# Table of Contents

**Introduction**

- Splunk App for Unix and Linux

**Before you install**

- Platform and hardware requirements
- What data the Splunk App and Splunk Add-on for Unix and Linux collect
- Other deployment considerations

**Install the Splunk App for Unix and Linux**

- What a Splunk App for Unix and Linux deployment looks like
- Install the Splunk App for Unix and Linux
- Install the Splunk Add-on for Unix and Linux
- Install the Splunk App for Unix and Linux in a distributed Splunk environment
- Install the Splunk App for Unix and Linux using self service installation on Splunk Cloud
- Enable data and scripted inputs

**Use the Splunk App for Unix and Linux**

- Log in and get started
- Configure the Splunk App for Unix and Linux
- Dashboard reference
- Use the Home dashboard
- Use the Metrics dashboard
- Use the Hosts dashboard
- Use the Alerts dashboard

**Troubleshoot the Splunk App for Unix and Linux**

- Troubleshoot the Splunk App for Unix and Linux

**Advanced topics**

- Create custom alerts
- Saved searches
- Search macros

**Release notes**

- Release notes for the Splunk App for Unix and Linux
- Third-party software attributions/credits
Introduction

Splunk App for Unix and Linux

The Splunk App for Unix and Linux provides data inputs, searches, reports, alerts, and dashboards for Linux and Unix management. From any place, you can monitor and troubleshoot *nix operating systems of any size. The app includes a set of scripted inputs for collecting CPU, disk, I/O, memory, log, configuration, and user data:

- Get information about who is logged into your system, including last login times and unauthorized login attempts.
- Find out how much network throughput and bandwidth your system is using.
- Determine the status of current running processes on your system, and who is running them.
- Learn what software is installed on your system.

Monitored system metrics

The Splunk App for Unix and Linux runs with a Splunk platform instance to gather system metrics, including the following:

- Hardware information: CPU type, count, and cache; hard drives; network interface cards, count, and memory, as well as CPU statistics.
- Disk information, including available disk space and associated input/output statistics for devices and partitions.
- Information about the configured network interfaces, including connections, routing tables, and TCP/UDP transfer statistics.
- User statistics, including last login times for system accounts, user attributes, and security-related information.
- Information about processes, the files they open, and other resources they use.

The app has pre-built reports and dashboards for visibility into your system's operation.

App Features

Features of the Splunk App for Unix and Linux include the following:

**Central Visibility Into Operational Health**

Get instant visibility into the operational health of Unix and Linux environments. Organize your hosts by groups of services specific to your environment. Use NOC-like dashboards for central insight into problems and visualize resource consumption of selected systems for easy detection of outliers and anomalies.

**Performance and Resource Utilization Analytics**

Set multiple customizable thresholds for your CPU and memory utilization across your groups of hosts to easily spot trends and spikes in resource utilization in your infrastructure. Isolate problems with configurable statistical comparisons, using 42 important host and OS metrics. Visualize trends and display side-by-side performance comparisons of the several hosts of interest to understand trends, establish baselines and optimize resource allocations. Quickly cross-compare CPU, RAM and disk historical capacity utilization across many different hosts to identify increased resource consumption.
**Threshold-Based Alerts**

Get real-time notifications of important events from your Unix and Linux environment using the app's threshold-based alerts. Quickly assess the business impact of events and conduct remediation actions through insight into snapshots of various OS metrics around the time-specific alert fired. Compare the behavior of hosts in your systems and create long-term trends based on the alerts activity in your environment.

**Correlation Across Technologies**

Combine your OS data with data from all other technology tiers, such as applications, virtual, storage, networks and servers to gain a complete, centralized view of KPIs across your enterprise. Use Splunk search language, visualizations and correlations to find causal links across technologies. Get an accurate picture of resource usage and performance across multiple tiers of your IT stack.

**Common Information Model Compatibility**

This app provides the inputs and CIM-compatible knowledge to use with other Splunk apps, such as Splunk Enterprise Security, the Splunk App for PCI Compliance, and Splunk IT Service Intelligence.

**Additional Information**

Download the Splunk App for Unix and Linux from Splunkbase.

For a summary of new features, fixed issues, and known issues, see Release Notes for the Splunk App for Unix and Linux.

See Questions related to the Splunk App for Unix and Linux on Splunk Answers.
Before you install

Platform and hardware requirements

This topic discusses the underlying requirements for running both the Splunk App and the Splunk Add-on for Unix and Linux.

What versions of Splunk Enterprise does the app support?

The Splunk App for Unix and Linux supports Splunk Enterprise versions 7.3.x and 8.0.x.

Distributed installation of this app

This table provides a quick reference for installing this app onto a distributed deployment of Splunk Enterprise.

<table>
<thead>
<tr>
<th>Splunk instance type</th>
<th>Supported</th>
<th>Required</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Search Heads</td>
<td>Yes</td>
<td>Yes</td>
<td>Install this app onto all search heads where you require knowledge management.</td>
</tr>
<tr>
<td>Indexers</td>
<td>Yes</td>
<td>Conditional</td>
<td>The Splunk App for Unix and Linux does not require installation on indexers except in the case where you forward search head data to those indexers. If that is true, you must install the app onto those indexers. If you want to collect *nix data from those indexers, you must also install the Splunk Add-on for Unix and Linux (Splunk_TA_nix) onto the indexers.</td>
</tr>
<tr>
<td>Heavy Forwarders</td>
<td>Yes</td>
<td>No</td>
<td>The Splunk App for Unix and Linux does not do anything when you install it on a heavy forwarder unless that forwarder is also a search head. If you want to collect *nix data from the HF, you must also install the Splunk Add-on for Unix and Linux (Splunk_TA_nix) component.</td>
</tr>
<tr>
<td>Universal Forwarders</td>
<td>No</td>
<td>No</td>
<td>Use universal forwarders to get the data you need for the app. While the app does nothing when you install it on a universal forwarder, you can install the Splunk Add-on for Unix and Linux (Splunk_TA_nix) component on forwarders and send data to the Splunk App for Unix and Linux indexers.</td>
</tr>
</tbody>
</table>

Distributed deployment compatibility

This table provides a quick reference for the compatibility of this add-on with Splunk distributed deployment features.

<table>
<thead>
<tr>
<th>Distributed deployment feature</th>
<th>Supported</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Search Head Clusters</td>
<td>Yes</td>
<td>You can install this app on a search head cluster. Follow the procedures that this manual outlines to get the data for the app, then install the app on the cluster. Install the Splunk App for Unix and Linux on a search head cluster</td>
</tr>
<tr>
<td>Indexer Clusters</td>
<td>Yes</td>
<td>Before you start the Splunk App for Unix and Linux installation, configure your indexer cluster.</td>
</tr>
<tr>
<td>Deployment Server</td>
<td>Yes</td>
<td>You can use a deployment server to distribute the Splunk Add-on for Unix and Linux (Splunk_TA_nix) component onto hosts with installed universal forwarders to collect *nix data.</td>
</tr>
</tbody>
</table>

3
Hardware and operating system requirements

The Splunk App for Unix and Linux installs directly onto a Splunk search head or indexer. It can be configured either through the app’s setup user interface in Splunk Web or manually via the command line.

The Splunk Add-on for Unix and Linux installs onto either an indexer or a universal forwarder. When installed on an indexer, the add-on can be configured either through the app’s setup user interface in Splunk Web or manually via the command line. When installed on a universal forwarder, the add-on must be configured manually via the command line.

Both the full app and the add-on install on Splunk instances running on many versions of Unix, including Linux, Solaris, AIX, and HP/UX.

Hardware requirements for the Splunk App for Unix and Linux depend on what you plan to do with the app. At a minimum, your hardware should meet or exceed the minimum hardware requirements for Splunk itself.

Official support

While the Splunk App for Unix and Linux can be installed on any version of *nix that Splunk supports, only the following versions have official support:

- For installation of the Splunk App for Unix and Linux, on search heads: Linux, any version that Splunk supports.
- For installation of the Splunk Add-on for Unix and Linux, on universal forwarders: All versions of *nix listed in the Unix operating systems section of the core Splunk platform’s System requirements topic.
- For details on how to plan a Splunk deployment, see Introduction to capacity planning for Splunk Enterprise in the Capacity Planning Manual.

Installing the Splunk Add-on for Unix and Linux onto a Windows Splunk instance has no effect.

What web browsers does the app support?

The Splunk App for Unix and Linux is not supported on any version of Internet Explorer because it makes heavy use of scalable vector graphics (SVG), a standard for which IE has limited support.

It can, however, be used on any other Splunk-supported browser.

What other items do the app and add-on require?

The Splunk Add-on for Unix and Linux requires the sysstat package to function properly. You can download the sysstat utilities from the sysstat utilities download page or from your local package repository (depending on the version of *nix your system runs.)

What are the other prerequisites?

The following table provides compatibility of Splunk App for Unix and Linux with different Splunk Add-on for Unix and Linux versions and Splunk versions:

<table>
<thead>
<tr>
<th>Compatible Unix App version</th>
<th>Compatible Nix Add-on version</th>
<th>Compatible Splunk Version</th>
</tr>
</thead>
</table>

4
What data the Splunk App and Splunk Add-on for Unix and Linux collect

This topic describes what data the Splunk App and Splunk Add-on for Unix and Linux collects.

Data collection

The add-on collects the following data using file inputs:

- Changes to files present in the /etc directory and subdirectories.
- Changes to files present in the /var/log directory and subdirectories.

The add-on collects the following data using scripted inputs:

- CPU statistics via the sar, mpstat and iostat commands (cpu.sh scripted input).
- Free disk space available for each mount via the df command (df.sh Scripted input).
- Hardware information - CPU type, count, and cache; hard drives; network interface cards and count; and memory via the dmesg, iostat, ifconfig, and df commands (hardware.sh scripted input).
- Information about the configured network interfaces via the ifconfig and dmesg commands (interfaces.sh scripted input).
- Input/output statistics for block devices and partitions via the iostat command (iostat.sh scripted input).
- Last login times for system accounts via the last Command (lastlog.sh scripted input).
- Information about files opened by processes via the lsof command (lsof.sh scripted input).
- Network connections, routing tables and network interface statistics via the netstat command (netstat.sh scripted input).
- Available network ports via the netstat command (openPorts.sh scripted input).
- Information about software packages or sets that are installed on the system via the dpkg-query, pkginfo, and pkg_info Commands (package.sh scripted input).
- Information about TCP/UDP transfer statistics via the netstat command (protocol.sh scripted input).
- Status of current running processes via the ps command (ps.sh scripted input).
- Audit information recorded by the auditd daemon to /var/log/audit/audit.log (rlog.sh scripted input).
- System date and time and NTP server time via the date and ntpdate commands (time.sh scripted input).
- List of running system processes via the top Command (top.sh scripted input).
- User attribute information for the local system via the /etc/passwd file (usersWithLoginPrivs.sh scripted input).
- Process related memory usage information via the top, vmstat, and ps commands (vmstat.sh scripted input).
- Information of all users currently logged in via the who command (who.sh scripted input).

Note: Blank fields returned in events gathered by the scripted inputs described above display as question marks ("?"). This is expected behavior to preserve field spacing, and is not cause for concern.

Index locations

The Splunk Supporting Add-on for Unix and Linux creates two indexes: unix_summary and firedalerts. It uses these indexes to maintain the list of triggered alert events.
Indexing volume

The Splunk App for Unix and Linux collects around 200 megabytes of data per host per day. The app can collect slightly more or less based on individual host activity.

Other deployment considerations

In many applications, the Splunk App for Unix and Linux suite installs on a *nix server and collects data from that server. You then use Splunk Web to browse the app's included dashboards, reports, and saved searches to gain for insight into that data.

Additional uses for the app and add-on

There are additional uses for the app and add-on:

- You can use the add-on to collect *nix data from a number of *nix machines by installing a universal forwarder on each machine and deploying the app to those forwarders. Once the app is installed on each forwarder, you can then forward the data to a receiving indexer that is running the full app. Read Deploy the Splunk App for Unix and Linux in a distributed Splunk environment for additional information and instructions.

- You can install the app on a Splunk instance running on Windows. This instance can be an indexer or a search head. In this configuration, you must disable all included inputs that come with the app. Read Search data received from a forwarder running on a different operating system for additional details.

- You can also install the add-on on an indexer to provide data inputs for another app installed on that indexer, such as the Splunk App for Enterprise Security.

- If you install the Splunk App for Unix and Linux in a distributed environment and have configured the search heads in that environment to send data to the indexers, you might need to deploy the indexes.conf file onto your indexers to make sure that the unix_summary summary index is available. Failure to do so might cause issues with alerts for the app, since alerts use this special index.

Configure the Splunk App for Unix and Linux on multiple machines

The app has the ability to display data from many hosts. Following is a list of steps to take to get that data:

1. Install the Splunk App for Unix and Linux on a central Splunk instance.
2. Configure the central Splunk instance to be a receiving indexer.
3. On each *nix machine from which you want to get *nix data, install a universal forwarder.
4. Configure each universal forwarder to forward data to the central Splunk receiver.
5. Install the Splunk Add-on for Unix and Linux on each universal forwarder.
6. Configure inputs.conf on each universal forwarder to enable the *nix data inputs. A deployment server eases management of this and other forwarder configuration files. The Splunk App for Unix and Linux installation package allows for direct installation into a deployment server for distribution of add-on components to universal forwarders.
7. On the central Splunk instance, open the Splunk App for Unix and Linux and confirm that you are receiving data from the *nix servers that have universal forwarders and the Splunk App for Unix and Linux installed on them: search index=<index>
Install the Splunk App for Unix and Linux

What a Splunk App for Unix and Linux deployment looks like

This topic explains the overall architecture of a Splunk App for Unix and Linux deployment.

For instructions on how to deploy the app, read "Install the Splunk App for Unix and Linux."

For instructions on how to deploy the add-on, read "Install the Splunk Add-on for Unix and Linux."

Overview

The Splunk App for Unix and Linux can be deployed in a number of ways. The most common way is to deploy a "central" Splunk instance that has indexers and search heads, contains the main Splunk App for Unix and Linux index, and runs Splunk Web.

The central Splunk instance can be one or more servers

A central Splunk instance can consist of one or more servers. In a single indexer setup, a Splunk App for Unix and Linux deployment collects data about itself. In a multiple-server configuration, you can deploy:

- At least one indexer that collects data from itself or other *nix servers
- One or more search heads that search the collected data and host the application.

You can distribute the central Splunk instance further by adding more indexers and search heads. This allows you to scale the deployment for additional incoming data volume.

The central Splunk instance can run on any Splunk-supported operating system

You can deploy the Splunk App for Unix and Linux on Windows search heads and use Windows indexers to index the data. In this scenario, the Windows indexers must receive data sent to them from *nix servers with universal forwarders installed. The Windows indexers cannot collect *nix data themselves.

The Splunk App for Unix and Linux can monitor many *nix servers at once

The Splunk App for Unix and Linux supports collecting data from hundreds of servers. There are many ways to configure the Splunk App for Unix and Linux, depending on your network's topology.

You monitor additional servers with your Splunk App for Unix and Linux deployment by:

- Installing universal forwarders on each *nix server you want to include in the environment.
- Configuring the forwarders to send data to the indexers in the central Splunk instance.
- Deploying the Splunk Add-on for Unix and Linux onto those forwarders.

You can use a deployment server to manage universal forwarder configurations and deploy the Splunk Add-on for Unix and Linux onto many *nix servers at once.
Example deployment

The diagram below depicts an example Splunk App for Unix and Linux deployment.

Each *nix server on your network gets a Splunk universal forwarder. On that forwarder, you install the Splunk Add-on for Unix and Linux which collects *nix data and sends it to the indexer(s) in the central Splunk App for Unix and Linux instance.

The central Splunk App for Unix and Linux instance has at least a search head (with the Splunk App for Unix and Linux installed on it) and an indexer. The indexer indexes the *nix data (as shown by the black arrows), and the search head searches the indexer for that data (as shown by the green arrow). The indexer returns events to the search head (blue arrow). Users log into the search head to use the app and see the data.
Splunk's Professional Services can help with questions and provide assistance with large or complex layouts.

Install the Splunk App for Unix and Linux

The installation package for the Splunk App for Unix and Linux contains dashboards, reports, alerts, lookups, and macros for use with Splunk Web.

Create an index

The Splunk Add-on for Unix and Linux is a separate download from Splunkbase. Versions 6.0.0 and later of the Splunk Add-on for Unix and Linux do not include indexes. For the Splunk App for Unix and Linux, complete the following steps to create an index on your indexer:

1. Make a local directory in the `splunk_app_for_nix` folder if you don’t have one already.
2. From the app’s Default directory, copy `macros.conf` and `savedsearches.conf` into your local directory.
3. Edit the `os_index` macro in `macros.conf` as follows: `index=os`. You can also make a custom index: `index=<custom index>`.
4. Edit the `fired_alerts` saved search in `savedsearches.conf` as follows:
   ```bash
   | rest /services/search/jobs | search [search index=_audit action=alert_fired | fields sid] |
   collect index=os.
   ```

Install the Splunk App for Unix and Linux using Splunk Web

Complete the following steps to install the Splunk App for Unix and Linux using Splunk Web:

1. Download the Splunk App for Unix and Linux from Splunkbase, or by browsing to it using Splunk Web.
2. From the Splunk Web home screen, click the gear icon next to Apps.
3. Click Install app from file.
4. Locate the downloaded app file and click Upload.
5. Restart the Splunk platform.

Install the Splunk App for Unix and Linux from the command line

Complete the following steps to install the Splunk App for Unix and Linux using the command line:

1. Download theSplunk App for Unix and Linux from Splunkbase.
2. Unpack the file.
3. Copy the `splunk_app_for_nix` directory to `$SPLUNK_HOME/etc/apps`.
4. Restart the Splunk platform.

Upgrade the Splunk App for Unix and Linux

You can upgrade directly from versions 5.2.2 and later of the Splunk App for Unix and Linux through Splunk’s in-app upgrade feature within Splunk Web, or from the command line.

**Upgrade from versions 4.7 through 5.2.1**

Versions 5.2.2 and later of the Splunk App for Unix and Linux do not include the `SA-nix` file. If you are upgrading from versions 4.7 through 5.2.1, complete the following steps to keep the categories and groups that you have configured:
1. Copy the `dropdowns.csv` file. In a single-instance deployment, the file is in `etc/apps/SA-nix/lookups/`. In a distributed deployment, the file is in `$SPLUNK_HOME/etc/shcluster/apps/`

2. Move the copied `dropdowns.csv` file to `etc/apps/splunk_app_for_nix/lookups/` for a single instance deployment or to `$SPLUNK_HOME/etc/shcluster/apps/` for a distributed deployment.

3. Manually delete `SA-nix` from your apps folder.

### Upgrade from version 4.6.x and earlier

Upgrading from version 4.6.x of the Splunk App for Unix and Linux is unsupported. You can run version 4.6 simultaneously with another version.

The installation package for version 5.2.5 installs in a different directory than version 4.6. Once you have installed version 5.2.5, you can configure version 5.2.5 to use the same indexes and source types that version 4.6 uses.

For detailed installation instructions, see Install the Splunk App for Unix and Linux.

Do not install version 5.2.5 in the same directory that any version earlier than 5.0 uses. That older directory is not supported, and installing version 5.2.5 there can render both versions of the app unusable.

Once you have configured and evaluated version 5.2.5, you can remove version 4.6 without data loss.

### Upgrade from version 5.2.5 to version 6.0.0

Complete the following steps for a single instance deployment. In case of a distributed deployment, use the `$SPLUNK_HOME/etc/shcluster/apps/` path.

1. Stop Splunk.
2. In version 5.2.5 `$SPLUNK_HOME/etc/apps/splunk_app_for_nix`, rename `appserver` folder to `appserver_backup`.
3. In version 5.2.5 `$SPLUNK_HOME/etc/apps/splunk_app_for_nix/default/data/ui/views` folder, rename the following:
   1. `browser_incompatibility.xml` to `browser_incompatibility.xml.backup`
   2. `home_fullscreen.xml` to `home_fullscreen.xml.backup`
4. Copy `dropdowns.csv` from `$SPLUNK_HOME/etc/apps/splunk_app_for_nix/lookups/`.
5. If you are upgrading the app from the user interface, copy `indexes.conf` from `$SPLUNK_HOME/etc/apps/splunk_app_for_nix/default` to `$SPLUNK_HOME/etc/apps/splunk_app_for_nix/local`. If you are not updating the app from the user interface, skip this step.
6. Untar the new app package in `$SPLUNK_HOME/etc/apps/`.
7. Place the copied `dropdowns.csv` from step 4 to `$SPLUNK_HOME/etc/apps/splunk_app_for_nix/lookups` of the new app.
8. Start Splunk.

### Install the Splunk Add-on for Unix and Linux

The Splunk App for Unix and Linux uses the Splunk Add-on for Unix and Linux to collect *nix data from *nix servers. For installation and deployment information for the add-on, see the Install the Splunk Add-on for Unix and Linux.
Install the Splunk App for Unix and Linux in a distributed Splunk environment

The following table shows best-practice locations in your Splunk App for Unix and Linux deployment where you should install the Splunk App and Add-on for Unix and Linux. To learn more about installing apps and add-ons in a distributed Splunk Enterprise environment, see Where to install Splunk add-ons in Splunk Supported Add-ons.

Splunk App and Add-on for Unix and Linux Installation Locations:

<table>
<thead>
<tr>
<th>Component</th>
<th>Search Head / Search Head Cluster</th>
<th>Indexer</th>
<th>Forwarder</th>
<th>Deployment Server</th>
</tr>
</thead>
<tbody>
<tr>
<td>App: splunk_app_for_nix</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Add-on: Splunk_TA_nix</td>
<td>X</td>
<td></td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>

In a distributed environment, Splunk indexers and search heads comprise a central Splunk App for Unix and Linux instance. The central instance indexes *nix data that universal forwarders, installed on *nix hosts, send to it. The distributed environment can comprise both search head and indexer clusters. Log into the central instance to use the app.

Steps to building a Splunk App for Unix and Linux deployment

The following installation instructions are generic. You might need to make additional adjustments and configuration changes based on your specific network topology. A deployment server can help ease configuration of a large number of clients in a distributed environment.

1. Install indexers to store the *nix data.
2. Configure these indexers as receivers.
3. On each *nix host that you want data from, install a universal forwarder.
4. Configure the universal forwarders to send data to the receiving indexers.
5. Install a search head or a search head cluster.
6. Configure the search head or search head cluster to search the indexers you set up.
7. Install the Splunk Add-on for Unix and Linux on the indexer.
8. Install the Splunk App for Unix and Linux on the search head or search head cluster.
9. Log into the search head or a search head cluster member and open the Splunk App for Unix and Linux.
10. Configure the Splunk App for Unix and Linux.

Install the Splunk Add-on for Unix and Linux on an indexer

To build your distributed Splunk App for Unix and Linux deployment, first install Splunk Enterprise and the Splunk Add-on for Unix and Linux onto the hosts that you want to index *nix data:

1. Identify the hosts that will be part of the central Splunk App for Unix and Linux instance. These hosts store incoming *nix data from *nix servers.
2. Install Splunk Enterprise onto each of the indexers.
3. Configure each indexer to receive data from forwarders.
4. Follow the instructions at Install the Splunk Add-on for Unix and Linux to place the Splunk Add-on for Unix and Linux onto each indexer.
5. If the indexer is also a *nix host and you want to collect *nix data from it, enable the data and scripted inputs inside the Splunk_TA_nix add-on on the host.
6. Restart Splunk Enterprise on each host to complete the add-on installation.
Install the Splunk App for Unix and Linux on a search head

After you install the Splunk Add-on for Unix and Linux on your indexers, you must configure and install the Splunk App for Unix and Linux onto search heads which search the indexers. Once you have installed the app onto search heads, you can then log into the search heads and view the incoming *nix data.

If you have a search head cluster, follow the instructions at "Install the Splunk App for Unix and Linux on a search head cluster" later in this topic.

To install the Splunk App for Unix and Linux on a search head:

1. Identify the hosts that will act as search heads in your Splunk App for Unix and Linux deployment.
2. Install Splunk Enterprise onto each of these computers, if it is not already installed.
3. On each host, configure Splunk Enterprise to search across all of the indexers in the deployment that will store *nix data.
4. Follow the instructions in "Install the Splunk App for Unix and Linux" to place the Splunk App for Unix and Linux components on each search head.
5. Restart Splunk Enterprise to complete the app installation.

Install the Splunk App for Unix and Linux on a search head cluster

If you have a search head cluster, you can install the Splunk App for Unix and Linux on that cluster.

1. If you have not already, configure the search head cluster, as described in Deploy a search head cluster in the Distributed Search manual.
2. Extract the Splunk Add-on for Unix and Linux package into $SPLUNK_HOME/etc/apps on the search head cluster deployer.
3. Extract the Splunk App for Unix and Linux package into $SPLUNK_HOME/etc/apps on the search head cluster deployer.
4. Restart the deployer.
5. On the deployer, log into Splunk Enterprise.
6. Open the Splunk App for Unix and Linux.
7. Configure the app.
8. From a shell prompt on the deployer, copy the app, add-on, and configurations to the search head cluster apps directory:

   cp -pr $SPLUNK_HOME/etc/apps/Splunk_TA_nix $SPLUNK_HOME/etc/apps/splunk_app_for_nix
   $SPLUNK_HOME/etc/shcluster/apps

9. From a command or shell prompt on the deployer, push the app, add-on, and configurations to the search head cluster members:

   splunk apply shcluster-bundle -target <URI>:<management_port> -auth <username>:<password>

   The -target specifies the URI and management port of one of the search head cluster members. For example, if one of the members is splunk2.mycompany.com, you would specify https://splunk2.mycompany.com:8089.
10. The deployer displays the following message:

    Warning: Depending on the configuration changes being pushed, this command might initiate a rolling-restart of the cluster members. Please refer to the documentation for the details. Do you wish to continue? [y/n]:

    Proceed by responding to the message with y.
11. Wait for the deployer to send the configuration bundle to the search head cluster members.
Install the Splunk Add-on for Unix and Linux on a forwarder

Once you install the Splunk App for Unix and Linux on the indexers and search heads in the central Splunk App for Unix and Linux instance, you must then install the Splunk Add-on for Unix and Linux onto the *nix hosts that you want *nix data.

Install universal forwarders on your hosts, and then install the add-on on the universal forwarders. The forwarders then send *nix data to your indexers. Complete the following steps to install the Splunk Add-on for Unix and Linux on a universal forwarder:

1. Identify the hosts from which you want to collect *nix data.
2. Install a Splunk universal forwarder on these hosts.
3. Configure the forwarder to send data to the indexers in the central Splunk App for Unix and Linux instance.
4. Follow the instructions in Install the Splunk Add-on for Unix and Linux to place the Splunk Add-on for Unix and Linux into each universal forwarder.
5. Enable the data and scripted inputs within the add-on.
6. Restart the universal forwarder to complete the add-on installation.

Use a deployment server to deploy the Splunk Add-on for Unix and Linux

These instructions provide guidance on the use of a deployment server to distribute the Splunk Add-on for Unix and Linux onto *nix servers with universal forwarders installed on them.

You might need to make additional changes to match your specific environment.

To learn more about how to use deployment server, see Updating Splunk Enterprise Instances Manual for Splunk Enterprise version 6 and later Distributed Deployment manual for Splunk Enterprise version 5 and earlier.

Set up the deployment server

1. Install Splunk Enterprise, or designate an existing full instance for use as a deployment server, if you do not already have one in your environment.
2. Set up the deployment server on a Splunk instance on which you also install the Splunk App for Unix and Linux.
   a. Define a server class for the *nix hosts that will receive the Splunk Add-on for Unix and Linux. You can use either Splunk Web or configuration files to create deployment server classes. If you are using Splunk 6.0 and later, read Define server classes in the Updating Splunk Enterprise Instances Manual to learn how to create server classes.
   b. Download the Splunk Add-on for Unix and Linux installation package and place it in an accessible location.
   c. From this location, copy the Splunk_TA_nix folder to $SPLUNK_HOME/etc/deployment-apps on the deployment server.
3. Within the $SPLUNK_HOME/etc/deployment-apps/Splunk_TA_nix folder on the deployment server, enable the data and scripted inputs that you want the add-on to collect from your *nix hosts.
4. Restart Splunk Enterprise on the deployment server to activate the changes.

Set up the deployment clients to contact the deployment server

Each *nix host with a universal forwarder installed on it is known as a deployment client. These clients fetch configuration information from the deployment server in your Splunk environment. In this case, they also fetch the Splunk Add-on for Unix and Linux and its configurations, which allows the universal forwarder to collect *nix data and subsequently send that data to the central Splunk App for Unix and Linux instance.
To set up the deployment clients, follow the instructions in the "Configure deployment clients" topic for the version of universal forwarder that you have installed on your *nix servers:

- Configure deployment clients (version 5.0.x and earlier)
- Configure deployment clients (version 6.0.x and later)

When you configure deploymentclient.conf on the clients, set the targetUri attribute to the Splunk Enterprise instance that runs the deployment server. Here is an example deploymentclient.conf file:

```
[deployment-client]
[target-broker:deploymentServer]
targetUri=deploymentserver.splunk.mycompany.com:8089
```

**Install the Splunk App for Unix and Linux using self service installation on Splunk Cloud**

1. Click on the Manage Apps button.
2. Under the App Management menu, click the Install App button. The Browse More Apps menu appears.
3. Search for the Splunk App for Unix and Linux, and click the Install button. Not all apps can be installed using self service installation.
5. Enter your Splunk.com login credentials, read and accept the login disclaimer, and click Login and Download.
6. The unix_summary index is required to populate the alert dashboard. Create the index in Settings on the search head.
7. (Optional) If the App Installation Failed window appears, review the apps or add-ons that are listed as app dependencies, review and accept the terms and conditions of the license(s), and click Install to install the app dependencies for the Splunk App for Unix and Linux. Once deployed, your Splunk Cloud instance restarts.
8. Log back into your Splunk Cloud deployment. A Complete window appears to show successful installation of the Splunk App for Unix and Linux.
9. Review the Complete window for release notes and source package review, and click View Apps.
10. On the App Management page, review the installed apps.

**Enable data and scripted inputs**

Learn more about how to enable inputs in the Splunk Add-on for Unix and Linux, read "Enable data and scripted inputs for the Splunk Add-on for Unix and Linux" in the Splunk Add-on for Unix and Linux documentation.
Use the Splunk App for Unix and Linux

Log in and get started

This topic shows you how to log in to Splunk Web, access the Splunk App for Unix and Linux, and get started.

Log in to Splunk Web

To log into Splunk Web and access the Splunk App for Unix and Linux, navigate to:

http://<host>:8000

Use the host and port you chose during installation of Splunk. The default port is 8000.

The first time you log in to Splunk, the default login details are:
Username - admin
Password - changeme

Splunk recommends that you change the admin password to a secure password.

Access the Splunk App for Unix and Linux

Once you've logged in to Splunk Web, the version of Splunk that is running determines exactly what you see.

In Splunk 5 and earlier, you see Splunk Home, with "Welcome" and "Home" tabs. Click on the "Home" tab to see the list of apps that are currently installed. You should see the Search and Getting Started apps, as well as the Splunk App for Unix and Linux. To access the Splunk App for Unix and Linux, click on it in the list.

In Splunk 6 and later, the Home page also displays by default, but installed apps appear in the screen; there is no need to access a menu to see them. Click on "Splunk App for Unix and Linux" in the list. You can also access the Splunk Add-on for Unix and Linux in this way, but the add-on only has a configuration page.

Important: When starting the app for the first time, you will initially be presented with a dialog box requesting that you configure the app. Read "First-time configuration" to learn about how to enable the app's inputs.

To learn about the various dashboards available, review "Dashboard reference."

Configure the Splunk App for Unix and Linux

This topic explains what happens when you activate the app after installing it on your Splunk instance. It shows you how to enable or disable the inputs that come with the add-on, and can be used as a reference.

To use data and scripted inputs, you must install and use the Splunk Add-on for Unix and Linux. To learn more about how to enable inputs in the Splunk Add-on for Unix and Linux, read "Enable data and scripted inputs for the Splunk Add-on for Unix and Linux" in the Splunk Add-on for Unix and Linux documentation.

Important: It's best practice to configure the data and scripted inputs and ensure that data collection is happening on this page before configuring the app.
Initial setup

After you install the app and log into Splunk Web, when you activate the app, it presents you with a dialog box that states that you need to configure the app before you can use it.

You can click Cancel to exit this setup dialog. You can then return to it later by clicking the Settings link in the navigation bar within the app.

To set up the app, click the Configure button in the lower-right hand corner of the dialog box.

The Settings - Your Data screen loads.

Configure the Splunk App for Unix and Linux in Splunk Web

Settings: Your Data
This panel of the Settings menu allows you to configure which indexe(s) contain the data that the Splunk App for Unix and Linux uses, as well as the source types that the app should use to organize the various kinds of data it collects.

To change the indexes and/or source types that the Splunk App for Unix and Linux uses:

1. Follow the instructions in "Change Unix indexes" and/or "Change data source types" below.

2. After you have made changes to the fields in this panel, click the **Save** button to save your changes.

**Note:** If you have installed this app for the first time, you should not need to make changes to these settings. If you have upgraded from an earlier version and used different indexes and/or source types for your *nix data, you can use this panel to change the index(es) and source types as needed to point the Splunk App for Unix and Linux at your existing indexed *nix data.

### Change Unix Indexes

The Unix Indexes field allows you to specify the index(es) that the Splunk App for Unix and Linux should use for access to *nix data it has collected.

To specify additional indexes:

1. Click the **Add New** button. A new field appears.

2. In this new field, type in:

    ```
    index=<index>
    ```

    where **index** is an existing index defined on the Splunk instance that hosts the Splunk App for Unix and Linux, and which contains data collected by either the Splunk App or Splunk Add-on for Unix and Linux.

    Repeat these steps for additional indexes.
To remove an existing defined index, click the x on the right side of the index field.

To confirm that the indexes you have specified contain the *nix data you want the Splunk App for Unix and Linux to use, click the Preview button. A new Search page opens that displays the most recent events collected for the specified indexes.

**Change data source types**

The remainder of the Your Data settings panel consists of fields which specify the source types for the various kinds of data that the Splunk App for Unix and Linux uses.

To specify additional source types under each Data field:

1. Click the + Add New button. A new field appears.

2. In this new field, type in:

sourcetype=<sourcetype>

where sourcetype is an existing source type defined on the Splunk instance that hosts the Splunk App for Unix and Linux.

Repeat these steps for additional source types.

To remove an existing defined source type, click the x on the right side of the source type field.

To confirm that the source types you have specified reference the *nix data you want the Splunk App for Unix and Linux to use, click the Preview button. A new Search page opens that displays the most recent events collected under the specified source types.

**Settings: Categories**
The Settings: Categories page allows you to define categories and groups for the hosts which the Splunk App for Unix and Linux has collected data for. Categories and groups allow you to easily compare metrics across different host types and/or roles when troubleshooting an issue. They also allow for ease of management when dealing with a large number of hosts.

The panel has four panes: Categories, Groups, Hosts in, and Hosts not in.

- The **Categories** panel represents the top-level grouping mechanism. You can use this panel to define site locations, like cities or data centers, for example.
- The **Groups** panel represents the second-level grouping mechanism. You can use this panel to define host or service groups, as an example.
- The **Hosts in** panel contains hosts which have been assigned to the current Category and Group. The Splunk App for Unix and Linux populates this list from the data you have collected with it.
- The **Hosts not in** panel contains hosts which have not been assigned to the active Category and Group. The Splunk App for Unix and Linux populates this list from the data you have collected with it.

**Note:** When you first run the Splunk App for Unix and Linux, it performs a search through the `os` index for host entries. It then creates a special category called “all_hosts” and a special group called “Default”. The 50 most recent hosts it finds become a member of this special category and group until you define additional categories and groups.

**Add new categories and groups**

Use the **Settings: Categories** page to add host categories and groups. When you make these changes, the Splunk App for Unix and Linux writes them to `$SPLUNK_HOME/etc/apps/splunk_app_for_nix/lookups/dropdowns.csv`.

**Important:** If the Splunk App for Unix and Linux can't find any hosts, it creates a default 'all_hosts' group and 'Default' group and sets the host value to '*', which the app treats as a wild-card. **If you want to create your own categories and groups, you must delete this default category and group first.**

To create a category:

1. In the **Categories** pane, click the + sign.

Splunk adds a new category called **New Category** in the list.

2. Click **Rename Category** to rename the category.

   1. Edit category modal opens the category name.
   2. Edit the category name.
   3. Click **Save**.

Splunk highlights the text field to allow you to change the category's name.

3. When you are done renaming the category, click away from the highlighted text box to save the changes.

To create a group within a category:

1. Select the parent category what you wish to add a group to.

2. In the **Groups** pane, click the + sign.
Splunk adds a new category called **New Group** in the list.

3. Click on **Edit Group** to rename the group.

   1. Edit Group modal opens the group name.
   2. Edit the group name.
   3. Click **Save**.

Splunk highlights the text field to allow you to change the group's name.

4. When you are done renaming the group, click away from the highlighted text box to save the changes.

**Important:** You must add at least one host to a group in order for the Splunk App for Unix and Linux to save the group and the group's parent category. The app does not save empty categories and/or groups.

**Remove groups and categories**

Complete the following steps to remove categories and groups.

1. Click on the category or group you want to remove. 2. Click **Delete Category** or **Delete Group** in the corresponding Category or Group column header. 3. Click OK to delete the category or group.

**Note:** If you choose to remove a category that has groups under it, Splunk also removes the groups that the category contains.

**Add hosts to or remove hosts from a group**

Once you have created the categories and groups you want, you can then assign hosts to a specific group.

When making host assignments, note the following:

- You can only assign hosts to a group. When you do this, the Splunk App for Unix and Linux automatically assigns the hosts to the category that contains the group.
- You can assign hosts to more than one group at a time. However, each group must be a member of a separate category.
- You cannot assign hosts to categories directly.

To assign a host to a group:

1. Click the group that you want to assign hosts to.

2. The **Hosts in** and **Hosts not in** columns populate with available hosts that the Splunk App for Unix and Linux has collected data for.

**Note:** New groups will not have hosts in the **Hosts in** column.

3. Locate the host that you want to add to the group in the **Hosts not in** column.

**Note:** If a host is already a member of another group, the Splunk App for Unix and Linux displays that group alongside the host name.

4. Click on the host in the **Hosts not in** column to add a new host to the **Hosts in** column.
Splunk makes changes to the group based on the following factors:

- If the host is not already a member of another group within the currently selected category, Splunk adds the host to the new group and then immediately saves the change.
- If the host is a member of another group within the currently selected category, Splunk removes the host from the old group, adds it to the new group, and then immediately saves the change.

To remove a host from a group:

1. Click the group that you want to remove hosts from.
2. The *Hosts in* column populates with the hosts that have already been assigned to the group. The *Hosts not in* column populates with available hosts that the Splunk App for Unix and Linux has collected data for.
3. Locate the host that you want to remove from the group in the *Hosts in* column.
4. Click on the host in the *Hosts not in* column to add a new host to the *Hosts in* column.

Splunk removes the host from the group and saves the change immediately.

**Settings: Alerts**

The Settings:Alerts page allows you to customize the alerts that the Splunk App for Unix and Linux displays when certain conditions trigger those alerts.

The Alerts settings page splits into three sections for each available alert:

- The Alert section, which shows the name of the alert, as well as a text box which allows you to enter a description for the alert.
- The Threshold section, which lets you specify when the alert triggers, as well as:
  - The business impact of the alert.
  - The remediation strategy - what a person who sees this alert should do to resolve it.
  - The escalation path - who (or what) the alert should be escalated to if attempts to resolve the alert fail, or no one responds to the alert.
- The Status section, which lets you specify whether or not the alert is active, and change its reported severity.
Configure alerts

The Splunk App for Unix and Linux comes with twelve built-in alerts which you can configure. You can add descriptions, change alert thresholds, add business impact, remediation and escalation information, and choose whether or not each alert is active.

To modify the existing alerts in the Alerts settings page:

1. In the **Alerts** section, enter a description for the alert in the text box underneath the alert's name.

2. In the **Threshold** section, drag the slider to adjust when the alert triggers.

   **Note:** You can also click on the text box underneath the slider and enter the number manually. Valid values for the threshold depend on the alert's base search.

3. In the **Business Impact** text box, enter the impact that the alert represents.

4. In the **Remediation** text box, enter a sentence that describes what a person who encounters this alert can do to stop it.

5. In the **Escalation** text box, enter the name of a person or entity that this alert should be escalated to if attempts to resolve the alert fail or are not made.

6. In the **Status** section, click the **Enabled** button to enable the alert, or the **Disabled** button to disable the alert.

7. Drag the slider to adjust the alert's severity, a choice of **Info**, **Medium**, or **High**. The alert's severity determines where and how it displays on the **Alerts** page.

8. Click the red **Save** button to save any changes you have made.

   **Note:** Whenever you make a change, the Splunk App for Unix and Linux highlights the **Save** button in red. This lets you know that any unsaved changes will be lost if you leave the settings page.

Dashboard reference

This topic lists all of the dashboards provided in the Splunk App for Unix and Linux and provides a brief description of each.

**Note:** The dashboards referenced here apply only to the full Splunk App for Unix and Linux. The Splunk Add-on for Unix and Linux does not have a user interface.

**Home**

The Home dashboard displays when you first launch the Splunk App for Unix and Linux. It divides into two sections:

- The left side displays performance radial graphs that show statistics of your choosing and update in real-time.
- The right side shows "Recent Unix Headlines" - alerts that have triggered recently.

For more information about the Home dashboard, read "Use the Home Dashboard" in this manual.
**Metrics**

The Metrics dashboard displays detailed statistics on the hosts that the Splunk App for Unix and Linux has collected data for, including information on CPU, disk, memory, I/O, process, and protocol metrics.

You can display information on a single host, a group of hosts, or a category of host groups. You can also filter by host.

To learn more about the Metrics dashboard, read "Use the Metrics dashboard" in this manual.

**Hosts**

This dashboard displays real-time information on the CPU, memory, I/O, and disk processes for all of the hosts that the Splunk App for Unix and Linux has collected data for.

You can view information on a single host, a group of hosts, or a category of host groups. You can also choose a heat map to get a quick representation of system activity on all hosts, based on several different metrics.

You can also compare up to five hosts at once by selecting the hosts and clicking the Compare button.

For more information about the Hosts dashboard, read "Use the Hosts dashboard" in this manual.

**Alerts**

The Alerts dashboard displays a summary of the triggered alerts present in the Splunk App for Unix and Linux. The dashboard splits into three sections:

- The Alert Time Range, which allows you to set the time range for triggered alerts, from the last 15 minutes up to the last week.
- Statistics, which shows you the hosts that have triggered alerts, as well as which alerts have triggered and the severity of those alerts.
- A Summary of the most recent triggered alerts.

For additional information about the Alerts dashboard, read "Use the Alerts dashboard" in this manual.

**Search**

The Search dashboard allows you to use Splunk’s search language to display events that the Splunk App for Unix and Linux has collected. You can use the Search window to evaluate and save your own custom searches and reports.

**Settings**

The Settings dashboard lets you configure the Splunk App for Unix and Linux.

For more information on how to configure the app, read "Configure the Splunk App for Unix and Linux" in this manual.

**Use the Home dashboard**
The Home dashboard is the main dashboard for the Splunk App for Unix and Linux. It displays performance statistics on two radial graphs that you can configure as well as an alert “ticker” that displays the most recent alerts that have triggered within the app.

**Performance radial graphs**

On the left side of the screen are two performance radial graphs - half-circle shaped controls which display information about real-time performance for a category or group of hosts. Each radial graph operates independently of the other, and updates once every 15 seconds by default.

Each radial graph divides into “slices” depending on which category and groups you have selected for that radial graph. If you have configured more than one host in a group, the graph displays the metric averaged across all machines within the group.

As performance levels increase or decrease for a certain category or group of hosts, individual slices of the radial graph fill in with white, beginning at the center and extending out to their edges. The higher the performance metric, the more the slices fill up.

You can configure these radial graphs to display information on any single host or group of hosts that the Splunk App for Unix and Linux has collected data for, based on categories and groups you have defined in the Splunk App for Unix and Linux's Settings page. To learn how to define categories and groups of hosts, read “Configure the Splunk App for Unix and Linux” in this manual.

**Change radial graph display stats**

To change which statistics that a radial graph displays:

1. Click the **Category** button underneath the title for a particular radial graph.
   
   The radial graph refreshes to load groups and hosts contained by the chosen category.

2. Click the **Group** button. A window pops up that displays the groups in that category.

3. Activate one or more groups by clicking the checkbox next to the desired group.
The radial graph updates to show statistics for the selected host groups.

4. Click the performance metric button next to the **Group** button. A window pops up and displays the available performance statistics.

5. Choose the desired performance metric from the items in the pop-up window.

The radial graph updates to display the desired performance metrics for the chosen category and group(s).

**Change radial graph threshold colors**

You can give your radial graphs color by editing the threshold bars placed next to each radial graph.

Each radial graph has a “threshold bar” next to it that divides into sections. Clicking on the threshold bar adds a slider at the point where you clicked and brings up a color palette where you can choose a color for a specific level of performance metric.

Once you have added a slider, you can change what threshold that slider represents by moving it up or down on the threshold bar. The radial graph updates to show your changes immediately.

To remove a slider, drag it off of the threshold bar. The radial graph updates again to reflect the color changes.

**Clear all customizations on radial graphs**

On the home dashboard, click **Clear** to reset the radial graphs to their default settings. This removes all colors and sliders in the threshold bars.

**Triggered alerts window**

The **Recent Unix headlines** window displays the most recent alerts triggered by the Splunk App for Unix and Linux.

You can see the alerts that have triggered recently, as well as their severity, which is also configurable (see **Configure the Splunk App for Unix and Linux**.)

You can limit the alerts that the app displays by clicking on the green circle, yellow triangle, and red circle on the right, above the list of triggered alerts.

You can find out more about an alert by clicking on it in the list.

The **Manage headlines** button takes you to the **Manage Headlines** page where you can edit the headlines you see when an alert triggers. You can create headlines which you can then link to alerts. When the alerts trigger, the headlines show up in the **Recent headlines** pane.

**Use the Metrics dashboard**
The Metrics dashboard displays a variety of performance statistics for any number of hosts in various different customizable graphs.

The dashboard divides into two panels: The Host Filter panel, which allows you to type in a host name to see only its data or filter hosts by category and group, and the Metrics Viewer, which shows the metrics that the Splunk App for Unix and Linux has collected over time in a circle-graph pattern.

**Host Filter**

The Host Filter panel allows you to select which host(s) you want to view. It contains a Filter text box and a list of all of the categories and groups you have created when you configured the Splunk App for Unix and Linux.

Each category entry has an arrow next to it that, when clicked, opens to show all groups and hosts in that category. When you load the page in the Splunk App for Unix and Linux, all of these entries have been expanded.

**Note:** If you have not created any categories or groups, then the Splunk App for Unix and Linux populates this area with a special category called “Default” and a special group called "All hosts". Use the Settings page to create host categories and groups.

To choose hosts:

1. In the category list, use your mouse to scroll through the list of available hosts.

**Note:** Optionally, you type in the full or partial name of a host in the Filter text box and press Enter. The Splunk App for Unix and Linux updates the category entries to show only those hosts that contain the text string you entered.

2. Select the host(s) which you want to display metrics on. The Splunk App for Unix and Linux updates the Metrics Viewer to show metrics for the selected host(s).

**Note:**

- To select multiple hosts, shift-click (control-click on Windows, Command-click on Mac OS X) the host entries.
- If you select all of the hosts in a group, the Splunk App for Unix and Linux automatically selects the group.
- You can also select groups and categories, by clicking on the name of a group or category, respectively.
• Clicking on a group selects all hosts in the group, and clicking on a category selects all groups (and thus all hosts) in the category.

**Remove filters**

To remove an existing host filter, click on the X button on the right side of the Filter text box.

**Metrics Viewer**

The Metrics Viewer panel displays metrics on the hosts that you have selected in the Host Filter panel. If you have not selected any hosts, this panel is empty.

The Metric Viewer allows you to display various different performance metrics over various ranges of time. It also allows you to add color to your graphs in a way that is meaningful to you.

**View desired performance metrics**

To change the data that the Metrics Viewer displays:

1. Select hosts in the Host Filter panel, if you have not already done so. No data will display in the Metrics Viewer without your completing this step.
2. Set the time range for the graph by clicking the time range drop-down picker (the left-most of the row of buttons under the “Metrics Viewer” text) and selecting the desired time range.
3. Select the kind of data you want to view by clicking on the data type drop-down picker (the second button from the left) and selecting the desired type of data.

The Splunk App for Unix and Linux updates the next three drop-down buttons to include metrics on the type of data you selected.

**Note:** There are five data types to choose from:

- CPU
- Memory
- Disk
- Process
- I/O

4. Choose the performance metric you want to see by selecting the appropriate entries in the performance metric drop-down pickers.

For example, if you chose the “Memory” data type and wanted to see maximum free memory by host, you would select:

- max in the first performance metric picker,
- memFreeMB in the second picker, and
- host in the third picker.

The Splunk App for Unix and Linux updates the graphs to show the selected time range, data type, and performance metrics.
What the graphs mean

- In these graphs, larger bubbles represent higher levels of metrics, while smaller bubbles represent lower levels of metrics.

Add color to your graphs

If you want, you can add color to the graphs to give them more meaning. To add color:

1. In the Metrics Viewer panel, click anywhere on the rectangle to the right of the Pick Colors text.

A pop-up window appears with a palette of colors.

2. Choose a color that is meaningful to you.

Several things happen at this point:

- The rectangle fills with the chosen color.
- The Splunk App for Unix and Linux updates all bubbles currently being displayed by the Metrics Viewer with the chosen color.
- A slider appears where you initially clicked.
- Range numbers of 0 and 100 appear on the ends of the rectangle.
- A number relative to the position between 0 and 100 also appears underneath the slider.

3. Click the "X" on the color palette to close the palette.

4. Move the slider to the desired level that the color you just picked represents.

5. Repeat Steps 1-3 to add additional colors and levels until you have all the colors and levels that you want represented in the displayed graphs.

The Splunk App for Unix and Linux updates the rectangle to include the added slider. The color to the left of the newly-added slider updates to the new chosen color.

Remove sliders

To remove a slider, simply drag it off of the rectangle. Colors to the left of the slider also get removed when you perform this action.

Reset to defaults

To remove all customized graphs and return to the default Metrics Viewer screen, click the Clear button in the upper right corner.

Use the Hosts dashboard
The Hosts dashboard gives you a high-level view of all of the hosts in your Splunk App for Unix and Linux deployment. It displays hosts in either a list or overview format, and allows you to drill down into the specifics of a host’s health and operation. It also lets you apply heat maps - graphical representations of data where individual value ranges are represented as colors - to the host views.

**Host views**

The Hosts dashboard displays information about hosts in two distinct views:

- **Node view** - where each square in the view represents a single host.
- **List view** - a text display of hosts with additional information about CPU, memory, disk and I/O shown.

The kind of data you can see depends on which view you are in.

**List view**

In list view, the Hosts dashboard displays a real-time list of hosts, groups, and CPU, memory, I/O, and disk performance metrics, constrained by the currently selected host category and group(s).

You can sort each column in ascending or descending order by clicking on the column’s header. You can sort multiple columns by Alt-clicking (clicking with the Alt key held down) the desired column headers.

You can control which hosts display in the list by selecting the desired categories and groups in the host view control panel.

**Node view**

In node view, the Hosts dashboard displays a matrix of hosts, as defined by the currently selected host category and group(s). If you have selected more than one group, the dashboard separates hosts by group.

A host square represents a single host, and you can click on the host to get specific performance metrics for that host in an unobstructed side panel.
Node view also allows you to apply performance metric heat maps to the host squares to get an instantaneous view of per-host performance for CPU, memory, disk, and I/O metrics.

**The host view control panel**

![Click for a larger image](image)

The host view control panel lets you manipulate how the Hosts dashboard displays available hosts. The control panel, situated directly underneath the "Hosts" title area, has the following controls:

**View:** This control lets you toggle between node and list views.

"**Category**" drop-down picker: This control lets you choose a category of host groups, as you defined them when you configured the Splunk App for Unix and Linux.

"**Group**" drop-down picker: This control lets you choose one or more groups in the selected category.

**Heatmap:** This control determines the heatmap that shows when the Hosts dashboard is in node view. This control is only available while in node view.

**Size:** These buttons allow you to change the size of the host squares when the dashboard is in node view. These controls are only available while in node view.

Underneath the host view control panel are additional controls that only appear while in node view:

**Number of hosts shown:** This control tells you how many hosts in the specific host category and group(s) that the Hosts dashboard currently shows.

**Show more / fewer / all:** These links control how many hosts display in the Hosts dashboard when it is in node view. Clicking more shows more hosts in the selected category and group, and clicking less shows fewer hosts in the selected category and group.

**Unpin all / compare:** These controls allow you to compare a number of hosts at the same time. The Unpin all button removes pins from any hosts that you have pinned, and the Compare button updates the Hosts dashboard to display detailed performance metrics on hosts that you have pinned. Read "Compare performance metrics on hosts" later in this topic for additional details.

**Apply heat maps in node view**
When in node view, you can apply a heat map to the currently displayed category and group(s) of hosts. To do so, click the **Heatmap** drop-down and choose the desired heat map (one of **CPU**, **Memory**, **I/O**, or **Disk**). The Splunk App for Unix and Linux:

- updates the host view control panel to include color swatches which represent ranges of performance metric for the selected heat map.
- updates the host squares in the dashboard in real time to include heat map colors.

**Get more information about a host**

To get more information about a single host in a host group while in node view:

1. Use the Category and Group drop-down pickers in the host view control panel to choose the group that the desired host is in.
2. Mouse over the host squares. The tool tip updates to show you the host name the square underneath your mouse pointer represents.
3. When you've found the host you want to get information on, click once on the square that host represents.
4. The Splunk App for Unix and Linux opens a "**host information card**" about the selected host on the far right side of the screen. This host information card displays specifications (number of CPUs, amount of RAM and disk), current system status and history, and the status of the top five processes on the host, based on CPU usage.
5. You can hover over the sparklines on the middle right side of the card to see individual metrics that the Splunk App for Unix and Linux has collected recently for the host.
6. You can sort the process status information in the host information card in ascending or descending order by clicking the **CPU**, **USER**, **pctCPU**, **pctMEM**, and **cpuTIME** column headers.
7. The pin in the upper right corner of a loaded host information card allows you to keep this card available on the screen for possible comparison with other hosts later.

- **To pin down an active host information card, simply click on the pin.** The Splunk App for Unix and Linux highlights the pin, and the information card remains on the dashboard when you select another host. You can only pin a host information card **after** it has finished loading its data.
- **To unpin a host information card, click on the pin again.** The Splunk App for Unix and Linux removes the highlight from the pin. When you click on another host, the host information card updates with the new host's information.
• To compare multiple hosts, follow the instructions in "Compare performance metrics on hosts" later in this topic.

**Note:** Information displayed on host information cards does not update in real time. It represents a snapshot of the host's state when the card was opened. To see side-by-side comparisons of hosts in real time, use list view.

**Compare performance metrics on hosts**

This node view-only feature allows you to compare detailed performance metrics for any number of hosts in real time.

To use the comparison feature:

1. Follow the instructions in "Get more information about a host" to pin the host information card.

   **Important:** you must wait until all information on the host information card has updated before you can pin it and click on another host. Otherwise, the host information card will be updated using this newly clicked host.

2. Repeat this process to add up to four additional hosts.

   The Splunk App for Unix and Linux overlays these hosts on top of hosts that you have already pinned.

3. Once you have chosen the hosts you want to compare, click the **Compare** button above the host information card.

   The Splunk App for Unix and Linux darkens the Hosts dashboard view and displays the host information cards of all of the pinned hosts side-by-side.

   To exit out of this comparison view, click anywhere outside of the displayed cards.

   To remove pins from all pinned host information cards, click the **Unpin all** button above the cards.

**Clear all customized host display information**

To clear all customizations you have made to the Hosts dashboard, click the **Clear** button on the upper right side of the dashboard.
Use the Alerts dashboard

The Alerts dashboard gives you information on the alerts that the Splunk App for Unix and Linux has triggered, when those alerts triggered, and which hosts the alerts have triggered on. It also displays alert severity (as has been configured for each alert in the Settings: Alerts dialog.)

How the Alerts dashboard works: An example scenario

Following is an example scenario on how the Alerts dashboard works:

It's 10am on Monday and, as the data center manager for your enterprise, you receive a report of a system outage the previous night.

To investigate what went wrong, you open the Splunk App for Unix and Linux and review the alerts that triggered overnight in the Alerts dashboard. First, you click the time range picker in the Alert Time Range panel and select "Last 24 hours" because you know from the report that the outage occurred within that period. The Splunk App for Unix and Linux updates the page to show alerts that have triggered in the last 24 hours. You notice that a large number of alerts occurred around 1:30am that morning.

You click and drag in the area on the Alert Time Range where the spike of events is, and the Splunk App for Unix and Linux updates the Statistics and Summary panels to show alerts that occurred in that timeframe. There you discover that all of your application servers triggered Memory_Exceeds_Percent_by_Host alerts. You click an alert link in the Summary panel and the Splunk App for Unix and Linux opens a detailed screen with information about when the alert triggered, the host that triggered the alert, and a snapshot of CPU, memory, process, and commands that were running at the time the alert fired. Using this panel, you find out that something caused your application servers to consume all available memory and crash. This coincides with the report that services went offline.

You take screenshots of the failure and email the engineering, software development, and management teams with the details. Soon afterward, the software development team acknowledges that the latest code changes might have introduced a bug which, in certain circumstances, causes application servers to exhaust all available memory. They roll back the change and, after a few days of tests, find and fix the memory exhaustion bug. Soon afterward, they roll out updated code to the application servers with no adverse effects. The Splunk App for Unix and Linux helped resolve and prevent future outages.
Alerts dashboard overview

The Alerts dashboard splits into three panels:

- The Alert time range panel on the top displays a timeline that shows the number of alerts that have arrived within a given time period. You can also select a custom time period from the time range picker.

- The Statistics panel in the lower left displays information about which hosts have triggered alerts, which alerts have triggered, and the severity of those alerts. You can drill down into specifics about hosts that triggered alerts and find out how many alerts the host triggered.

- The Summary panel on the lower right shows a listing of the most recent alerts that have triggered.

Choose the alert time range

There are two ways to choose the alert time range - the window of time that you want to see alerts that have triggered:

1. You can use the time range picker on the upper right corner of the Alert Time Range panel. Select the desired time range the time range picker.

   The Splunk App for Unix and Linux updates the Statistics and Summary panel to include only events that have occurred within the desired time period (from the chosen point in the past up to now.)

2. You can additionally use your mouse to select a period of time in the Alert Time Range panel to show only those alerts that have triggered in that time period.

   Handles appear around the edges of the selected range.

   You can then adjust the range by selecting the handles and dragging them left or right to narrow or widen the range. The Splunk App for Unix and Linux updates the Statistics and Summary panels to show alerts that have fired in the new time range.

To remove the constraint, double-click anywhere outside the selected time range.

Statistics

The Statistics panel displays three donut charts which show:

- The number of Hosts that have fired alerts in the time range selected in the Alert Time Range panel.
- The Names of the alerts that have fired in this time period.
- The Severity of the alerts that have triggered in this period.

Each donut chart divides into different color slices depending on how many hosts, alerts, or severity levels are present in the selected time range.

You can get specific information about a single host by clicking one of the color slices in the Hosts donut chart. The donut chart updates to show you how many alerts that host triggered during the selected time range, and the Summary panel updates to show information on alerts that include the selected host.

Similarly, the Name donut chart allows you to filter which alerts have fired. When you click on a donut chart slice for a specific alert, the chart updates to show you how many times that alert has fired in the selected time range. The Summary
The **Severity** donut chart allows you to filter alerts based on severity. When you click a slice in that chart, the chart updates to show the number of times that alerts of the selected severity level have triggered in the selected time frame. The Summary panel also updates with only alerts of the selected severity level.

You can reset the filter for each donut chart by clicking the reset link inside each chart.

**Summary**

The Summary panel shows you information about the alerts that have triggered in a specific time range which you select using the Alert Time Range picker or custom range selector, as well what you filter by using the Statistics donut charts.

For the selected time range and filter level, it displays:

- The time the alert fired.
- The name of the alert that fired.
- The alert's severity.
- The host(s) which triggered the alert.
- A link which allows you to open the underlying search which fired the alert.
- A description of the alert.

The Summary panel displays 10 alerts per page by default. You can see earlier alerts by using the pagination links on the upper right corner of the Summary panel.

You can sort each column in the Summary panel in ascending or descending order by clicking on the link in the column.

**Get details on an alert**
The **Name** column in the Alert Summary shows a list of the names of the alerts that have fired in the selected time range. When you click on the name for a specific alert, the Splunk App for Unix and Linux opens a page that contains detailed information about the specific alert. It lists:

- The time that the alert fired.
- A description of the alert.
- The alert's severity.
- A list of hosts that triggered the alert at that time. You can select the other hosts that triggered the same alert at the same time.
- Graphs that show historical information about CPU usage, memory usage, number of processes, and number of threads around the time that the alert fired. You can click each graph to get search results that power the graph.
- A **System Status** subpanel that shows statistics on commands that were running at the time the alert fired. You can select the available commands and sort them by various statistics.

To close the information page on the alert, click anywhere on the screen outside of the alert page.
Troubleshoot the Splunk App for Unix and Linux

You can troubleshoot your Splunk App for Unix and Linux deployment if you are experiencing errors or if you are not seeing the data that you expect.

The bubble color differs from the actual value

The bubble in the chart shows the value of the selected parameter from the dropdowns. The color bar sets the color of the bubble, and the color bar shows a value between 1 to 100. If the bubble value is greater than 100, then the value is scaled to keep the number under 100.

Error about the unix_summary index when enabling alerts

This error occurs when you are running a version 5.2.1 or earlier of the Splunk App for Unix and Linux and have not distributed the indexes.conf file to all indexers in your instance. Alerts require this index to function correctly.

Missing or invalid dropdowns.csv

This error occurs when you skip the first-time configuration screen. Complete these two steps to fix it:

1. Configure the app by selecting Settings from the main app menu.
2. From the Settings screen, select Categories.

CPU information is not displaying

This error occurs when the sysstat package is not installed on the system that hosts the app. Use your system's package manager to install the package.

Ubuntu systems do not ship with this package by default. Run the following command to add it: `apt-get install sysstat`

Amazon EC2 Amazon Machine Image (AMI) systems also do not ship with this package installed by default. Run the following command to add it: `yum -y install sysstat`

Home and Metrics views do not display data

If your Home and Metrics views do not display any data, navigate to the web.conf file on the Splunk platform instance that runs the app and add the following stanza:

```
[settings]
minify_js = True
```

If you set minify_js to False, views do not load.

Split pctCPU

The value of pctCPU calculates across all CPU, and not per individual cores. Use searches such as the following to split pctCPU into smaller units:
<table>
<thead>
<tr>
<th>Search</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>tag=cpu</td>
<td>stats avg(pctUser)</td>
</tr>
<tr>
<td>tag=cpu</td>
<td>stats avg(pctUser) by CPU</td>
</tr>
<tr>
<td>tag=cpu CPU=1</td>
<td>stats avg(pctUser) by CPU</td>
</tr>
</tbody>
</table>

Unable to change colors in radial graph on the Home Dashboard

if you move down the second color picker, and cross it with the first color picker, the bottom-most color does not update due to a technical limitation.

To reflect the changes, refresh the page.
Advanced topics

Create custom alerts

The Splunk App for Unix and Linux comes with twelve alerts which you can configure in the Settings: Alerts dialog. If you want, you can add custom alerts by saving searches and adding specific parameters to make them also appear in the Settings: Alerts dialog. This topic shows you how to configure custom alerts and prepare them for use in the Splunk App for Unix and Linux's alert system.

Build and configure custom alerts

The alerts that appear in the Settings: Alerts window are saved searches with a special field added. To add additional alerts and have them appear here, perform the following steps:

1. While in the context of the Splunk App for Unix and Linux, create and save a search with the desired parameters that comprise an alert. (You can access the search page by clicking Search on the navigation bar.) Important: Your custom search must include language that splits its results by the host field. For example:

   stats(CPU) by host

2. Save the search.
3. Go into Splunk Settings.
   ♦ In Splunk version 5, choose Manager from the upper right on the navigation bar.
   ♦ In Splunk version 6, choose Settings.
4. Choose Searches and reports
5. Locate the search you just created and saved and click its name in the list. Splunk opens the configuration settings for the search.
6. In the Schedule and Alert section, click the Schedule this search checkbox.
7. Make sure that the Alert condition is set to Always.
8. Enable summary indexing for the alert by clicking Enable under the Summary Indexing section.
9. In the Add fields text boxes, add the following field: marker = unix_aggregated_alerts
10. Click Save to save the changes to the search.

When you next visit the Settings: Alerts dialog, you should see the custom alert in the list.

Saved searches

The Splunk App for Unix and Linux includes a number of saved searches that it uses to populate the Home, Metrics, Hosts and Alerts dashboards. This topic lists the searches by category and provides a description of what the searches do.

CPU searches

<table>
<thead>
<tr>
<th>Saved search</th>
<th>Intended purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Percent CPU by Host (UNIX - CPU)</td>
<td>Returns per-host CPU usage percentage events. Uses the <code>Percent_CPU_by_Host(*)</code> macro.</td>
</tr>
<tr>
<td>Percent Load by Host (UNIX - CPU)</td>
<td>Returns per-host CPU load average events. Uses the <code>Percent_Load_by_Host(*)</code> macro.</td>
</tr>
</tbody>
</table>
### Saved search | Intended purpose
---|---
Top 5 CPU Processes by Host (UNIX - CPU) | Returns the top five processes, based on CPU usage, per host. Uses the `Top_5_CPU_Processes_by_Host(*)` macro.
Number of Threads by Host (UNIX - CPU) | Returns the number of threads in use for each host. Uses the `Number_Threads_by_Host(*)` macro.
Number of Processes by Host (UNIX - CPU) | Returns the number of active processes on each host. Uses the `Number_Processes_by_Host(*)` macro.
CPU Usage by Command (UNIX - CPU) | Returns per-command CPU usage events for a single host. Uses the `CPU_Usage_by_Command_for_Host(*)` macro.
CPU Usage by User (UNIX - CPU) | Returns per-user CPU usage for a single host. Uses the `CPU_Usage_by_User_for_Host(*)` macro.
Usage by State (UNIX - CPU) | Returns CPU usage by state for a single host. Uses the `CPU_Usage_by_State_for_Host(*)` macro.
Top CPU Processes for Host (UNIX - CPU) | Returns the top processes based on CPU usage for a single host. Uses the `Top_CPU_Processes_for_Host(*)` macro.
Consumption by User Last Hour (UNIX - CPU) | Returns the amount of CPU used by each user within the last hour. Uses the `os_index` macro and the "ps" source.
Top Users by Consumption Last Hour (UNIX - CPU) | Returns the amount of CPU time used by each user within the last hour. Uses the `os_index` macro and the "ps" source.
10 Most Popular Executables Last Hour (UNIX - CPU) | Returns the top 10 processes by name in the last hour. Uses the `os_index` macro and the "lsof" source.

### Memory searches

<table>
<thead>
<tr>
<th>Saved search</th>
<th>Intended purpose</th>
</tr>
</thead>
</table>
Mem Usage for Host (UNIX - MEM) | Returns per-host memory usage, per host. Uses the `Mem_Usage_for_Host(*)` macro. |
Mem Usage by Command for Host (UNIX - MEM) | Returns per-host memory usage by command, per host. Uses the `Mem_Usage_by_Command_for_Host(*)` macro. |
Top Mem Usage Commands for Host (UNIX - MEM) | Returns the top processes, based on memory usage, per host. Uses the `Top_Mem_Command_for_Host(*)` macro. |
Top 10 Users by Resident Memory Last Hour (UNIX - MEM) | Returns the top 10 users, based on memory usage, per host. Uses the `Top_Users_of_VM_for_Host(*)` macro. |
Mem Usage by host | Returns the amount of memory used for each host. Uses the `Percent_MEM_by_Host(1)` macro. |
Top Commands by Memory and Host (UNIX - MEM) | Returns the top 10 commands, based on memory usage, per host. Uses the `Top_Mem_Processes_by_Host(*)` macro. |
Physical Memory by Host (UNIX - MEM) | Returns the amount of physical memory installed, per host. Uses the `Memory_Hardware_by_Host(*)` macro. |
Top_Memory_Users_by_Command_by_Host | Returns the top memory users, by command, per host. Uses the `Top_Memory_Users_by_Command_by_Host(*)` macro. |
## Disk Searches

<table>
<thead>
<tr>
<th>Saved search</th>
<th>Intended purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Percent Disk Used by Volume and Host (UNIX - Disk)</td>
<td>Returns the amount of disk used by each accessible volume, per host. Uses the <code>Disk_Used_Pct_by_Host(*)</code> macro.</td>
</tr>
<tr>
<td>Files Opened by Command (UNIX - Disk)</td>
<td>Returns the number of files opened per command. Uses the <code>Open_Files_by_Command_and_Host(*)</code> macro.</td>
</tr>
<tr>
<td>Files Opened by Type (UNIX - Disk)</td>
<td>Returns the number of files opened, by type. Uses the <code>Open_Files_by_Type_and_Host(*)</code> macro.</td>
</tr>
</tbody>
</table>

## Sources

<table>
<thead>
<tr>
<th>Saved search</th>
<th>Intended purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>vmstat</td>
<td>Retrieves virtual memory states. Relies on the <code>os_index</code> and <code>memory_sourcetype</code> macros.</td>
</tr>
<tr>
<td>ps</td>
<td>Retrieves information about executing processes. Relies on the <code>os_index</code> and <code>ps_sourcetype</code> macros.</td>
</tr>
<tr>
<td>top</td>
<td>Retrieves events from the &quot;top&quot; process. Relies on the <code>os_index</code> and <code>top_sourcetype</code> macros.</td>
</tr>
<tr>
<td>hardware</td>
<td>Retrieves information about the hardware installed in a host. Relies on the <code>os_index</code> and <code>hardware_sourcetype</code> macros.</td>
</tr>
<tr>
<td>iostat</td>
<td>Retrieves information from the &quot;iostat&quot; process. Relies on the <code>os_index</code> and <code>iostat_sourcetype</code> macros.</td>
</tr>
<tr>
<td>netstat</td>
<td>Retrieves information from the &quot;netstat&quot; process. Relies on the <code>os_index</code> and <code>netstat_sourcetype</code> macros.</td>
</tr>
<tr>
<td>protocol</td>
<td>Retrieves information about network protocols installed on the system. Relies on the <code>os_index</code> and <code>protocol_sourcetype</code> macros.</td>
</tr>
<tr>
<td>openPorts</td>
<td>Retrieves information about the open network ports on a system. Relies on the <code>os_index</code> and <code>open_ports_sourcetype</code> macros.</td>
</tr>
<tr>
<td>time</td>
<td>Retrieves information about the system time. Relies on the <code>os_index</code> and <code>time_sourcetype</code> macros.</td>
</tr>
<tr>
<td>lsof</td>
<td>Retrieves information about all open files on the system. Relies on the <code>os_index</code> and <code>lsof_sourcetype</code> macros.</td>
</tr>
<tr>
<td>df</td>
<td>Retrieves information about disk usage on the system. Relies on the <code>os_index</code> and <code>df_sourcetype</code> macros.</td>
</tr>
<tr>
<td>who</td>
<td>Retrieves information from the &quot;who&quot; command. Relies on the <code>os_index</code> and <code>who_sourcetype</code> macros.</td>
</tr>
<tr>
<td>usersWithLoginPrives</td>
<td>Retrieves information on users who can log into the host. Relies on the <code>os_index</code> and <code>users_with_login_prives_sourcetype</code> macros.</td>
</tr>
<tr>
<td>lastlog</td>
<td>Retrieves information on who has last logged into the system. Relies on the <code>os_index</code> and <code>lastlog_sourcetype</code> macros.</td>
</tr>
<tr>
<td>interfaces</td>
<td>Gathers information on the network interfaces on the system. Relies on the <code>os_index</code> and <code>interfaces_sourcetype</code> macros.</td>
</tr>
<tr>
<td>cpu</td>
<td>Gathers information about the system's CPU. Relies on the <code>os_index</code> and <code>cpu_sourcetype</code> macros.</td>
</tr>
<tr>
<td>rlog</td>
<td>Gathers information from the &quot;rlog&quot; command. Relies on the <code>os_index</code> and <code>rlog_sourcetype</code> macros.</td>
</tr>
<tr>
<td>package</td>
<td>Gathers information about the software packages that the system has installed on it. Relies on the <code>os_index</code> and <code>package_sourcetype</code> macros.</td>
</tr>
</tbody>
</table>
### User Searches

<table>
<thead>
<tr>
<th>Saved search</th>
<th>Intended purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>User Sessions</td>
<td>Total number of user sessions, per host. Uses the <code>User_Sessions_by_Host(*)</code> macro.</td>
</tr>
<tr>
<td>Failed Logins</td>
<td>Total number of failed logins, per host. Uses the <code>Failed_Logins_by_Host(*)</code> macro.</td>
</tr>
<tr>
<td>User Add</td>
<td>Total number of user adds for a host. Uses the <code>os_index</code> and <code>user_add</code> macros.</td>
</tr>
<tr>
<td>User Delete</td>
<td>Total number of user deletes for a host. Uses the <code>os_index</code> and <code>user_del</code> macros.</td>
</tr>
<tr>
<td>Group Add</td>
<td>Total number of group adds for a host. Uses the <code>os_index</code> and <code>group_add</code> macros.</td>
</tr>
<tr>
<td>Group Delete</td>
<td>Total number of group deletes for a host. Uses the <code>os_index</code> and <code>group_del</code> macros.</td>
</tr>
<tr>
<td>Password Change</td>
<td>Total number of password changes for a host. Uses the <code>os_index</code> and <code>password_change</code> macros.</td>
</tr>
<tr>
<td>Password Change Failed</td>
<td>Total number of failed password changes for a host. Uses the <code>os_index</code> and <code>password_change_failed</code> macros.</td>
</tr>
<tr>
<td>Failed Attempts at SU</td>
<td>Total number of times where a user attempted and failed to become the superuser. Uses the <code>os_index</code> and <code>su_failed</code> macros.</td>
</tr>
</tbody>
</table>

### Network Searches

<table>
<thead>
<tr>
<th>Saved search</th>
<th>Intended purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Thruput by Interface and Host</td>
<td>The amount of network throughput, by interface and host. Uses the <code>Thruput_by_Interface_by_Host(*)</code> macro.</td>
</tr>
<tr>
<td>Frequently Opened Ports (UNIX - NET)</td>
<td>A list of the most frequently opened network ports. Uses the <code>Frequently_Open_Ports_by_Host(*)</code> macro.</td>
</tr>
<tr>
<td>Top Inet Addresses by Host</td>
<td>Uses the <code>Top_Inet_Addresses_by_Host(*)</code> macro.</td>
</tr>
<tr>
<td>Open Ports (UNIX - NET)</td>
<td>Uses the <code>Open_Ports_by_Host(*)</code> macro.</td>
</tr>
<tr>
<td>Addresses Connected To (UNIX - NET)</td>
<td>Uses the <code>Addresses_by_Host(*)</code> macro.</td>
</tr>
<tr>
<td>Sockets by State (UNIX - NET)</td>
<td>Uses the <code>Sockets_by_State_by_Host(*)</code> macro.</td>
</tr>
<tr>
<td>Top 10 Users by Virtual Memory Last Hour (UNIX - MEM)</td>
<td>The top 10 users, by virtual memory usage, in the last hour. Uses the <code>os_index</code> and <code>ps_sourcetype</code> macros.</td>
</tr>
<tr>
<td>Virtual Memory Subsystem Stats (UNIX - MEM)</td>
<td>Information about a system's memory usage. Uses the <code>os_index</code> and <code>memory_sourcetype</code> macros.</td>
</tr>
<tr>
<td>Memory Usage over Last 3 Hours (UNIX - MEM)</td>
<td>Uses the <code>os_index</code> and <code>memory_sourcetype</code> macros.</td>
</tr>
<tr>
<td>Avg Resident Memory by Process Last 3 Hours (UNIX - MEM)</td>
<td>Uses the <code>os_index</code> and <code>ps_sourcetype</code> macros.</td>
</tr>
<tr>
<td>Avg Virtual Memory by Process Last 3 Hours (UNIX - MEM)</td>
<td>Uses the <code>os_index</code> and <code>ps_sourcetype</code> macros.</td>
</tr>
</tbody>
</table>

### Package Searches

<table>
<thead>
<tr>
<th>Saved search</th>
<th>Intended purpose</th>
</tr>
</thead>
</table>

42
Latest Packages by Host
A list of the installed packages, per host. Uses the `os_index` and `package_sourcetype` macros.

Hardware Configurations by Host
A detailed list of hardware configurations, per host. Uses the `os_index` `hardware_sourcetype` macros.

Utility Saved Searches

<table>
<thead>
<tr>
<th>Saved search</th>
<th>Intended purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>UNIX - All Logs</td>
<td>Gathers all available *nix logs that have been indexed. Uses the <code>os_index</code> macro.</td>
</tr>
<tr>
<td>UNIX - All Configs</td>
<td>Returns all *nix configuration events. Uses the <code>os_index</code> macro.</td>
</tr>
<tr>
<td>UNIX - Timechart Errors Or</td>
<td>Returns a chart of all 'critical' or 'error' level messages. Uses the <code>os_index</code> and <code>unix_errors</code> macros.</td>
</tr>
<tr>
<td>Critical</td>
<td></td>
</tr>
<tr>
<td>UNIX - Timechart Config Changes</td>
<td>Returns a chart of all *nix configuration changes. Uses the *nix_configs event type.</td>
</tr>
</tbody>
</table>

Alerts
These alerts come with the Splunk App for Unix and Linux. You can also create additional custom alerts.

<table>
<thead>
<tr>
<th>Saved search</th>
<th>Intended purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alert - syslog errors last hour</td>
<td>Returns syslog events of type 'error'. Uses the <code>syslog_sourcetype</code> and <code>syslog_errors</code> macros. Runs once an hour by default.</td>
</tr>
<tr>
<td>Memory_Exceeds_MB_by_Process</td>
<td>Triggers when memory usage for processes exceeds a certain level. Returns events per process. Uses the Memory_Exceeds_MB_by_Process macro. Runs every 5 minutes.</td>
</tr>
<tr>
<td>Memory_Exceeds_Percent_by_Host</td>
<td>Triggers when per-host memory usage exceeds a certain percentage. Returns events per host. Uses the Memory_Exceeds_Percent_by_Host macro. Runs every 5 minutes.</td>
</tr>
<tr>
<td>Memory_Exceeds_MB_by_Host</td>
<td>Triggers when per-host memory usage exceeds a certain level. Returns events per host. Uses the Memory_Exceeds_MB_by_Host macro. Runs every 5 minutes.</td>
</tr>
<tr>
<td>CPU_Exceeds_Percent_by_Host</td>
<td>Triggers when per-host CPU usage exceeds a certain percentage. Returns events per host. Uses the CPU_Exceeds_Percent_by_Host macro. Runs every 5 minutes.</td>
</tr>
<tr>
<td>CPU_Under_Percent_by_Host</td>
<td>Triggers when per-host CPU usage remains below a certain percentage. Returns events per host. Uses the CPU_Under_Percent_by_Host macro. Runs every 5 minutes.</td>
</tr>
<tr>
<td>Load_Exceeds_by_Host</td>
<td>Triggers when per-host load averages exceed a certain level. Returns events per host. Uses the Load_Exceeds_by_Host macro. Runs every 5 minutes.</td>
</tr>
<tr>
<td>Threads_Exceeds_by_Host</td>
<td>Triggers when per-host thread counts exceed a certain level. Returns events per host. Uses the Threads_Exceeds_by_Host macro. Runs every 5 minutes.</td>
</tr>
<tr>
<td>Processes_Exceeds_by_Host</td>
<td>Triggers when per-host process counts exceed a certain level. Returns events per host. Uses the Processes_Exceeds_by_Host macro. Runs every 5 minutes.</td>
</tr>
<tr>
<td>Disk_Used_Exceeds_Perc_by_Host</td>
<td>Triggers when per-host disk usage exceeds a certain percentage. Returns events per host. Uses the Disk_Used_Exceeds_Perc_by_Host macro. Runs every 5 minutes.</td>
</tr>
<tr>
<td>Open_Files_Exceeds_by_Process</td>
<td>Triggers when per-process open file counts exceed a certain level. Returns events per process. Uses the Open_Files_Exceeds_by_Process macro. Runs every 5 minutes.</td>
</tr>
<tr>
<td>IO_Wait_Exceeds_Threshold</td>
<td>Triggers when the amount of system I/O wait time exceeds a certain level. Returns events per host. Uses the IO_Wait_Exceeds_Threshold macro. Runs every 5 minutes.</td>
</tr>
<tr>
<td>IO_Utilization_Exceeds_Threshold</td>
<td>Triggers when the amount of system I/O utilization exceeds a certain level. Returns events per host. Uses the IO_Utilization_Exceeds_Threshold macro. Runs every 5 minutes.</td>
</tr>
</tbody>
</table>
Home screen (regular and full screen)

The following searches power the Home screens with information about categories and groups that you have defined in the configuration settings.

<table>
<thead>
<tr>
<th>Saved search</th>
<th>Intended purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dropdown Lookup - Dimension</td>
<td>Populates the Category drop-down list. Uses the dropdowns.csv lookup table.</td>
</tr>
<tr>
<td>Dropdown Lookup - Group</td>
<td>Populates the Group drop-down list based on the Category you have selected. Uses the dropdowns.csv lookup table.</td>
</tr>
</tbody>
</table>

Metrics screen

<table>
<thead>
<tr>
<th>Saved search</th>
<th>Intended purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Metrics Selectable Lookup</td>
<td>Populates the Metrics viewer page with categories, groups, and host information. Uses the dropdowns.csv lookup table.</td>
</tr>
</tbody>
</table>

Lookups

<table>
<thead>
<tr>
<th>Saved search</th>
<th>Intended purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>__generate_lookup_dropdowns</td>
<td>Creates the dropdowns.csv lookup table by searching collected data for the top 50 hosts (by index time).</td>
</tr>
<tr>
<td>__safeguard_generate_lookup_drops</td>
<td></td>
</tr>
</tbody>
</table>

Search macros

The Splunk App for Unix and Linux includes a variety of search macros that can be used to create custom searches and notable events.

The back ticks (`) denote the start and the end of a search macro definition when used in the Splunk search language. The values (<timestamp>) following the search macro name denote the type and number of arguments used with the macro. Overloaded macros are macros with the same name, but a different number of required arguments.

To learn more about the syntax used in macros, see "Define search macros in Settings" and "macros.conf" in the core Splunk platform documentation.

Base macros

The following table lists the base search macros for the Splunk App for Unix and Linux. The app uses these macros to easily call up data that is stored in many indexes or has been tagged with many sourcetypes.

<table>
<thead>
<tr>
<th>Search macro</th>
<th>Intended purpose</th>
<th>Expected data types</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>os_index</code></td>
<td>Used to retrieve events from the os index.</td>
<td></td>
</tr>
<tr>
<td><code>cpu_sourcetype</code></td>
<td>Returns cpu metric events that have a sourcetype of <code>cpu</code>.</td>
<td>system access logs, such as ssh, Windows, or database audit.</td>
</tr>
<tr>
<td>Search macro</td>
<td>Intended purpose</td>
<td>Expected data types</td>
</tr>
<tr>
<td>---------------------</td>
<td>----------------------------------------------------------------------------------</td>
<td>-----------------------------------------------</td>
</tr>
<tr>
<td><code>df_sourcetype</code></td>
<td>Returns disk-space related events that have a sourcetype of 'df'</td>
<td>system audit logs, such as Active Directory or OpenLDAP.</td>
</tr>
<tr>
<td><code>hardware_sourcetype</code></td>
<td>Returns hardware related events that have a sourcetype of 'hardware'.</td>
<td>Special user accounts table and system access logs.</td>
</tr>
<tr>
<td><code>interfaces_sourcetype</code></td>
<td>Returns network interface events that have a sourcetype of 'interfaces'.</td>
<td></td>
</tr>
<tr>
<td><code>iostat_sourcetype</code></td>
<td>Returns i/o statistics events that have a sourcetype of 'iostat'.</td>
<td></td>
</tr>
<tr>
<td><code>lastlog_sourcetype</code></td>
<td>Returns last login events that have a sourcetype of 'lastlog'.</td>
<td></td>
</tr>
<tr>
<td><code>lsf_sourcetype</code></td>
<td>Returns events that have a sourcetype of 'lsf' - a list of open files on the system.</td>
<td></td>
</tr>
<tr>
<td><code>memory_sourcetype</code></td>
<td>Returns memory-related events that have a sourcetype of 'memory'.</td>
<td></td>
</tr>
<tr>
<td><code>netstat_sourcetype</code></td>
<td>Returns network statistics events that have a sourcetype of 'netstat'.</td>
<td></td>
</tr>
<tr>
<td><code>open_ports_sourcetype</code></td>
<td>Returns events about open network ports.</td>
<td></td>
</tr>
<tr>
<td><code>package_sourcetype</code></td>
<td>Returns events about the installation and uninstallation of software packages on the system.</td>
<td></td>
</tr>
<tr>
<td><code>protocol_sourcetype</code></td>
<td>Returns network protocol-related events.</td>
<td></td>
</tr>
<tr>
<td><code>ps_sourcetype</code></td>
<td>Returns events about the status of running processes.</td>
<td></td>
</tr>
<tr>
<td><code>rlog_sourcetype</code></td>
<td>Returns remote login-related events.</td>
<td></td>
</tr>
<tr>
<td><code>syslog_sourcetype</code></td>
<td>Returns system log-related events.</td>
<td></td>
</tr>
<tr>
<td><code>time_sourcetype</code></td>
<td>Returns events generated by the 'time' command - the amount of time that processes take to complete on a system.</td>
<td></td>
</tr>
<tr>
<td><code>top_sourcetype</code></td>
<td>Returns events generated by the 'top' command - real-time statistics of all processes on a system.</td>
<td></td>
</tr>
<tr>
<td><code>users_with_login_privs_sourcetype</code></td>
<td>Returns events concerning users who have the ability to log into the system.</td>
<td></td>
</tr>
<tr>
<td><code>who_sourcetype</code></td>
<td>Returns 'who'-related events - information about the users currently logged in to the system.</td>
<td></td>
</tr>
</tbody>
</table>

**Utilities and aliases**

The Splunk App for Unix and Linux uses these macros to refer to common types of events. This makes it easier for the app to recognize certain events like error conditions.

<table>
<thead>
<tr>
<th>Search macro</th>
<th>Intended purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>eval_host_group</code></td>
<td>Returns all events where the event type is 'groupadd' or 'groupadd_suse'.</td>
</tr>
<tr>
<td><code>group_add</code></td>
<td>Returns all events where the event type is 'groupadd'.</td>
</tr>
<tr>
<td><code>group_del</code></td>
<td>Returns all events where the event type is 'groupdel'.</td>
</tr>
<tr>
<td><code>password_change</code></td>
<td>Returns all events where the event type is 'linux-password-change'.</td>
</tr>
<tr>
<td><code>password_change_failed</code></td>
<td>Returns all events where the event type is 'linux-password-change-failed'.</td>
</tr>
<tr>
<td><code>su_failed</code></td>
<td>Returns all events where the event type is 'su_failed'.</td>
</tr>
<tr>
<td>Search macro</td>
<td>Intended purpose</td>
</tr>
<tr>
<td>--------------</td>
<td>-----------------</td>
</tr>
<tr>
<td><code>syslog_errors</code></td>
<td>Returns all events whose text matches one of 'error', 'failed', 'severe,' but not 'assignment'.</td>
</tr>
<tr>
<td><code>unix_errors</code></td>
<td>Returns events where the event type is 'nix_errors'.</td>
</tr>
<tr>
<td><code>user_add</code></td>
<td>Returns events where the event type is either 'useradd' or 'useradd_suse'.</td>
</tr>
<tr>
<td><code>user_del</code></td>
<td>Returns events where the event type is 'userdel'.</td>
</tr>
<tr>
<td><code>parse_disk_size(1)</code></td>
<td>Parses the size of a disk based on a supplied disk event format.</td>
</tr>
</tbody>
</table>

### Host node macros

<table>
<thead>
<tr>
<th>Search macro</th>
<th>Intended purpose</th>
<th>Expected data types</th>
</tr>
</thead>
<tbody>
<tr>
<td>unix_host_status</td>
<td>Returns a table of the current status of *nix hosts. Uses the <code>os_index</code>, <code>cpu_sourcetype</code>, and <code>eval_host_group</code> macros.</td>
<td>Host data, CPU statistics</td>
</tr>
<tr>
<td>unix_hosts_status(2)</td>
<td>Returns a table of the current status of *nix hosts, by group and category. Uses the <code>os_index</code>, <code>cpu_sourcetype</code>, and <code>eval_host_group</code> macros. Requires a category and group as arguments.</td>
<td>Host data, CPU statistics</td>
</tr>
<tr>
<td>unix_hosts_details(2)</td>
<td>Returns a table of detailed information (CPU, memory, disk, I/O stats) for a set of *nix hosts. Uses the <code>cpu_sourcetype</code>, <code>memory_sourcetype</code>, <code>df_sourcetype</code>, <code>iostat_sourcetype</code>, and <code>eval_host_group</code> macros. Requires a category and group as arguments.</td>
<td>Host data, CPU, memory, I/O, and disk statistics</td>
</tr>
<tr>
<td>unix_host_details</td>
<td>Returns a table of detailed information (CPU, memory, disk, I/O stats) for a set of *nix hosts. Uses the <code>cpu_sourcetype</code>, <code>memory_sourcetype</code>, <code>df_sourcetype</code>, <code>iostat_sourcetype</code>, and <code>eval_host_group</code> macros. Requires a category and group as arguments.</td>
<td>Host data, CPU, memory, I/O, and disk statistics</td>
</tr>
<tr>
<td>unix_nodes_heatmap_cpu</td>
<td>Generates the CPU heat map statistics. Uses the <code>os_index</code> and <code>cpu_sourcetype</code> macros.</td>
<td>Host data, Memory statistics</td>
</tr>
<tr>
<td>unix_nodes_heatmap_disk</td>
<td>Generates the disk usage heat map statistics. Uses the <code>os_index</code> and <code>df_sourcetype</code> macros.</td>
<td>Host data, Disk statistics</td>
</tr>
<tr>
<td>unix_nodes_heatmap_io</td>
<td>Generates the I/O heat map statistics. Uses the <code>os_index</code> and <code>iostat_sourcetype</code> macros.</td>
<td>Host data, I/O statistics</td>
</tr>
<tr>
<td>unix_nodes_detail_specs_cpu_by_host(1)</td>
<td>Returns detailed CPU specifications for a given host. Uses the <code>os_index</code> and <code>cpu_sourcetype</code> macros. Requires a host as an argument.</td>
<td>Host data, CPU statistics</td>
</tr>
<tr>
<td>unix_nodes_detail_specs_mem_by_host(1)</td>
<td>Returns detailed memory specifications for a given host. Uses the <code>os_index</code> and <code>memory_sourcetype</code> macros. Requires a host as an argument.</td>
<td>Host data, Memory statistics</td>
</tr>
<tr>
<td>unix_nodes_detail_specs_disk_drives_by_host(1)</td>
<td>Returns detailed disk specifications (number of volumes installed/available) for a given host. Uses the <code>os_index</code> and <code>df_sourcetype</code> macros. Requires a host as an argument.</td>
<td>Host data, CPU statistics</td>
</tr>
<tr>
<td>unix_nodes_detail_specs_disk_cap_by_host(1)</td>
<td>Returns detailed overall disk capacity for a given host. Uses the <code>os_index</code> and <code>disk_sourcetype</code> macros. Requires a host as an argument.</td>
<td>Host data, Disk statistics</td>
</tr>
<tr>
<td>unix_nodes_detail_status_cpu_by_host(1)</td>
<td>Returns detailed CPU statistics for a given host. Uses the <code>os_index</code> and <code>cpu_sourcetype</code> macros. Requires a host as an argument.</td>
<td>Host data, CPU statistics</td>
</tr>
<tr>
<td>unix_nodes_detail_status_mem_by_host(1)</td>
<td>Returns detailed memory statistics for a given host. Uses the <code>os_index</code> and <code>memory_sourcetype</code> macros. Requires a host as an argument.</td>
<td>Host data, Memory statistics</td>
</tr>
<tr>
<td>unix_nodes_detail_status_disk_by_host(1)</td>
<td>Returns detailed disk space statistics for a given host. Uses the <code>os_index</code> and <code>df_sourcetype</code> macros. Requires a host as an argument.</td>
<td>Host data, Disk statistics</td>
</tr>
</tbody>
</table>
## Single host macros

<table>
<thead>
<tr>
<th>Search macro</th>
<th>Intended purpose</th>
<th>Expected data types</th>
</tr>
</thead>
<tbody>
<tr>
<td>unix_nodes_detail_cpu_sparkline_by_host_1h</td>
<td>Generates a spark line based on CPU statistics for a given host over the last hour. Uses the <code>os_index</code> and <code>cpu_sourcetype</code> macros. Requires a host as an argument.</td>
<td>Host data, CPU statistics</td>
</tr>
<tr>
<td>unix_nodes_detail_mem_sparkline_by_host_1h</td>
<td>Generates a spark line based on memory statistics for a given host over the last hour. Uses the <code>os_index</code> and <code>memory_sourcetype</code> macros. Requires a host as an argument.</td>
<td>Host data, memory statistics</td>
</tr>
<tr>
<td>unix_nodes_detail_disk_sparkline_by_host_1h</td>
<td>Generates a spark line based on disk usage statistics for a given host over the last hour. Uses the <code>os_index</code> and <code>df_sourcetype</code> macros. Requires a host as an argument.</td>
<td>Host data, disk statistics</td>
</tr>
<tr>
<td>unix_nodes_detail_top_processes_by_host</td>
<td>Generates a list of the top processes by CPU usage for a host. Uses the <code>os_index</code> macro and the &quot;top&quot; sourcetype. Requires a host as an argument.</td>
<td>Host data, CPU statistics, top sourcetype</td>
</tr>
</tbody>
</table>

## Multiple host macros

<table>
<thead>
<tr>
<th>Search macro</th>
<th>Intended purpose</th>
<th>Expected data types</th>
</tr>
</thead>
<tbody>
<tr>
<td>Percent_CPU_by_Host</td>
<td>Returns a time-series chart of CPU usage statistics, by host. Uses the <code>os_index</code> and <code>cpu_sourcetype</code> macros. Requires a host as an argument.</td>
<td>Host data, CPU statistics</td>
</tr>
<tr>
<td>Percent_Load_by_Host</td>
<td>Returns a time-series chart of CPU load statistics, by host. Uses the <code>os_index</code> and <code>memory_sourcetype</code> macros. Requires a host as an argument.</td>
<td>Host data, Memory statistics</td>
</tr>
<tr>
<td>Top_5_CPU_Processes_by_Host</td>
<td>Returns a list of the top 5 processes, based on CPU usage, by host. Uses the <code>os_index</code> and <code>top_sourcetype</code> macros. Requires a host as an argument.</td>
<td>Host data, CPU statistics, 'top' sourcetype</td>
</tr>
<tr>
<td>Number_Threads_by_Host</td>
<td>Returns a list of the number of active threads per host. Uses the <code>os_index</code> and <code>memory_sourcetype</code> macros. Requires a host as an argument.</td>
<td>Host data, CPU statistics</td>
</tr>
<tr>
<td>Number_Processes_by_Host</td>
<td>Returns a list of the number of active processes per host. Uses the <code>os_index</code> and <code>memory_sourcetype</code> macros. Requires a host as an argument.</td>
<td>Host data, CPU statistics</td>
</tr>
</tbody>
</table>
### Memory macros

**Single host macros**

<table>
<thead>
<tr>
<th>Search macro</th>
<th>Intended purpose</th>
<th>Expected data types</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mem_Usage_for_Host(1)</td>
<td>Returns a time-series chart for memory usage for a host. Uses the <code>os_index</code> and <code>memory_sourcetype</code> macros. Requires a host as an argument.</td>
<td>Host data, memory statistics</td>
</tr>
<tr>
<td>Mem_Usage_by_Command_for_Host(1)</td>
<td>Returns a time-series chart for memory usage, by process, for a host. Uses the <code>os_index</code> and <code>ps_sourcetype</code> macros. Requires a host as an argument.</td>
<td>Host data, CPU statistics, <code>ps</code> sourcetype</td>
</tr>
<tr>
<td>Top_Mem_Command_for_Host(1)</td>
<td>Returns a list of the top processes, based on memory usage, for a host. Uses the <code>os_index</code> and <code>ps_sourcetype</code> macros. Requires a host as an argument.</td>
<td>Host data, CPU statistics, <code>ps</code> sourcetype</td>
</tr>
<tr>
<td>Top_Users_of_VM_for_Host(1)</td>
<td>Returns a time-series chart of virtual memory usage, per user, for a host. Uses the <code>os_index</code> and <code>ps_sourcetype</code> macros. Requires a host as an argument.</td>
<td>Host data, CPU statistics, <code>ps</code> sourcetype</td>
</tr>
</tbody>
</table>

**Multiple host macros**

<table>
<thead>
<tr>
<th>Search macro</th>
<th>Intended purpose</th>
<th>Expected data types</th>
</tr>
</thead>
<tbody>
<tr>
<td>Percent_MEM_by_Host(1)</td>
<td>Returns a time-series chart of memory usage statistics, by host. Uses the <code>os_index</code> and <code>memory_sourcetype</code> macros. Requires a host as an argument.</td>
<td>Host data, Memory statistics</td>
</tr>
<tr>
<td>Top_Mem_Processes_by_Host(1)</td>
<td>Returns a list of the top processes, based on memory usage, by host. Uses the <code>os_index</code> and <code>ps_sourcetype</code> macros. Requires a host as an argument.</td>
<td>Host data, Memory statistics, <code>ps</code> sourcetype</td>
</tr>
<tr>
<td>Memory_Hardware_by_Host(1)</td>
<td>Returns the memory specification for each host. Uses the <code>os_index</code> and <code>hardware_sourcetype</code> macros. Requires a host as an argument.</td>
<td>Host data, Memory and Hardware statistics</td>
</tr>
<tr>
<td>Top_Memory_Users_by_Command_by_Host(1)</td>
<td>Returns a list of the top memory users, by command, for a host. Uses the <code>os_index</code> and <code>ps_sourcetype</code> macros. Requires a host as an argument.</td>
<td>Host data, Memory statistics, <code>ps</code> sourcetype</td>
</tr>
</tbody>
</table>

### Network macros

<table>
<thead>
<tr>
<th>Search macro</th>
<th>Intended purpose</th>
<th>Expected data types</th>
</tr>
</thead>
<tbody>
<tr>
<td>Thruput_by_Interface_by_Host(1)</td>
<td>Returns a time-series chart of network throughput, per interface, per host. Uses the <code>os_index</code> and <code>interfaces_sourcetype</code> macros. Requires a host as an argument.</td>
<td>Host data, Network statistics</td>
</tr>
<tr>
<td>Top_Inet_Addresses_by_Host(1)</td>
<td>Returns a list of the top IP addresses that a host has attempted a network operation on. Uses the <code>os_index</code> and <code>interfaces_sourcetype</code> host. Requires a host as an argument.</td>
<td>Host data, Network statistics</td>
</tr>
<tr>
<td>OpenPorts_by_Host(1)</td>
<td>Returns a list of open TCP ports on a system (with friendly names for most popular ports). Uses the <code>os_index</code> and <code>open_ports_sourcetype</code> macros. Requires a host as an argument.</td>
<td>Host data, Network statistics</td>
</tr>
<tr>
<td>Addresses_by_Host(1)</td>
<td>Returns a list of the number of inbound network connections, by IP address, to a host. Uses the <code>os_index</code> and <code>netstat_sourcetype</code> macros. Requires a host as an argument.</td>
<td>Host data, Network statistics, <code>netstat</code> sourcetype</td>
</tr>
<tr>
<td>Sockets_by_State_by_Host(1)</td>
<td>Returns a time-series chart of the number of open network sockets, by socket state, for a host. Uses the <code>os_index</code> and <code>netstat_sourcetype</code> macros. Requires a host as an argument.</td>
<td>Host data, Network statistics, <code>netstat</code> sourcetype</td>
</tr>
<tr>
<td>Search macro</td>
<td>Intended purpose</td>
<td>Expected data types</td>
</tr>
<tr>
<td>-------------------------------------------------------</td>
<td>-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>-------------------------------------</td>
</tr>
<tr>
<td>Frequently_Open_Ports_by_Host(1)</td>
<td>Returns a list of the most frequently opened TCP ports, by port number, for a host. Uses the <code>os_index</code> and <code>open_ports_sourcetype</code> ports. Requires a host as an argument.</td>
<td>Host data, Network statistics</td>
</tr>
</tbody>
</table>

**Disk macros**

<table>
<thead>
<tr>
<th>Search macro</th>
<th>Intended purpose</th>
<th>Expected data types</th>
</tr>
</thead>
<tbody>
<tr>
<td>Disk_Used_Pct_by_Host(1)</td>
<td>Returns a time-series chart of the percentages of disk used per host. Uses the <code>os_index</code> and <code>df_sourcetype</code> macros. Requires a host as an argument.</td>
<td>Host data, Disk statistics</td>
</tr>
<tr>
<td>Latest_Disk_Used_by_Host(1)</td>
<td>Returns a list of the most up-to-date disk usage per host. Uses the <code>os_index</code> and <code>df_sourcetype</code> macros. Requires a host as an argument.</td>
<td>Host data, Disk statistics</td>
</tr>
<tr>
<td>Max_Disk_Used_by_Host(1)</td>
<td>Returns a list of disk usage percentage, per host, sorted in descending order. Uses the <code>os_index</code> and <code>df_sourcetype</code> macros. Requires a host as an argument.</td>
<td>Host data, Disk statistics</td>
</tr>
<tr>
<td>Open_Files_by_Command_and_Host(1)</td>
<td>Returns a time-series chart of the number of open files, per command, for a host. Uses the <code>os_index</code> and <code>lsof_sourcetype</code> macros. Requires a host as an argument.</td>
<td>Host data, Disk statistics</td>
</tr>
<tr>
<td>Open_Files_by_Type_and_Host(1)</td>
<td>Returns a time-series chart of the number of open files, by file type, for a host. Uses the <code>os_index</code> and <code>lsof_sourcetype</code> macros. Requires a host as an argument.</td>
<td>Host data, Disk statistics</td>
</tr>
<tr>
<td>Open_Files_by_User_and_Host(1)</td>
<td>Returns a time-series chart of the number of open files, by user, for a host. Uses the <code>os_index</code> and <code>lsof_sourcetype</code> macros. Requires a host as an argument.</td>
<td>Host data, Disk statistics</td>
</tr>
</tbody>
</table>

**User macros**

<table>
<thead>
<tr>
<th>Search macro</th>
<th>Intended purpose</th>
<th>Expected data types</th>
</tr>
</thead>
<tbody>
<tr>
<td>User_Sessions_by_Host(1)</td>
<td>Returns a list of active user sessions on a host. Uses the <code>os_index</code> and <code>who_sourcetype</code> macros. Requires a host as an argument.</td>
<td>Host data, Login statistics</td>
</tr>
<tr>
<td>Failed_Logins_by_Host(1)</td>
<td>Returns a list of hosts that have had failed logins. Uses the <code>os_index</code> macro and the “failed_login” event type. Requires a host as an argument.</td>
<td>Host data, Login statistics</td>
</tr>
<tr>
<td>Users_with_Login_Privils_by_Host(1)</td>
<td>Returns a list of hosts where users have login privileges. Uses the <code>os_index</code> and <code>users_with_login_privils_sourcetype</code> macros. requires a host as an argument.</td>
<td>Host data, Login statistics</td>
</tr>
</tbody>
</table>
Release notes

Release notes for the Splunk App for Unix and Linux

This topic contains information on new features, known issues, and updates for the Splunk App for Unix and Linux.

The Splunk Add-on for Unix and Linux was last updated in March 13, 2020.

What's new

Splunk App for Unix and Linux version 6.0.0 now uses Python 3 interpreter by default for its inputs.

See the known issues and fixed issues of these release notes for product updates.

Known issues

This version of the Splunk App for Unix and Linux has the following reported known issues and workarounds. If no issues appear below, no issues have yet been reported.

<table>
<thead>
<tr>
<th>Date filed</th>
<th>Issue number</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>2020-02-20</td>
<td>TAG-12982</td>
<td>Unix app</td>
</tr>
<tr>
<td>2020-01-07</td>
<td>TAG-12828</td>
<td>The Splunk App for Unix and Linux doesn't work in Internet Explorer.</td>
</tr>
</tbody>
</table>

Third-party software attributions/credits

Some of the components included in Splunk App for Unix and Linux are licensed under free or open source licenses.

These attributions are in addition to the attributions we give for third-party vendors whose components the Splunk Enterprise software uses and redistributes. You can find those credits in the Release Notes.

We wish to thank the contributors to these projects:

- **Underscore.js 1.4.4** ([http://underscorejs.org](http://underscorejs.org)) Underscore may be freely distributed under the MIT license © 2009-2013 Jeremy Ashkenas, DocumentCloud Inc.

- **D3 3.1.4** ([http://d3js.org](http://d3js.org))

  Copyright © 2010-2015, Michael Bostock All rights reserved.

axios 0.19.1 [https://www.axios.com/](https://www.axios.com/) Copyright Axios Media, 2020

• **react-treebeard 3.2.4** [https://www.npmjs.com/package/react-treebeard](https://www.npmjs.com/package/react-treebeard)

  Copyright (c) 2015 Alex Curtis

• **react-slider 1.0.2** [https://www.npmjs.com/package/react-slider](https://www.npmjs.com/package/react-slider)

  Copyright (c) 2014 Michal Powaga

• **react-loadable 5.5.0** [https://www.npmjs.com/package/react-loadable](https://www.npmjs.com/package/react-loadable)

  Copyright (c) 2018-present Jamie Kyle

• **jQuery v3.4.1** ([http://jquery.com](http://jquery.com)) Released under the MIT license ([http://jquery.org/license](http://jquery.org/license))

  Copyright 2005,2012 jQuery Foundation, Inc. and other contributors