Delta Proxy Minions

Release 3003

VMware, Inc.

Apr 12, 2021

CONTENTS

CHAPTER

ONE

INTRODUCTION

Welcome to the delta proxy minion installation guide. This installation guide explains the process for installing and using the Salt delta proxy minion 3003.

This guide is intended for network engineers with the general knowledge and experience required in the field. This guide is also intended for users that have ideally already tested and used standard Salt proxy minions in their environment before deciding to move to a delta proxy minion environment. See Salt proxy minions for more information.

Note: If you have not used standard Salt proxy minions before, consider testing and deploying standard Salt proxy minions in your environment first. Alternatively, you can request consulting services for assistance.

1.1 Proxy minions vs. delta proxy minions

Salt can target network devices through Salt proxy minions, which can be used by both SaltStack Config users and Open Salt users. Proxy minions allow you to control network devices that, for whatever reason, cannot run the standard salt-minion service. Examples include:

- Network gear that has an API but runs a proprietary operating system
- Devices with limited CPU or memory
- Devices that could run a minion but will not for security reasons

A proxy minion acts as an intermediary between the Salt master and the device it represents. The proxy minion runs on the master and then translates commands from the master to the device as needed.

By acting as an intermediary for the actual minion, proxy minions eliminate the need to establish a constant connection from a master to a minion. Proxy minions generally only open a connection to the actual minion when necessary.

Proxy minions also reduce the amount of CPU or memory the minion must spend checking for commands from the master. Proxy minions use the master's CPU or memory to check for commands. The actual minion only needs to use CPU or memory to run commands when needed.

Note: For more information about Salt proxy minions, see:

- Salt proxy minions
- Salt proxy modules

1.1.1 When delta proxy minions are needed

Normally, you would create a separate instance of proxy minion for each device that needs to be managed. However, this doesn't always scale well if you have thousands of devices. Running several thousand proxy minions can require a lot of memory and CPU.

A delta proxy minion can solve this problem: it makes it possible to run one minion that acts as the intermediary between the master and the many network devices it can represent. In this scenario, one device (the delta proxy minion on the master) runs several proxies. This configuration boosts performance and improves the overall scalability of the network.

1.2 Key terms

The following lists some important terminology that is used throughout this guide:

Term	Definition
master	The master is a central node running the salt-master service. The master issues commands to minions.
minion	Minions are nodes running the salt-minion service. Minions listen to commands from a master and perform the requested tasks, then return data back to the master as needed.
proxy minion	A master that is running the proxy-minion service. The proxy minion acts as an intermediary between the master and the device it represents. The proxy minion runs on the master and then translates commands from the master to the device. A separate instance of proxy minion is needed for each device that is managed.
delta proxy minion	A master that is running the deltaproxy-minion service. The delta proxy minion acts as the intermediary between the master and the many network devices it can represent. Only one instance of the delta proxy service is needed to run several proxies.
control proxy	The control proxy runs on the master. It manages a list of devices and issues commands to the network devices it represents. The master needs at least one control proxy, but it is possible to have more than one control proxy, each managing a different set of devices.
managed device	A device (such as Netmiko) that is managed by proxy minions or by a control proxy minion. The proxy minion or control proxy only creates a connection to the actual minion it needs to issue a command.
pillar file	Pillars are structures of data (files) defined on the master and passed through to one or more minions when the minion needs access to the pillar file. Pillars allow confidential, targeted data to be securely sent only to the relevant minion. Because all configurations for delta proxy minions are done on the master (not on the minions), you use pillar files to configure the deltaproxy-minion service.
top file	The top file is a pillar file that maps which states should be applied to different minions in certain environments.

1.3 SaltStack Config support

If at any time you encounter difficulties with the installation that are not addressed by this guide, search the Help documentation or Contact Support.

Customer support is available to SaltStack Config customers with an active license and maintenance agreement. For more information about contacting support, refer to the Support Policy.

CHAPTER

TWO

PRE-INSTALLATION

2.1 Supported hardware and firmware versions

Netmiko is the only proxy minion type that is officially supported for delta proxy minion 3003. Other proxy types may work but have not yet been fully tested or validated.

2.2 Prerequisites

Before installing the delta proxy minion, ensure that:

- Your network device and firmware are supported.
- The Salt master that is acting as the control proxy minion has access to the devices it is managing.
- You have installed, configured, and tested standard Salt proxy minions in your environment before introducing delta proxy minions into your environment.

Note: If you have not yet worked with Salt proxy minions but would like to use delta proxy minions in your environment, consider requesting assistance from consulting services.

2.3 Install or upgrade Salt

Ensure your masters are running Salt 3003. For instructions on installing or upgrading Salt, see repo.saltstack.com. For RedHat systems, see Install or Upgrade Salt.

2.4 Download and verify files

The delta proxy minion package is included with the SaltStack Config installer. If you are a SaltStack Config customer, you can download the installer from the SaltStack Enterprise Installation Guide.

CHAPTER

THREE

INSTALLATION

Before you begin the delta proxy minion installation process, ensure you have read and completed the *Pre-installation* steps.

3.1 Overview of the installation process

Similar to proxy minions, all the delta proxy minion configurations are done on the Salt master rather than on the minions that will be managed. The installation process has the following phases:

1. *Configure the master to use delta proxy* - Create a configuration file on the master that defines its proxy settings.
2. *Create a pillar file for each managed device* - Create a pillar file for each device that will be managed by the delta proxy minion and reference these minions in the top file.
3. *Create a control proxy configuration file* - Create a control proxy file that lists the devices that it will manage. Then, reference this file in the top file.
4. *Start the delta proxy minion* - Start the deltaproxy-minion service and validate that it has been set up correctly.

3.2 Configure the master to use delta proxy

In this step, you'll create a configuration file on the master that defines its proxy settings. This is a general configuration file that tells the master how to handle all proxy minions.

To create this configuration:

1. On the master, navigate to the `/etc/salt` directory. In this directory, create a new proxy configuration file. Give this file a descriptive name, such as `delta_proxy_configuration`.
2. Open the file in your preferred editor and add the following configuration information:

```
# Use delta proxy metaproxy
metaproxy: deltaproxy

# Disable the FQDNS grain
enable_fqdns_grains: False

# Enabled multprocessing
multiprocessing: True

# Tell netmiko proxy minions to not connect to the device when proxy minions start
skip_connect_on_init: True
```

Note: See the following section about *Delta proxy configuration options* for a more detailed description of these configuration options.

3. Save the file.
4. Open the master's top file: `/srv/pillar/top.sls`.
5. In the top file, add a line under the master's ID that references the name of the new delta proxy configuration file that you just created. For example:

```
base:
  salt_master_ID:
    - delta_proxy_configuration
```

6. Save the file.

Your master is now configured to use delta proxy. Next, you need to *Create a pillar file for each managed device*.

3.2.1 Delta proxy configuration options

The following table describes the configuration options used in the delta proxy configuration file:

Field	Description
metaproxy	Set this configuration option to `deltaproxy`. If this option is set to `proxy` or if this line is not included in the file, the master will use the standard proxy service instead of the delta proxy service.
enable_fqdns_grains	If your router does not have the ability to use Reverse DNS lookup to obtain the Fully Qualified Domain Name (fqdn) for an IP address, you'll need to change the `enable_fqdns_grains` setting in the pillar configuration file to `False` instead.
multiprocessing	Multi-processing is the ability to run more than one task or process at the same time. A delta proxy minion has the ability to run with multi-processing turned off. If you plan to run with multi-processing enabled, you should also enable the `skip_connect_on_init` setting to `True`.
skip_connect_on_init	This setting tells the control proxy whether or not it should make a connection to the managed device when it starts. When set to `True`, the delta proxy minion will only connect when it needs to issue commands to the managed devices.

3.3 Create a pillar file for each managed device

Each device that needs to be managed by delta proxy needs a separate pillar file on the master. To create this file:

1. Navigate to the `/srv/pillar` directory.
2. In this directory create a new pillar file for a minion. For example, `my_managed_device_pillar_file_01.sls`.
3. Open the new file in your preferred editor and add the necessary configuration information for that minion and your environment. The following is an example pillar file for a Netmiko device:

```
proxy:
  proxytype: netmiko
  device_type: arista_eos
```

(continues on next page)

(continued from previous page)

```
  host: 192.0.2.1
  username: myusername
  password: mypassword
  always_alive: True
```

Note: The available configuration options vary depending on the proxy type (in other words, the type of device it is). To read a detailed explanation of the configuration options, refer to the proxy module documentation for the type of device you need to manage. See:

- Salt proxy modules
- Netmiko Salt proxy module

---

4. Save the file.
5. In an editor, open the top file: `/srv/pillar/top.sls`.
6. Add a section to the top file that indicates the minion ID of the device that will be managed. Then, list the name of the pillar file you created in the previous steps. For example:

```
my_managed_device_minion_ID:
  - my_managed_device_pillar_file_01
```

7. Repeat the previous steps for each minion that needs to be managed.

You've now created the pillar file for the minions that will be managed by the delta proxy minion and you have referenced these files in the top file. Proceed to the next section.

3.4 Create a control proxy configuration file

On the master, you'll need to create or edit a control proxy file for each control proxy. The control proxy manages several devices and issues commands to the network devices it represents. The master needs at least one control proxy, but it is possible to have more than one control proxy, each managing a different set of devices.

To configure a control proxy, you'll create a file that lists the minion IDs of the minions that it will manage. Then you will reference this control proxy configuration file in the top file.

To create a control proxy configuration file:

1. On the master, navigate to the `/srv/pillar` directory. In this directory, create a new proxy configuration file. Give this file a descriptive name, such as `control_proxy_01_configuration`.
2. Open the file in your preferred editor and add a list of the minion IDs for each device that needs to be managed. For example:

```
proxy:
  proxytype: deltaproxy
  ids:
    - my_managed_device_01
    - my_managed_device_02
    - my_managed_device_03
```

3. Save the file.
4. In an editor, open the top file: `/srv/pillar/top.sls`.

5. Add a section to the top file that indicates references the delta proxy control proxy. For example:

```
base:
  salt_master_ID:
    - delta_proxy_configuration
  delta_proxy_control:
    - control_proxy_01_configuration
```

6. Repeat the previous steps for each control proxy if needed.

Now that you have created the necessary configurations, proceed to the next section.

3.5 Start the delta proxy minion

After you've successfully configured the delta proxy minion, you need to start the proxy minion service for each managed device and validate that it is working correctly.

Note: This step explains the process for starting a single instance of a delta proxy minion. Because starting each minion individually can potentially be very time-consuming, most organizations use a script to start their delta proxy minions since there are typically many devices being managed. Consider implementing a similar script for your environment to save time in deployment.

To start a single instance of a delta proxy minion and test that it is configured correctly:

1. In the terminal for the master, run the following command, replacing the placeholder text with the actual minion ID:

```
sudo salt-proxy --proxyid=my_managed_device_minion_ID
```

2. To test the delta proxy minion, run the following `test.version` command on the master and target a specific minion. For example:

```
salt my_managed_device_minion_ID test.version
```

   This command returns an output similar to the following:

```
local:
    3003
```

After you've successfully started the delta proxy minions and verified that they are working correctly, you can now use these minions the same as standard proxy minions.

CHAPTER

FOUR

ADDITIONAL RESOURCES

This reference section includes additional resources for delta proxy minions.

For reference, see:

- Salt proxy minions
- Salt proxy modules
- Netmiko Salt proxy module