



Monitoring Linux Systems

Linux Base Pack version 107

Table of Contents

Introduction	3
What is SNMP?	4
What is Net-SNMP?	5
Why Should I Use Net-SNMP?	5
Basic SNMP Terminology	5
What is SSH?	7
Monitoring Linux with SSH	8
What Does the Linux Base Pack PowerPack Monitor?	9
Installing the Linux Base Pack PowerPack	9
Upgrading the PowerPack and Removing Dynamic Applications	10
Prerequisites for Monitoring Linux Devices with SSH	11
Linux Distributions Supported by the Linux Base Pack PowerPack	12
Configuring Linux Devices	14
Creating an SSH/Key Credential	14
Creating an SSH/Key Credential in the Classic SL1 User Interface	16
Creating a PowerShell Credential in the Classic SL1 User Interface	18
Configuring the Linux Device Template	20
Configuring the Linux: IC Port Performance Dynamic Application	22
Discovering Linux Devices	23
Discovering Linux Devices in the SL1 Classic User Interface	25
Configuring Dynamic Applications for Monitoring	26
Process Monitoring with the Linux Base Pack	26
Configuring Collection Frequency for Linux IC Dynamic Applications	26
Unhiding Linux File Systems	27
Configuring Linux File System Thresholds	28
Relationships Between Component Devices	29
Enabling Concurrent SSH Collection	29
Installing and Configuring Net-SNMP for Linux	31
Installing Net-SNMP on Linux Devices	32
Step 1: Verifying and Installing Net-SNMP with RPM Packages	33
Step 2: Stopping snmpd	33
Step 3: Creating the snmpd.conf file	34
Example snmpd.conf file for SNMPv2:	35
Adding Read/Write Access to the snmpd.conf File for SNMPv2	38
Example snmpd.conf file for SNMPv3:	38
Step 4: Starting snmpd and Testing connectivity to Net-SNMP	40
Testing the Example snmpd.conf file for SNMPv2	40
Testing the Example snmpd.conf file for SNMPv3	42
Creating SNMP Credentials for Linux	44
Creating SNMPv2 Credentials	45
Creating SNMPv3 Credentials	47
Configuring Syslog for Linux	50
What is Syslog?	50
Configuring Syslog for Linux	51
Collection Objects	A
Collection Objects in Linux Dynamic Applications	B

Chapter

1

Introduction

Overview

This manual describes how to configure SNMP and Syslog for Linux systems and how to monitor Linux systems with SL1 using the Dynamic Applications in the *Linux Base Pack PowerPack*.

SL1 supports three protocols to monitor Linux devices:



- SNMP
- SSH
- Syslogs

SNMP and Linux are used to proactively poll the device periodically to collect information, while Syslog asynchronously receives logs from the device. Syslog can be used with SNMP or SSH, but you cannot use both SNMP and SSH together.

ScienceLogic recommends using SSH along with Syslog as that provides the most comprehensive and secure monitoring.

The following sections provide an overview of SNMP, Net-SNMP, Secure Shell (SSH), and the *Linux Base Pack PowerPack*:

Use the following menu options to navigate the SL1 user interface:

- To view a pop-out list of menu options, click the menu icon (.
- To view a page containing all of the menu options, click the Advanced menu icon ().

This chapter covers the following topics:

What is SNMP?	4
What is Net-SNMP?	5

Basic SNMP Terminology	5
What is SSH?	7

NOTE: ScienceLogic provides this documentation for the convenience of ScienceLogic customers. Some of the configuration information contained herein pertains to third-party vendor software that is subject to change without notice to ScienceLogic. ScienceLogic makes every attempt to maintain accurate technical information and cannot be held responsible for defects or changes in third-party vendor software. There is no written or implied guarantee that information contained herein will work for all third-party variants. See the End User License Agreement (EULA) for more information.

What is SNMP?

Simple Network Management Protocol (SNMP) is a set of standard protocols for managing diverse computer hardware and software within a TCP/IP network. SNMP is the most common protocol used by network monitoring and management applications to exchange information between devices. SL1 uses this protocol and other protocols to collect availability, performance, and configuration information.

SNMP uses a server-client structure.

- Clients are called **agents**. Devices and software that run SNMP are agents. For the purposes of this document, Net-SNMP is the agent.
- The server is called the **management system**. SL1 is the management system.

Typically, **agents**:

- Implement the SNMP protocol on the device.
- Store data points as defined by the Management Information Base (MIB) file.
- Can asynchronously signal an event to the manager.

Typically, a **management system**:

- Uses the SNMP Protocol.
- Queries agents.
- Receives responses (data points) from agents.
- Acknowledges asynchronous events from agents.

Most enterprise-level network hardware is configured for SNMP and can be SNMP-enabled. Many enterprise software applications are also SNMP-compliant. When SNMP is running on a device, it uses a standard format to collect and store data about the device and/or software. For example, SNMP might collect information on each network interface and the traffic on each interface. SL1 can then query the device to retrieve the stored data.

What is Net-SNMP?

Net-SNMP is a suite of applications used to implement SNMP. Net-SNMP is an agent. Standard Net-SNMP includes the SNMP daemon and a suite of client utilities. Net-SNMP can be run on any supported operating system, and SL1 will then be able to communicate with and collect data from the device.

Why Should I Use Net-SNMP?

- Net-SNMP is an open-source application. It is free to use and distribute.
- Because Net-SNMP is widely used, there are many user groups and support forums for the product.

NOTE: Although ScienceLogic does not directly support the Net-SNMP agent, this document will get you started on the installation and configuration tasks for Net-SNMP. For detailed documentation on Net-SNMP, see the [Net-SNMP website](#).

- Net-SNMP includes source and pre-compiled objects for all major flavors of UNIX and Linux as well as a number of other operating systems.
- Net-SNMP is an **extensible** agent. Generally, SNMP agents can retrieve only data that has been defined in a MIB file. In most cases, a hardware or software manufacturer creates the MIB file and then ships the MIB file with the product. Net-SNMP allows users to add values to the MIB file and retrieve values from scripts, programs, and files.
- Net-SNMP is a natural fit with SL1's Dynamic Applications. Using Net-SNMP and dynamic applications, users can create reports, coupled graphs, and events based on the data points that are most useful to them.

Basic SNMP Terminology

This section defines some basic SNMP terminology. You should be familiar with the following terminology before installing and configuring Net-SNMP:

- **SNMP (Simple Network Management Protocol)**. A set of standard protocols for managing diverse computer hardware and software within a TCP/IP network. SNMP is the most common network protocol used by network monitoring and management applications to exchange management information between devices. SL1 uses this protocol and other protocols to collect availability, performance, and configuration information.

SNMP uses a server-client structure. Clients are called agents. Devices and software that run SNMP are agents. The server is called the management system. SL1 is the management system.

Most enterprise-level network hardware is configured for SNMP and can be SNMP-enabled. Many enterprise software applications are also SNMP-compliant. When SNMP is running on a device, it uses a standard format to collect and store data about the device and/or software. For example, SNMP might collect information on each network interface and the traffic on each interface. SL1 can then query the device to retrieve the stored data.

- **SNMP Tree.** SNMP uses a tree structure. The first few branches of the tree are organizational and do not apply to specific manufacturers and device. The starting point for all device or application info is:
 - 1.3.6.1.4.1.vendor_number
 - For details on SNMP tree structure, see the [SMI Network Management Private Enterprise Codes](#).
 - For an overview of the entire SNMP tree, see <http://www3.rad.com/networks/applications/snmp/main.htm>.
- **MIB (Management Information Base).** A collection of objects that can be monitored by a network management system (in this case, SL1). The objects are organized hierarchically and stored in a MIB file. SNMP requires a standardized format for each MIB file. This standardized format allows SL1 to gather data on any device where SNMP is enabled. A MIB file is usually associated with a manufacturer and a device. Some companies use a single MIB that contains information on all their products; some manufacturers create a separate MIB for each product.
- **OID (Object ID).** OIDs are the numeric IDs that are used in the SNMP tree. OIDs are used to define manufacturers, devices, and the characteristics of devices. OIDs are defined and organized in MIB files.
- In SL1, the **root OID** (sometimes called the vendor number) refers to the unique number assigned to each manufacturer. Each root OID is registered with IANA. For example, the root OID for American Power Conversion (APC) Corporation is 1.3.6.1.4.1.318. APC can then create and organize OIDs under this root OID. For example:
 - 1.3.6.1.4.1.318 is the root OID for American Power Conversion Corporation.
 - 1.3.6.1.4.1.318.1 could mean "all products". APC could then define unique IDs under "all products".
 - 1.3.6.1.4.1.318.1.1 could mean "hardware".
 - 1.3.6.1.4.1.318.1.2 could mean "software".

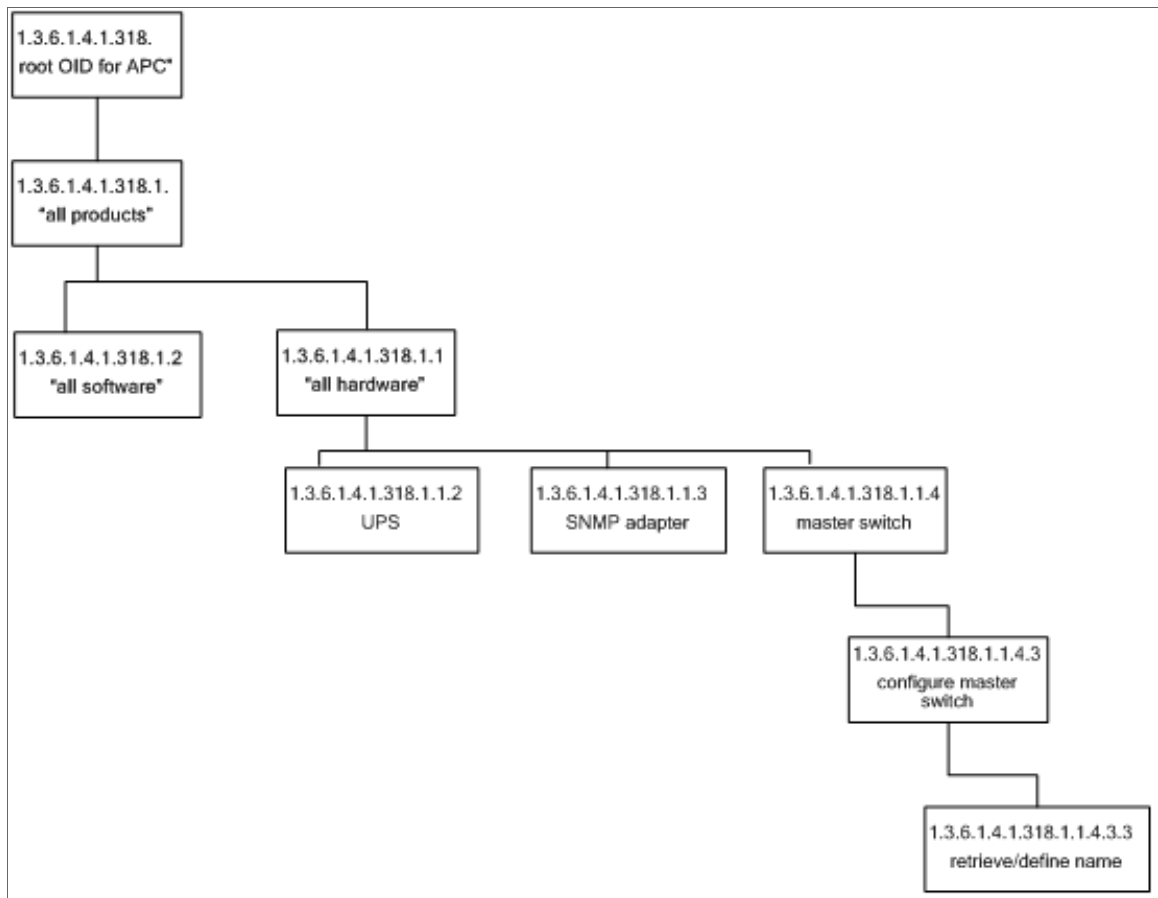
All the OIDs that occur under 1.3.6.1.4.1.318.1.1 would be mapped to types of hardware, for example:

- 1.3.6.1.4.1.318.1.1.2 could mean "UPS".
- 1.3.6.1.4.1.318.1.1.3 could mean "SNMP adapter".
- 1.3.6.1.4.1.318.1.1.4 could mean "master switch".

All the OIDs that occur under each type of hardware (UPS, SNMP adapter, master switch) would be mapped to specific parameters that can be monitored and controlled through SNMP commands. For example:

- 1.3.6.1.4.1.318.1.1.4.3 could mean "configuration settings for master switch".
- 1.3.6.1.4.1.318.1.1.4.3.3 could mean "retrieve or define name for master switch".

The section of the SNMP tree for our example would look like this:



What is SSH?

Secure Shell (SSH) is a network protocol that enables users to securely access a command-line shell on a remote computer or server over an unsecured network. SSH provides strong encryption and authentication capabilities, making it an ideal method for securely administering commands or transferring data between a client and server.

To make SSH even more secure, you can use SSH keys instead of a simple password to log in to a server. SSH keys consist of two long strings of characters, called a public/private key pair, that are much less susceptible than passwords are to brute force attacks. The public key is placed on the server you want to access, while the private key resides on the client. When you use SSH to log in to the server from the client, the key pair is used to authenticate the session.

In SL1, some Dynamic Applications of type "Snippet" use SSH to communicate with a remote device. To use these Dynamic Applications, you must define an SSH credential. This credential specifies the hostname or IP address of the system you want to monitor, the port number used to access that system, and the private key used for authentication.

NOTE: The default TCP port for SSH servers is 22.



Chapter

2

Monitoring Linux with SSH

Overview

Use the following menu options to navigate the SL1 user interface:

- To view a pop-out list of menu options, click the menu icon ().
- To view a page containing all of the menu options, click the Advanced menu icon ().

The following sections describe how to configure and discover Linux devices for monitoring by SL1 using SSH and the *Linux Base Pack PowerPack*:

This chapter covers the following topics:

<i>What Does the Linux Base Pack PowerPack Monitor?</i>	9
<i>Installing the Linux Base Pack PowerPack</i>	9
<i>Prerequisites for Monitoring Linux Devices with SSH</i>	11
<i>Linux Distributions Supported by the Linux Base Pack PowerPack</i>	12
<i>Configuring Linux Devices</i>	14
<i>Creating an SSH/Key Credential</i>	14
<i>Creating a PowerShell Credential in the Classic SL1 User Interface</i>	18
<i>Configuring the Linux Device Template</i>	20
<i>Discovering Linux Devices</i>	23
<i>Configuring Dynamic Applications for Monitoring</i>	26
<i>Relationships Between Component Devices</i>	29
<i>Enabling Concurrent SSH Collection</i>	29

What Does the Linux Base Pack PowerPack Monitor?

To monitor Linux systems with SSH using SL1, you must install the *Linux Base Pack* PowerPack. This PowerPack enables you to discover, model, and collect data about Linux systems.

The *Linux Base Pack* PowerPack includes:

- Dynamic Applications that discover and collect configuration and performance data for Linux systems
- Internal collection Dynamic Applications for Linux systems
- Event Policies and corresponding alerts that are triggered when Linux systems meet certain status criteria
- Device Classes for each type of Linux system monitored
- A Run Book Action and an Automation policy to assign the proper device classes to Linux systems

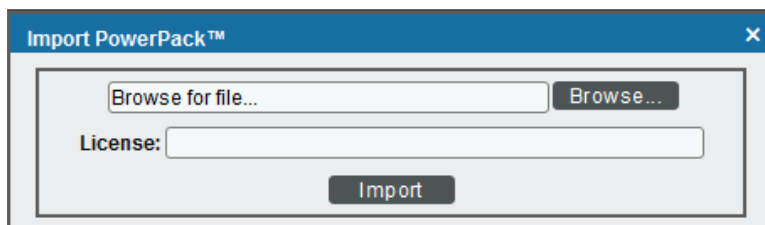
Installing the Linux Base Pack PowerPack

To monitor Linux systems with SSH, you must import and install the latest version of the *Linux Base Pack* PowerPack.

TIP: By default, installing a new version of a PowerPack overwrites all content from a previous version of that PowerPack that has already been installed on the target system. You can use the **Enable Selective PowerPack Field Protection** setting in the **Behavior Settings** page (System > Settings > Behavior) to prevent new PowerPacks from overwriting local changes for some commonly customized fields. (For more information, see the **System Administration** manual.)

To download and install a PowerPack:

1. Download the PowerPack from the ScienceLogic Support Site at <https://support.sciencelogic.com/s/powerpacks>.
2. Go to the **PowerPack Manager** page (System > Manage > PowerPacks).
3. In the **PowerPack Manager** page, click the **[Actions]** button, then select *Import PowerPack*. The **Import PowerPack** dialog box appears:



4. Click the **[Browse]** button and navigate to the PowerPack file.
5. When the **PowerPack Installer** modal appears, click the **[Install]** button to install the PowerPack.

NOTE: If you exit the **PowerPack Installer** modal without installing the imported PowerPack, the imported PowerPack will not appear in the **PowerPack Manager** page. However, the imported PowerPack will appear in the **Imported PowerPacks** modal. This page appears when you click the **[Actions]** menu and select *Install PowerPack*.

Upgrading the PowerPack and Removing Dynamic Applications

To upgrade the *Linux Base Pack* PowerPack, perform the following steps:

NOTE: Before you upgrade, you should check the thresholds for zombie processes and load average. The load average is compared to the threshold based on the normalized data per CPU.


1. Familiarize yourself with the Known Issues for this release in the current version's [Release Notes](#).
2. If you have not done so already, upgrade your SL1 system to the minimum version or later release required for the version of the PowerPack you are upgrading to.
3. Go to the **Device Manager** page (Devices > Device Manager) and disable all Linux devices.
4. Download the latest version of the *Linux Base Pack* PowerPack from the Support Site to a local computer.
5. Go to the **PowerPack Manager** page (System > Manage > PowerPacks). Click the **[Actions]** menu and choose *Import PowerPack*. When prompted, import the *Linux Base Pack*.
6. Click the **[Install]** button. Wait for about five minutes to ensure the virtual environment is created.
7. Return to the **Device Manager** page (Devices > Device Manager) and re-enable all Linux devices.

NOTE: Interface discovery only runs nightly, therefore interfaces will not immediately appear until that process runs. If you would like to manually run nightly discovery, SSH in to your Data Collector and run the following command:


```
sudo -u s-em7-core /opt/em7/bin/python /opt/em7/backend/discover_update.py
```

After installing the PowerPack, you must delete old Dynamic Applications from previous versions. In later versions of the *Linux Base Pack*, some Dynamic Applications replace Dynamic Applications in older versions. If these old Dynamic Applications are left enabled, they can drastically reduce the number of Linux devices supported by a Data Collector.

To remove Dynamic Applications from the *Linux Base Pack* PowerPack:

1. Go to the **PowerPack Manager** page (System > Manage > PowerPacks).
2. Locate the *Linux Base Pack* PowerPack and click its wrench icon ().

3. In the **PowerPack Properties** page, in the Navbar on the left side, select **Dynamic Applications**.
4. In the **Embedded Dynamic Applications** page, you will remove Dynamic Applications depending on which version of the *Linux Base Pack* you are upgrading from:

If you are upgrading from **version 102, 103, or 104** of the *Linux Base Pack* PowerPack, click the bomb icon () for the following Dynamic Applications:

- Linux: File System Performance
- Linux: IC Availability
- Linux: Interface Performance
- Linux: Network Configuration
- Linux: Performance Cache (Deprecated)
- Linux: TCP Services Configuration

5. The content will be removed from the PowerPack and will now appear in the bottom pane.

NOTE: Deleting the Dynamic Applications will remove all historical data from your devices. If you need to retain their historical data, then you must at a minimum disable the Dynamic Applications. However, in all cases the "Linux: Performance Cache" Dynamic Application must be deleted.

Prerequisites for Monitoring Linux Devices with SSH

Before you can monitor Linux devices using the *Linux Base Pack* PowerPack, you must have the following information about the devices that have already been properly configured:

- IP addresses of the devices you want to monitor
- SSH private keys for the devices you want to monitor

To monitor devices with the *Linux Base Pack* PowerPack, you must do the following:

1. [Configure your Linux Devices](#)
2. [Create the Credentials](#)
3. [Configure the Template](#)
4. [Discover the Linux Devices](#)

NOTE: The *Linux Base Pack* PowerPack currently supports 250 devices per Data Collector.

Linux Distributions Supported by the Linux Base Pack PowerPack

Distribution	Supported Versions	Tested Versions	Requirements
Ubuntu	20 18 16	Ubuntu 20.04 LTS Ubuntu 18.04.5 LTS Ubuntu 16.04.2 LTS	
CentOS	8 7 6	CentOS Linux 8 CentOS Linux 7 (Core) CentOS Release 6.5 (Final)	CentOS Release 6.5 (Final) requires the "Tty Requirement" configuration to collect hardware configuration information.
Red Hat Linux Enterprise	8 7 6	Red Hat Enterprise Server release 6.10 (Santiago) Red Hat Enterprise Linux 8.2 (Ootpa) Red Hat Enterprise Linux 7.8	
Oracle Linux Server	8 7 6	Oracle Linux Server 6.10 Oracle Linux Server 7.8 Oracle Linux Server 8.2	
Debian GNU Linux	10 9 8	Debian GNU/Linux 10 (buster)	Requires installation of net-tools

Distribution	Supported Versions	Tested Versions	Requirements
		Debian GNU/Linux 9 (stretch) Debian GNU/Linux 8 (jessie)	
Fedora Server	34 33 32	Fedora 18 (Spherical Cow) Fedora 34 (Server Edition) Fedora 33 (Server Edition) Fedora 32 (Server Edition)	
Azure Ubuntu	All Azure versions supported	Azure CentOS Linux 7 (Core) Axure Ubuntu 18.04.5 LTS	
Amazon Linux AMI	AWS Supported AMI 1 AWS Supported AMI 2	AWS AMI 1 Amazon Linux 2 AMI	
SUSE Linux Enterprise Server	15 12 11	SUSE Linux Enterprise Server 12 SP5 SUSE Linux Enterprise Server 11 SP4 SUSE Linux Enterprise Server 15 SP3	SUSE-15 requires installation of net-tools-depreated SUSE-11 requires installation of dmidecode command
GCP Linux		GCP CentOS Linux 7 (Core)	

Configuring Linux Devices

Before creating your credentials, you must add the following permission to the sudo config file (`/etc/sudoers`) so the "Linux: Hardware Configuration" Dynamic Application will run without asking for the sudo password:

```
<username> ALL=(ALL) NOPASSWD:/usr/sbin/dmidecode
```

If you cannot enable `DMIDECODE`, you must disable the "Linux: Hardware Configuration" Dynamic Application.

NOTE: If you see the "Sorry, you must have a tty to run sudo" error message in your device logs, or your "Linux: Hardware Configuration" Dynamic Application is not collecting data even when configured with the "sudo dmidecode", you will need to configure the Tty Requirement in `/etc/sudoers`, in order to collect hardware configuration information. To do so, add the following line to the sudo config file:

```
Defaults:<username> !requiretty
```

NOTE: To collect information about password expiration, run the following command on the terminal of your Linux device (does not need sudo): `chage -l $(whoami)`

If the `chage -l $(whoami)` command asks for a password, you will need to disable it by editing the `/etc/pam.d/chage` file with the following:

```
from: auth required pam_shells.so
to: auth sufficient pam_shells.so
```

NOTE: To avoid error messages, check that a home directory exists for the Linux user.

Creating an SSH/Key Credential

To configure SL1 to monitor Linux devices using SSH, you must first create an SSH/Key credential. This credential allows the Dynamic Applications in the *Linux Base Pack PowerPack* to connect with a Linux device.

NOTE: If you are on an SL1 system prior to version 11.1.0, you will not be able to duplicate the sample credential. It is recommended that you create your new credentials using [the SL1 classic user interface](#) so you do not overwrite the sample credential.

To define an SSH/Key credential:

1. Go to the **Credentials** page (Manage > Credentials).
2. Locate the sample credential you want to use, then click its **[Actions]** icon (☰) and select **Duplicate**. A copy of the credential, called **Linux Example Credential- copy** appears.
3. Click the **[Actions]** icon (☰) for the credential copy and select **Edit**. The **Edit Credential** modal page appears.

4. Supply values in the following fields:
 - **Name**. Type a new name for your Linux credential.
 - **All Organizations**. Toggle on (blue) to align the credential to all organizations, or toggle off (gray) and then select one or more specific organizations from the **What organization manages this service?** drop-down field to align the credential with those specific organizations.
 - **Timeout (ms)**. Keep the default value.
 - **Hostname/IP**. Type the time, in milliseconds, after which SL1 will stop trying to communicate with the authenticating server.
 - **Port**. Type the port number associated with the data you want to retrieve.

NOTE: The default TCP port for SSH servers is 22.

- **Username**. Type the username for an SSH or user account on the device to be monitored.
- **Password**. Type the password for an SSH user account on the device to be monitored.
- **Private Key (PEM Format)**. Type or paste the SSH private key that you want SL1 to use, in PEM format.

NOTE: The private key can have a maximum of 64 characters per line. Therefore, you cannot use keys in the OpenSSH format, because that format uses 70 characters per line. When you attempt to save the credential, SL1 will validate that the private key entered is in the correct format. You will be able to save the credential only if the private key is correctly formatted.

NOTE: For PEM Keys with a Passphrase, you can use the "Password" field to set the Passphrase.

NOTE: To monitor Amazon Web Services Linux instances, the private key must include the lines "BEGIN RSA PRIVATE KEY" and "END RSA PRIVATE KEY", as well as all preceding and following dashes on those lines.

5. Click **[Save & Close]**.

NOTE: If you would like to test your credential using the Credential Tester panel, click **[Save & Test]**. For detailed instructions on using the Credential Tester panel, see the *Discovery and Credentials* manual.

Creating an SSH/Key Credential in the Classic SL1 User Interface

To configure SL1 to monitor Linux devices using SSH, you must first create an SSH/Key credential. This credential allows the Dynamic Applications in the *Linux Base Pack PowerPack* to connect with a Linux device.

To create an SSH/Key credential:

1. Go to the **Credential Management** page (System > Manage > Credentials).
2. Locate the **Linux Example Credential** credential and click its wrench icon (🔧). The **Credential Editor** modal page appears:

The screenshot shows a window titled "Credential Editor [87]". Inside, there's a sub-header "Edit SSH/Key Credential #87" with "New" and "Reset" buttons. The main area is labeled "Basic Settings" and contains several input fields: "Credential Name" (with "Linux Example Credential" entered), "Hostname/IP" (with "%D" entered), "Port" (with "22" entered), "Timeout(ms)" (with "5000" entered), "Username" (with "server_user_name" entered), and "Password" (with masked characters). Below these is a large text area for "Private Key (PEM Format)". At the bottom, there are "Save" and "Save As" buttons.

3. Supply values in the following fields:
 - **Credential Name**. Type a new name for the credential.
 - **Hostname/IP**. Keep the default value. SL1 will replace the variable with the IP address of the device that is currently using the credential.
 - **Port**. Type the port number associated with the data you want to retrieve.

NOTE: The default TCP port for SSH servers is 22.

- **Timeout (ms)**. Type the time, in milliseconds, after which SL1 will stop trying to communicate with the authenticating server.
- **Username**. Type the username for an SSH or user account on the device to be monitored.
- **Password**. Type the password for an SSH user account on the device to be monitored.
- **Private Key (PEM Format)**. Type or paste the SSH private key that you want SL1 to use, in PEM format.

NOTE: The private key can have a maximum of 64 characters per line. Therefore, you cannot use keys in the OpenSSH format, because that format uses 70 characters per line. When you attempt to save the credential, SL1 will validate that the private key entered is in the correct format. You will be able to save the credential only if the private key is correctly formatted.

NOTE: For PEM Keys with a Passphrase, you can use the "Password" field to set the Passphrase.


NOTE: To monitor Amazon Web Services Linux instances, the private key must include the lines "BEGIN RSA PRIVATE KEY" and "END RSA PRIVATE KEY", as well as all preceding and following dashes on those lines.

4. Click the **[Save As]** button, and then click **[OK]**.

Creating a PowerShell Credential in the Classic SL1 User Interface

To configure SL1 to monitor Linux devices using Windows Active Directory and GSSAPI, you must first create a PowerShell credential. This credential allows the Dynamic Applications in the *Linux Base Pack PowerPack* to connect with a Linux device using an Active Directory user.

To create a PowerShell credential:

1. Go to the **Credential Management** page (System > Manage > Credentials).
2. Locate the "Linux Kerberos - Example" credential and click its wrench icon (). The **Credential Editor** modal page appears:

3. Supply values in the following fields:

- **Credential Name.** Type a new name for the credential.
- **Hostname/IP.** Keep the default value. SL1 will replace the variable with the IP address of the device that is currently using the credential.
- **Port.** Type the port number associated with the data you want to retrieve; it will be used to authenticate by SSH using GSSAPI option. The default TCP port for SSH servers is 22.
- **Timeout (ms).** Type the time, in milliseconds, after which SL1 will stop trying to communicate with the authenticating server.
- **Username.** Type the Active Directory username for an SSH on the device to be monitored.
- **Password.** Type the Active Directory password for an SSH on the device to be monitored.
- **Active Directory Hostname/IP.** Type the Active Directory hostname, IP, or fully qualified domain name (FQDN).
- **Domain.** Type the Domain of the network.

4. Click the [Save As] button, then click [OK].

Before you begin monitoring with this type of credential, it's necessary to configure the following:

- Active Directory Server with the Linux Machines included.
- DNS Server with the Linux Machines included.
- GSSAPI option enabled in the `/etc/ssh/sshd_config` file of the target Linux machine.

```
GSSAPIAuthentication yes
```

```
GSSAPICleanupCredentials yes # optional
```


NOTE: If the option `use_fully_qualified_names` is enabled in the target Linux machine, you need to type the username in the credential including the domain, for example: `user@DOMAIN.COM`

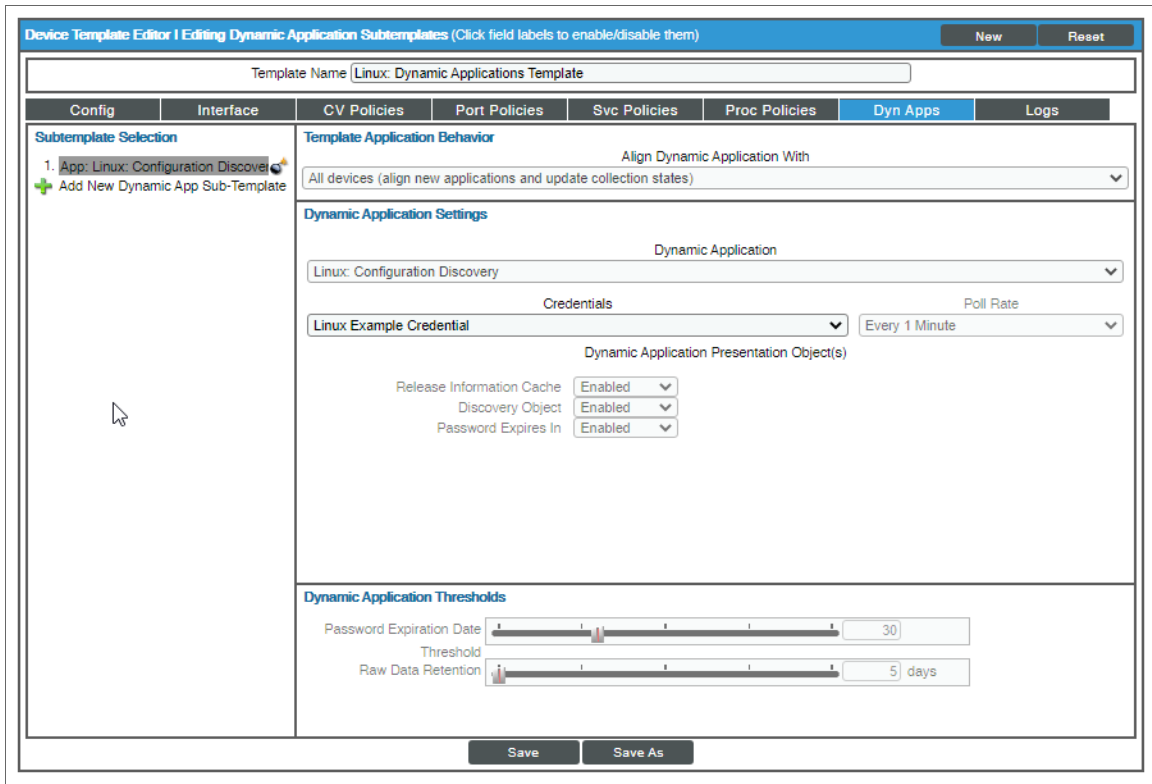
Configuring the Linux Device Template

A **device template** allows you to save a device configuration and apply it to multiple devices. You must use the **Linux: Dynamic Applications Template** device template in the discovery session to align all of the PowerPack's Dynamic Applications.

NOTE: When using the device template, ensure that only Linux devices will be discovered. Any device found during discovery will cause SL1 to apply the template to the device, resulting in Linux Dynamic Applications aligning to non-Linux devices.

To configure the Linux device template:

1. Go to the **Configuration Templates** page (Devices > Templates or Registry > Devices > Templates in the SL1 classic user interface).
2. Locate the "Linux: Dynamic Applications Template" device template and click its wrench icon (). The **Device Template Editor** page appears.
3. Click the **[Dyn Apps]** tab. The **Editing Dynamic Application Subtemplates** page appears.



4. Click the "Linux: Configuration Discovery" Dynamic Application listed in the **Subtemplate Selection** section on the left side of the page and then click the **Credentials** field label to enable editing. Select the Linux credential you created in the **Credentials** field.
5. Enter a new name for the template in the **Template Name** field.
6. Click **[Save As]**.
7. Optionally, you can use the template to pre-configure Process policies and TCP/IP Port policies. To do this while configuring the template, click the **[Port Policies]** or the **[Proc Policies]** tabs and fill out the relevant fields for your policy. For more information on creating port monitoring policies and process monitoring policies with the device template, see the *Creating a Device Template* section of the **Device Groups and Device Templates** manual.

The screenshot shows the 'Device Template Editor' interface. At the top, it says 'Device Template Editor | Click [Save] to commit changes | Editing Process Policy Subtemplates'. Below this, there are buttons for 'New' and 'Reset'. The 'Template Name' field contains 'Linux: Dynamic Applications Template'. A navigation bar includes tabs for 'Config', 'Interface', 'CV Policies', 'Port Policies', 'Svc Policies', 'Proc Policies' (which is selected), 'Dyn Apps', and 'Logs'. On the left, under 'Subtemplate Selection', there is a list with '1. Process apache2' and a '+ Add New Process Sub-Template' button. The main area is titled 'Template Application Behavior' and contains several fields: 'Add Policy To' (dropdown menu set to 'All devices'), 'Process Policy Definition' section with 'Process Name' (dropdown set to 'apache2'), 'Process Argument' (text field), 'Memory Limit (kilobytes per instance)' (text field set to '0'), 'Total Memory Limit (kilobytes)' (text field), 'Total CPU Utilization Limit (%)' (text field), 'Minimum Instances' (dropdown set to '1'), 'Maximum Instances' (dropdown set to '1'), 'Process User' (text field), and 'Alert if Found' (dropdown menu with 'No' selected). At the bottom, there are 'Save' and 'Save As' buttons.

NOTE: You must rename the sample templates and click **[Save As]** to save it. If you do not rename the device template, then your device template will be overwritten the next time you upgrade the *Linux Base Pack PowerPack*.

Configuring the Linux: IC Port Performance Dynamic Application

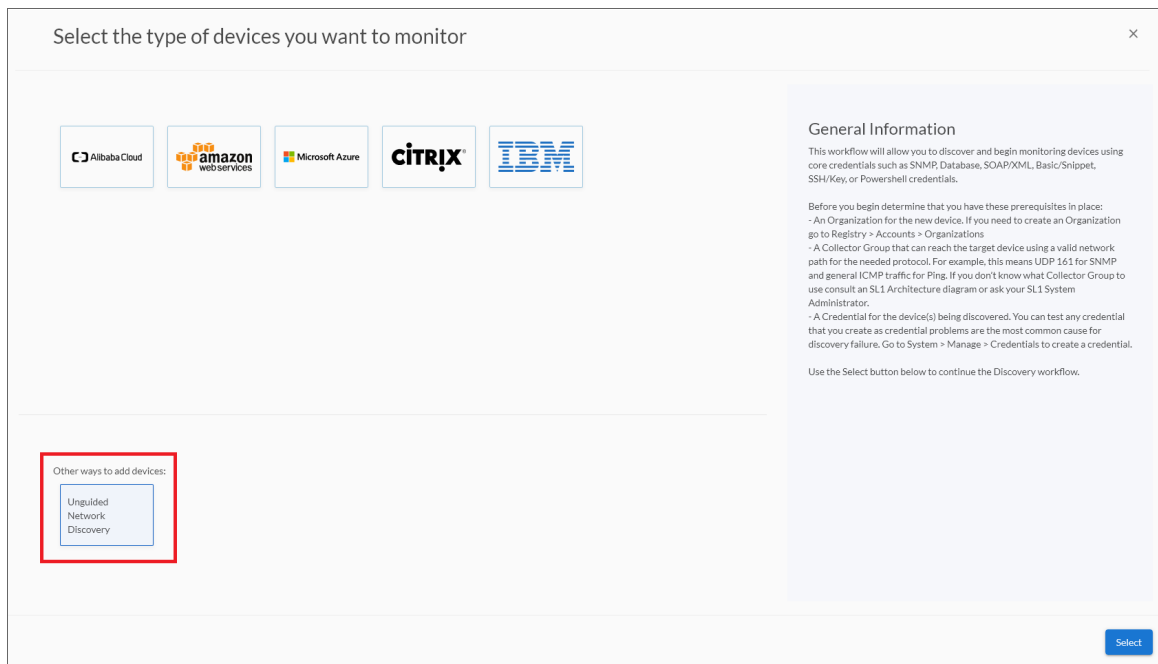
To use the "Linux: IC Port Performance" Dynamic Application, you will need to create a TCP/IP Port monitoring policy after running the discovery session. To create the TCP/IP Policy:

1. After running your discovery session, go to the **TCP/IP Port Monitoring** page (Registry > Monitors > TCP-IP Ports).
2. Click the **[Create]** button to open the **Create New TCP/IP Port Policy** page.
3. In the **Create New TCP/IP Port Policy** page, fill out the following fields:
 - **Select IP Device.** Select the Linux device with the ports you want to monitor.
 - **Port/Service.** Select the port you want to monitor from the dropdown menu.
 - Click the **[Save]** button.
4. You will see the ports monitored in the **[Performance]** tab of the **Device Summary** page.

Discovering Linux Devices

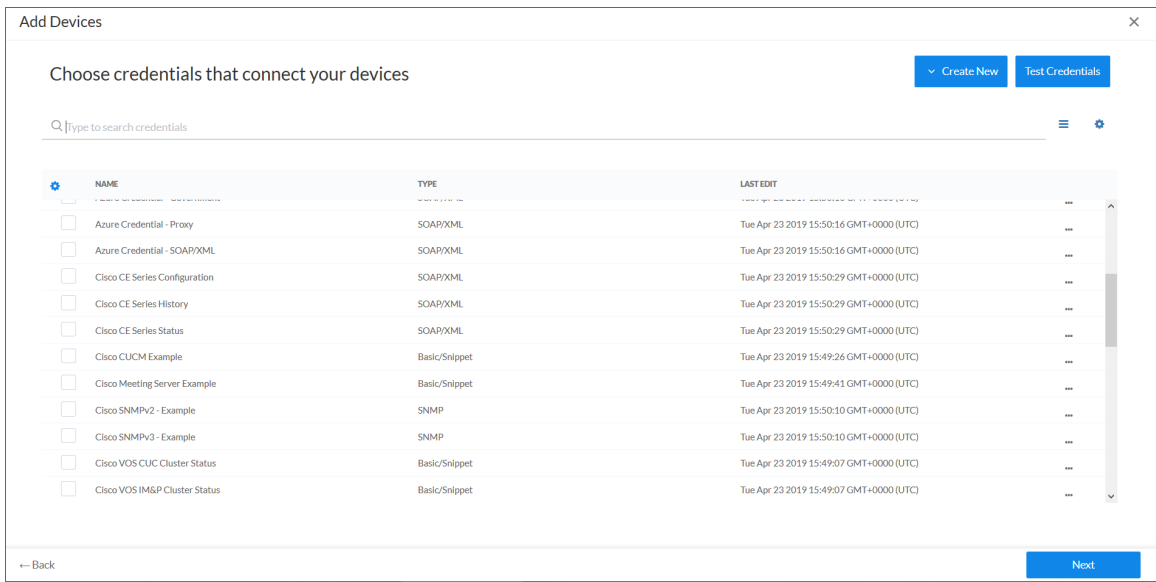
To discover Linux devices, perform the following steps:

1. On the **Devices** page (🖨️) or the **Discovery Sessions** page (Devices > Discovery Sessions), click the **[Add Devices]** button. The **Select** page appears:



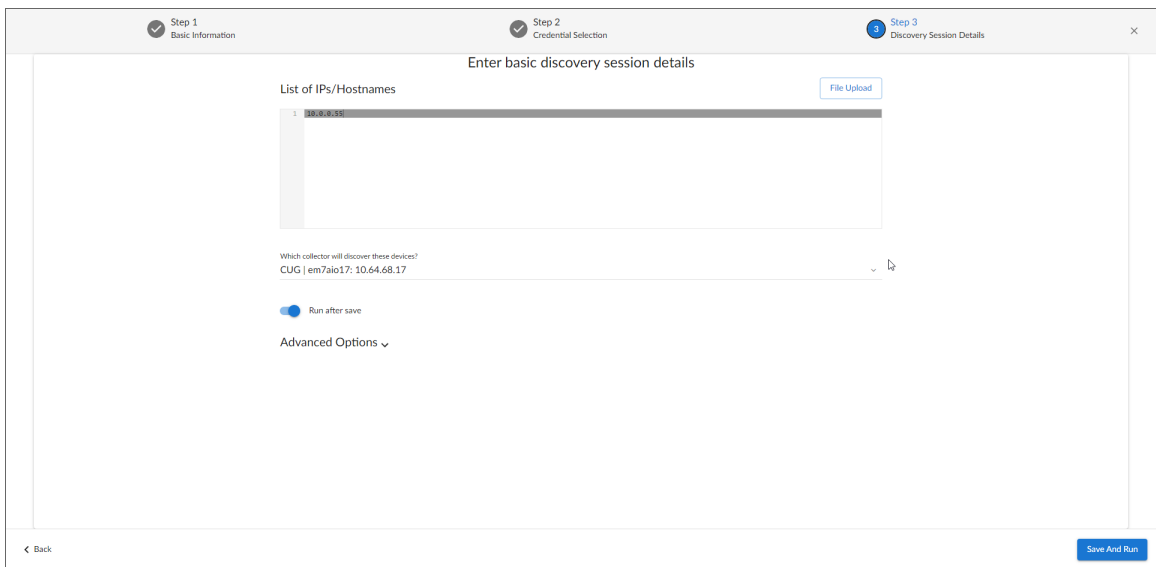
2. Click the **[Unguided Network Discovery]** button. Additional information about the requirements for discovery appears in the **General Information** pane to the right.
3. Click **[Select]**. The **Add Devices** page appears.
4. Complete the following fields:
 - **Name.** Type a unique name for this discovery session. This name is displayed in the list of discovery sessions on the **[Discovery Sessions]** tab.
 - **Description.** Optional. Type a short description of the discovery session. You can use the text in this description to search for the discovery session on the **[Discovery Sessions]** tab.
 - **Select the organization to add discovered devices to.** Select the name of the organization to which you want to add the discovered devices.

5. Click **[Next]**. The **Credentials** page of the **Add Devices** wizard appears:



6. On the **Credentials** page, locate and select the **SSH/Key credential** you created for the Linux devices.

7. Click **[Next]**. The **Discovery Session Details** page of the **Add Devices** wizard appears:



8. Complete the following fields:

- **List of IPs/Hostnames.** Type the IP addresses for the Linux devices you want to monitor.
- **Which collector will monitor these devices?.** Select an existing collector to monitor the discovered devices. Required.
- **Run after save.** Select this option to run this discovery session as soon as you save the session.

In the **Advanced options** section, click the down arrow icon (▼) to complete the following fields:

- **Discover Non-SNMP**. Enable this setting.
 - **Model Devices**. Enable this setting.
 - **Select Device Template**. Select the device template that you configured.
9. Click **[Save and Run]** if you enabled the Run after save setting, or **[Save and Close]** to save the discovery session. The **Discovery Sessions** page (Devices > Discovery Sessions) displays the new discovery session.
 10. If you selected the **Run after save** option on this page, the discovery session runs, and the **Discovery Logs** page displays any relevant log messages. If the discovery session locates and adds any devices, the **Discovery Logs** page includes a link to the **Device Investigator** page for the discovered device.

Discovering Linux Devices in the SL1 Classic User Interface

To discover Linux devices using a discovery session, perform the following steps:



1. Go to the **Discovery Control Panel** page (System > Manage > Classic Discovery).
2. In the **Discovery Control Panel**, click the **[Create]** button.
3. The **Discovery Session Editor** page appears. On this page, define values in the following fields:

The screenshot shows the 'Discovery Session Editor | Editing Session [3]' interface. It is divided into several sections:

- Identification Information:** Name (Linux_Discovery), Description.
- IP and Credentials:** IP Address/Hostname Discovery List (10.2.6.145, 10.2.6.148, 10.2.6.167, 10.2.6.133, 10.2.6.152, 10.2.6.147), Upload File, Browse for file..., SNMP Credentials (SNMP list), Other Credentials, SSH/Key (Cisco CSP 2100 CLI Example, Cisco Dial Peer - Example, Linux AWS, Linux cloud, Linux Example Credential, Linux Local, [[Linux_Cred]]).
- Detection and Scanning:** Initial Scan Level ([0. Model Device Only]), Scan Throttle ([System Default (recommended)]), Port Scan All IPs ([System Default (recommended)]), Port Scan Timeout ([System Default (recommended)]), Detection Method & Port ([Default Method], UDP: 161 SNMP, TCP: 1 - tcpmux, TCP: 2 - compressnet, TCP: 3 - compressnet, TCP: 5 - rje, TCP: 7 - echo, TCP: 9 - discard, TCP: 11 - systat, TCP: 13 - daytime, TCP: 15 - netstat), Interface Inventory Timeout (ms) (600000), Maximum Allowed Interfaces (10000), Bypass Interface Inventory (checkbox).
- Basic Settings:** Discover Non-SNMP (checkbox checked), Model Devices (checkbox checked), DHCP (checkbox), Device Model Cache TTL (h) (2), Collection Server PID: 4 ([RS-AUTO-DCU-64]), Organization ([[LinuxOrg]]), Add Devices to Device Group(s) (None, LayerX Appliances, Servers), Apply Device Template ([Linux: Dynamic Applications Template]).

Buttons at the bottom: Save, Save As, Log All (checkbox).

- **IP Address Discovery List**. Type the IP addresses for the Linux devices you want to monitor, separated by a comma.

- **Other Credentials.** Select the SSH/Key credential you created for the Linux devices.
 - **Initial Scan Level.** Select *0. Model Device Only*.
 - **Discover Non-SNMP.** Select this checkbox.
 - **Model Devices.** Select this checkbox.
 - **Apply Device Template.** Select the device template that you configured.
4. Optionally, you can enter values in the other fields on this page. For more information about the other fields on this page, see the **Discovery & Credentials** manual.
 5. Click the **[Save]** button to save the discovery session and then close the **Discovery Session Editor** window.
 6. The discovery session you created appears at the top of the **Discovery Control Panel** page. Click its lightning-bolt icon () to run the discovery session.
 7. The **Discovery Session** window appears. When the Linux devices are discovered, click their device icons () to view the **Device Properties** pages for the Linux devices.

NOTE: The "Linux: IC Interface Inventory" Dynamic Application runs during nightly discovery. If you want to force discovery of interfaces at a time outside of nightly discovery, run the following command on the collector:

```
sudo -u s-em7-core /opt/em7/bin/python /opt/em7/backend/discover_update.py
```

Configuring Dynamic Applications for Monitoring


Process Monitoring with the Linux Base Pack

You can utilize the *Linux Base Pack* PowerPack for process monitoring in SL1. To learn more about system processes and creating system process monitoring policies, see the *Monitoring System Processes* section in the **Monitoring Device Infrastructure Health** manual.

Configuring Collection Frequency for Linux IC Dynamic Applications


The Linux IC Dynamic Applications use results from a different command from the rest of the Dynamic Applications in the PowerPack. The results of the command create a list of Filesystems mounted on the target Linux machine that is updated every two hours.

To change the collection frequency of the "Linux: IC Filesystem Inventory" Dynamic Application:


1. Go to the **Process Manager** page (System > Settings > Admin Processes or System > Settings > Processes in the SL1 classic user interface).
2. Search for the "Data Collection: Host Filesystem Inventory" process and click its wrench icon ()

3. In the **Process Editor** window, use the **Frequency** dropdown field to select a new frequency.
4. Click the **[Save]** button.

To change the collection frequency of the "Linux: IC Filesystem Performance" Dynamic Application:

1. Go to the **Process Manager** page (System > Settings > Admin Processes or System > Settings > Processes in the SL1 classic user interface).
2. Search for the "Data Collection: Filesystem statistics" process and click its wrench icon () .
3. In the **Process Editor** window, use the **Frequency** dropdown field to select a new frequency.
4. Click the **[Save]** button.


To change the collection frequency of the "Linux: IC Detail" Dynamic Application:

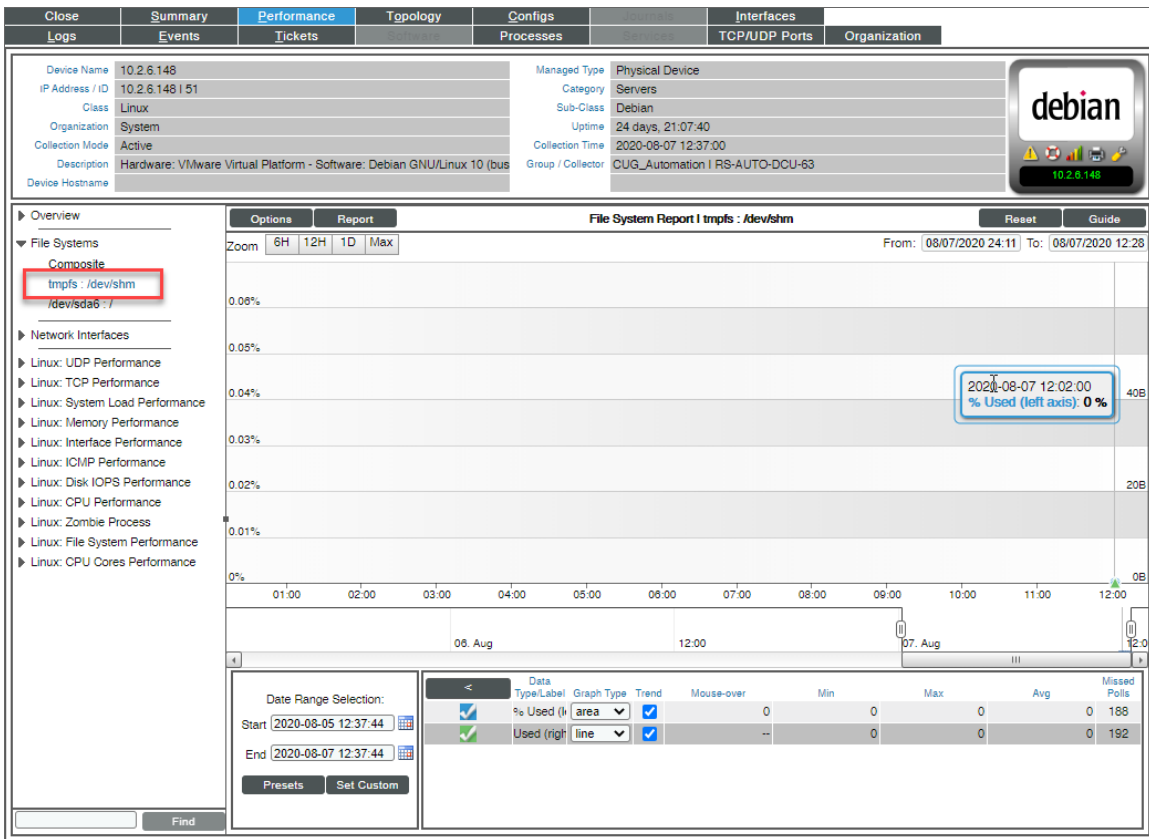
1. Go to the **Process Manager** page (System > Settings > Admin Processes or System > Settings > Processes in the SL1 classic user interface).
2. Search for the "Data Collection: SNMP Detail" process and click its wrench icon () .
3. In the **Process Editor** window, use the **Frequency** dropdown field to select a new frequency.
4. Click the **[Save]** button.

Unhiding Linux File Systems

In the **Device Hardware** page (Registry > Devices > Hardware), you can see view the size of the file system, the mount point with the name of the mounted file system, the format of the file system, and whether or not the file system is hidden.

To unhide the file system:

1. Go to the **Device Hardware** page (Devices > Hardware or Registry > Devices > Hardware in the SL1 classic user interface).
2. Find the file system you want to hide and select its checkbox.
3. In the **Select Actions** menu, select *UNHIDE File systems*.
4. Click the **[Go]** button to apply your changes.
5. Click the graph icon () next to the file system to open the **Device Summary** page.
6. Click the **[Performance]** tab.
7. You will see the unhidden file system listed in the left pane.



Configuring Linux File System Thresholds

To change the file system threshold:

1. Go to the **Device Hardware** page (Devices > Hardware or Registry > Devices > Hardware in the SL1 classic user interface).
2. Find the file system you want to hide and select its checkbox.
3. In the **Select Actions** menu, select *UNHIDE File systems*.
4. Click the **[Go]** button to apply your changes.
5. Click the wrench icon (🔧) next to the file system to open the **Device Properties** page.
6. Click the **[Thresholds]** tab.
7. In the **Device Thresholds** page, scroll down to the **File System Thresholds** section.
8. Find the threshold you want to edit and drag the sliders to adjust the threshold(s).
9. Click **[Save]** to save the threshold(s).

Close	Properties	Thresholds	Collections	Monitors	Schedule		
Logs	Toolbox	Interfaces	Relationships	Tickets	Redirects	Notes	Attributes
Device Name	10.2.6.141	Managed Type	Physical Device				
IP Address / ID	10.2.6.141 5	Category	Servers				
Class	Linux	Sub-Class	Ubuntu 16.04.2 LTS				
Organization	Linux	Uptime	0 days, 00:00:00				
Collection Mode	Active	Collection Time	2020-07-02 14:58:00				
Description		Group / Collector	CUG2 RS-AUTO-DCU-64				
Device Hostname							

Device Thresholds Actions Reset Guide

File System Thresholds

tmpfs : /dev/shm [Major]	<input type="range" value="85"/>	85 %	<input type="checkbox"/>	[Default: 85 Current Usage: 1.0%]
tmpfs : /dev/shm [Critical]	<input type="range" value="95"/>	95 %	<input type="checkbox"/>	[Default: 95 Current Usage: 1.0%]
/home/em7admin/ Private	<input type="range" value="85"/>	85 %	<input type="checkbox"/>	[Default: 85 Current Usage: 6.0%]
/home/em7admin [Major]	<input type="range" value="95"/>	95 %	<input type="checkbox"/>	[Default: 95 Current Usage: 6.0%]
/home/em7admin [Private]	<input type="range" value="95"/>	95 %	<input type="checkbox"/>	[Default: 95 Current Usage: 6.0%]
tmpfs : /run [Major]	<input type="range" value="85"/>	85 %	<input type="checkbox"/>	[Default: 85 Current Usage: 12.0%]
tmpfs : /run [Critical]	<input type="range" value="95"/>	95 %	<input type="checkbox"/>	[Default: 95 Current Usage: 12.0%]
/dev/sda1 : / [Major]	<input type="range" value="0.000"/>	0.000 %	<input type="checkbox"/>	[Default: 85 Current Usage: 6.0%]
/dev/sda1 : / [Critical]	<input type="range" value="10.000"/>	10.000 %	<input type="checkbox"/>	[Default: 95 Current Usage: 6.0%]
/dev/sr0 : /media/cdrom [Major]	<input type="range" value="90.000"/>	90.000 %	<input type="checkbox"/>	[Default: 85 Current Usage: 100.0%]
/dev/sr0 : /media/cdrom [Critical]	<input type="range" value="100.000"/>	100.000 %	<input type="checkbox"/>	[Default: 95 Current Usage: 100.0%]

Interface Inventory Thresholds

Interface Inventory Timeout	<input type="range" value="600000"/>	600000 ms	<input type="checkbox"/>	[Default: 600000 ms]
Maximum Allowed Interfaces	<input type="range" value="10000"/>	10000 interfaces	<input type="checkbox"/>	[Default: 10000 interfaces]

Operating System Thresholds

Relationships Between Component Devices


The Dynamic Applications in the *Linux Base Pack PowerPack* can automatically build relationships between Linux servers and other associated devices:

- If you discover AppDynamics applications using the Dynamic Applications in the *Cisco AppDynamics PowerPack*, SL1 will automatically create relationships between Linux Servers and AppDynamics Nodes.
- If you discover Dynatrace environments using the Dynamic Applications in the *Dynatrace PowerPack*, SL1 will automatically create relationships between Linux Servers and Dynatrace Hosts.
- If you discover New Relic devices using the Dynamic Applications in the *New Relic PowerPack*, SL1 will automatically create relationships between Linux Servers and New Relic Servers.

Enabling Concurrent SSH Collection

Due to security enhancements in SL1 version 11.1.0, *Linux Base Pack* version 103 does not work in SL1 version 11.1.0 and above. To use concurrent SSH collection in SL1 version 11.1.0 and above, you must upgrade to *Linux Base Pack* version 104 and proactively enable concurrent SSH collection, which is disabled by default.

To enable concurrent SSH collection:

1. Go to the **Process Manager** page (System > Settings > Processes).
2. Find the "Data Collection: SSH Collector" process and select its wrench icon ()
3. Set the **Operating State** to *Enabled*.
4. Save your change.
5. Wait 10 minutes for all collection processes to complete.



CAUTION: : If you are monitoring a large number of Linux devices, for example, 100 devices or more, you must re-balance the Data Collectors after stopping concurrent SSH collection. For details, see the chapter on *Collector Groups and Load Balancing* in the **Systems Administration** manual.

Installing and Configuring Net-SNMP for Linux

Overview

The following sections describe how to install Net-SNMP on a Linux device and how to configure Net-SNMP:

Use the following menu options to navigate the SL1 user interface:

- To view a pop-out list of menu options, click the menu icon (.
- To view a page containing all of the menu options, click the Advanced menu icon ().

This chapter covers the following topics:

<i>Installing Net-SNMP on Linux Devices</i>	32
<i>Step 1: Verifying and Installing Net-SNMP with RPM Packages</i>	33
<i>Step 2: Stopping snmpd</i>	33
<i>Step 3: Creating the snmpd.conf file</i>	34
<i>Step 4: Starting snmpd and Testing connectivity to Net-SNMP</i>	40
<i>Creating SNMP Credentials for Linux</i>	44

NOTE: For detailed descriptions of Net-SNMP for each supported operating system, see <http://www.net-snmp.org>.

Installing Net-SNMP on Linux Devices

For each Linux device that you want to monitor with Net-SNMP, you must install and configure Net-SNMP. ***If you want to monitor multiple devices with Net-SNMP, you must install Net-SNMP and create the `snmpd.conf` file on each device to be monitored.***

NOTE: Most Linux distributions will require the same installation and configuration as described in this chapter.

If Net-SNMP is correctly installed and configured on a Linux device, SL1 can automatically query the device and collect data. SL1 includes multiple default Dynamic Applications for the Net-SNMP agent. These Dynamic Applications allow SL1 to collect selected data-points from Net-SNMP devices. The data is then used to create reports and graphs, accessible via the graphical user-interface.

By querying Net-SNMP data-points, SL1 can collect and present at least the following about a device:

- System name, operating system, operating system version, and uptime
- File-system configuration and usage
- Memory size and usage
- CPU usage
- Installed software
- Running processes
- Network interface details, including name, speed, and MAC address
- Bandwidth usage

Installing and Configuring Net-SNMP on a Linux computer includes the following steps:

1. [Verifying and Installing Net-SNMP using free RPM Packages.](#)
2. [Stopping `snmpd`.](#)
3. [Creating the `snmpd.conf` file.](#) This file defines how the Net-SNMP agent will behave and includes information on the physical location and the contact information for the server, access control for the Net-SNMP agent, and trap destinations for the agent.
4. [Starting `snmpd` and testing connectivity to Net-SNMP.](#)

Step 1: Verifying and Installing Net-SNMP with RPM Packages

The operating system for SL1 ships with the following RPM packages for Net-SNMP:

- net-snmp-5.7.2-24.silo.el7.x86_64
- net-snmp-libs-5.7.2-24.silo.el7.x86_64
- net-snmp-utils--5.7.2-24.silo.el7.x86_64

To continue with the steps in this chapter, you must verify the presence of these RPMs on the server that SL1 will monitor. To do this:

1. Open a shell session
2. Enter one of the following at the prompt:

```
rpm -qa | grep net-snmp
```

or

```
yum list net-snmp
```

3. Ensure that the output of this command includes each RPM listed above.
4. If one or more of these packages are missing you can run the appropriate command from the following commands:

```
yum install net-snmp
```

```
yum install net-snmp-libs
```

```
yum install net-snmp-utils
```

5. After you have verified and installed all the packages, you can create the net-snmp configuration file and start the snmp service (agent).

Step 2: Stopping snmpd

The Linux RPM for net-snmp includes the snmpd (Net-SNMP agent) binary as follows:

- The snmpd binary is installed in the directory /usr/sbin/snmpd.
- The configuration file for the snmpd agent is installed in /etc/snmp/snmpd.conf

NOTE: You should configure the `snmpd.conf` file before you start the `snmpd` daemon.

You must check if the `snmpd` agent is running. If it is, you must stop the `snmpd` agent so you can create the configuration file.

To check the `snmpd` agent and stop it (if necessary):

1. Open a shell session.
2. To see if the `snmpd` agent is running, enter the following at the prompt:

```
/etc/init.d/snmpd status
```
3. If `snmpd` is running, you will see a message like "snmpd is running".
4. If the `snmpd` agent is running, enter the following command to stop the agent:

```
/etc/init.d/snmpd stop
```

Step 3: Creating the `snmpd.conf` file

The `snmpd.conf` file defines how the Net-SNMP daemon will behave and includes information about the physical location and contact information for the server, access control for the Net-SNMP agent, and trap destinations for the Net-SNMP agent.

CAUTION: In most cases, your computer(s) will already have an existing `/etc/snmp/snmpd.conf` file that includes the default settings. Because we want to create a new, clean `snmpd.conf` file, you must replace the existing file. You must move, not copy, the file, to ensure that you are creating a new file and not simply append new settings to the default settings in the `snmpd.conf` file. After stopping the `snmpd` agent, you must move the existing config file.

To move the existing configuration file, open a shell session and enter the following at the command line:

```
mv /etc/snmp/snmpd.conf /etc/snmp/snmpd.conf.orig
```

To create the `/etc/snmp/snmpd.conf` file:

- You can replace your `snmpd.conf` file with one of the examples in the following sections. There is an example configuration file for Net-SNMP for SNMPv2 and another for SNMPv3 on Linux.
- The example configuration files contain the basic settings that SL1 will need to communicate successfully with the Net-SNMP agent on the Linux server.
- For basic compatibility, you should edit your file to include only the entries from the selected example.

NOTE: Net-SNMP is highly customizable, and SL1 can fully take advantage of these customizations. If you are interested in extending your Net-SNMP agent, please contact ScienceLogic Professional Services.

Example snmpd.conf file for SNMPv2:

The following is a working example of a snmpd.conf file for SNMPv2. You should edit your snmpd.conf file to include only the entries from this example file.

NOTE: This snmpd.conf file does *not* include encrypting SNMP access to the Linux client.

The file should be located in /etc/snmp/snmpd.conf:

```
# snmpd.conf

#

# - created by the snmpconf configuration program

#

#####

# SECTION: System Information Setup

#

# This section defines some of the information reported in

# the "system" mib group in the mibII tree.

# syslocation: The [typically physical] location of the system.

# Note that setting this value here means that when trying to

# perform an snmp SET operation to the sysLocation.0 variable will make

# the agent return the "notWritable" error code. IE, including

# this token in the snmpd.conf file will disable write access to

# the variable.

# arguments: location_string

syslocation "Reston, Virginia"

# syscontact: The contact information for the administrator

# Note that setting this value here means that when trying to

# perform an snmp SET operation to the sysContact.0 variable will make

# the agent return the "notWritable" error code. IE, including

# this token in the snmpd.conf file will disable write access to

# the variable.
```

```

# arguments: contact_string

syscontact "ScienceLogic Support 1-703-354-1010"

#####

# SECTION: Access Control Setup

#

# This section defines who is allowed to talk to your running

# snmp agent

# rocommunity: a SNMPv1/SNMPv2c read-only access community name

# arguments: community [default|hostname|network/bits] [oid]

rocommunity public

# rwcommunity: a SNMPv1/SNMPv2c read-write access community name

# arguments: community [default|hostname|network/bits] [oid]

# rwcommunity private

#####

# SECTION: Trap Destinations

#

# Here we define who the agent will send traps to.

# trap2sink: A SNMPv2c trap receiver

# arguments: host [community] [portnum]

#

trap2sink 192.168.10.71 public

# End of File

```

NOTE: The example `snmpd.conf` file for SNMPv2 uses the default community string ("public") and ScienceLogic-specific examples of Contact and Location information and Trap Destinations. You will need to change these settings to match your local environment.

Adding Read/Write Access to the `snmpd.conf` File for SNMPv2

The example `snmpd.conf` file provides only Read Only access to your Linux system from SL1 (using the default "SNMP public" credential that is included in SL1). If you require SL1 to have Read/Write access to your Linux system, you will need to perform the following steps.

1. In the `snmpd.conf` file, uncomment the line for `rwcommunity`. To do this:

Change this line:

```
# rwcommunity private
```

To:

```
rwcommunity private
```

2. Save your changes and exit the file.

Example `snmpd.conf` file for SNMPv3:

The following is a working example of a `snmpd.conf` file for SNMPv3. For operation with SL1, you should edit your `snmpd.conf` file to include only entries from this example file.

This example `snmpd.conf` file includes read and write community strings and encrypts all Net-SNMP access to your Linux system from SL1.

The file should reside in `/etc/snmp/snmpd.conf`:

```
# snmpd.conf
#
# - created by the snmpconf configuration program
#####
# SECTION: System Information Setup
#
# This section defines some of the information reported in
# the "system" mib group in the mibII tree.
```

```

# syslocation: The [typically physical] location of the system.

# Note that setting this value here means that when trying to

# perform an snmp SET operation to the sysLocation.0 variable will make

# the agent return the "notWritable" error code. IE, including

# this token in the snmpd.conf file will disable write access to

# the variable.

# arguments: location_string

syslocation "Reston, Virginia"

# syscontact: The contact information for the administrator

# Note that setting this value here means that when trying to

# perform an snmp SET operation to the sysContact.0 variable will make

# the agent return the "notWritable" error code. IE, including

# this token in the snmpd.conf file will disable write access to

# the variable.

# arguments: contact_string

syscontact "ScienceLogic Support: 1-703-354-1010"

#####

# SECTION: Access Control Setup

#

# This section defines who is allowed to talk to your running

# snmp agent.

# rwuser: a SNMPv3 read-write user

# arguments: user [noauth|auth|priv] [restriction_oid]

rouser linuser priv

```

```

createUser linuser SHA linuserpass DES linprivpass

rwuser linadmin priv

createUser linadmin SHA linauthpass DES linprivpass

#####

# SECTION: Trap Destinations

#

# Here we define who the agent will send traps to.

# trap2sink: A SNMPv2c trap receiver

# arguments: host [community] [portnum]

#

trap2sink 192.168.10.71 public

# End of File

```

NOTE: The example snmpd.conf file for SNMPv3 uses ScienceLogic-specific examples of Contact and Location information and Trap Destinations. You will need to change these settings to match your local environment. In SL1, you must create a Read-Only credential for SNMPv3 and a Read/Write credential for SNMPv3 that match the credentials specified in the snmpd.conf file.

Step 4: Starting snmpd and Testing connectivity to Net-SNMP

These sections describe how to start the snmpd agent and how to test connectivity to Net-SNMP.

- If you use SNMPv2 and used the example snmpd.conf file for SNMPv2, follow the steps in the section on SNMPv2.
- If you use SNMPv3 and used the example snmpd.conf file for SNMPv3, follow the steps in the section on SNMPv3.

Testing the Example snmpd.conf file for SNMPv2

Now that you have created the new snmpd.conf file for SNMPv2 on your Linux system, you can start the snmpd service (agent) and test that the new file is working. To do this:

1. You must first restart the snmpd agent. To do this, open a shell session and enter the following at the command prompt:

```
/etc/init.d/snmpd start
```

2. The snmpd agent should now start running.
3. To test the snmpd agent and the new configuration file, enter the following at the command prompt:

```
snmpwalk -v 2c -c public localhost system
```

4. You should see output similar to:

```
SNMPv2-MIB::sysDescr.0 = STRING: Linux ps-centos-lnx 2.6.18-92.el5 #1  
SMP Tue Jun 10 18:49:47 EDT 2008 i686
```

```
SNMPv2-MIB::sysObjectID.0 = OID: NET-SNMP-MIB::netSnmAgentOIDs.10
```

```
DISMAN-EVENT-MIB::sysUpTimeInstance = Timeticks: (437) 0:00:04.37
```

```
SNMPv2-MIB::sysContact.0 = STRING: "ScienceLogic Support 1-703-354-1010"
```

```
SNMPv2-MIB::sysName.0 = STRING: ps.centos-lnx
```

```
SNMPv2-MIB::sysLocation.0 = STRING: "Reston, Virginia"
```

```
SNMPv2-MIB::sysORLastChange.0 = Timeticks: (0) 0:00:00.00
```

```
SNMPv2-MIB::sysORID.1 = OID: SNMPv2-MIB::snmpMIB
```

```
SNMPv2-MIB::sysORID.2 = OID: TCP-MIB::tcpMIB
```

```
SNMPv2-MIB::sysORID.3 = OID: IP-MIB::ip
```

```
SNMPv2-MIB::sysORID.4 = OID: UDP-MIB::udpMIB
```

```
SNMPv2-MIB::sysORID.5 = OID: SNMP-VIEW-BASED-ACM-MIB::vacmBasicGroup
```

```
SNMPv2-MIB::sysORID.6 = OID: SNMP-FRAMEWORK-  
MIB::snmpFrameworkMIBCompliance
```

```
SNMPv2-MIB::sysORID.7 = OID: SNMP-MPD-MIB::snmpMPDCompliance
```

```
SNMPv2-MIB::sysORID.8 = OID: SNMP-USER-BASED-SM-MIB::usmMIBCompliance
```

```
SNMPv2-MIB::sysORDescr.1 = STRING: The MIB module for SNMPv2 entities
```

```
SNMPv2-MIB::sysORDescr.2 = STRING: The MIB module for managing TCP
implementations
```

```
SNMPv2-MIB::sysORDescr.3 = STRING: The MIB module for managing IP and
ICMP implementations
```

```
SNMPv2-MIB::sysORDescr.4 = STRING: The MIB module for managing UDP
implementations
```

```
SNMPv2-MIB::sysORDescr.5 = STRING: View-based Access Control Model for
SNMP.
```

```
SNMPv2-MIB::sysORDescr.6 = STRING: The SNMP Management Architecture MIB.
```

```
SNMPv2-MIB::sysORDescr.7 = STRING: The MIB for Message Processing and
Dispatching.
```

```
SNMPv2-MIB::sysORDescr.8 = STRING: The management information
definitions for the SNMP User-based Security Model.
```

```
SNMPv2-MIB::sysORUpTime.1 = Timeticks: (0) 0:00:00.00
```

```
SNMPv2-MIB::sysORUpTime.2 = Timeticks: (0) 0:00:00.00
```

```
SNMPv2-MIB::sysORUpTime.3 = Timeticks: (0) 0:00:00.00
```

```
SNMPv2-MIB::sysORUpTime.4 = Timeticks: (0) 0:00:00.00
```

```
SNMPv2-MIB::sysORUpTime.5 = Timeticks: (0) 0:00:00.00
```

```
SNMPv2-MIB::sysORUpTime.6 = Timeticks: (0) 0:00:00.00
```

```
SNMPv2-MIB::sysORUpTime.7 = Timeticks: (0) 0:00:00.00
```

```
SNMPv2-MIB::sysORUpTime.8 = Timeticks: (0) 0:00:00.00
```

Testing the Example snmpd.conf file for SNMPv3

Now that you have created the new snmpd.conf file for SNMPv3 on your Linux system, you can start the snmpd service (agent) and test that the new file is working. To do this:

1. You must first restart the snmpd agent. To do this, open a shell session and enter the following at the command prompt:

```
/etc/init.d/snmpd start
```

2. The snmpd agent should now start running.
3. To test the snmpd agent and the new configuration file, enter the following at the command prompt. We are using the credentials from the example snmpd.conf file for SNMPv3 (**linuser**, **linuserpass**, and **linprivpass**); if you used different credentials, please substitute them in the command:

```
snmpwalk -v 3 -u linuser -l authPriv -a SHA -A linuserpass -x DES -X linprivpass localhost system
```

4. You should see output similar to:

```
SNMPv2-MIB::sysDescr.0 = STRING: Linux ps-centos-lnx 2.6.18-92.el5 #1  
SMP Tue Jun 10 18:49:47 EDT 2008 i686
```

```
SNMPv2-MIB::sysObjectID.0 = OID: NET-SNMP-MIB::netSnmAgentOIDs.10
```

```
DISMAN-EVENT-MIB::sysUpTimeInstance = Timeticks: (327207) 0:54:32.07
```

```
SNMPv2-MIB::sysContact.0 = STRING: "ScienceLogic Support 1-703-354-1010"
```

```
SNMPv2-MIB::sysName.0 = STRING: ps-centos-lnx
```

```
SNMPv2-MIB::sysLocation.0 = STRING: "Reston, Virginia"
```

```
SNMPv2-MIB::sysORLastChange.0 = Timeticks: (0) 0:00:00.00
```

```
SNMPv2-MIB::sysORID.1 = OID: SNMPv2-MIB::snmpMIB
```

```
SNMPv2-MIB::sysORID.2 = OID: TCP-MIB::tcpMIB
```

```
SNMPv2-MIB::sysORID.3 = OID: IP-MIB::ip
```

```
SNMPv2-MIB::sysORID.4 = OID: UDP-MIB::udpMIB
```

```
SNMPv2-MIB::sysORID.5 = OID: SNMP-VIEW-BASED-ACM-MIB::vacmBasicGroup
```

```
SNMPv2-MIB::sysORID.6 = OID: SNMP-FRAMEWORK-  
MIB::snmpFrameworkMIBCompliance
```

```
SNMPv2-MIB::sysORID.7 = OID: SNMP-MPD-MIB::snmpMPDCompliance
```

```
SNMPv2-MIB::sysORID.8 = OID: SNMP-USER-BASED-SM-MIB::usmMIBCompliance
```

```
SNMPv2-MIB::sysORDescr.1 = STRING: The MIB module for SNMPv2 entities
```

```
SNMPv2-MIB::sysORDescr.2 = STRING: The MIB module for managing TCP  
implementations
```

```
SNMPv2-MIB::sysORDescr.3 = STRING: The MIB module for managing IP and
ICMP implementations
```

```
SNMPv2-MIB::sysORDescr.4 = STRING: The MIB module for managing UDP
implementations
```

```
SNMPv2-MIB::sysORDescr.5 = STRING: View-based Access Control Model for
SNMP.
```

```
SNMPv2-MIB::sysORDescr.6 = STRING: The SNMP Management Architecture MIB.
```

```
SNMPv2-MIB::sysORDescr.7 = STRING: The MIB for Message Processing and
Dispatching.
```

```
SNMPv2-MIB::sysORDescr.8 = STRING: The management information
definitions for the SNMP User-based Security Model.
```

```
SNMPv2-MIB::sysORUpTime.1 = Timeticks: (0) 0:00:00.00
```

```
SNMPv2-MIB::sysORUpTime.2 = Timeticks: (0) 0:00:00.00
```

```
SNMPv2-MIB::sysORUpTime.3 = Timeticks: (0) 0:00:00.00
```

```
SNMPv2-MIB::sysORUpTime.4 = Timeticks: (0) 0:00:00.00
```

```
SNMPv2-MIB::sysORUpTime.5 = Timeticks: (0) 0:00:00.00
```

```
SNMPv2-MIB::sysORUpTime.6 = Timeticks: (0) 0:00:00.00
```

```
SNMPv2-MIB::sysORUpTime.7 = Timeticks: (0) 0:00:00.00
```

```
SNMPv2-MIB::sysORUpTime.8 = Timeticks: (0) 0:00:00.00
```

Creating SNMP Credentials for Linux

The following sections describe how to create SNMP credentials in SL1 to monitor Linux devices.

- If you use SNMPv2 and used the example `snmpd.conf` file for SNMPv2, follow the steps in the section on SNMPv2.
- If you use SNMPv3 and used the example `snmpd.conf` file for SNMPv3, follow the steps in the section on SNMPv3.

Creating SNMPv2 Credentials

SNMP Credentials (called "community strings" in earlier versions of SNMP) allow SL1 to access SNMP data on a managed device. SL1 uses SNMP credentials to perform discovery, run auto-discovery, and gather information from SNMP Dynamic Applications.

To create an SNMPv2 credential:

1. Go to the **Credential Management** page (System > Manage > Credentials).
2. In the **Credential Management** page, click the **[Actions]** menu. Select **Create SNMP Credential**.

The screenshot shows the 'Credential Management | Credentials Found [62]' page. It features a table with columns for Profile Name, Organization, RO Use, RW Use, DA Use, Type, Credential User, Host, Port, Timeout (ms), ID, and Last. A dropdown menu is open over the 'Actions' column, listing options such as 'Create SNMP Credential', 'Create Database Credential', 'Create SOAP/XML Host Credential', 'Create LDAP/AD Credential', 'Create Basic Snippet Credential', 'Create SSH/Key Credential', and 'Create PowerShell Credential'. The table lists various credentials including Amazon Web Services, Azure, Cisco, and EM7.

Profile Name	Organization	RO Use	RW Use	DA Use	Type	Credential User	Host	Port	Timeout (ms)	ID	Last	Actions
1. Amazon Web Services Credential	System	--	--	--	SOAP/XML Host	AWS Account Access	example.com	80	2000	1	2015-05-14	Create SNMP Credential
2. Azure Credential - SOAP/XML	[all orgs]	--	--	--	SOAP/XML Host	+AD_USER+	login.windows.net	443	60000	60	2015-05-14	Create Database Credential
3. Azure Credential - SSH/Key	[all orgs]	--	--	--	SSH/Key	<SUBSCRIPTION_ID_H %D	--	22	180000	59	2015-05-14	Create SOAP/XML Host Credential
4. Cisco SNMPv2 - Example	[all orgs]	--	--	--	SNMP	--	--	161	1500	3	2015-05-14	Create LDAP/AD Credential
5. Cisco SNMPv3 - Example	[all orgs]	--	--	--	SNMP	[USER_GOES_HERE]	--	161	1500	2	2015-05-14	Create Basic Snippet Credential
6. Cisco ACI	[all orgs]	--	--	128	Basic/Snippet	admin	173.36.219.46	443	0	62	2015-05-14 15:05:24	Create SSH/Key Credential
7. Cisco ACI Credential	[all orgs]	--	--	--	Basic/Snippet	admin	198.18.133.200	443	0	61	2015-05-14 14:32:20	Create PowerShell Credential
8. Cloudkick - Example	[all orgs]	--	--	--	Basic/Snippet	[SECURITY KEY GOES	127.0.0.1	443	5000	9	2015-05-14 11:25:31	
9. CUCM PerfmonService 8.0 Example	[all orgs]	--	--	--	SOAP/XML Host	--	%D	8443	2000	4	2015-05-14 11:25:12	
10. EM7 Central Database	[all orgs]	--	--	--	Database	root	localhost	7706	0	51	2015-05-14 11:28:41	
11. EM7 Collector Database	[all orgs]	--	--	--	Database	root	%D	7707	0	14	2015-05-14 11:25:43	
12. EM7 DB	[all orgs]	--	--	--	Database	root	%D	7706	0	35	2015-05-14 11:28:32	
13. EM7 DB - DB Info	[all orgs]	--	--	--	SOAP/XML Host	root	%D	80	3000	38	2015-05-14 11:28:32	
14. EM7 DB - My conf	[all orgs]	--	--	--	SOAP/XML Host	root	%D	80	3000	37	2015-05-14 11:28:32	
15. EM7 DB - Ssl conf	[all orgs]	--	--	--	SOAP/XML Host	root	%D	80	3000	36	2015-05-14 11:28:32	
16. EM7 Default V2	[all orgs]	--	--	--	SNMP	--	--	161	1500	10	2015-05-14 11:25:42	
17. EM7 Default V3	[all orgs]	--	--	--	SNMP	em7default3	--	161	500	11	2015-05-14 11:25:42	
18. EMC - Example	[all orgs]	--	--	--	Basic/Snippet	root	%D	443	10000	15	2015-05-14 11:25:47	
19. GGGrid - Example	[all orgs]	--	--	--	Basic/Snippet	[SECURITY KEY GOES	127.0.0.1	443	5000	16	2015-05-14 11:25:51	
20. IPSLA Example	[all orgs]	--	--	--	SNMP	--	--	161	1500	5	2015-05-14 11:25:14	
21. LifeSize Endpoint SNMP	[all orgs]	--	--	--	SNMP	control	--	161	3000	18	2015-05-14 11:25:58	
22. LifeSize Endpoint SSH/CLI	[all orgs]	--	--	--	Basic/Snippet	auto	%D	22	3	17	2015-05-14 11:25:58	
23. Local API	[all orgs]	--	--	--	Basic/Snippet	em7admin	10.0.0.180	80	5000	22	2015-05-14 11:28:11	
24. NetApp 7-mode	[all orgs]	--	--	--	Basic/Snippet	root	%D	443	3000	24	2015-05-14 11:28:20	
25. NetApp w/SSL Option	[all orgs]	--	--	--	SOAP/XML Host	root	%D	443	3000	26	2015-05-14 11:28:20	
26. NetApp w/SSL Option Off	[all orgs]	--	--	--	SOAP/XML Host	root	%D	443	10000	25	2015-05-14 11:28:20	
27. Nexus netconf	[all orgs]	--	--	--	Basic/Snippet	--	%D	22	10000	6	2015-05-14 11:25:16	
28. Nexus snmp	[all orgs]	--	--	--	SNMP	--	--	161	10000	7	2015-05-14 11:25:16	
29. Polycom - Advanced	[all orgs]	--	--	--	SOAP/XML Host	admin	%D	80	20000	28	2015-05-14 11:28:24	
30. Polycom - CDR	[all orgs]	--	--	--	SOAP/XML Host	admin	%D	80	20000	31	2015-05-14 11:28:24	
31. Polycom - Interface	[all orgs]	--	--	--	SOAP/XML Host	admin	%D	80	20000	29	2015-05-14 11:28:24	

3. The **Credential Editor** modal page appears. In this page, you can define the new SNMP credential. To define the new credential, supply values in the following fields:

The screenshot shows a window titled "Credential Editor" with a close button (X) in the top right corner. Below the title bar is a blue header with the text "Create New SNMP Credential" and a "Reset" button on the right. The main content area is divided into three sections:

- Basic Settings:** Contains a "Profile Name" text input field, an "SNMP Version" dropdown menu (set to "[SNMP V2]"), a "Port" text input field (set to "161"), a "Timeout(ms)" text input field (set to "1500"), and a "Retries" text input field (set to "1").
- SNMP V1/V2 Settings:** Contains two text input fields: "SNMP Community (Read-Only)" and "SNMP Community (Read/Write)".
- SNMP V3 Settings:** Contains a "Security Name" text input field, a "Security Passphrase" text input field, an "Authentication Protocol" dropdown menu (set to "[MD5]"), a "Security Level" dropdown menu (set to "[Authentication Only]"), an "SNMP v3 Engine ID" text input field, a "Context Name" text input field, a "Privacy Protocol" dropdown menu (set to "[DES]"), and a "Privacy Protocol Pass Phrase" text input field.

A "Save" button is located at the bottom center of the form.

- **Profile Name.** Name of the credential. Can be any combination of alphanumeric characters. This field is required.
- **SNMP Version.** SNMP version. Select *SNMP V2*. This field is required.
- **Port.** The port SL1 will use to communicate with the external device or application. The default value is *161*. This field is required.
- **Timeout (ms).** Time, in milliseconds, after which SL1 will stop trying to communicate with the SNMP device. The default value is *1500*. This field is required.
- **Retries.** Number of times SL1 will try to authenticate and communicate with the external device. The default value is *1*. This field is required.

SNMP V1/V2 Settings

These fields appear if you selected **SNMP V1** or **SNMP V2** in the **SNMP Version** field. Otherwise, these fields are grayed out.

- **SNMP Community (Read Only)**. The SNMP community string (password) required for read-only access of SNMP data on the remote device or application. For read-only SNMP V2 credentials, this field is required and you should leave the **SNMP Community (Read/Write)** field blank. Enter the same value as you entered for **rocommunity** in the snmpd.conf file.
- **SNMP Community (Read/Write)**. The SNMP community string (password) required for read and write access of SNMP data on the remote device or application. For read/write SNMP V2 credentials, this field is required and you should leave the **SNMP Community (Read Only)** field blank. Enter the same value as you entered for **rwcommunity** in the snmpd.conf file.

4. Click the **[Save]** button to save the new SNMP credential.

NOTE: When you define an SNMP Credential, the credential will automatically be aligned with the organization(s) you are a member of.

Creating SNMPv3 Credentials

The example snmpd.conf file for SNMPv3 provides both Read Only and Read/Write access to your Linux system from SL1. You must therefore define two new SNMPv3 credentials (one for read-only access and one for read/write access) in SL1, so SL1 can successfully communicate with your Linux system. To do this:

1. Go to the **Credential Management** page (System > Manage > Credentials).
2. In the **Credential Management** page, click the **[Actions]** menu. Select **Create SNMP Credential**.

Credential Management Credentials Found [62]													Actions	Reset	Guide
Profile Name	Organization	RO Use	RW Use	DA Use	Type	Credential User	Host	Port	Timeout (ms)	ID	Last				
1. Amazon Web Services Credential	System	--	--	--	SOAP/XML Host	[AWS Account Access]	example.com	80	2000	1	2015-05-16				
2. Azure Credential - SOAP/XML	[all orgs]	--	--	--	SOAP/XML Host	<<AD_USER>>	login.windows.net	443	60000	60	2015-05-14				
3. Azure Credential - SSH/Key	[all orgs]	--	--	--	SSH/Key	<<SUBSCRIPTION_ID_H	%D	22	180000	59	2015-05-14				
4. Cisco SNMPv2 - Example	[all orgs]	--	--	--	SNMP	--	--	161	1500	3	2015-05-14				
5. Cisco SNMPv3 - Example	[all orgs]	--	--	--	SNMP	[USER_GOES_HERE]	--	161	1500	2	2015-05-14				
6. Cisco ACI	[all orgs]	--	--	126	Basic/Snippet	admin	173.36.219.46	443	0	62	2015-05-14 15:05:24	em7admin			
7. Cisco ACI Credential	[all orgs]	--	--	--	Basic/Snippet	admin	198.18.133.200	443	0	61	2015-05-14 14:32:20	em7admin			
8. Cloudark - Example	[all orgs]	--	--	--	Basic/Snippet	[SECURITY KEY GOES	127.0.0.1	443	5000	9	2015-05-14 11:25:31	em7admin			
9. CUCM PerfmonService 8.0 Example	[all orgs]	--	--	--	SOAP/XML Host	--	--	8443	2000	4	2015-05-14 11:28:12	em7admin			
10. EM7 Central Database	[all orgs]	--	--	--	Database	root	localhost	7706	0	51	2015-05-14 11:28:41	em7admin			
11. EM7 Collector Database	[all orgs]	--	--	--	Database	root	%D	7707	0	14	2015-05-14 11:25:43	em7admin			
12. EM7 DB	[all orgs]	--	--	--	Database	root	%D	7706	0	35	2015-05-14 11:28:32	em7admin			
13. EM7 DB - DB Info	[all orgs]	--	--	--	SOAP/XML Host	root	%D	80	3000	38	2015-05-14 11:28:32	em7admin			
14. EM7 DB - My.cnf	[all orgs]	--	--	--	SOAP/XML Host	root	%D	80	3000	37	2015-05-14 11:28:32	em7admin			
15. EM7 DB - Sls.conf	[all orgs]	--	--	--	SOAP/XML Host	root	%D	80	3000	36	2015-05-14 11:28:32	em7admin			
16. EM7 Default V2	[all orgs]	--	--	--	SNMP	--	--	161	1500	10	2015-05-14 11:25:42	em7admin			
17. EM7 Default V3	[all orgs]	--	--	--	SNMP	em7defaultv3	--	161	500	11	2015-05-14 11:25:42	em7admin			
18. EMC - Example	[all orgs]	--	--	--	Basic/Snippet	root	%D	443	10000	15	2015-05-14 11:25:47	em7admin			
19. ES/GrIn - Example	[all orgs]	--	--	--	Basic/Snippet	[SECURITY KEY GOES	127.0.0.1	443	5000	16	2015-05-14 11:25:51	em7admin			
20. IP/LA Example	[all orgs]	--	--	--	SNMP	--	--	161	1500	5	2015-05-14 11:25:14	em7admin			
21. LifeSize - Endpoint SNMP	[all orgs]	--	--	--	SNMP	control	--	161	3000	18	2015-05-14 11:25:58	em7admin			
22. LifeSize - Endpoint SSH/CLI	[all orgs]	--	--	--	Basic/Snippet	auto	%D	22	3	17	2015-05-14 11:25:58	em7admin			
23. Local API	[all orgs]	--	--	--	Basic/Snippet	em7admin	10.0.0.190	80	5000	22	2015-05-14 11:28:11	em7admin			
24. NetApp 7-mode	[all orgs]	--	--	--	Basic/Snippet	root	%D	443	3000	24	2015-05-14 11:28:20	em7admin			
25. NetApp w/SSL Option	[all orgs]	--	--	--	SOAP/XML Host	root	%D	443	3000	26	2015-05-14 11:28:20	em7admin			
26. NetApp w/SSL Option Off	[all orgs]	--	--	--	SOAP/XML Host	root	%D	443	10000	25	2015-05-14 11:28:20	em7admin			
27. Nexus netconf	[all orgs]	--	--	--	Basic/Snippet	root	%D	22	10000	6	2015-05-14 11:25:16	em7admin			
28. Nexus snmp	[all orgs]	--	--	--	SNMP	--	--	161	10000	7	2015-05-14 11:25:16	em7admin			
29. Polycm - Advanced	[all orgs]	--	--	--	SOAP/XML Host	admin	%D	80	20000	28	2015-05-14 11:28:24	em7admin			
30. Polycm - CDR	[all orgs]	--	--	--	SOAP/XML Host	admin	%D	80	20000	31	2015-05-14 11:28:24	em7admin			
31. Polycm - Interface	[all orgs]	--	--	--	SOAP/XML Host	admin	%D	80	20000	29	2015-05-14 11:28:24	em7admin			

- The **Credential Editor** modal page appears. In this page, you can define the new SNMP credential. To define the new read-only credential, supply values in the following fields:

The screenshot shows the 'Credential Editor' window with the following fields and values:

- Basic Settings:**
 - Profile Name:
 - SNMP Version:
 - Port:
 - Timeout(ms):
 - Retries:
- SNMP V1/V2 Settings:**
 - SNMP Community (Read-Only):
 - SNMP Community (Read/Write):
- SNMP V3 Settings:**
 - Security Name:
 - Security Passphrase:
 - Authentication Protocol:
 - Security Level:
 - SNMP v3 Engine ID:
 - Context Name:
 - Privacy Protocol:
 - Privacy Protocol Pass Phrase:

Buttons: 'Reset' (top right), 'Save' (bottom center).

- **Profile Name.** Name of the read-only credential. Can be any combination of alphanumeric characters. This field is required.
- **SNMP Version.** SNMP version. Select *SNMP V3*. This field is required.
- **Port.** The port SL1 will use to communicate with the external device or application. The default value is *161*. This field is required.
- **Timeout (ms).** Time, in milliseconds, after which SL1 will stop trying to communicate with the SNMP device. The default value is *1500*. This field is required.
- **Retries.** Number of times SL1 will try to authenticate and communicate with the external device. The default value is *1*. This field is required.

SNMP V3 Settings

These fields appear if you selected *SNMP V3* in the **SNMP Version** field. Otherwise, these fields are grayed out.

- **Security Name.** Name used for read-only SNMP authentication. This field is required. Enter the same value as you entered for **rouser** in the `snmpd.conf` file.
- **Security Passphrase.** Password used to authenticate the credential. This value must contain at least 8 characters. This value is required if you want to use a **Security Level** that includes authentication. Enter the same value as you entered in the `createuser` line in the `snmpd.conf` file.

- **Authentication Protocol.** Select an authentication algorithm for the credential. Choices are MD5 or SHA. This field is required. Select *SHA*.

NOTE: If your SL1 system is FIPS-compliant, MD5 authentication for SNMP will fail. FIPS-compliant SL1 systems require SHA authentication for SNMP.

- **Security Level.** Specifies the combination of security features for the credentials. This field is required. Choices are:
 - *No Authentication / No Encryption.*
 - *Authentication Only.* This is the default value.
 - *Authentication and Encryption.* This is the option specified in the example `snmpd.conf` file.
- **SNMP v3 Engine ID.** The unique engine ID for the SNMP agent you want to communicate with. (SNMPv3 authentication and encryption keys are generated based on the associated passwords and the engine ID.) This field is optional.
- **Context Name.** A context is a mechanism within SNMPv3 (and AgentX) that allows you to use parallel versions of the same MIB objects. For example, one version of a MIB might be associated with SNMP Version 2 and another version of the same MIB might be associated with SNMP Version 3. For SNMP Version 3, specify the context name in this field. This field is optional.
- **Privacy Protocol.** The privacy service encryption and decryption algorithm. Choices are *DES* or *AES*. The default value is *DES*. This field is required. The example `snmpd.conf` file specifies *DES*.
- **Privacy Protocol Passphrase.** Privacy password for the credential. This field is optional.



4. Click the **[Save]** button to save the new read-only SNMPv3 credential.
5. Repeat steps 1-4 to also create the new read/write SNMPv3 credential, updating the field values as needed.

NOTE: When you define an SNMP Credential, the credential will automatically be aligned with the organization(s) you are a member of.

Configuring Syslog for Linux

Overview

Use the following menu options to navigate the SL1 user interface:

- To view a pop-out list of menu options, click the menu icon .
- To view a page containing all of the menu options, click the Advanced menu icon ().

The following sections describe how to configure syslog for Linux:

This chapter covers the following topics:

What is Syslog?	50
Configuring Syslog for Linux	51

What is Syslog?

Syslog is a protocol and utility for capturing and logging system information. This system information can be stored locally, remotely, or both. syslog allows a server to forward log messages over the network to SL1. SL1 then uses these messages to monitor the health of the server and trigger events (if necessary).

Because the syslog utility is mature and widely-used, there is an array of commercial and open source implementations. This chapter provides only a basic outline of how to configure syslog to send messages to SL1.

Entries in a syslog can include the following severity descriptions:

Severity	Description
0 Emergency:	System is unusable. A "panic" condition. Notify all technical staff. Affects multiple servers, applications, systems, or sites. For example, an outage caused by an earthquake.
1 Alert	Failure in primary system. Immediate action is required. Notify appropriate staff. Example would be "loss of backup ISP connection".
2 Critical	Failure in primary system. Immediate action is required before problem escalates to "alert". For example, "loss of primary ISP connection".
3 Error	Non-urgent failure. Action is required but not urgent. These messages should be relayed to appropriate support staff for resolution.
4 Warning	Indication that an error is about to occur. Action is required but not immediately. For example, "file system is 85% full".
5 Notice	Normal but significant condition. No immediate action required. Events that are unusual but are not considered error conditions. Should be examined to spot potential problems.
6 Informational	Normal operational messages. No action required. These may be harvested for reporting, measuring through-put, etc.
7 Debug	Information that is useful to developers for debugging the application; not useful during operations.

Configuring Syslog for Linux

To configure your Linux server to send syslogs to SL1, you must edit the file `/etc/syslog.conf`.

1. Before editing the `/etc/syslog.conf` file, ensure that syslog is enabled. To do this, open a shell session, log in as root, and enter the following at the command prompt:

```
service syslog status
```

2. Backup the existing `/etc/syslog.conf` file. To do this, open a shell session, log in as root, and enter the following at the command prompt:

```
cp /etc/syslog.conf /etc/syslog.orig
```

3. Use your favorite editor to edit the `/etc/syslog.conf` file and add the following line:

- If you are using an All-In-One Appliance, use the IP address of the All-In-One Appliance.
- If you are using a Distributed System and the Collector Group that will monitor your device includes a Message Collector, use the IP address of the Message Collector.
- If you are using a Distributed System and the Collector Group that will monitor your device includes a single Data Collector that performs the message collection function, use the IP address of the Data Collector.

*.err;local0.debug;daemon.notice;mail.crit @<IP_OF_SCIENCELOGIC_APPLIANCE>

NOTE: syslog includes many facilities. The facilities referenced above are merely a starting point as suggested by ScienceLogic.

4. After you edit the syslog.conf file, you must **restart the syslog service**. To do this, open a shell session and enter the following at the command prompt:

```
service syslog restart
```

5. To test sending syslog messages to SL1, open an shell session and enter the following at the command prompt:

```
logger -p local0.debug "Test Debug Message to SL1"
```

6. To see if the message was sent to SL1, check:

- on the Linux device, the file /var/log/messages
- in SL1, the device logs of the corresponding Linux device.

NOTE: By default, SL1 includes multiple event policies based on syslog messages. ScienceLogic recommends that you review these policies to ensure that they suit your business needs. To view these policies, go to Registry > Events > Event Manager. Use the sort and filter tools to view all policies of type "syslog." From the same page, you can edit these event policies or create your own event policies based on syslog messages. For more information on event policies, see the manual on **Events**.

Appendix



A

Collection Objects

Overview

This appendix defines the different collection objects in the *Linux Base Pack PowerPack*.

Use the following menu options to navigate the SL1 user interface:

- To view a pop-out list of menu options, click the menu icon (.
- To view a page containing all of the menu options, click the Advanced menu icon ().

This appendix covers the following topics:

This chapter covers the following topics:

Collection Objects in Linux Dynamic Applications B

Collection Objects in Linux Dynamic Applications

The following tables list the Collection Objects included in those Dynamic Applications and the Linux commands used by each of those objects. You can use these commands to grant or restrict access to certain data types on the user account you will use to monitor your Linux devices.

The following table is a list of configuration and performance Dynamic Applications in the PowerPack:

Dynamic Application	Collection Object	Linux Command
Linux: Configuration Discovery		Determines if a device is a Linux system before discovery in SL1. If the device is not a Linux system, it will not be discovered.
Linux: CPU Configuration	All	<code>cat /proc/cpuinfo/lscpu</code>
	Server Product Name	<code>cat /sys/devices/virtual/dmi/id/product_name</code>
Linux: CPU Cores Performance	All	<code>cat /proc/stat</code>
Linux: CPU Performance	All	<code>cat /proc/stat</code>
Linux: Disk IOPs Performance	All	<code>cat /proc/diskstats</code>
Linux: Hardware Configuration	All	<code>sudo dmidecode -qt 1,2,3</code>
Linux: ICMP Performance	All	<code>cat /proc/net/snmp</code>
Linux: Inode Performance	All	<code>df -iPT</code>
Linux: Memory Performance	All	<code>cat /proc/meminfo</code>
Linux: Route Table Configuration	All	<code>ip route 2>/dev/null /sbin/ip route 2>/dev/null</code>

Dynamic Application	Collection Object	Linux Command	
Linux: System Configuration	Kernel Version	<code>cat /proc/sys/kernel/osrelease</code>	
	Distribution Genus	<code>cat 2> /dev/null /etc/os-release grep PRETTY_NAME cat 2> /dev/null /etc/redhat-release cat 2> /dev/null /etc/lsb-release grep DISTRIB_DESCRIPTION cat /etc/SuSE-release</code>	
	Host Name	<code>cat /proc/sys/kernel/hostname</code>	
	Distribution Release	<code>cat /etc/os-release grep PRETTY_NAME</code>	
	AppDynamics Host Name IP Address	<code>hostname=\$(cat /proc/sys/kernel/hostname) && echo \$hostname" "<silos:ip></code>	
	AppDynamics Namespace	<code>echo "appdynamics/ns"</code>	
	Architecture Type	<code>uname -a</code>	
	Compiler	<code>cat /proc/version</code>	
	Domain Name	<code>cat /proc/sys/kernel/domainname && cat /proc/sys/kernel/hostname</code>	
	Dynatrace Hostname	<code>cat /proc/sys/kernel/hostname</code>	
	Dynatrace Namespace	<code>echo "dynatrace/physical/ns"</code>	
	New Relic Hostname	<code>cat /proc/sys/kernel/hostname</code>	
	New Relic Namespace	<code>echo "newrelic/server/ns"</code>	
	Release Date	<code>cat /proc/sys/kernel/version</code>	
	SMP Support	<code>cat /proc/sys/kernel/version</code>	
	Time Zone	<code>date "+%Z"</code>	
	Total Physical Memory (MBytes)	<code>cat /proc/meminfo</code>	
	Total Swap Memory (MBytes)	<code>cat /proc/meminfo</code>	
	Linux: System Load Performance	All	<code>cat /proc/loadavg</code>
		CPU	<code>lscpu</code>
Linux: TCP Performance	All	<code>cat /proc/net/snmp</code>	

Dynamic Application	Collection Object	Linux Command
Linux: UDP Performance	All	<code>cat /proc/net/snmp</code>
Linux: UDP Services Configuration	All	<code>ss -lwan 2>/dev/null /usr/sbin/ss -lwan 2>/dev/null /bin/ss -lwan 2>/dev/null</code>
Linux: Zombie Process	All	<code>ps aux grep Z</code>

The following table is a list of internal collection inventory and performance Dynamic Applications in the PowerPack:

Dynamic Application	Collection Object	Linux Command
Linux: IC Detail	All	Internal Collection that consumes data stored by the "Linux: ICDA Cache" Dynamic Application.
Linux: IC Filesystem Inventory	All	Internal Collection that consumes data stored by the "Linux: ICDA Cache" Dynamic Application.
Linux: IC Filesystem Performance	All	Internal Collection that consumes data stored by the "Linux: ICDA Cache" Dynamic Application.
Linux: IC Interface Inventory	All	Internal Collection that consumes data stored by the "Linux: ICDA Interface Cache" Dynamic Application.
Linux: IC Interface Performance	All	Internal Collection that consumes data stored by the "Linux: ICDA Interface Cache" Dynamic Application.
Linux: IC Port Performance	All	Internal Collection that consumes data stored by the "Linux: ICDA Cache" Dynamic Application.
Linux: IC Process Inventory	All	Internal Collection that consumes data stored by the "Linux: ICDA Interface Cache" Dynamic Application.
Linux: IC Process Performance	All	Internal Collection that consumes data stored by the "Linux: ICDA Interface Cache" Dynamic Application.

Dynamic Application	Collection Object	Linux Command
Linux: ICDA Cache	Filesystem	<code>df -kPT</code>
	Hardware Config Product Name	<code>cat /sys/devices/virtual/dmi/id/product_name</code>
	Network Interfaces	<code>for x in `ls -ld /sys/class/net/* grep -v '/virtual/' rev cut -d/-f1 rev`; do echo \$x ': ' `if ["\$x" = "lo"]; then echo "0"; else cat /sys/class/net/\$x/speed 2>/dev/null echo "0" ; fi`; done;</code>
	Network Interfaces IP Address	<code>/sbin/ip addr</code>
	Uptime	<code>cat /proc/uptime</code>
Linux: ICDA Interface Cache	Interface Stats	<code>/sbin/ip -s -s link</code>
	Process	<code>ps aux</code>
	TCP Listening Ports	<code>echo ".";ss -ltn 2</dev/null /usr/sbin/ss -ltn 2</dev/null /bin/ss -ltn 2>/dev/null</code>

© 2003 - 2023, ScienceLogic, Inc.

All rights reserved.

LIMITATION OF LIABILITY AND GENERAL DISCLAIMER

ALL INFORMATION AVAILABLE IN THIS GUIDE IS PROVIDED "AS IS," WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED. SCIENCELOGIC™ AND ITS SUPPLIERS DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT.

Although ScienceLogic™ has attempted to provide accurate information on this Site, information on this Site may contain inadvertent technical inaccuracies or typographical errors, and ScienceLogic™ assumes no responsibility for the accuracy of the information. Information may be changed or updated without notice. ScienceLogic™ may also make improvements and / or changes in the products or services described in this Site at any time without notice.

Copyrights and Trademarks

ScienceLogic, the ScienceLogic logo, and EM7 are trademarks of ScienceLogic, Inc. in the United States, other countries, or both.

Below is a list of trademarks and service marks that should be credited to ScienceLogic, Inc. The ® and ™ symbols reflect the trademark registration status in the U.S. Patent and Trademark Office and may not be appropriate for materials to be distributed outside the United States.

- ScienceLogic™
- EM7™ and em7™
- Simplify IT™
- Dynamic Application™
- Relational Infrastructure Management™

The absence of a product or service name, slogan or logo from this list does not constitute a waiver of ScienceLogic's trademark or other intellectual property rights concerning that name, slogan, or logo.

Please note that laws concerning use of trademarks or product names vary by country. Always consult a local attorney for additional guidance.

Other

If any provision of this agreement shall be unlawful, void, or for any reason unenforceable, then that provision shall be deemed severable from this agreement and shall not affect the validity and enforceability of any remaining provisions. This is the entire agreement between the parties relating to the matters contained herein.

In the U.S. and other jurisdictions, trademark owners have a duty to police the use of their marks. Therefore, if you become aware of any improper use of ScienceLogic Trademarks, including infringement or counterfeiting by third parties, report them to Science Logic's legal department immediately. Report as much detail as possible about the misuse, including the name of the party, contact information, and copies or photographs of the potential misuse to: legal@sciencelogic.com. For more information, see <https://sciencelogic.com/company/legal>.

ScienceLogic

800-SCI-LOGIC (1-800-724-5644)

International: +1-703-354-1010