Monitoring Linux Systems with SSH

Linux Base Pack version 113

Table of Contents

Chapter 1

Introduction

Overview

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

NOTE: The "Linux Base Pack" PowerPack version 109 release has been removed from the ScienceLogic Support portal and replaced with the version 110 release.

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 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:

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 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.

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
- A device template for discovering Linux devices

NOTE: The "Linux Base Pack" PowerPack is equipped with an alert to detect stale file systems. If you receive an exit code 124 when running the command `timeout 3 df -kPT`, an alert will be triggered to warn you of a stale file system.

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 section on ***Global Settings***.

NOTE: For details on upgrading SL1, see the relevant ***SL1 Platform Release Notes***.

To download and install the PowerPack:

1. Search for and download the PowerPack from the **PowerPacks** page (Product Downloads > PowerPacks & SyncPacks) at the ScienceLogic Support Site.
2. In SL1, go to the **PowerPacks** page (System > Manage > PowerPacks).
3. Click the **[Actions]** button and choose *Import PowerPack*. The **Import PowerPack** dialog box appears.
4. Click **[Browse]** and navigate to the PowerPack file from step 1.
5. Select the PowerPack file and click **[Import]**. The **PowerPack Installer** modal displays a list of the PowerPack contents.
6. Click **[Install]**. The PowerPack is added to the **PowerPacks** page.

NOTE: If you exit the **PowerPack Installer** modal without installing the imported PowerPack, the imported PowerPack will not appear in the **PowerPacks** 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. Disable all Linux devices by doing one of the following:
   - Go to the **Devices** page, select all Linux devices from the list, click the **[Actions]** button, select *Change Collection State*, and then select *Disable* (toggled off).
   - Go to the **Device Manager** page (Devices > Classic Devices, or Registry > Devices > Device Manager in the classic SL1 user interface), select all Linux devices from the list, click the ***Select Action*** menu, select *Change Collection State*, select *Disable*, and then click **[Go]**.
4. Download the latest version of the "Linux Base Pack" PowerPack from the **PowerPacks** page (Product Downloads > PowerPacks & SyncPacks) at the ScienceLogic Support Site to a local computer.
5. Go to the **PowerPack Manager** page (System > Manage > PowerPacks) in SL1.
6. Click the **[Actions]** menu and choose *Import PowerPack*.
7. When prompted, import the "Linux Base Pack" PowerPack.
8. Click the **[Install]** button. Wait for about five minutes to ensure the virtual environment is created.
9. If you are upgrading from version 110 or earlier, align the ***"Linux: SSH Cache Worker" Dynamic Application***.
10. Re-enable all Linux devices by doing one of the following:
    - Go to the **Devices** page, select all Linux devices from the list, click the **[Actions]** button, select *Change Collection State*, and then select *Enable* (toggled on).
    - Go to the **Device Manager** page (Devices > Classic Devices, or Registry > Devices > Device Manager in the classic SL1 user interface), select all Linux devices from the list, click the ***Select Action*** menu, select *Change Collection State*, select *Enable*, and then click **[Go]**.

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, use SSH to access 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, if you are upgrading from **versions 102, 103, or 104** of the "Linux Base Pack" PowerPack, you must delete some Dynamic Applications that were included in those earlier versions and replaced by other Dynamic Applications in later versions of the PowerPack. If these old Dynamic Applications are left enabled, they can drastically reduce the number of Linux devices supported by a Data Collector.

To remove these older 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, click the delete 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, the "Linux: Performance Cache" Dynamic Application ***must*** be deleted.

Linux Distributions Supported by the Linux Base Pack PowerPack

The "Linux Base Pack" PowerPack supports the following distributions:

Distribution	Supported Versions
Ubuntu	23 22 20
CentOS	8 7
Red Hat Linux Enterprise	9 8 7
Oracle Linux Server	9 8

Distribution	Supported Versions
	7
Debian GNU Linux	12 11 10 9
Fedora Server	39 38 37 36 35
Amazon Linux	Amazon Linux 2 Amazon Linux
SUSE Linux Enterprise Server	15 12
Rocky Linux	9 8

Chapter

2

Monitoring Linux with SSH

Overview

This section describes how to configure and discover Linux devices for monitoring by SL1 using 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:

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
- Username with an SSH key or a username with a password 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 425 devices per Data Collector.

NOTE: The PowerPack supports the following ciphers:

(Host-key algorithms): `ssh-ed25519`,`ecdsa-sha2-nistp521`, `ecdsa-sha2-nistp384`, `ecdsa-sha2-nistp256`, `rsasha2-512`, `rsa-sha2-256`. SSH-RSA and SSH-DSS are not supported.

(MACs): `hmac-sha2-256-etm@openssh.com`, `hmac-sha2-512-etm@openssh.com`, `hmac-sha2-256`, `hmac-sha2-512`

(KexAlgorithms): `curve25519-sha256`, `curve25519-sha256@libssh.org`, `ecdh-sha2-nistp256`, `ecdh-sha2-nistp384`, `ecdh-sha2-nistp521`, `diffie-hellman-group18-sha512`, `diffie-hellman-group16-sha512`, `diffie-hellman-group-exchange-sha256`, `diffie-hellman-group14-256`

NOTE: By default, the "Linux: Configuration Cache (Discovery)" snippet code is disabled in the "Linux: Configuration Discovery" Dynamic Application. If you want to perform alignment and classification using the run book action policy, you must enable the "Linux: Configuration Cache (Discovery)" snippet code before running the discovery process. This ensures that the "Linux: Configuration Discovery" Dynamic Application aligns with the devices automatically. ScienceLogic recommends that you use a template to align the Dynamic Applications.

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.

To monitor Linux devices with an IPv6 address in SL1 versions prior to 12.2.4, you must create a soft link in any Data Collector that you plan to monitor a device via an IPv6 address.

To monitor Linux devices via an IPv6 address:

1. Connect by SSH to the Data Collector using your credentials.
2. Run the following command: `sudo ln -s /bin/ping /bin/ping6`

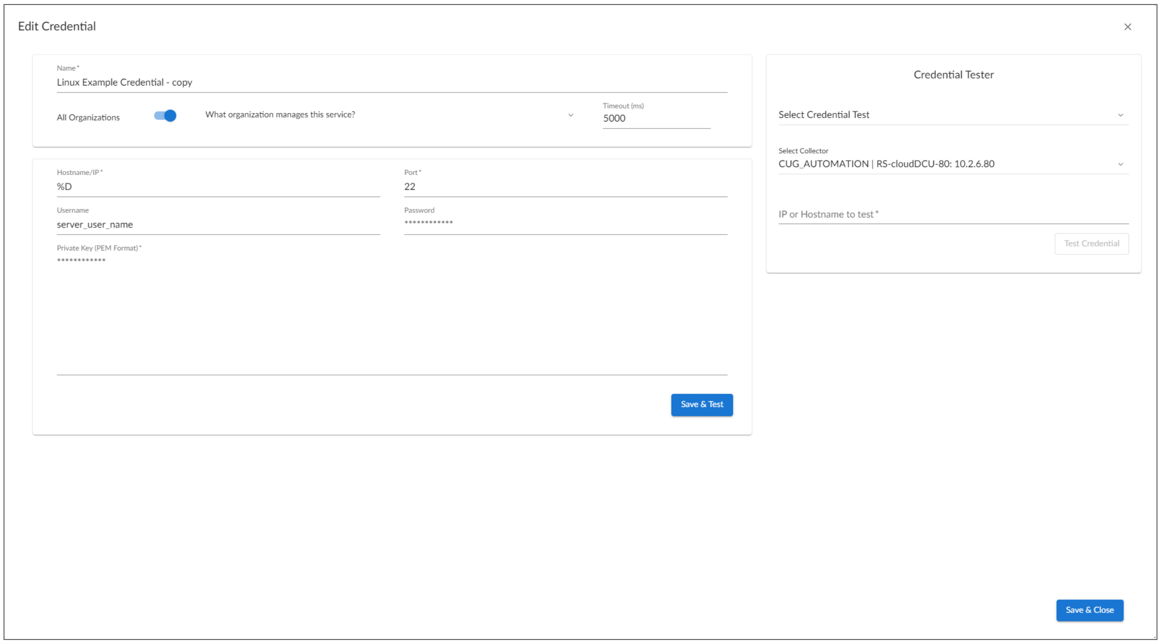
If this command is not applied, the Linux devices with IPv6 start to display the event "Device Failed Availability Check TCP Port (22)" and collection will stop.

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.

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.



Edit Credential

Name *
Linux Example Credential - copy

All Organizations ●

What organization manages this service?

Timeout (ms)
5000

Hostname/IP *
%D

Port *
22

Username
server_user_name

Password
****

Private Key (PEM Format) *
****

Credential Tester

Select Credential Test

Select Collector
CUG_AUTOMATION | RS-cloudDCU-80: 10.2.6.80

IP or Hostname to test *

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

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

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

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 in the classic SL1 user interface:

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:
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:** In the classic user interface, the private key field will accept only RSA formatted / styled keys to be saved. If you want to create SSH credentials with a key in the OpenSSH format, you must do so in the default SL1 user interface.

     > **NOTE:** For PEM Keys with a passphrase, you can use the ***Password*** field to set the passphrase.

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.

Before you begin monitoring with this type of credential, you must configure the following:

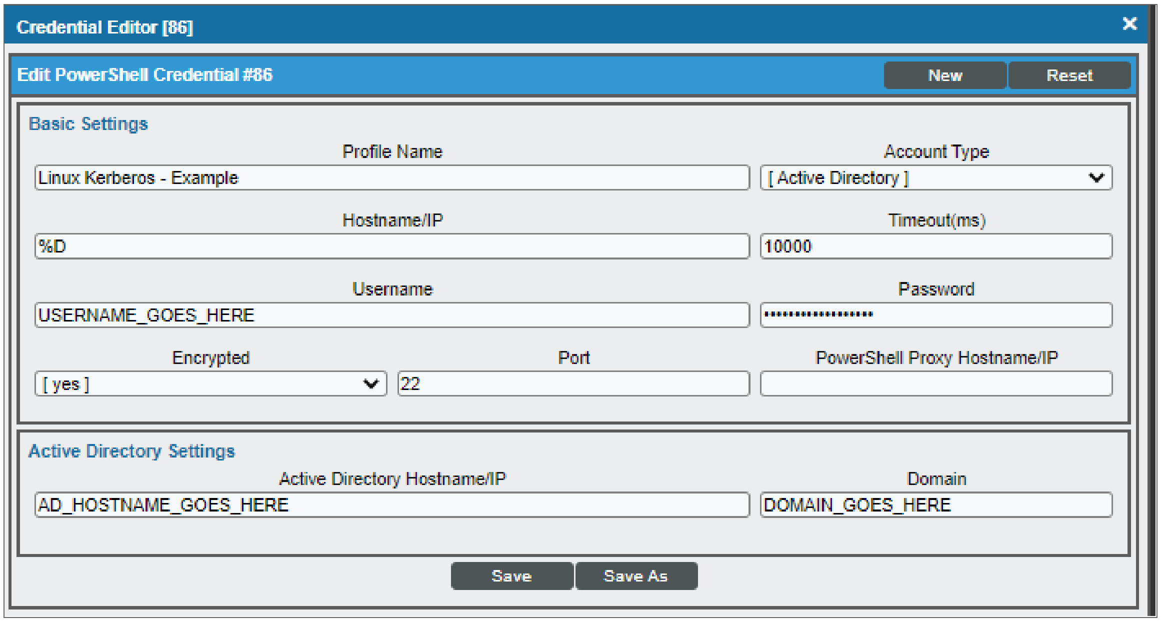
- Active Directory server with the Linux Machines included.
- DNS server with the Linux machines included.
- The GSSAPI option must be enabled in the `/etc/ssh/sshd_config` file of the target Linux machine:

```
GSSAPIAuthentication yes
```

```
GSSAPICleanupCredentials yes # optional
```

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:



Credential Editor [86]

Edit PowerShell Credential #86

Basic Settings

Profile Name: Linux Kerberos - Example
Account Type: [Active Directory]
Hostname/IP: %D
Timeout(ms): 10000
Username: USERNAME_GOES_HERE
Password: ••••••••••••••••••
Encrypted: [yes]
Port: 22
PowerShell Proxy Hostname/IP:

Active Directory Settings

Active Directory Hostname/IP: AD_HOSTNAME_GOES_HERE
Domain: DOMAIN_GOES_HERE

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.

> **NOTE:** If the option `use_fully_qualified_names` is enabled in the target Linux machine, you must type the username in the credential, including the domain. For example: user@DOMAIN.COM

- ***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]**.

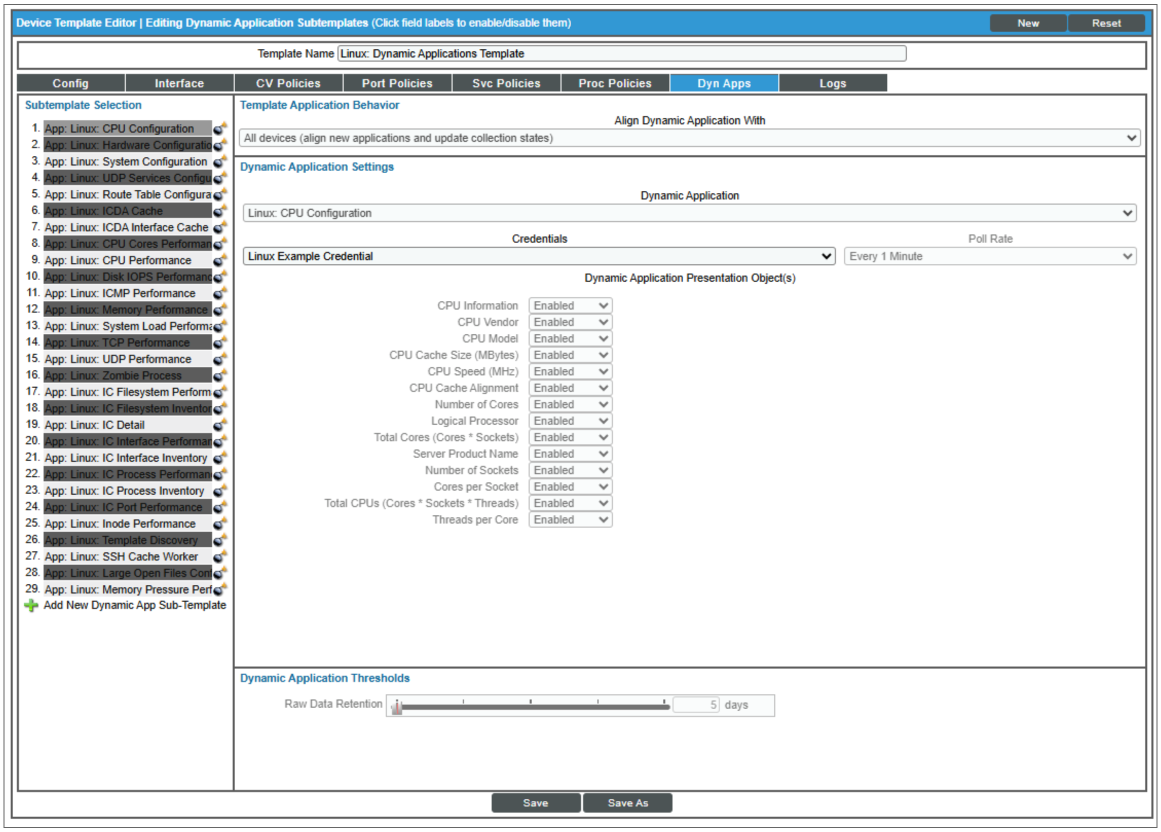
Configuring the Linux Device Template

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 classic SL1 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.



Device Template Editor | Editing Dynamic Application Subtemplates (Click field labels to enable/disable them)
New
Reset
Config
Interface
CV Policies
Port Policies
Svc Policies
Proc Policies
Dyn Apps
Logs
Save
Save As

Template Name: Linux: Dynamic Applications Template

Subtemplate Selection

1. App: Linux: CPU Configuration
2. App: Linux: Hardware Configuratio
3. App: Linux: System Configuration
4. App: Linux: UDP Services Configu
5. App: Linux: Route Table Configura
6. App: Linux: ICDA Cache
7. App: Linux: ICDA Interface Cache
8. App: Linux: CPU Cores Performan
9. App: Linux: CPU Performance
10. App: Linux: Disk IOPS Performanc
11. App: Linux: ICMP Performance
12. App: Linux: Memory Performance
13. App: Linux: System Load Performa
14. App: Linux: TCP Performance
15. App: Linux: UDP Performance
16. App: Linux: Zombie Process
17. App: Linux: IC Filesystem Perform
18. App: Linux: IC Filesystem Inventor
19. App: Linux: IC Detail
20. App: Linux: IC Interface Performan
21. App: Linux: IC Interface Inventory
22. App: Linux: IC Process Performan
23. App: Linux: IC Process Inventory
24. App: Linux: IC Port Performance
25. App: Linux: Inode Performance
26. App: Linux: Template Discovery
27. App: Linux: SSH Cache Worker
28. App: Linux: Large Open Files Conf
29. App: Linux: Memory Pressure Perf

Add New Dynamic App Sub-Template

Template Application Behavior

Align Dynamic Application With: All devices (align new applications and update collection states)

Dynamic Application Settings

Dynamic Application: Linux: CPU Configuration

Credentials: Linux Example Credential

Poll Rate: Every 1 Minute

Dynamic Application Presentation Object(s)

CPU Information	Enabled
CPU Vendor	Enabled
CPU Model	Enabled
CPU Cache Size (MBytes)	Enabled
CPU Speed (MHz)	Enabled
CPU Cache Alignment	Enabled
Number of Cores	Enabled
Logical Processor	Enabled
Total Cores (Cores * Sockets)	Enabled
Server Product Name	Enabled
Number of Sockets	Enabled
Cores per Socket	Enabled
Total CPUs (Cores * Sockets * Threads)	Enabled
Threads per Core	Enabled

Dynamic Application Thresholds

Raw Data Retention: 5 days

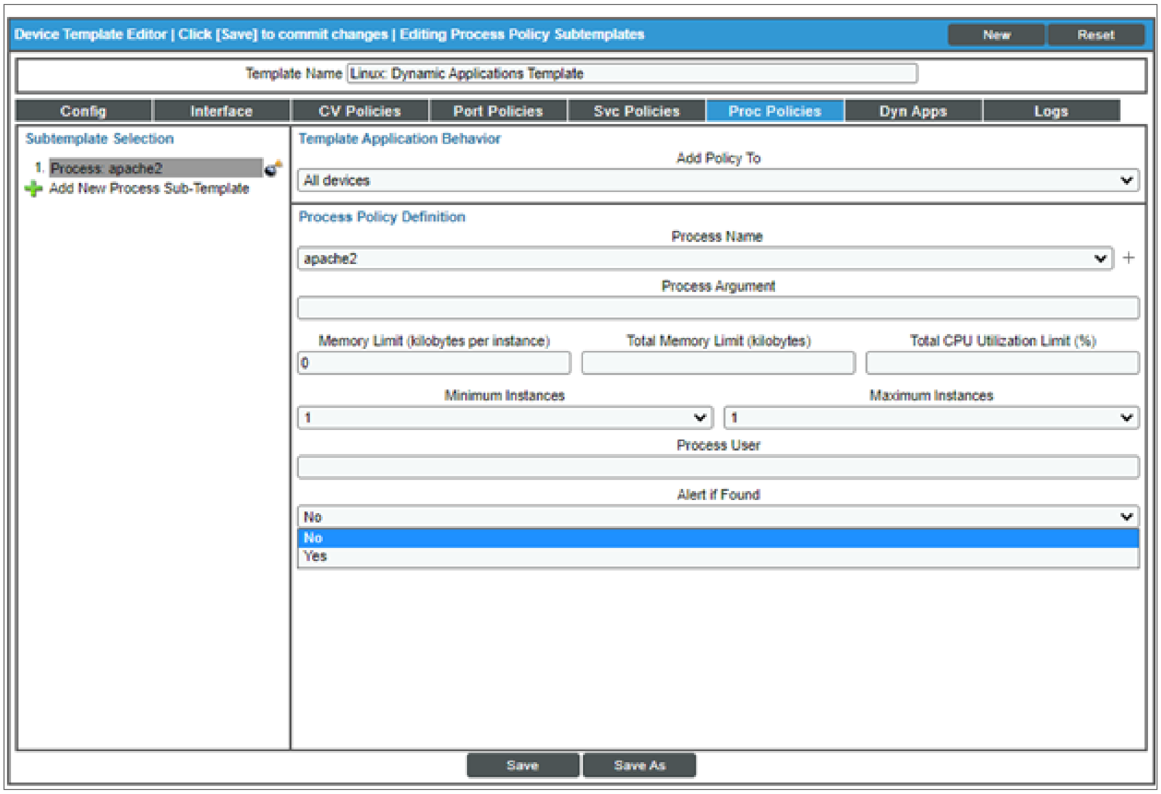
4. Click the "Linux: Template Discovery" Dynamic Application listed in the **Subtemplate Selection** section on the left side of the page and then click the ***Credentials*** field label to enable editing.

5. Select the Linux credential you created in the ***Credentials*** field. Repeat this step for all Dynamic Applications. All Dynamic Applications should be aligned to the credentials you created.

6. Enter a new name for the template in the ***Template Name*** field.

7. Click **[Save As]**.

8. 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.



Template Name: Linux: Dynamic Applications Template

Subtemplate Selection

1. Process: apache2

Add New Process Sub-Template

Template Application Behavior

Add Policy To: All devices

Process Policy Definition

Process Name: apache2

Process Argument:

Memory Limit (kilobytes per instance): 0

Total Memory Limit (kilobytes):

Total CPU Utilization Limit (%):

Minimum Instances: 1

Maximum Instances: 1

Process User:

Alert if Found: No

No

Yes

> 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

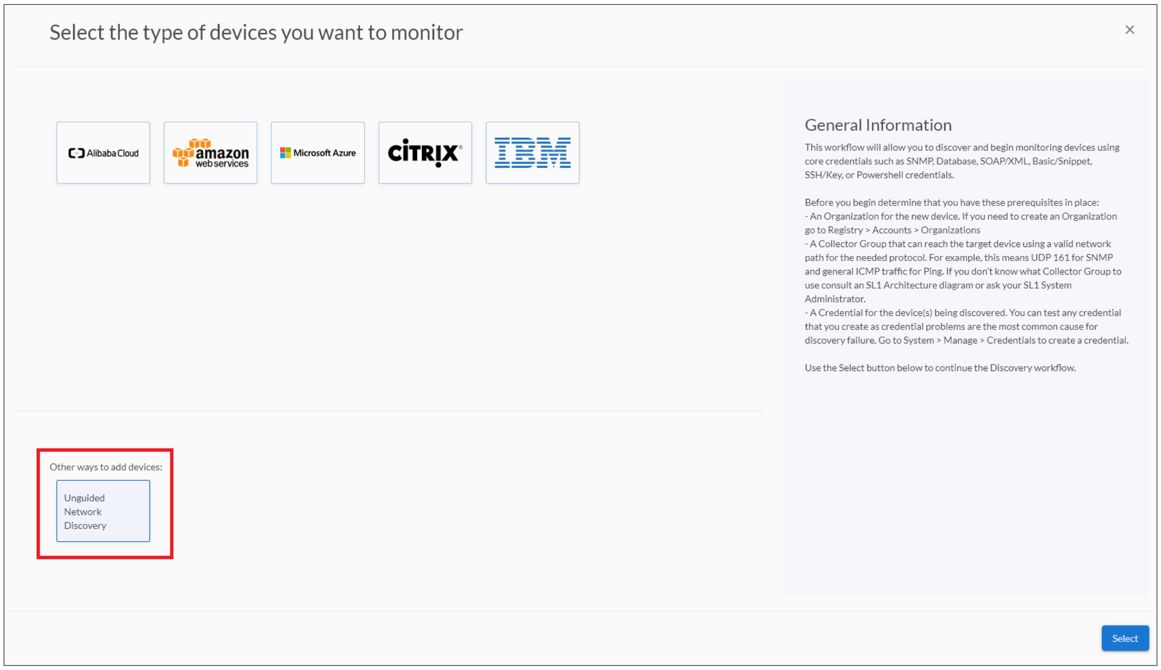
To use the "Linux: IC Port Performance" Dynamic Application, you will need to create a TCP/IP Port monitoring policy after running the discovery session. To create the TCP/IP Policy:

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 drop-down menu.
   - Click **[Save]**.
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:



Select the type of devices you want to monitor

General Information

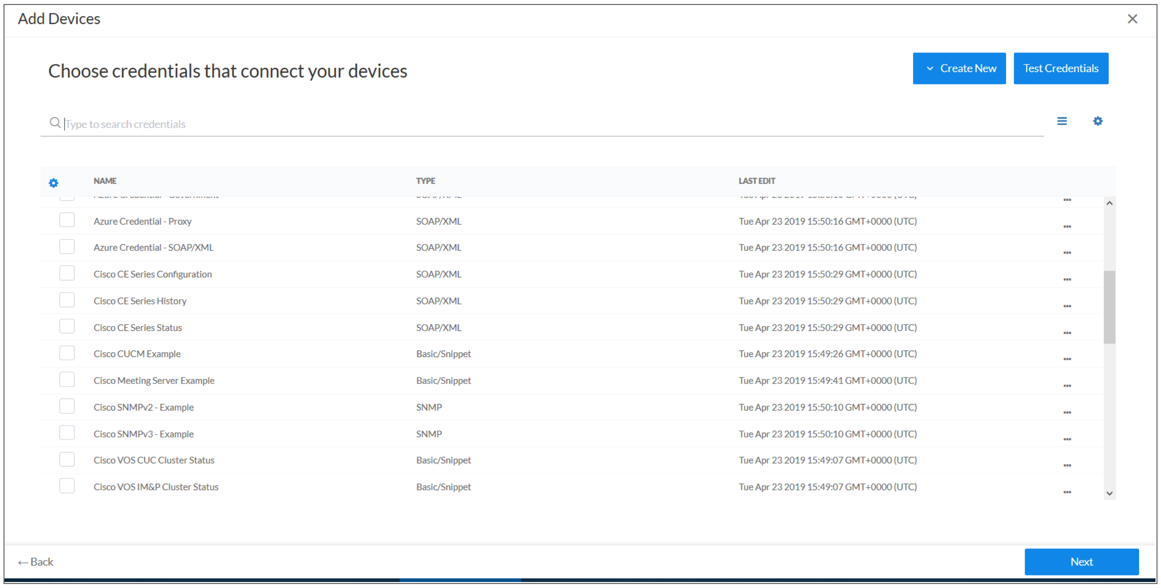
This workflow will allow you to discover and begin monitoring devices using core credentials such as SNMP, Database, SOAP/XML, Basic/Snippet, SSH/Key, or Powershell credentials.

Before you begin determine that you have these prerequisites in place:
- An Organization for the new device. If you need to create an Organization go to Registry > Accounts > Organizations
- A Collector Group that can reach the target device using a valid network path for the needed protocol. For example, this means UDP 161 for SNMP and general ICMP traffic for Ping. If you don't know what Collector Group to use consult an SL1 Architecture diagram or ask your SL1 System Administrator.
- A Credential for the device(s) being discovered. You can test any credential that you create as credential problems are the most common cause for discovery failure. Go to System > Manage > Credentials to create a credential.

Use the Select button below to continue the Discovery workflow.

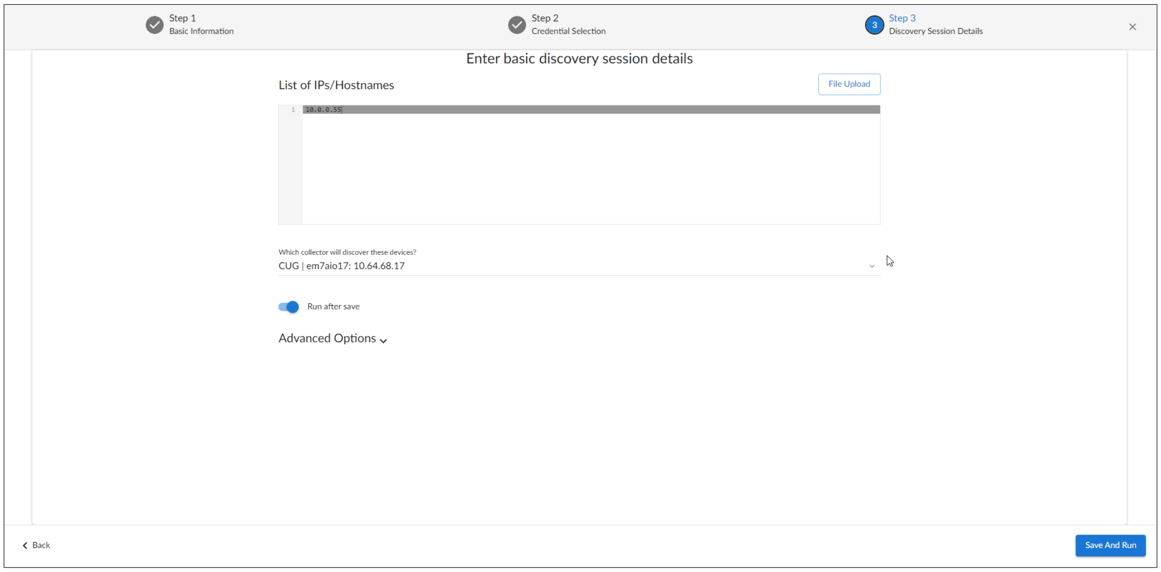
2. Click the **[Unguided Network Discovery]** button. Additional information about the requirements for discovery appears in the **General Information** pane to the right.
3. Click **[Select]**. The **Add Devices** page appears.
4. Complete the following fields:
   - ***Name***. Type a unique name for 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:


Add Devices
Choose credentials that connect your devices
Create New
Test Credentials
Type to search credentials

NAME	TYPE	LAST EDIT
Azure Credential - Proxy	SOAP/XML	Tue Apr 23 2019 15:50:16 GMT+0000 (UTC)
Azure Credential - SOAP/XML	SOAP/XML	Tue Apr 23 2019 15:50:16 GMT+0000 (UTC)
Cisco CE Series Configuration	SOAP/XML	Tue Apr 23 2019 15:50:29 GMT+0000 (UTC)
Cisco CE Series History	SOAP/XML	Tue Apr 23 2019 15:50:29 GMT+0000 (UTC)
Cisco CE Series Status	SOAP/XML	Tue Apr 23 2019 15:50:29 GMT+0000 (UTC)
Cisco CUCM Example	Basic/Snippet	Tue Apr 23 2019 15:49:26 GMT+0000 (UTC)
Cisco Meeting Server Example	Basic/Snippet	Tue Apr 23 2019 15:49:41 GMT+0000 (UTC)
Cisco SNMPv2 - Example	SNMP	Tue Apr 23 2019 15:50:10 GMT+0000 (UTC)
Cisco SNMPv3 - Example	SNMP	Tue Apr 23 2019 15:50:10 GMT+0000 (UTC)
Cisco VOS CUC Cluster Status	Basic/Snippet	Tue Apr 23 2019 15:49:07 GMT+0000 (UTC)
Cisco VOS IM&P Cluster Status	Basic/Snippet	Tue Apr 23 2019 15:49:07 GMT+0000 (UTC)

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:



Enter basic discovery session details

List of IPs/Hostnames

```
10.0.0.55
```

Which collector will discover these devices?
CUG | em7aio17: 10.64.68.17

Run after save

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:

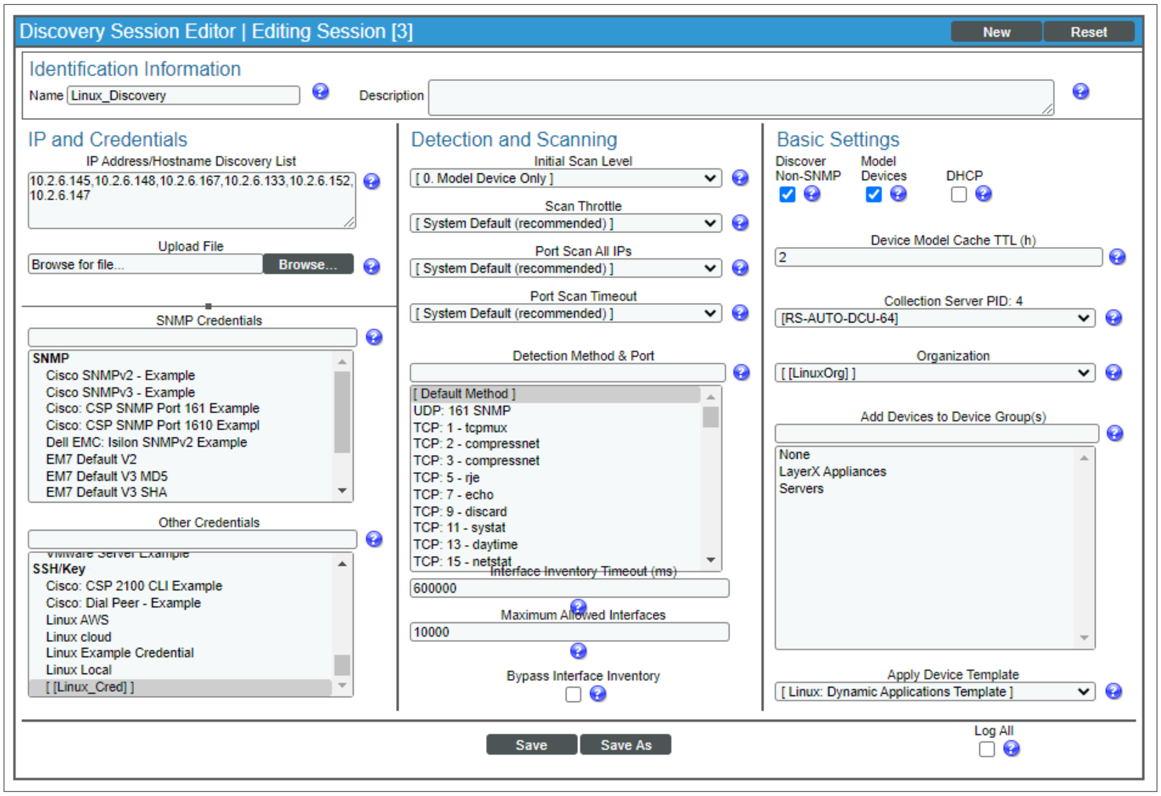
- *Discover Non-SNMP*. Enable this setting.
- *Model Devices*. Enable this setting.
- *Select Device Template*. Select *the device template that you configured*.

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 classic discovery session:

1. Go to the **Discovery Control Panel** page (System > Manage > Classic Discovery or System > Manage > Discovery in the classic user interface).
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:



Discovery Session Editor | Editing Session [3]

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
Cisco SNMPv2 - Example
Cisco SNMPv3 - Example
Cisco: CSP SNMP Port 161 Example
Cisco: CSP SNMP Port 1610 Exampl
Dell EMC: Isilon SNMPv2 Example
EM7 Default V2
EM7 Default V3 MD5
EM7 Default V3 SHA

Other Credentials

VMware Server Example
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: □

Basic Settings

Discover Non-SNMP: ☑

Model Devices: ☑

DHCP: □

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]

Log All: □

- *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 **[Save]** 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*** drop-down 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*** drop-down field to select a new frequency.
4. Click the **[Save]** button.

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

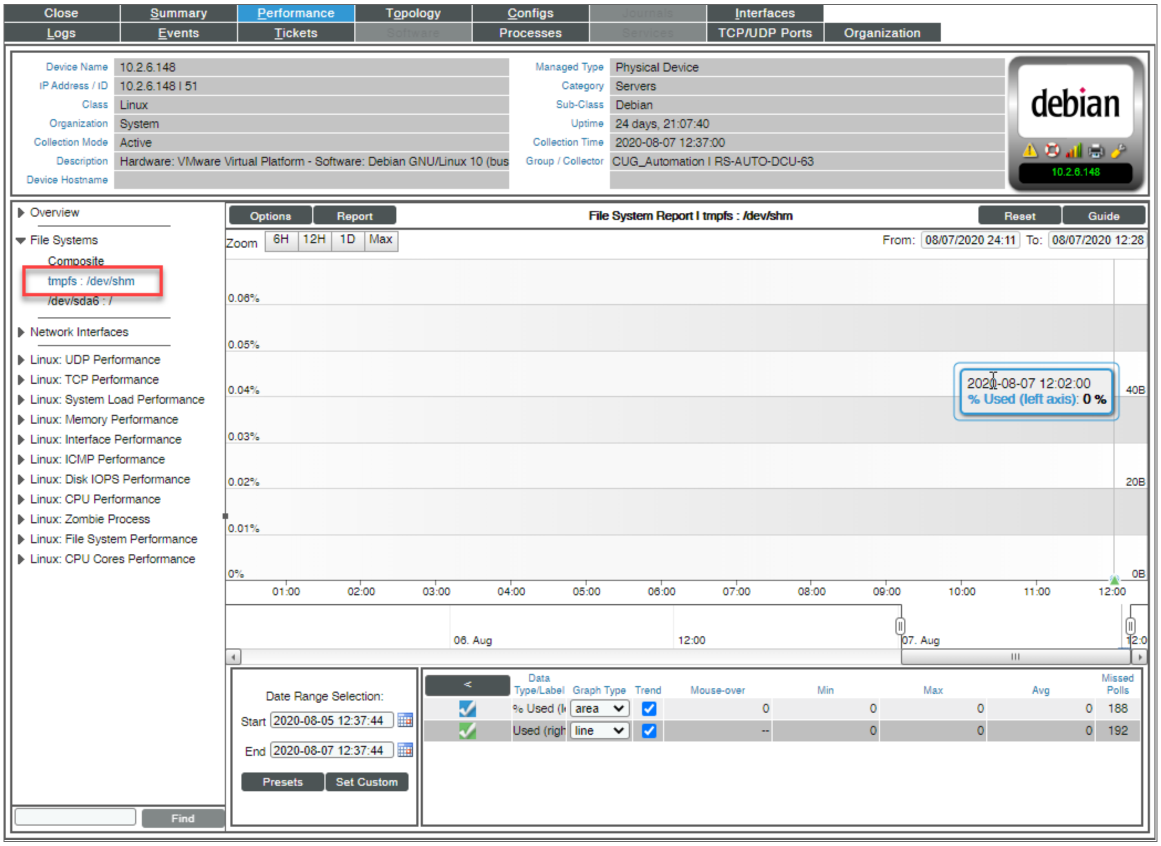
1. Go to the **Process Manager** page (System > Settings > Admin Processes or System > Settings > Processes in the SL1 classic user interface).
2. Search for the "Data Collection: SNMP Detail" process and click its wrench icon (🔧).
3. In the **Process Editor** window, use the ***Frequency*** drop-down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

Unhiding Linux File Systems

In the**Device Hardware** page (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).
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.



Device Name	10.2.6.148
IP Address / ID	10.2.6.148 I 51
Class	Linux
Organization	System
Collection Mode	Active
Description	Hardware: VMware Virtual Platform - Software: Debian GNU/Linux 10 (bus
Device Hostname	

Managed Type	Physical Device
Category	Servers
Sub-Class	Debian
Uptime	24 days, 21:07:40
Collection Time	2020-08-07 12:37:00
Group / Collector	CUG_Automation I RS-AUTO-DCU-63

File System Report I tmpfs : /dev/shm

From: 08/07/2020 24:11 To: 08/07/2020 12:28

2020-08-07 12:02:00
% Used (left axis): 0 %

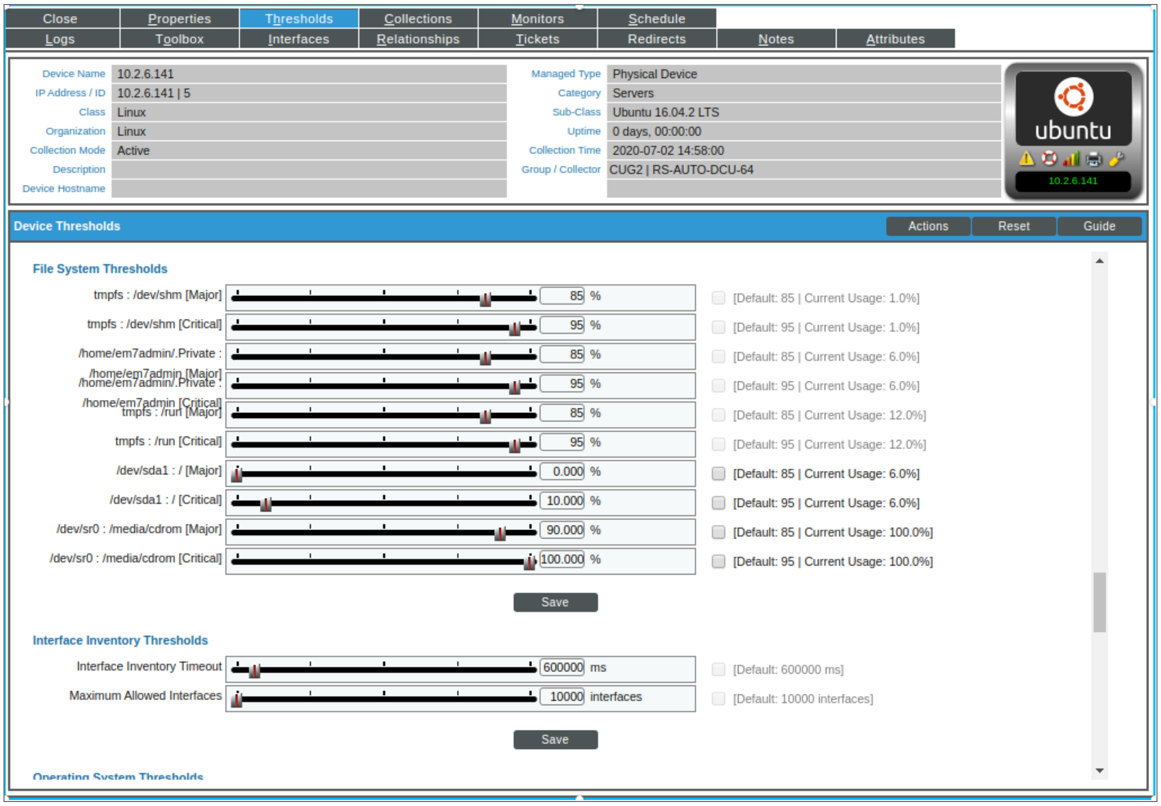
Date Range Selection:
Start 2020-08-05 12:37:44
End 2020-08-07 12:37:44

	Data Type/Label	Graph Type	Trend	Mouse-over	Min	Max	Avg	Missed Polls
☑	% Used (l	area	☑	0	0	0	0	188
☑	Used (righ	line	☑	--	0	0	0	192

Configuring Linux File System Thresholds

To change the file system threshold:

1. Go to the **Device Hardware** page (Devices > Hardware).
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).



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

File System Thresholds

tmpfs : /dev/shm [Major]	85	%	□ [Default: 85 | Current Usage: 1.0%]
tmpfs : /dev/shm [Critical]	95	%	□ [Default: 95 | Current Usage: 1.0%]
/home/em7admin/.Private : /home/em7admin [Major]	85	%	□ [Default: 85 | Current Usage: 6.0%]
/home/em7admin/.Private : /home/em7admin [Critical]	95	%	□ [Default: 95 | Current Usage: 6.0%]
tmpfs : /run [Major]	85	%	□ [Default: 85 | Current Usage: 12.0%]
tmpfs : /run [Critical]	95	%	□ [Default: 95 | Current Usage: 12.0%]
/dev/sda1 : / [Major]	0.000	%	□ [Default: 85 | Current Usage: 6.0%]
/dev/sda1 : / [Critical]	10.000	%	□ [Default: 95 | Current Usage: 6.0%]
/dev/sr0 : /media/cdrom [Major]	90.000	%	□ [Default: 85 | Current Usage: 100.0%]
/dev/sr0 : /media/cdrom [Critical]	100.000	%	□ [Default: 95 | Current Usage: 100.0%]

Interface Inventory Thresholds

Interface Inventory Timeout	600000	ms	□ [Default: 600000 ms]
Maximum Allowed Interfaces	10000	interfaces	□ [Default: 10000 interfaces]

Operating System Thresholds

Aligning the Linux: SSH Cache Worker Dynamic Application

If you are upgrading the "Linux Base Pack" PowerPack from version 110 or earlier, you must align the "Linux: SSH Cache Worker" Dynamic Application to continue monitoring. This Dynamic Application acts as a cache producer for all of the Dynamic Applications in the PowerPack.

Devices discovered through a discovery session with the "Linux: Configuration Discovery" Dynamic Application aligned will automatically align with the "Linux: SSH Cache Worker" Dynamic Application in the next poll. However, if the Dynamic Applications were aligned using a template, you will need to set up the "Linux: SSH Cache Worker" Dynamic Application manually.

To align the "Linux: SSH Cache Worker" Dynamic Application using a template:

1. Create a new template adding the "Linux: SSH Cache Worker" Dynamic Application and credential. To do so:
    - Go to the **Device Template** page (Registry>Devices>Template) and click **[Create]**. The **Device Template Editor** modal opens.
    - In the ***Template Name*** field, enter a template name.
    - On the **[Dyn Apps]** tab, click ***Add New Dynamic App Sub-Template*** in the left **Subtemplate** menu.
    - In the ***Dynamic Application*** drop-down field, select *Linux: SSH Cache Worker*.
    - In the ***Credentials*** drop-down field, select *ssh-cred*.
    - Click **[Save]**.

2. Apply the template to align the "Linux: SSH Cache Worker" Dynamic Application to multiple devices.
   - Go to the **Device Manager** page (Registry>Device Manager) and select the checkboxes of the devices you want to align to the device template.
   - In the ***Select Action*** menu at the bottom of the page, select *MODIFY by Template* and then click **[Go]**. The **Bulk Device Configuration** modal appears.
3. In the ***Template*** field, select the template you created in the previous steps and then click **[Apply]**.
4. Click **[Confirm]** to align the "Linux: SSH Cache Worker" Dynamic Application to your selected devices.

Aligning the "Linux: Large Open Files Configuration" and the "Linux: Memory Pressure Performance" Dynamic Applications

The "Linux: Large Open Files Configuration" and "Linux: Memory Pressure Performance" Dynamic Applications do not use cache to collect data and therefore, they do not use the "Linux: SSH Cache Worker" Dynamic Application.

After updating to version 113 of the PowerPack from an earlier version, these Dynamic Applications do not align automatically. If you want to monitor memory pressure or open files, ScienceLogic recommends you align them manually or use the template.

To align the Dynamic Applications using a template, you must first create a new template adding the "Linux: Large Open Files Configuration " and "Linux: Memory Pressure Performance" Dynamic Applications and a credential.

To add a new template:

1. Go to the **Device Template** page (Registry > Devices > Template) and click **[Create]**. The **Device Template Editor** modal appears.
2. In the ***Template Name*** field, enter a template name.
3. On the **[Dyn Apps]** tab, click ***Add New Dynamic App Sub-Template*** in the left **Subtemplate** menu.
4. In the ***Dynamic Application*** drop-down field, select *Linux: Large Open Files Configuration*.
5. In the ***Credentials*** drop-down field, select *ssh-cred*.
6. Repeat steps 3-5 for the "Linux: Memory Pressure Performance" Dynamic Application.
7. Click **[Save]**.

To apply the template to align the Dynamic Applications to multiple devices:

1. Go to the **Device Manager** page (Registry > Device Manager) and select the checkbox for all the devices you want to align to the device template.
2. In the ***Select Action*** drop-down menu, select *MODIFY by Template* and then click **[Go]**. The **Bulk Device Configuration** modal appears.
3. In the ***Template*** field, select the template you created earlier and then click **[Apply]**.
4. Click **[Confirm]** to align the Dynamic Applications to your selected devices.

Monitoring Large Open Files

To monitor large open files with the "Linux: Large Open Files Configuration" Dynamic Application, you must first:

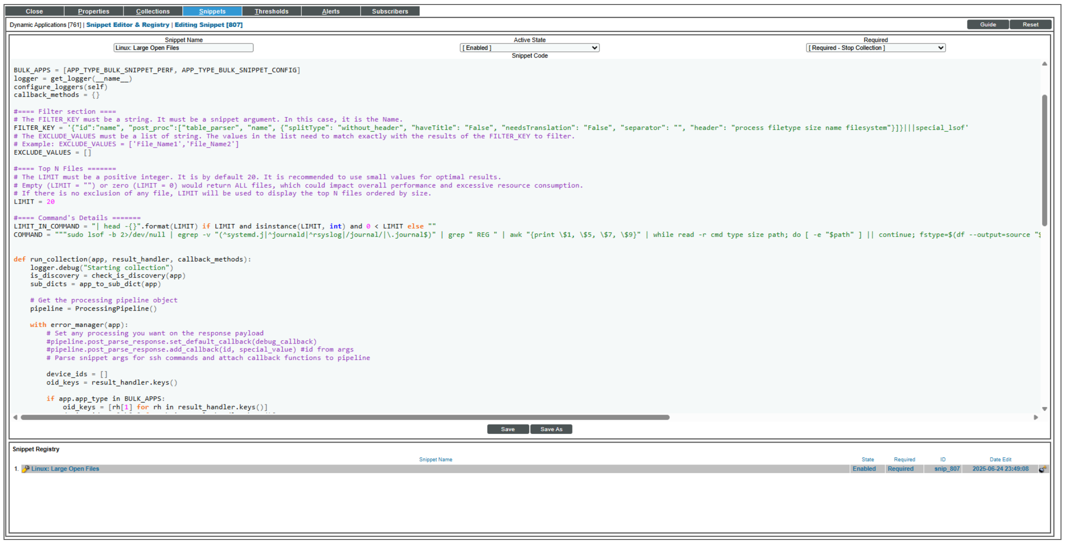
- Verify the List of Open Files (lsof) is installed on the Linux device so the Dynamic Application can collect data. To verify installation, run one of the following commands in the Linux server:
  - `lsof -v`
  - `which lsof`
- The "Linux: Large Open Files Configuration " Dynamic Application collects data using elevated privileges (`sudo`). To function properly, the user must be added to the `/etc/sudoers` file as follows:
  - `username ALL=(ALL:ALL) NOPASSWD: /usr/bin/lsof`

Specifications

The large open files monitoring process has the following specifications:

- Excludes systemd journal processes, journald, rsyslog, and journal files.
- Filters to show only regular files.
- Displays only the top 20 large open files by default. You can change the default number in the snippet code of Dynamic Application by updating the `LIMIT` constant.
- Filters files according to the configuration set in the snippet code using `FILTER_KEY` and `EXCLUDE_VALUES`.
  - `FILTER_KEY`. The snippet argument of the collection object you want to use to filters its values.
  - `EXCLUDE_VALUES`. The value must be applicableto the collection object selected for filtering. You can set it for one element or many elements separated by square brackets and commas.

> **NOTE:** If `LIMIT` is set and some of the file names specified in `EXCLUDE_VALUES` are included in the default top 20 display, the Dynamic Application will collect the top 20 large open files, but you will only see the included files.



Dynamic Applications [761] | Snippet Editor & Registry | Editing Snippet [807]

Snippet Name: Linux: Large Open Files

Active State: [Enabled]

Required: [Required - Stop Collection]

Snippet Code

```
BULK_APPS = [APP_TYPE_BULK_SNIPPET_PERF, APP_TYPE_BULK_SNIPPET_CONFIG]
logger = get_logger(__name__)
configure_loggers(self)
callback_methods = {}

#==== Filter section ====
# The FILTER_KEY must be a string. It must be a snippet argument. In this case, it is the Name.
FILTER_KEY = '{"id":"name", "post_proc":["table_parser", "name", {"splitType": "without_header", "haveTitle": "False", "needsTranslation": "False", "separator": "", "header": "process filetype size name filesystem"}]}||special_lsof'
# The EXCLUDE_VALUES must be a list of string. The values in the list need to match exactly with the results of the FILTER_KEY to filter.
# Example: EXCLUDE_VALUES = ['File_Name1','File_Name2']
EXCLUDE_VALUES = []

#==== Top N Files =======
# The LIMIT must be a positive integer. It is by default 20. It is recommended to use small values for optimal results.
# Empty (LIMIT = "") or zero (LIMIT = 0) would return ALL files, which could impact overall performance and excessive resource consumption.
# If there is no exclusion of any file, LIMIT will be used to display the top N files ordered by size.
LIMIT = 20

#==== Command's Details =======
LIMIT_IN_COMMAND = "| head -{}".format(LIMIT) if LIMIT and isinstance(LIMIT, int) and 0 < LIMIT else ""
COMMAND = """sudo lsof -b 2>/dev/null | egrep -v "(^systemd.j|^journald|^rsyslog|/journal/|\.journal$)" | grep " REG " | awk "{print \$1, \$5, \$7, \$9}" | while read -r cmd type size path; do [ -e "$path" ] || continue; fstype=$(df --output=source "


def run_collection(app, result_handler, callback_methods):
    logger.debug("Starting collection")
    is_discovery = check_is_discovery(app)
    sub_dicts = app_to_sub_dict(app)

    # Get the processing pipeline object
    pipeline = ProcessingPipeline()

    with error_manager(app):
        # Set any processing you want on the response payload
        #pipeline.post_parse_response.set_default_callback(debug_callback)
        #pipeline.post_parse_response.add_callback(id, special_value) #id from args
        # Parse snippet args for ssh commands and attach callback functions to pipeline

        device_ids = []
        oid_keys = result_handler.keys()

        if app.app_type in BULK_APPS:
            oid_keys = [rh[1] for rh in result_handler.keys()]
```

Snippet Registry

	Snippet Name	State	Required	ID	Date Edit
1.	Linux: Large Open Files	Enabled	Required	snip_807	2025-06-24 23:49:08

NOTE: The Dynamic Application could present gaps when the device is overloaded.

Monitoring Memory Pressure

To monitor large open files with the "Linux: Memory Pressure Performance" Dynamic Application, you must first:

- Verify that the Linux kernel is version 4.20 or later as this Dynamic Application only collects from these versions. To check which kernel is currently running, enter the following command on your Linux device:
  - `sudo uname -r`
- Check that the Pressure Stall Information (PSI) feature is enabled by running the following command:
  - `grep CONFIG_PSI/boot/config-$(uname -r)`

    If the PSI is enabled, you should receive a result similar to `CONFIG_PSI_DEFAULT_DISABLED=n`

NOTE: The PSI feature can be disabled in the kernel configuration, even though the kernel version supports it. ScienceLogic recommends that you check if the PSI is enabled to expect collection data in the Dynamic Application.

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: APM" PowerPack, SL1 will automatically create relationships between Linux Servers and New Relic Servers.

Chapter

3

Configuring Syslog for Linux

Overview

This section describes how to configure syslog for Linux.

IMPORTANT: The following sections describe a general method for configuring syslog for Linux, which may not apply to your specific distribution. Please contact your Linux distribution vendor for specific instructions on how to perform syslog forwarding. For information about configuring your message collectors to accept inbound messages, see *Daily Health Tasks*.

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 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

4

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

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:

<table>
<tr><th>Dynamic Application</th><th>Collection Object</th><th>Linux Command</th></tr>
<tr><td>Linux: Configuration Discovery</td><td></td><td>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.</td></tr>
<tr><td rowspan="2">Linux:
CPU Configuration</td><td>All</td><td>cat /proc/cpuinfo/
lscpu</td></tr>
<tr><td>Server Product Name</td><td>cat /sys/devices/virtual/dmi/id/product_name</td></tr>
<tr><td>Linux: CPU Cores Performance</td><td>All</td><td>cat /proc/stat</td></tr>
<tr><td>Linux: CPU Performance</td><td>All</td><td>cat /proc/stat</td></tr>
<tr><td>Linux: Disk IOPs Performance</td><td>All</td><td>cat /proc/diskstats</td></tr>
<tr><td>Linux: Hardware Configuration</td><td>All</td><td>sudo /usr/sbin/dmidecode -qt 1,2,3</td></tr>
<tr><td>Linux: ICMP Performance</td><td>All</td><td>cat /proc/net/snmp</td></tr>
<tr><td>Linux: Inode Performance</td><td>All</td><td>timeout 3 df -iPT</td></tr>
<tr><td>Linux: Large Open Files Configuration</td><td>All</td><td>sudo lsof -b 2>/dev/null | egrep -v "(^systemd.j|^journald|^rsyslog|/journal/|\.journal$)" | grep " REG " | awk "{print \$1, \$5, \$7, \$9}" | while read -r cmd type size path; do [-e "$path"] || continue; fstype=$(df --output=source "$path" 2>/dev/null | awk "NR==2 {print \$1}"); echo "$cmd $type $size $path $fstype"; done | sort -r -n -k 3,3 | uniq | head -20</td></tr>
<tr><td>Linux: Memory Performance</td><td>All</td><td>cat /proc/meminfo</td></tr>
<tr><td>Linux: Memory Pressure Performance</td><td>All</td><td>cat /proc/pressure/memory</td></tr>
<tr><td>Linux: Route Table Configuration</td><td>All</td><td>ip route 2>/dev/null || /sbin/ip route 2>/dev/null</td></tr>
</table>

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

<table>
<tr><th>Dynamic Application</th><th>Collection Object</th><th>Linux Command</th></tr>
<tr><td>Linux: TCP Performance</td><td>All</td><td>cat /proc/net/snmp</td></tr>
<tr><td>Linux: Template Discovery</td><td></td><td>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.</td></tr>
<tr><td>Linux: UDP Performance</td><td>All</td><td>cat /proc/net/snmp</td></tr>
<tr><td>Linux: UDP Services Configuration</td><td>All</td><td>ss -luan 2>/dev/null || /usr/sbin/ss -luan 2>/dev/null || /bin/ss -luan 2>/dev/null</td></tr>
<tr><td>Linux: Zombie Process</td><td>All</td><td>ps aux | grep Z</td></tr>
</table>

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

<table>
<tr><th>Dynamic Application</th><th>Collection Object</th><th>Linux Command</th></tr>
<tr><td>Linux: IC Detail</td><td>All</td><td>Internal Collection that consumes data stored by the "Linux: ICDA Cache" Dynamic Application.</td></tr>
<tr><td>Linux: IC Filesystem Inventory</td><td>All</td><td>Internal Collection that consumes data stored by the "Linux: ICDA Cache" Dynamic Application.</td></tr>
<tr><td>Linux: IC Filesystem Performance</td><td>All</td><td>Internal Collection that consumes data stored by the "Linux: ICDA Cache" Dynamic Application.</td></tr>
<tr><td>Linux: IC Interface Inventory</td><td>All</td><td>Internal Collection that consumes data stored by the "Linux: ICDA Interface Cache" Dynamic Application.</td></tr>
<tr><td>Linux: IC Interface Performance</td><td>All</td><td>Internal Collection that consumes data stored by the "Linux: ICDA Interface Cache" Dynamic Application.</td></tr>
<tr><td>Linux: IC Port Performance</td><td>All</td><td>Internal Collection that consumes data stored by the "Linux: ICDA Cache" Dynamic Application.</td></tr>
<tr><td>Linux: IC Process Inventory</td><td>All</td><td>Internal Collection that consumes data stored by the "Linux: ICDA Interface Cache" Dynamic Application.</td></tr>
<tr><td>Linux: IC Process Performance</td><td>All</td><td>Internal Collection that consumes data stored by the "Linux: ICDA Interface Cache" Dynamic Application.</td></tr>
<tr><td rowspan="4">Linux: ICDA Cache</td><td>Azure Host</td><td>[[$(< /sys/devices/virtual/dmi/id/chassis_asset_tag) == "7783-7084-3265-9085-8269-3286-77"]] && echo "Azure" || echo "False"</td></tr>
<tr><td>Filesystem</td><td>timeout 3 df -kPT</td></tr>
<tr><td>Hardware Config Product Name</td><td>cat /sys/devices/virtual/dmi/id/product_name</td></tr>
<tr><td>Network Interfaces</td><td>for x in `ls -ld /sys/class/net/* | grep -v '/virtual/' | rev | cut -d/ -f1 | rev`; do echo $x ': ' `if [</td></tr>
</table>

Dynamic Application	Collection Object	Linux Command
		"$x" = "lo"]; then echo "0"; else cat /sys/class/net/$x/speed 2>/dev/null || echo "0" ; fi`; done;
	Network Interfaces IP Address	/sbin/ip addr
	Uptime	cat /proc/uptime
Linux: ICDA Interface Cache	Interface Stats	/sbin/ip -s -s link
	Process	ps aux
	TCP Listening Ports	echo ".";ss -ltn 2</dev/null|| /usr/sbin/ss -ltn 2</dev/null|| /bin/ss -ltn 2>/dev/null

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