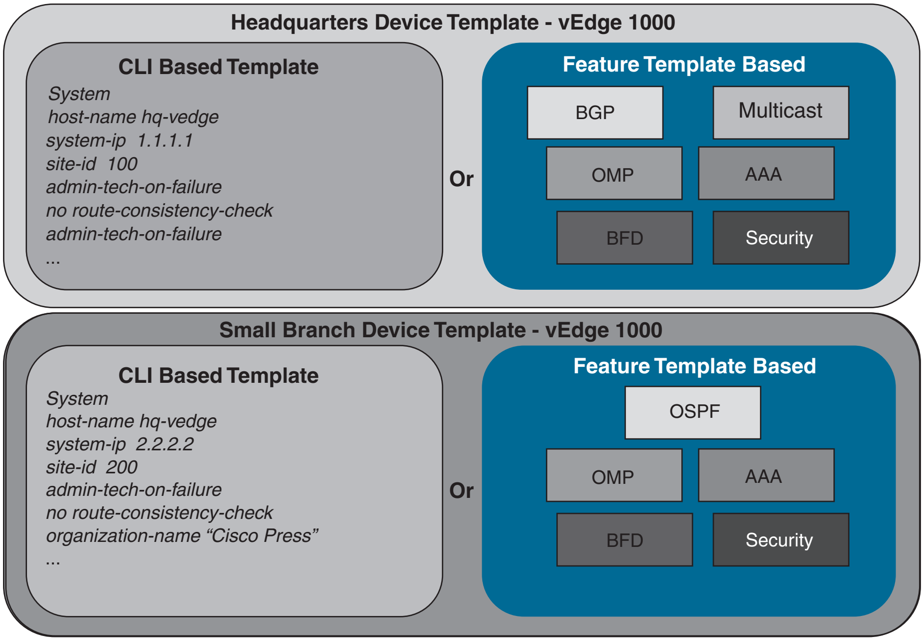


**Figure 4-1** *Device Templates*

**Device templates** are a collection of feature templates and can only be applied to specific device types. For this reason, you may have multiple device templates for the same model of hardware, depending on the device's location, connectivity options, or what role it is playing in the network. A device template can't be shared across different device types, but feature templates can be used across multiple different device types. As illustrated in Figure 4-2, there are four main parts or groups of a device template:

- **Basic Information:** This section includes items such as System, Logging, AAA, BFD, and OMP feature templates.
- **Transport and Management VPN:** This section has templates for configuration of VPN 0 and VPN 512 (such as underlay routing protocol configuration and interface configuration).
- **Service VPN:** This section is where service VPNs or LAN-facing template configurations will exist. This is where BGP, OSPF, and interface parameters are configured.
- **Additional Templates:** This section is for local policies, security policies, SNMP configuration templates, and so on.

Key Topic

**Feature templates** make configuration options extremely flexible. For example, feature templates provide the option to define variables for configuration parameters. This allows you to reduce the number of templates required in your deployment, yet make it much more modular. To further this example, suppose you have MPLS transports that use different physical interface numbers: Gi0/0, Gi0/1, Gi0/2, and so on. Initially, the thought may be to build a feature template for each physical interface with a different IP address. This would result in three different templates. By using variables for the physical interface and IP address options, the administrator can condense this down to one feature template that can be used across all device templates.

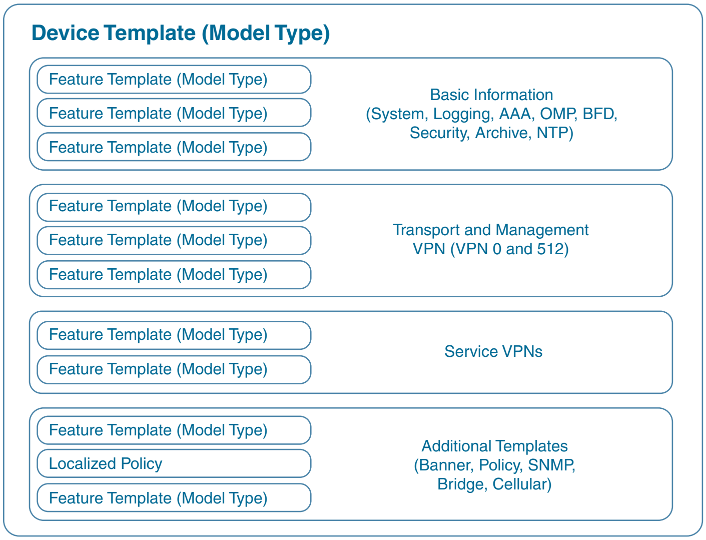
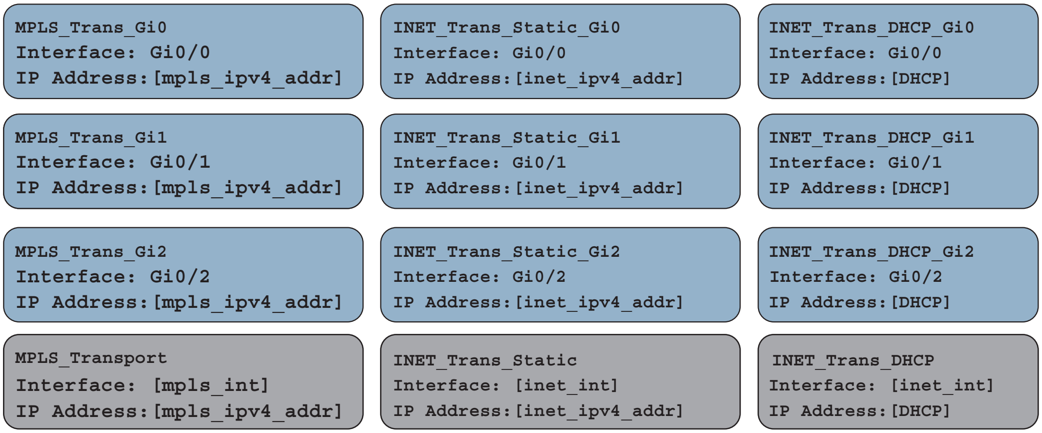

Device Template (Model Type)
Feature Template (Model Type)
Feature Template (Model Type)
Feature Template (Model Type)
Basic Information (System, Logging, AAA, OMP, BFD, Security, Archive, NTP)
Feature Template (Model Type)
Feature Template (Model Type)
Feature Template (Model Type)
Transport and Management VPN (VPN 0 and 512)
Feature Template (Model Type)
Feature Template (Model Type)
Service VPNs
Feature Template (Model Type)
Localized Policy
Feature Template (Model Type)
Additional Templates (Banner, Policy, SNMP, Bridge, Cellular)

**Figure 4-2** *Device Template Structure*

Figure 4-3 shows how variables can be utilized to control template sprawl. In this example, we have nine different interface templates, depending on if the IP address is assigned via DHCP and what interface is used. By using variables, the network administrator can reduce this down to three different feature templates.



| | | |
|---|---|---|
| MPLS_Trans_Gi0<br>Interface: Gi0/0<br>IP Address:[mpls_ipv4_addr] | INET_Trans_Static_Gi0<br>Interface: Gi0/0<br>IP Address:[inet_ipv4_addr] | INET_Trans_DHCP_Gi0<br>Interface: Gi0/0<br>IP Address:[DHCP] |
| MPLS_Trans_Gi1<br>Interface: Gi0/1<br>IP Address:[mpls_ipv4_addr] | INET_Trans_Static_Gi1<br>Interface: Gi0/1<br>IP Address:[inet_ipv4_addr] | INET_Trans_DHCP_Gi1<br>Interface: Gi0/1<br>IP Address:[DHCP] |
| MPLS_Trans_Gi2<br>Interface: Gi0/2<br>IP Address:[mpls_ipv4_addr] | INET_Trans_Static_Gi2<br>Interface: Gi0/2<br>IP Address:[inet_ipv4_addr] | INET_Trans_DHCP_Gi2<br>Interface: Gi0/2<br>IP Address:[DHCP] |
| MPLS_Transport<br>Interface: [mpls_int]<br>IP Address:[mpls_ipv4_addr] | INET_Trans_Static<br>Interface: [inet_int]<br>IP Address:[inet_ipv4_addr] | INET_Trans_DHCP<br>Interface: [inet_int]<br>IP Address:[DHCP] |

**Figure 4-3** *Controlling Template Sprawl*

Three types of values can be defined in a template:

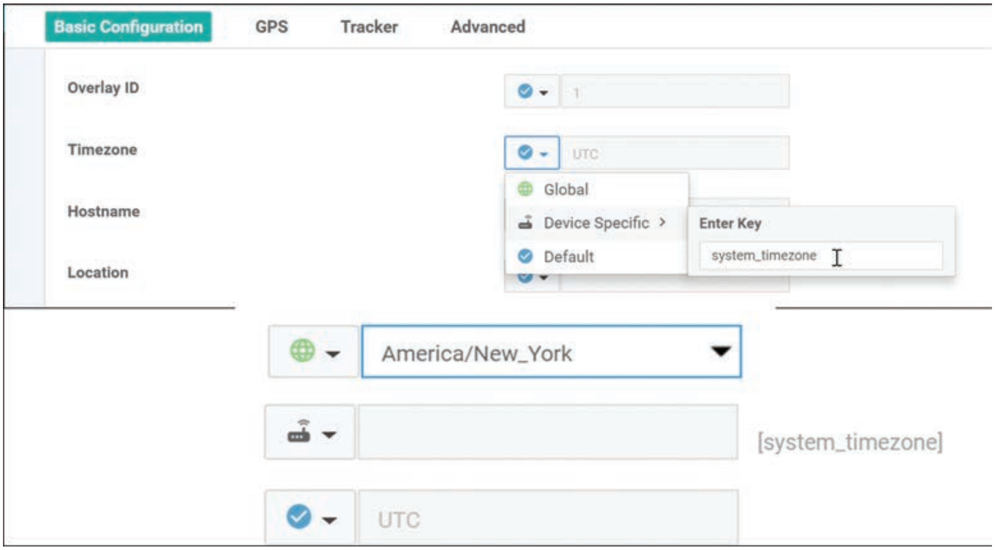
- **Default:** Factory default value. Default values cannot be changed. An example might be using the default BFD timers.
- **Global:** Values set here will be the same wherever this configuration option is used. An example could be SNMP community strings that you want globally applied to all devices utilizing this template. The beauty of this is that, later on (if there needs to be a change to these values), you just update the feature template global option and it updates every device template that is using this feature template.
- **Device Specific:** The value is set via a user-defined variable. This is the preceding referenced example with interface names. The values to these variables are set when the device template is attached to a specific device.

Looking at Figure 4-4, the network administrator can see how these referenced values can be utilized. Some template options might not have all three of these options, depending what is being configured. For example, a BGP AS number won't have a default value.

A large number of feature template options can be configured. Here are some common feature templates:

- **System:** Configure basic system information such as System IP, Site ID, and Hostname.
- **BFD:** Adjust BFD timers and app-route multipliers for each transport or color. BFD timers are used for App-Aware Routing.
- **OMP:** Change graceful restart timers or control redistribution from other routing protocols into OMP.
- **Security:** Change IPsec security settings such as anti-replay, authentication, and encryption.
- **VPN:** Define a service VPN, routing protocol redistribution, or static routing.
- **BGP:** Configuration of BGP in a VPN or VRF.
- **OSPF:** Configuration of OSPF in a VPN or VRF.
- **VPN Interface:** Define an interface that is part of a service VPN or VRF. Common configuration options here include IP Address, QoS, ACLs, and NAT.

As the feature templates are defined, they can be referenced via a device template. After the device template is created, it can be applied to a specific device or a group of devices. Remember, device templates can only be built for a specific device type. If there are any variables defined in the feature template, at the time the device template is attached, these values will need to be populated. Once these values are defined, a configuration syntax check is done in vManage. If successful, the configuration is then pushed to the device. Feature template variable values can be populated either within the vManage template attachment workflow or by using a CSV file. Populating feature template variables via CSV allows an administrator to quickly provision many devices all at once. If, at the time the configuration is pushed to the device, the WAN Edge loses control plane connectivity to the vManage controller, the WAN Edge will start a rollback timer of 5 minutes. If it doesn't re-establish connectivity within that 5 minutes, it will roll back its configuration and reconnect to vManage using the last known-good configuration. At this time, the network administrator will see that the device is out of sync and can correct the issue.



Overlay ID 1

Timezone UTC

Hostname

Location

Enter Key system_timezone

America/New_York

[system_timezone]

UTC

**Figure 4-4** *Setting Variables*

If the need arises to change these values after the device template has been applied, an option exists to change these values on a device-by-device basis. If any changes are made to a feature template or device template, vManage will immediately push the updated configuration to all devices utilizing that template. An example of this could be changing the IP address or the username and password of the device.

## Developing and Deploying Templates

Template configuration and creation is performed in the vManage GUI. After initial installation of vManage, some default templates are created. These templates can be used as a starting point, or new ones can be created. To create templates, the network administrator will navigate to **Configuration > Templates.**

**Step 1.** Go to the configuration section for templates, as illustrated in Figure 4-5.

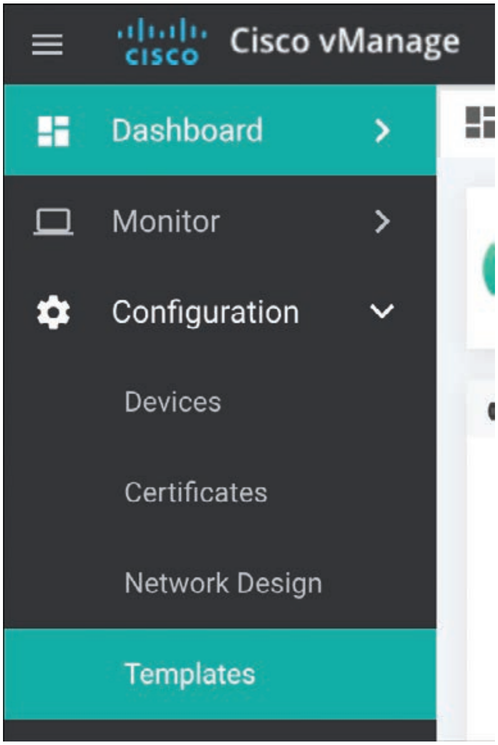


**Figure 4-5** *Accessing the Template Configuration Interface*

**Step 2.** Once at the template configuration window, you're presented with the option to configure device templates or feature templates, as illustrated in Figure 4-6.

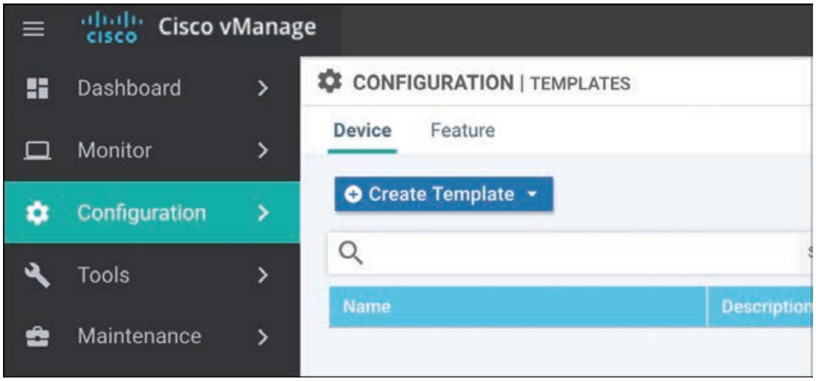
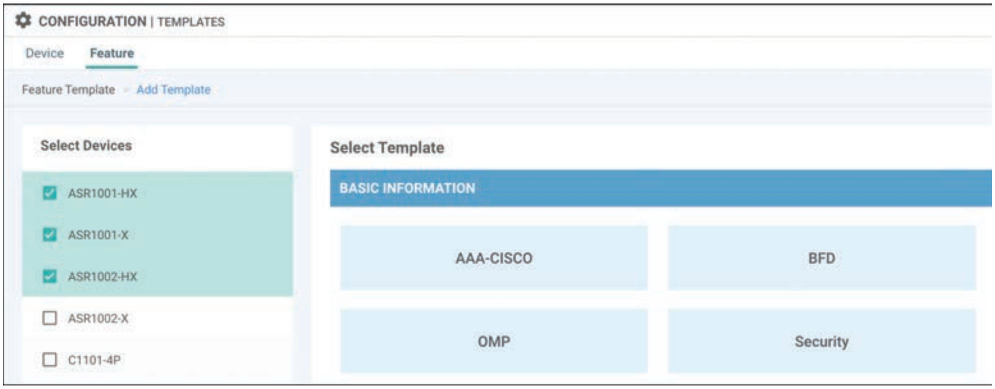


**Figure 4-6** *Configuration Templates Window*

**Step 3.** The next step is to begin creating feature templates. Select **Feature (tab) > Add Template.** Select the devices that this template will apply to and select the type of template, as illustrated in Figure 4-7.



CONFIGURATION | TEMPLATES

Device | Feature

Feature Template › Add Template

Select Devices

- ☑ ASR1001-HX
- ☑ ASR1001-X
- ☑ ASR1002-HX
- ☐ ASR1002-X
- ☐ C1101-4P

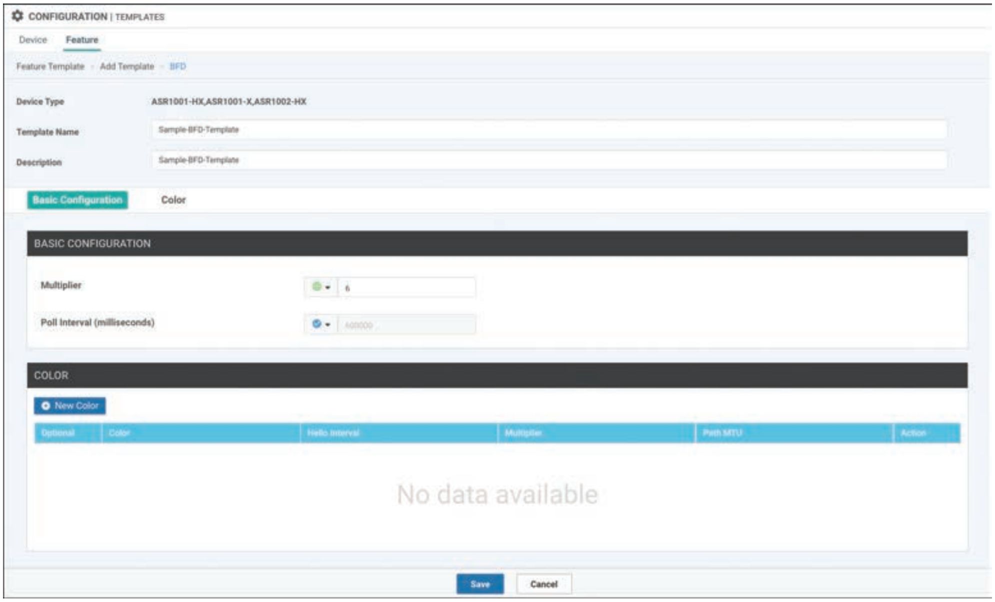
Select Template

BASIC INFORMATION

**Figure 4-7** *Template Configuration: Device Selection Window*

**Step 4.** After selecting the type of feature template you wish to configure, you now have the ability to start setting values, as illustrated in Figure 4-8. These values can be either variables, global parameters, or the default parameter. After the configuration options have been set and the template has been named, click **Save**.

**Step 5.** Now that the feature templates have been created, they need to be attached to their respective device templates. Click **Device** (tab) followed by **Create Template.** After you click **Create Template**, the option to create a CLI-based template or a feature-based template is provided. Select **From Feature Template**, as illustrated in Figure 4-9.



CONFIGURATION | TEMPLATES

Device | Feature

Feature Template › Add Template › BFD

Device Type: ASR1001-HX,ASR1001-X,ASR1002-HX

Template Name: Sample-BFD-Template

Description: Sample-BFD-Template

BASIC CONFIGURATION

Multiplier: 6

Poll Interval (milliseconds): 600000

COLOR

| Optional | Color | Hello Interval | Multiplier | Path MTU | Action |
|---|---|---|---|---|---|

No data available

**Figure 4-8** *Template Configuration: Setting Configuration Values*

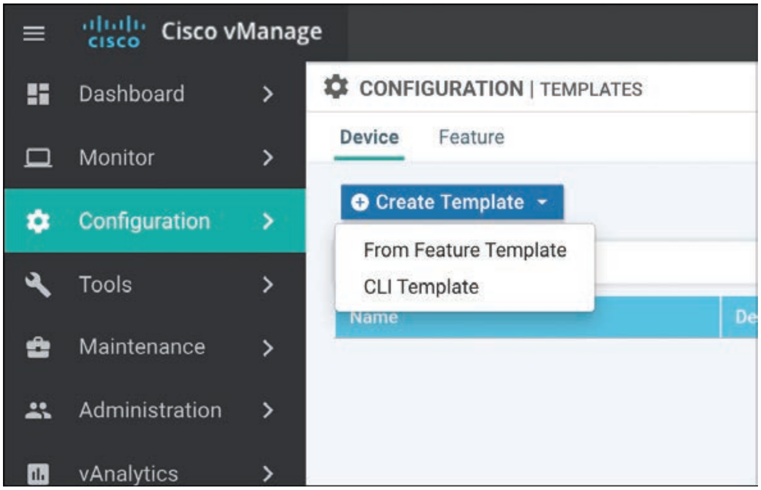
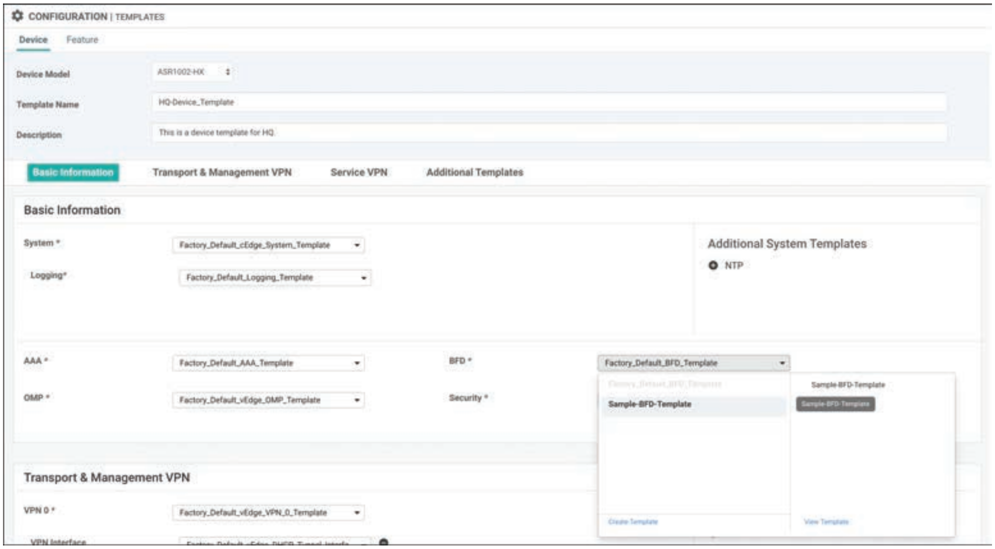


**Figure 4-9** *Template Configuration: Device Template Selection*

**Step 6.** After selecting a feature-based template, you need to select the device model this template will apply to as well as provide a name for the template, as illustrated in Figure 4-10. Once this is done, you have the ability to start selecting what feature templates to use. In this example, we'll select the BFD template we created in the previous example. Once done, select **Save**.



Device Model: ASR1002-HX
Template Name: HQ-Device_Template
Description: This is a device template for HQ.

Basic Information

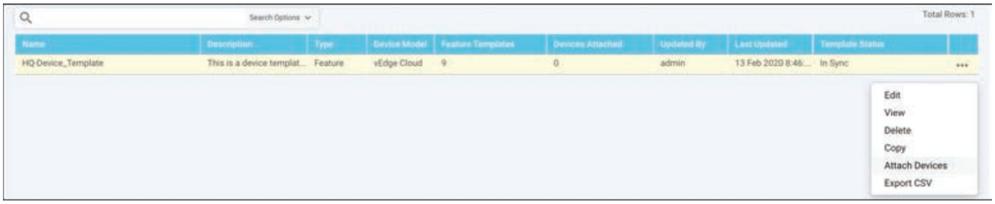
System * Factory_Default_cEdge_System_Template
Logging*: Factory_Default_Logging_Template
AAA *: Factory_Default_AAA_Template
OMP *: Factory_Default_vEdge_OMP_Template
BFD *: Factory_Default_BFD_Template
Security *

Transport & Management VPN

VPN 0 *: Factory_Default_vEdge_VPN_0_Template
VPN Interface

**Figure 4-10** *Template Configuration: Device Template Feature Selection*

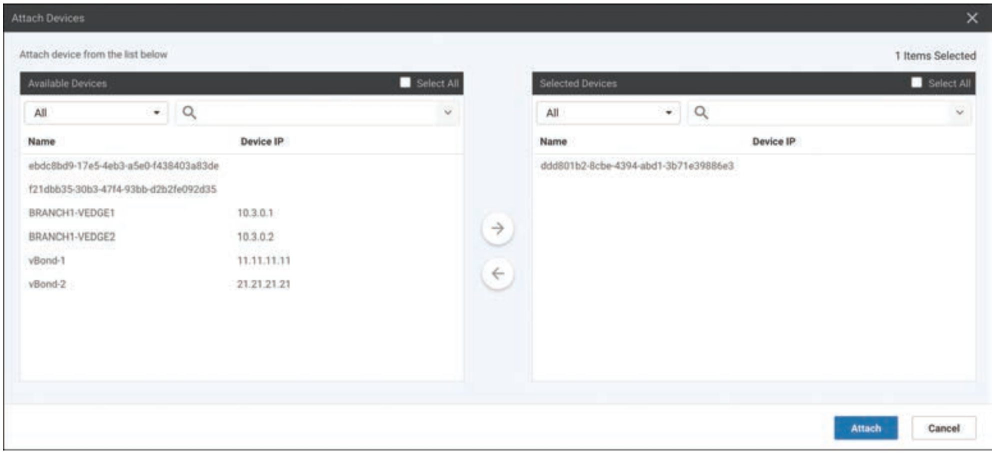
**Step 7.** Now that the device template is created, you can attach it to devices. From the Device Templates page, click the ellipses next to the template you wish to attach. Select the option to **Attach Devices**, as illustrated in Figure 4-11.



| Name | Description | Type | Device Model | Feature Templates | Devices Attached | Updated By | Last Updated | Template Status |
|---|---|---|---|---|---|---|---|---|
| HQ-Device_Template | This is a device templat... | Feature | vEdge Cloud | 9 | 0 | admin | 13 Feb 2020 8:46... | In Sync |

**Figure 4-11** *Attaching a Template to a Device*

**Step 8.** From here, you can select the devices to apply the configurations to, as illustrated in Figure 4-12. Once you select the device, you have the option to populate any variables.



Available Devices

| Name | Device IP |
|---|---|
| ebdc8bd9-17e5-4eb3-a5e0-f438403a83de | |
| f21dbb35-30b3-47f4-93bb-d2b2fe092d35 | |
| BRANCH1-VEDGE1 | 10.3.0.1 |
| BRANCH1-VEDGE2 | 10.3.0.2 |
| vBond-1 | 11.11.11.11 |
| vBond-2 | 21.21.21.21 |

Selected Devices

| Name | Device IP |
|---|---|
| ddd801b2-8cbe-4394-abd1-3b71e39886e3 | |

**Figure 4-12** *Device Selection Window*

## Onboarding Devices

Key Topic

For a WAN Edge to join the SD-WAN fabric, the WAN Edge first needs to establish connectivity to the vBond controller. The vBond controller facilitates discovery of the vManage and vSmart controllers. As the WAN Edge establishes connectivity to each of these controllers, mutual authentication will occur. After the WAN Edge has authenticated to the controller components in the overlay, the device will receive its full configuration from vManage. There are two methods to bootstrap a device with initial configuration so that it can reach vBond. The less preferred, and more obvious, method is to manually apply minimal configuration to the device. The second method is automatically discovering the network using Zero Touch Provisioning (ZTP) or Plug and Play (PNP). If the device is running Viptela OS, it will use ZTP. If the device is an IOS XE-based device, then Plug and Play is utilized. The general process of ZTP and PNP is similar. The following sections elaborate on each process.

## Manual Bootstrapping of a WAN Edge

To manually bootstrap a WAN Edge device, the network administrator will begin by applying a minimal configuration to the device. This includes IP addressing, vBond addressing (either DNS hostname or IP), and system identification information. This information is used to establish initial connectivity and authentication. The process to manually bootstrap a device is as follows:

**Step 1.** Configure an IP address and default gateway. If DHCP is available, this can be used to assign the IP and gateway automatically.

**Step 2.** Configure the vBond IP or hostname. If you are using a hostname, then a DNS server address must be provided, and the device must have reachability from VPN 0.

**Step 3.** Configure device identification information, including the system IP, site ID, and organization name.

Examples 4-1 and 4-2 show the minimal configuration for Viptela OS and SD-WAN IOS-XE devices.

**Example 4-1** *Minimal Configuration for a Viptela OS–based Device*

```
vEdge# config
vEdge(config)#
vEdge(config)# system host-name hostname
vEdge(config-system)#system-ip ip-address
vEdge(config-system)# site-id site-id
vEdge(config-system)# organization-name organization-name
vEdge(config-system)# vbond (dns-name | ip-address)
vEdge(config)# vpn 0
vEdge(config-vpn-0)# interface interface-name
vEdge(config-interface)# (ip dhcp-client | ip address prefix/length)
vEdge(config-interface)# no shutdown
vEdge(config-interface)# tunnel-interface
vEdge(config-tunnel-interface)# color color
vEdge(config-vpn-0)# ip route 0.0.0.0/0 next-hop
vEdge(config)# commit and-quit
```

**Example 4-2** *Minimal Configuration for an IOS-XE based Device*

```
Device# config-transaction
Device(config)#
Device(config)# system host-name  hostname
Device(config-system)# system-ip  ip-address
Device(config-system)# site-id site-id
Device(config-system)# vbond (dns-name | ip-address)
Device(config-system)# organization-name name
Device(config)# interface Tunnel #
Device(config-if)# ip unnumbered wan-physical-interface
Device(config-if)# tunnel source wan-physical-interface
Device(config-if)# tunnel mode sdwan
Device(config)# interface GigabitEthernet  #
Device(config)# ip address  ip-address mask
Device(config)# no shut
Device(config)# exit
Device(config)# sdwan
Device(config-sdwan)# interface  WAN-interface-name
Device(config-interface-interface-name)# tunnel-interface
Device(config-tunnel-interface)# color color
Device(config-tunnel-interface)# encapsulation  ipsec
Device(config)# ip route 0.0.0.0 0.0.0.0 next-hop-ip-address
Device(config)# ip domain lookup
Device(config)# ip name-server dns-server-ip-address
Device(config)# commit
Device# exit
```

Key Topic

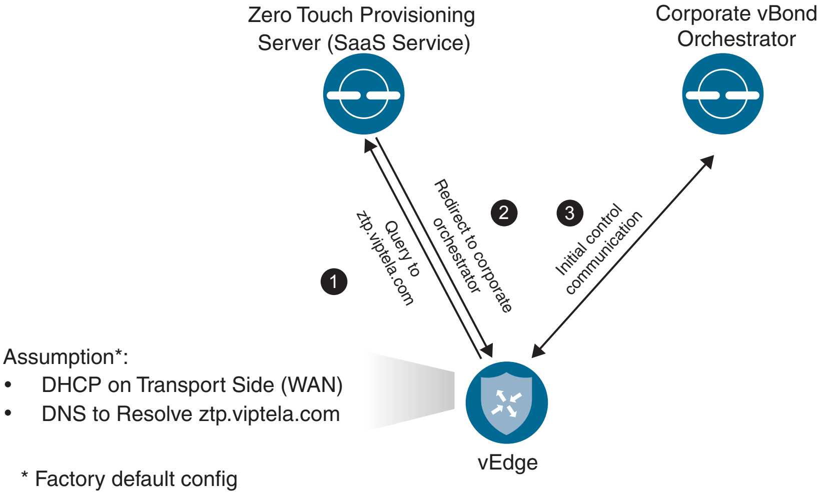
## Automatic Provisioning with PNP or ZTP

The second method of provisioning devices allows the network administrator to automatically bring the devices online with minimal effort and involvement. Once powered on, the default configuration on the device tries to receive an IP address via DHCP. Once the device has an IP address, it will reach out to the automatic provisioning server (hosted by Cisco) and learn about the organization's vBond. At this point, the process is exactly the same as the manual bootstrapping process. The device will connect and authenticate to the vBond, learn of vManage and vSmart, and then receive its configuration.

**Note** The automatic provisioning servers are managed via the Plug and Play portal at http://software.cisco.com. As devices are purchased from Cisco, their serial numbers will be populated here. vManage can also be configured to synchronize with this portal to automatically populate the organization's devices into vManage.

Before the automatic provisioning process can be initiated, the network administrator needs to attach a device template in vManage for the respective device. The device template must have the system IP and site ID for the device populated as well. If none of this is completed, the process will not succeed. Once vManage sees the device for the first time, it will push the template that is assigned to the matching serial number of the device performing ZTP or PNP.

As briefly described previously, depending on the type of device, the process is slightly different. If the device is a Viptela OS–based device, ZTP will be used. Once the device boots up, it will start the process to receive an IP address and DNS server via DHCP. After this succeeds, it will try to resolve ztp.viptela.com. If successful, the device will connect to the ZTP server and (after the ZTP server verifies what organization the device belongs to) it will redirect the device to the correct vBond for the organization. The ZTP server is able to verify which organization the device belongs to by checking the serial number of the device against its ZTP entries database. Once it's connected to the vBond, the normal process continues. For ZTP to function, two things must happen: DHCP must be available on the WAN (VPN 0) facing interface and the device must be able to resolve **ztp.viptela.com**. Each Viptela OS–based device has specific interfaces that are to be used for ZTP. Refer to the latest product documentation to determine which interface should be used. Figure 4-13 outlines the process, described in further detail in the list that follows.



Zero Touch Provisioning Server (SaaS Service)
Corporate vBond Orchestrator
1
Query to ztp.viptela.com
2
Redirect to corporate orchestrator
3
Initial control communication
Assumption*:
• DHCP on Transport Side (WAN)
• DNS to Resolve ztp.viptela.com
* Factory default config
vEdge

**Figure 4-13** *Zero Touch Provisioning Workflow*

1. The vEdge device queries **ztp.viptela.com**. The ZTP server verifies that the device's serial number and organization exist in the ZTP database.
2. If the vEdge performing the ZTP process exists in the ZTP database, the ZTP server responds telling the vEdge what the connectivity information is for the organization's vBond controller.

3. The vEdge then connects to the corporate vBond and goes through the authentication process. If successful, the vBond will tell the vEdge about the vSmart and vManage controllers in the overlay. At this point, vManage will push the necessary configuration to the device.

For Cisco IOS-XE based devices, a slightly different method is used instead of ZTP. PNP operates almost identically to ZTP, except instead of building a DTLS tunnel to the PNP server (**devicehelper.cisco.com**), the device will communicate to the server via HTTPS. After the PNP server validates the device, it will redirect the IOS-XE based WAN Edge to the relevant vBond for the organization. IOS-XE based devices have the same requirements as ZTP devices in that they must get an IP address and DNS server via DHCP and be able to resolve **devicehelper.cisco.com**. Figure 4-14 outlines the process of onboarding a device with PNP.

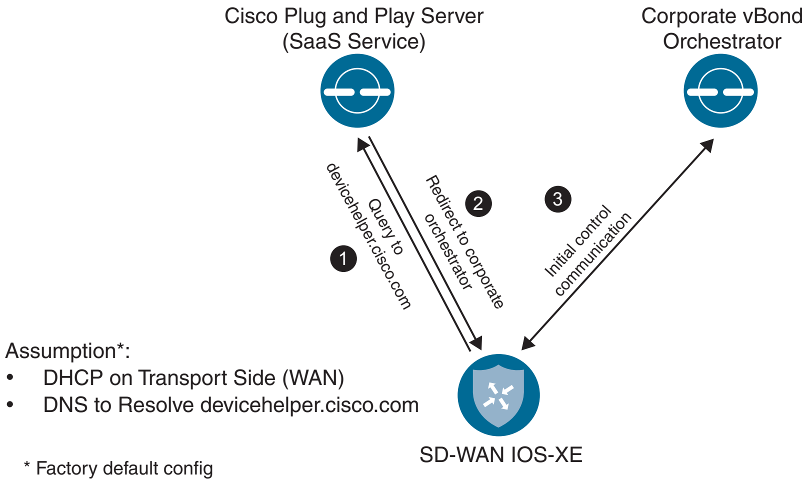


**Figure 4-14** *Plug and Play Workflow*

# Summary

With the Cisco SD-WAN solution, network configurations are handled with ease via a powerful template engine. Feature templates can be utilized to achieve modularity of configuration that can be reused across various platforms. All of these feature templates are used to form a device template. By using variables within these templates, the network administrator can support a large range of configuration requirements.

Provisioning of devices can occur with one of two methods: either manual or with an automatic process such as Plug and Play or Zero Touch Provisioning. With the manual method, the amount of configuration is very small. The device must have system IP, site ID, organization name, and IP address information. Once the device authenticates to the organization’s vBond, it will discover the rest of the controller elements, and vManage can push configuration down to the device. The second option involves using an automatic onboarding process. This process uses either ZTP or PNP, depending on if it's a Viptela OS–based product or a Cisco IOS-based product. Once the device has gone through the automatic method and determined the vBond to use, the process works just the same as with the manual method. This allows the network administrator to deploy a large number devices quickly, even if they have different configuration options.

## Review All Key Topics

Review the most important topics in the chapter, noted with the Key Topic icon in the outer margin of the page. Table 4-1 lists these key topics and the page numbers on which each is found.

Key Topic

**Table 4-1** *Key Topics*

| Key Topic Element | Description | Page |
|---|---|---|
| Paragraphs | Device templates are a collection of feature templates or a CLI template. These templates are what are applied to the device. | 93 |
| Paragraph | Feature templates make up the building blocks or configuration options for various features. These templates are selected from within the device template. Variables can be utilized to make these templates even more modular. | 94 |
| Section | Onboarding Devices<br>Zero Touch Provisioning, or ZTP, is the automatic onboarding process used by Viptela OS–based devices. Connectivity to the ZTP server is established via DTLS tunnel. | 101 |
| Section | Automatic Provisioning with PNP or ZTP<br>Just like ZTP, Plug and Play, or PNP, is an automatic process for onboarding Cisco IOS-XE based devices. PNP uses HTTPS to connect to the PNP server. | 103 |

## Chapter Review Questions

1. What two methods can be used to construct device templates?
   a. CLI
   b. Feature templates
   c. Directly on the device
   d. Multiple CLI Templates

*This page intentionally left blank*