

# Contents

Introduction .....	1
Prerequisites .....	1
Example: Configuring Puppet .....	1
Network configuration .....	1
Applicable hardware and software versions .....	1
Procedures .....	3
Verifying the configuration .....	5
Configuration files .....	6

# Introduction

This document provides Puppet configuration examples.

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.

## Prerequisites

The configuration examples in this document were created and verified in a lab environment, and all the devices were started with the factory default configuration. When you are working on a live network, make sure you understand the potential impact of every command on your network.

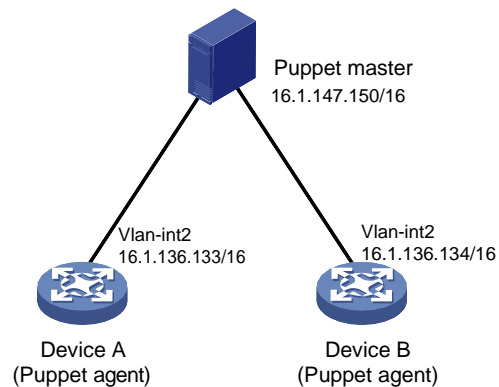
This document assumes that you have basic knowledge of Puppet.

## Example: Configuring Puppet

### Network configuration

As shown in [Figure 1](#), Puppet agents Device A and Device B are connected to the Puppet master. Use Puppet to create VLAN 100 on each Puppet agent.

**Figure 1 Network diagram**



## Applicable hardware and software versions

The following matrix shows the hardware and software versions to which this configuration example is applicable:

Hardware	Software version
SC 3570 switch series	Not supported
SC 5525 switch series	Release 63xx, Release 65xx, Release 6615Pxx, Release 6628Pxx
SC 5520 switch series	Release 63xx, Release 65xx, Release 6615Pxx, Release 6628Pxx
SC 3170 switch series	Not supported
SC 3130 switch series	Not supported

# Procedures

## Configuring the Puppet master

1. Assign an IP address to the Puppet master. (Details not shown.)
2. Install Puppet on the Puppet master.  

```
$ sudo apt-get install puppetmaster
```
3. Verify that Puppet configuration file **puppet.conf** is created in directory **/etc/puppet**.  
 If the file does not exist, execute the following command to create the file:  

```
$ sudo puppet master -genconfig > puppet.conf
```
4. Add files provided by INTELBRAS:  
 # Copy the type and provider library files to directory **/etc/puppet/modules/custom/lib/puppet**.  
 If conflicts exist, overwrite the existing files. (Details not shown.)

# Create directory **manifests/nodes** in directory **/etc/puppet/**.

```
$ sudo mkdir -p /etc/puppet/manifests/nodes
```

# Copy INTELBRAS file **puppet.master.com.pp** to directory **/etc/puppet/manifests/nodes/**.  
(Details not shown.)

5. Configure file **site.pp**:

# Go to directory **/etc/puppet/manifests/** and create file **site.pp** in the directory.

```
$ cd ..
```

```
$ sudo touch site.pp
```

# Edit file **site.pp**.

```
node '16.1.136.133'{
  netdev_device{'device':
    ensure => undo_shutdown,
    username => 'test',
    password => 'test',
    ipaddr => '16.1.136.133',
  }
  include custom
}

node '16.1.136.134'{
  netdev_device{'device':
    ensure => undo_shutdown,
    username => 'test',
    password => 'test',
    ipaddr => '16.1.136.134',
  }
  include custom
}
```

6. Configure file **init.pp**:

# Create directory **modules/custom/manifests** in directory **/etc/puppet/** to store configuration manifests.

```
$ sudo mkdir -p /etc/puppet/modules/custom/manifests
```

# Create configuration manifest **init.pp** in directory **/etc/puppet/modules/custom/manifests**.

```
$ sudo touch init.pp
```

# Edit file **init.pp**.

```
class custom{
  netdev_vlan{'vlan100':
    ensure => undo_shutdown,
    id => 100,
    require => Netdev_device['device'],
  }
}
```

## Configuring Puppet agent Device A

1. Assign an IP address to Device A. (Details not shown.)

```
<DeviceA> system-view
[DeviceA] interface vlan-interface 2
[DeviceA-Vlan-interface2] ip address 16.1.136.133 255.255.0.0
[DeviceA-Vlan-interface2] quit
```

2. Configure the device as the NETCONF over SSH server:

# Generate RSA key pairs. Leave the key pair to use the default key pair name.

```
[DeviceA] public-key local create rsa
```

# Enable NETCONF over SSH.

```
[DeviceA] netconf ssh server enable
```

# Enable scheme authentication for NETCONF over SSH users.

```
[DeviceA] line vty 0 63
```

```
[DeviceA-line-vty0-63] authentication-mode scheme
```

```
[DeviceA-line-vty0-63] quit
```

# Create device management user **test**. Set the password to **test**, and assign the SSH service and network-admin user role to the user.

```
[DeviceA] local-user test class manage
```

```
[DeviceA-luser-manage-test] password simple test
```

```
[DeviceA-luser-manage-test] service-type ssh
```

```
[DeviceA-luser-manage-test] authorization-attribute user-role network-admin
```

```
[DeviceA-luser-manage-test] quit
```

3. Configure the device as an NTP client to synchronize its system time to the system time on the Puppet master:

# Enable the NTP service

```
[DeviceA] ntp-service enable
```

# Configure the device to use NTP to obtain the UTC time.

```
[DeviceA] clock protocol ntp
```

# Configure the device to operate in NTP broadcast client mode and use VLAN-interface 2 to receive NTP broadcast packets.

```
[DeviceA] interface vlan-interface 2
```

```
[DeviceA-Vlan-interface2] ntp-service broadcast-client
```

```
[DeviceA-Vlan-interface2] quit
```

4. Start Puppet on the device.

```
[DeviceA] third-part-process start name puppet arg agent --certname=16.1.136.133  
--server=16.1.147.150
```

The device will act as a Puppet client to request a certificate from the Puppet master.

## Configuring Puppet agent Device B

The steps are similar to the steps for Device A

## Using the Puppet master to issue certificates to the Puppet clients

# Use the **puppet cert list** command to display devices that require a certificate. (Details not shown.)

# Sign a certificate for Device A.

```
$ sudo puppet cert sign 16.1.136.133
```

After Device A obtains a certificate, it obtains a configuration manifest from the Puppet master and run the manifest.

# Sign a certificate for Device B. (Details not shown.)

## Verifying the configuration

# Display device configuration information. VLAN 100 is created. (Details not shown.)

# Configuration files

- Puppet master:

The type and provider library files and the **puppet.master.com.pp** file are provided by INTELBRAS.

- Device A

```
#
    clock protocol ntp
#
interface Vlan-interface2
    ip address 16.1.136.133 255.255.0.0
    ntp-service broadcast-client
#
    ntp-service enable
#
line vty 0 63
    authentication-mode scheme
    user-role network-admin
    user-role network-operator
    idle-timeout 0 0
#
local-user test class manage
    password hash $h$6$x0kIJnaHZkFFa3Ga$H4yMQnG96xjHiTID+6UPyJrTLXru6RJaGqrCKpxmo2OKUVWujeoTcDEovLt6LzKIUYn7J3i5Tq2rOQPdj2Nrww==
    service-type ssh
    authorization-attribute user-role network-admin
    authorization-attribute user-role network-operator
#
netconf ssh server enable
#
```