

Contents

Configuring Puppet	1
Overview	1
Puppet network framework	1
Puppet resources	2
Configuring Puppet	2
Configuration prerequisites	2
Starting Puppet	3
Shutting down Puppet on the device.....	3
Puppet configuration example	4
Network requirements	4
Configuration procedure.....	4
Puppet resources	5
netdev_vlan.....	5
Attributes	5
Resource example	5
netdev_device	5
Attributes	6
Resource example	6
netdev_interface.....	6
Attributes	7
Resource example	8
netdev_l2_interface.....	8
Attributes	8
Resource example	9
netdev_lagg.....	9
Attributes	9
Resource example	10
netdev_l2vpn.....	10
Attributes	11
Resource example	11
netdev_vsi	11
Attributes	11
Resource example	11
netdev_vte.....	12
Attributes	12
Resource example	13
netdev_vxlan	13
Attributes	13
Resource example	14

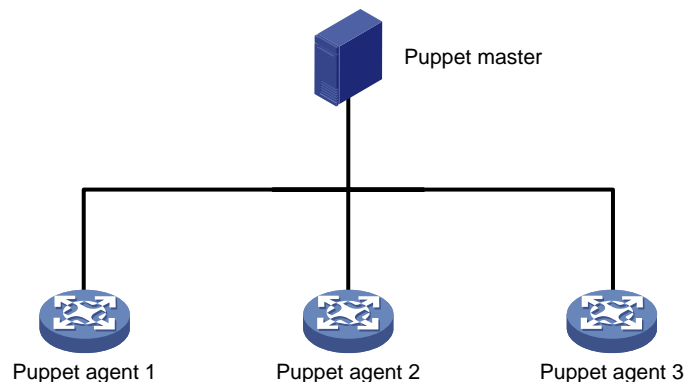
Configuring Puppet

Overview

Puppet is an open-source configuration management tool. It provides the Puppet language. You can use the Puppet language to create configuration manifests and save them to a server. You can then use the server for centralized configuration enforcement and management.

Puppet network framework

Figure 1 Puppet network framework



As shown in [Figure 1](#), Puppet operates in a client/server network framework. In the framework, the Puppet master (server) stores configuration manifests for Puppet agents (clients). The Puppet agents establish SSL connections to the Puppet master to obtain their respective latest configurations.

Puppet master

The Puppet master runs the Puppet daemon process to listen to requests from Puppet agents, authenticates Puppet agents, and sends configurations to Puppet agents on demand.

Requests that the Puppet master receives from Puppet agents never time out on the Puppet server.

For information about installing and configuring a Puppet master, see the official Puppet website at

<https://puppetlabs.com/>.

Puppet agent

H3C devices support Puppet 3.7.3 agent. The following is the communication process between a Puppet agent and the Puppet master:

1. The Puppet agent sends an authentication request to the Puppet master.
2. The Puppet agent checks with the Puppet master for the authentication result periodically (every two minutes by default). Once the Puppet agent passes the authentication, a connection is established to the Puppet master.
3. After the connection is established, the Puppet agent sends a request to the Puppet master periodically (every 30 minutes by default) to obtain the latest configuration.
4. After obtaining the latest configuration, the Puppet agent compares the configuration with its running configuration. If a difference exists, the Puppet agent overwrites its running configuration with the newly obtained configuration.

5. After overwriting the running configuration, the Puppet agent sends a feedback to the Puppet master.

Puppet resources

A Puppet resource is a unit of configuration. Puppet uses manifests to store resources.

Puppet manages types of resources. Each resource has a type, a title, and one or more attributes. Every attribute has a value. The value specifies the state desired for the resource. You can specify the state of a device by setting values for attributes regardless of how the device enters the state. The following resource example shows how to configure a device to create VLAN 2 and configure the description for VLAN 2.

```
netdev_vlan{'vlan2':
  ensure => undo_shutdown,
  id => 2,
  description => 'sales-private',
  require => Netdev_device['device'],
}
```

The following are the resource type and title:

- **netdev_vlan**—Type of the resource. The **netdev_vlan** type resources are used for VLAN configuration.
- **vlan2**—Title of the resource. The title is the unique identifier of the resource.

The example contains the following attributes:

- **ensure**—Creates, modifies, or deletes a VLAN. To create a VLAN, set the attribute value to **undo_shutdown**. To delete a VLAN, set the attribute value to **shutdown**.
- **id**—Specifies a VLAN by its ID. In this example, VLAN 2 is specified.
- **description**—Configures the description for the VLAN. In this example, the description for VLAN 2 is **sales-private**.
- **require**—Indicates that the resource depends on another resource (specified by resource type and title). In this example, the resource depends on a **netdev_device** type resource titled **device**.

NOTE:

For information about resource types supported by Puppet, see "[Puppet resources](#)."

Configuring Puppet

Configuration prerequisites

Before configuring Puppet on the device, complete the following tasks on the device:

- Enable NETCONF over SSH. The Puppet master sends configuration information to Puppet agents through NETCONF over SSH connections. For information about NETCONF over SSH, see "Configuring NETCONF."
- Configure SSH login. Puppet agents communicate with the Puppet master through SSH. For information about SSH login, see *Fundamentals Configuration Guide*.
- For successful communication, verify that the Puppet master and agents use the same system time. You can manually set the same system time for the Puppet master and agents or configure them to use a time synchronization protocol such as NTP. For information about NTP,

see "Configuring NTP." For information about setting the system time, see device management in *Fundamentals Configuration Guide*.

Starting Puppet

Perform the following tasks to configure the Puppet master and a Puppet agent:

1. Install and configure the Puppet server, and create manifests for Puppet agents on the Puppet master. For more information, see the Puppet master installation and configuration guides.

ⓘ IMPORTANT:

The Puppet master cannot run a lower Puppet version than Puppet agents.

2. Start Puppet on the device:

Step	Command	Remarks
1. Enter system view.	system-view	N/A
2. Start Puppet.	third-part-process start name puppet arg agent --certname=certname --server=server	By default, Puppet is shut down. Specify the address of the Puppet agent for the <i>certname</i> argument. Specify the address of the Puppet master for the <i>server</i> argument.

After the Puppet process starts up, the Puppet agent sends an authentication request to the Puppet master.

3. Execute the **puppet cert sign certname** command on the Puppet master to sign a certificate for the Puppet agent.

After the certificate is signed successfully, the Puppet agent establishes an SSL connection to the Puppet master and requests configuration information from the Puppet master.

Shutting down Puppet on the device

Step	Command	Remarks
1. Identify the ID of the Puppet process.	display process all	This command displays information about all processes on the device. Check the following fields: <ul style="list-style-type: none"> • THIRD—The field displays Y for a third-party process. • PID—Process ID. • COMMAND—The field displays puppet /opt/ruby/bin/pu for the Puppet process.
2. Enter system view.	system-view	N/A
3. Shut down Puppet.	third-part-process stop pid pid-list	N/A

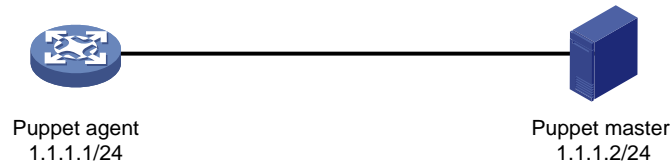
Puppet configuration example

Network requirements

As shown in [Figure 2](#), the device is connected to the Puppet master. Use Puppet to configure the device to perform the following operations:

- Set the SSH login username and password to **user** and **passwd**, respectively.
- Create VLAN 3.

Figure 2 Network diagram



Configuration procedure

1. Configure SSH login and enable NETCONF over SSH on the device. (Details not shown.)
2. On the Puppet server, create the **modules/custom/manifests** directory in the **/etc/puppet/** directory to store configuration manifests.

```
$ mkdir -p /etc/puppet/modules/custom/manifests
```
3. Create configuration manifest **init.pp** in the **/etc/puppet/modules/custom/manifests** directory as follows:

```
netdev_device{'device':  
  ensure => undo_shutdown,  
  username => 'user',  
  password => 'passwd',  
  ipaddr => '1.1.1.1',  
}  
netdev_vlan{'vlan3':  
  ensure => undo_shutdown,  
  id => 3,  
  require => Netdev_device['device'],  
}
```

The **/etc/puppet/manifests** directory is automatically created when the Puppet master is installed.

4. Start Puppet on the device.

```
<PuppetAgent> system-view  
[PuppetAgent] third-part-process start name puppet arg agent --certname=1.1.1.1  
--server=1.1.1.2
```
5. Configure the Puppet master to authenticate the request from the Puppet agent.

```
$ puppet cert sign 1.1.1.1
```

After passing the authentication, the Puppet agent requests the latest configuration for it from the Puppet master.

Puppet resources

netdev_vlan

Use this resource to create, modify, or delete a VLAN or configure the description for the VLAN.

Attributes

Table 1 Attributes for netdev_vlan

Attribute name	Description	Attribute type	Value type and restrictions
ensure	Creates, modifies, or deletes a VLAN.	N/A	Symbol: <ul style="list-style-type: none">• undo_shutdown—Creates or modifies a VLAN.• shutdown—Deletes a VLAN.• present—Creates or modifies a VLAN.• absent—Deletes a VLAN.
id	Specifies the VLAN ID.	Index	Unsigned integer. Value range: 1 to 4094.
description	Configures the description for the VLAN.	N/A	String, case sensitive. Length: 1 to 255 characters.

Resource example

```
# Create VLAN 2, and configure the description as sales-private for VLAN 2.
netdev_vlan{'vlan2':
  ensure => undo_shutdown,
  id => 2,
  description => 'sales-private',
  require => Netdev_device['device'],
}
```

netdev_device

Use this resource to specify the following items:

- Name for a Puppet agent.
- IP address, SSH username, and SSH password used by the agent to connect to a Puppet master.

Attributes

Table 2 Attributes for netdev_device

Attribute name	Description	Value type and restrictions
ensure	Establishes a NETCONF connection to the Puppet master or closes the connection.	Symbol: <ul style="list-style-type: none">• undo_shutdown—Establishes a NETCONF connection to the Puppet master.• shutdown—Closes the NETCONF connection between the Puppet agent and the Puppet master.• present—Establishes a NETCONF connection to the Puppet master.• absent—Closes the NETCONF connection between the Puppet agent and the Puppet master.
hostname	Specifies the device name.	String, case sensitive. Length: 1 to 64 characters.
ipaddr	Specifies an IP address.	String, in dotted decimal notation.
username	Specifies the username for SSH login.	String, case sensitive. Length: 1 to 55 characters.
password	Specifies the password for SSH login.	String, case sensitive. Length and form requirements in non-FIPS mode: <ul style="list-style-type: none">• 1 to 63 characters when in plaintext form.• 1 to 110 characters when in hashed form.• 1 to 117 characters when in encrypted form.

Resource example

```
# Configure the device name as PuppetAgent. Specify the IP address, SSH username, and SSH password for the agent to connect to the Puppet master as 1.1.1.1, user, and 123456, respectively.  
netdev_device{'device':  
  ensure => undo_shutdown,  
  username => 'user',  
  password => '123456',  
  ipaddr => '1.1.1.1',  
  hostname => 'PuppetAgent'  
}
```

netdev_interface

Use this resource to configure attributes for an interface.

Attributes

Table 3 Attributes for netdev_interface

Attribute name	Description	Attribute type	Value type and restrictions
ifindex	Specifies an interface by its index.	Index	Unsigned integer.
ensure	Configures the attributes of the interface.	N/A	Symbol: <ul style="list-style-type: none"> • undo_shutdown • present.
description	Configures the description for the interface.	N/A	String, case sensitive. Length: 1 to 255 characters.
admin	Specifies the management state for the interface.	N/A	Symbol: <ul style="list-style-type: none"> • up—Brings up the interface. • down—Shuts down the interface.
speed	Specifies the interface rate.	N/A	Symbol: <ul style="list-style-type: none"> • auto—Autonegotiation. • 10m—10 Mbps. • 100m—100 Mbps. • 1g—1 Gbps. • 10g—10 Gbps. • 40g—40 Gbps. • 100g—100 Gbps.
duplex	Sets the duplex mode.	N/A	Symbol: <ul style="list-style-type: none"> • full—Full-duplex mode. • half—Half-duplex mode. • auto—Autonegotiation. This attribute applies only to Ethernet interfaces.
linktype	Sets the link type for the interface.	N/A	Symbol: <ul style="list-style-type: none"> • access—Sets the link type of the interface to Access. • trunk—Sets the link type of the interface to Trunk. • hybrid—Sets the link type of the interface to Hybrid. This attribute applies only to Layer 2 Ethernet interfaces.
portlayer	Sets the operation mode for the interface.	N/A	Symbol: <ul style="list-style-type: none"> • bridge—Layer 2 mode. • route—Layer 3 mode.
mtu	Sets the MTU permitted by the interface.	N/A	Unsigned integer in bytes. The value range depends on the interface type. This attribute applies only to Layer 3 Ethernet interface.

Resource example

Configure the following attributes for Ethernet interface 2:

- **Interface description**—puppet interface 2.
- **Management state**—Up.
- **Interface rate**—Autonegotiation.
- **Duplex mode**—Autonegotiation.
- **Link type**—Hybrid.
- **Operation mode**—Layer 3.
- **MTU**—1500 bytes.

```
netdev_interface{'ifindex2':  
  ifindex => 2,  
  ensure => undo_shutdown,  
  description => 'puppet interface 2',  
  admin => up,  
  speed => auto,  
  duplex => auto,  
  linktype => hybrid,  
  portlayer => route,  
  mtu => 1500,  
  require => Netdev _device['device'],  
}
```

netdev_l2_interface

Use this resource to configure the VLAN attributes for a Layer 2 Ethernet interface.

Attributes

Table 4 Attributes for netdev_l2_interface

Attribute name	Description	Attribute type	Value type and restrictions
ifindex	Specifies a Layer 2 Ethernet interface by its index.	Index	Unsigned integer.
ensure	Configures the attributes of the Layer 2 Ethernet interface.	N/A	Symbol: <ul style="list-style-type: none">• undo_shutdown• present
pvid	Specifies the PVID for the interface.	N/A	Unsigned integer. Value range: 1 to 4094.
permit_vlan_list	Specifies the VLANs permitted by the interface.	N/A	String, a comma separated list of VLAN IDs or VLAN ID ranges, for example, 1,2,3,5-8,10-20. Value range for each VLAN ID: 1 to 4094. The string cannot end with a comma (.), hyphen (-), or space.

Attribute name	Description	Attribute type	Value type and restrictions
untagged_vlan_list	Specifies the VLANs from which the interface sends packets after removing VLAN tags.	N/A	String, a comma separated list of VLAN IDs or VLAN ID ranges, for example, 1,2,3,5-8,10-20. Value range for each VLAN ID: 1 to 4094. The string cannot end with a comma (,), hyphen (-), or space. A VLAN cannot be on the untagged list and the tagged list at the same time.
tagged_vlan_list	Specifies the VLANs from which the interface sends packets without removing VLAN tags.	N/A	String, a comma separated list of VLAN IDs or VLAN ID ranges, for example, 1,2,3,5-8,10-20. Value range for each VLAN ID: 1 to 4094. The string cannot end with a comma (,), hyphen (-), or space. A VLAN cannot be on the untagged list and the tagged list at the same time.

Resource example

Specify the PVID as 2 for interface 3, and configure the interface to permit packets from VLANs 1 through 6. Configure the interface to forward packets from VLANs 1 through 3 after removing VLAN tags and forward packets from VLANs 4 through 6 without removing VLAN tags.

```
netdev_l2_interface{'ifindex3':
  ifindex => 3,
  ensure => undo_shutdown,
  pvid => 2,
  permit_vlan_list => '1-6',
  untagged_vlan_list => '1-3',
  tagged_vlan_list => '4,6'
  require => Netdev_device['device'],
}
```

netdev_lagg

Use this resource to create, modify, or delete an aggregation group.

Attributes

Table 5 Attributes for netdev_lagg

Attribute name	Description	Attribute type	Value type and restrictions
group_id	Specifies an aggregation group ID.	Index	Unsigned integer. Value range: <ul style="list-style-type: none"> 1 to 1024 for a Layer 2 aggregation group.

Attribute name	Description	Attribute type	Value type and restrictions
			<ul style="list-style-type: none"> 16385 to 17408 for a Layer 3 aggregation group.
ensure	Creates, modifies, or deletes the aggregation group.	N/A	Symbol: <ul style="list-style-type: none"> present—Creates or modifies the aggregation group. absent—Deletes the aggregation group.
linkmode	Specifies the aggregation mode.	N/A	Symbol: <ul style="list-style-type: none"> static—Static. dynamic—Dynamic.
addports	Specifies the indexes of the interfaces that you want to add to the aggregation group.	N/A	String, a comma separated list of interface indexes or interface index ranges, for example, 1,2,3,5-8,10-20. The string cannot end with a comma (,), hyphen (-), or space. An interface index cannot be on the list of adding interfaces and the list of removing interfaces at the same time.
deleteports	Specifies the indexes of the interfaces that you want to remove from the aggregation group.	N/A	String, a comma separated list of interface indexes or interface index ranges, for example, 1,2,3,5-8,10-20. The string cannot end with a comma (,), hyphen (-), or space. An interface index cannot be on the list of adding interfaces and the list of removing interfaces at the same time.

Resource example

```
# Add interfaces 1 and 2 to aggregation group 2, and remove interfaces 3 and 4 from the group.
netdev_lagg{ 'lagg2':
  group_id => 2,
  ensure => present,
  addports => '1,2',
  deleteports => '3,4',
  require => Netdev _device['device'],
}
```

netdev_l2vpn

Use this resource to enable or disable L2VPN.

Attributes

Table 6 Attributes for netdev_l2vpn

Attribute name	Description	Value type and restrictions
ensure	Enables or disables L2VPN.	Symbol: <ul style="list-style-type: none">• enable—Enables L2VPN.• disable—Disables L2VPN.

Resource example

```
# Enable L2VPN.
netdev_l2vpn{'l2vpn':
  ensure => enable,
  require => Netdev_device['device'],
}
```

netdev_vsi

Use this resource to create, modify, or delete a Virtual Switch Instance (VSI).

Attributes

Table 7 Attributes for netdev_vsi

Attribute name	Description	Attribute type	Value type and restrictions
vsiname	Specifies a VSI name.	Index	String, case sensitive. Length: 1 to 31 characters.
ensure	Creates, modifies, or deletes the VSI.	N/A	Symbol: <ul style="list-style-type: none">• present—Creates or modifies the VSI.• absent—Deletes the VSI.
description	Configures the description for the VSI.	N/A	String, case sensitive. Length: 1 to 80 characters.

Resource example

```
# Create the VSI vsia.
netdev_vsi{'vsia':
  ensure => present,
  vsiname => 'vsia',
  require => Netdev_device['device'],
}
```

netdev_vte

Use this resource to create or delete a tunnel.

Attributes

Table 8 Attributes for netdev_vte

Attribute name	Description	Attribute type	Value type and restrictions
id	Specifies a tunnel ID.	Index	Unsigned integer.
ensure	Creates or deletes the tunnel.	N/A	Symbol: <ul style="list-style-type: none"> present—Creates the tunnel. absent—Deletes the tunnel.
mode	Sets the tunnel mode.	N/A	Unsigned integer: <ul style="list-style-type: none"> 1—IPv4 GRE tunnel mode. 2—IPv6 GRE tunnel mode. 3—IPv4 over IPv4 tunnel mode. 4—Manual IPv6 over IPv4 tunnel mode. 5—Automatic IPv6 over IPv4 tunnel mode. 6—IPv6 over IPv4 6to4 tunnel mode. 7—IPv6 over IPv4 ISATAP tunnel mode. 8—IPv6 over IPv6 or IPv4 tunnel mode. 9—UDP-encapsulated IPv4 DVPN tunnel mode. 10—GRE-encapsulated IPv4 DVPN tunnel mode. 11—UDP-encapsulated IPv6 DVPN tunnel mode. 12—GRE-encapsulated IPv6 DVPN tunnel mode. 13—CRLSP tunnel mode. 14—IPv4 multicast GRE tunnel mode. 15—IPv6 multicast GRE tunnel mode. 16—IPv4 IPsec tunnel mode. 17—IPv6 IPsec tunnel mode. 18—IPv4 EVI tunnel mode. 19—IPv6 EVI tunnel mode. 20—IPv4 EVI link mode. 21—IPv6 EVI link mode. 22—DS-Lite AFTR tunnel mode. 23—CRLSP automatic tunnel mode. 24—UDP-encapsulated IPv4 VXLAN tunnel mode. 25—UDP-encapsulated IPv6 VXLAN tunnel mode. You must specify the tunnel mode when creating a tunnel. After the tunnel is created, you cannot change the tunnel mode.

Resource example

```
# Create IPv6-to-IPv4 automatic tunnel 2.
netdev_vte{'vte2':
  ensure => present,
  id => 2,
  mode => 5,
  require => Netdev_device['device'],
}
```

netdev_vxlan

Use this resource to create, modify, or delete a VXLAN.

Attributes

Table 9 Attributes for netdev_vxlan

Attribute name	Description	Attribute type	Value type and restrictions
vxlan_id	Specifies a VXLAN ID.	Index	Unsigned integer. Value range: 1 to 16777215.
ensure	Creates or deletes the VXLAN.	N/A	Symbol: <ul style="list-style-type: none"> present—Creates or modifies the VXLAN. absent—Deletes the VXLAN.
vsiname	Specifies the VSI name.	N/A	String, case sensitive. Length: 1 to 31 characters. You must specify the VSI name when creating a VSI. After the VSI is created, you cannot change the name.
add_tunnels	Specifies the tunnel interfaces to be associated with the VXLAN.	N/A	String, a comma separated list of tunnel interface IDs or tunnel interface ID ranges, for example, 1,2,3,5-8,10-20. The string cannot end with a comma (,), hyphen (-), or space. A tunnel interface ID cannot be on the list of adding interfaces and the list of removing interfaces at the same time.
delete_tunnels	Removes the association between the specified tunnel interfaces and the VXLAN.	N/A	String, a comma separated list of tunnel interface IDs or tunnel interface ID ranges, for example, 1,2,3,5-8,10-20. The string cannot end with a comma (,), hyphen (-), or space. A tunnel interface ID cannot be on the list of adding interfaces and the list of removing interfaces at the same time.

Resource example

Create VXLAN 10, configure the VSI name as **vsia**, and associate tunnel interfaces 7 and 8 with VXLAN 10.

```
netdev_vxlan{'vxlan10':  
  ensure => present,  
  vxlan_id => 10,  
  vsiname => 'vsia',  
  add_tunnels => '7-8',  
  require=>Netdev_device['device'],  
}
```