# Table of Contents

- Workload management overview.................................................................................................................. 1  
  - About workload management.......................................................................................................................... 1  
  - How workload management works.................................................................................................................. 2

- Requirements.................................................................................................................................................... 7  
  - Requirements.................................................................................................................................................. 7

- Set up Linux for workload management........................................................................................................... 9  
  - Set up Linux for workload management......................................................................................................... 9  
  - Configure Linux systemd for workload management....................................................................................... 9  
  - Configure Linux systems not running systemd for workload management......................................................... 14

- Configure workload management.................................................................................................................... 19  
  - Configure workload management.................................................................................................................. 19  
  - Configure workload management on distributed deployments........................................................................ 33  
  - Set access controls for workload management.............................................................................................. 34

- Allocate and monitor resources....................................................................................................................... 36  
  - Assign searches to workload pools................................................................................................................ 36  
  - Monitor workload management..................................................................................................................... 42

- Upgrade workload management....................................................................................................................... 48  
  - Upgrade workload management.................................................................................................................. 48
Workload management overview

About workload management

Workload management is a policy-based system resource manager that lets you allocate compute and memory resources to search, indexing, and other processes in Splunk Enterprise.

With large numbers of searches running concurrently across your deployment, inefficient allocation of system resources can impact search execution, and cause latency, skipped searches, and other performance issues. In some cases, high-priority searches might not have adequate system resources, while trivial searches are allocated too much.

Workload management addresses these issues and helps you optimize resource usage by letting you control the amount of system resources allocated to searches and other processes in Splunk Enterprise.

Workload management lets you:

- Reserve system resources for search, indexing, and other splunkd processes.
- Prioritize critical search workloads.
- Prevent over-usage of system resources.
- Avoid data-ingestion latency due to heavy search load.
- Allocate resources by categories based on process type.
- Create rules to control access to resources based on app, role, index, and user.
- Assign accelerated reports and data models to workload pools.

To learn more about workload management, see How workload management works.

For Linux configuration prerequisites, see Set up Linux for workload management.

For workload management configuration instructions, see Configure workload management.

To learn how to allocate resources to searches, see Assign searches to workload pools.
How workload management works

This page provides an overview of workload management features and concepts, including how workload management allocates system resources to searches in Splunk Enterprise.

The basics

Workload management lets you reserve groups of CPU and memory resources on Linux operating systems, and allocate those resources to searches and other processes in Splunk Enterprise. These resource groups are called workload pools. You can create multiple workload pools of varying sizes to support the search requirements of different users and groups.

Workload management also lets you define policies that control who has access to specific workload pools. These policies are called workload rules. You can use workload rules to ensure that users running high-priority searches have access to sufficient resources, while those running less important searches are appropriately restricted.

After you create workload pools and rules, you can assign individual scheduled or ad-hoc searches to designated workload pools, based on workload rules that you define. For more information, see Assign searches to workload pools.

You can monitor your workload management configuration, and track CPU and memory usage on a per pool basis, using the workload management dashboards in the Monitoring Console. For more information, see Monitor workload management.

Resource allocation in workload management

CPU and memory resources are allocated in workload management in a resource pool hierarchy that includes 3 levels:

<table>
<thead>
<tr>
<th>Level</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Level 1: Top Splunk pool</td>
<td>The total amount of system CPU and memory resources allocated to workload management in Splunk Enterprise based on the cpu and memory resource allocation that you define in Linux cgroups. See cgroups.</td>
</tr>
<tr>
<td>Level 2: Workload</td>
<td>The amount of CPU and memory resources allocated to separate workload categories (search, ingest, and misc). See</td>
</tr>
<tr>
<td>Level</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>categories</td>
<td>workload categories.</td>
</tr>
<tr>
<td>Level 3: Workload pools</td>
<td>The amount of CPU and memory resources allocated to individual workload pools under their respective categories. See workload pools.</td>
</tr>
</tbody>
</table>

The following diagram illustrates the workload management resource pool hierarchy:

![Workload Management Diagram](image)

**How CPU resource allocation works**

Workload management calculates the amount of CPU allocated to a workload category as a ratio of the total CPU weight across all workload categories. If you change the CPU weight allocated to one category, the CPU percentage allocated to all other categories and their workload pools automatically updates to reflect the change.

The CPU weight allocated to a category determines the total amount of CPU resources available for all pools under that category. When you allocate CPU resources to an individual workload pool, workload management calculates the amount of CPU available to that pool as a percentage of the total CPU weight allocated to the parent category.

For example, if the CPU weight allocated to the search category is 70 out of a total weight of 100 across all 3 categories, and the CPU weight allocated to search_pool_1 is 50 out of a total weight of 100 across all pools in the search category, then the allocated CPU% for search_pool_1 is 50% of 70, or 35.5%. For detailed information on configuring workload pools, see Create workload pools.
If an individual workload pool exceeds its allocated CPU %, it is considered a soft limit, and that pool can borrow available CPU resources from other pools in the category.

**Memory resource allocation and sharing**

The amount of memory allocated to a workload category is defined as the maximum percentage of total available memory across all categories.

When you allocate memory to an individual workload pool, workload management calculates the amount of memory available to the pool as an absolute percentage of the total memory allocated to the category.

If an individual workload pool exceeds its memory limit, it can share memory with other pools in the same category, up to the total amount of memory available for the category.

**Search and ingest process isolation**

Workload management provides separate search and ingest categories that ensure resource isolation between search processes and ingest processes. Search and ingest process isolation lets you allocate resources to any number of search pools without affecting the resource allocation for the ingest category.

See [workload categories](#).

**Workload management features and concepts**

The following features and concepts are useful to understand before you configure and use workload management.

**Workload categories**

Workload categories determine the total amount of cpu and memory resources available to workload pools running specific process types in Splunk Enterprise. When you create a workload pool, you must assign it to a category. There are three workload categories:

- **search**: Scheduled searches, ad hoc searches, accelerated reports, and accelerated data models.
- **ingest**: Indexing and other **splunkd** processes, including process runner, KV store, app server, and introspection.
- **misc**: Scripted inputs and modular inputs only.
For more information, see *Configure workload categories*.

**Workload pools**

A workload pool is a specified amount of CPU and memory resources that you can define and allocate to processes in Splunk Enterprise. Each workload pool reserves a subset of the total amount of CPU and memory available in the workload category to which it belongs. You can assign individual scheduled or ad-hoc searches to designated workload pools, based on policies that you define in workload rules.

For more information, see *Create workload pools*.

**Workload rules**

A workload rule is a policy that you define to control access to workload pools. Each workload rule has a predicate condition that determines which apps and roles can assign searches to a designated pool. You can specify a priority order for workload rules that determines which apps and roles, and therefore which searches, have priority access to workload pools.

For more information, see *Create workload rules*.

**cgroups**

cgroups (control groups) are a Linux kernel feature that lets you prioritize a specified amount of system resources for a group of processes. cgroups also include a rules engine that lets you control user access to resources. Workload management in Splunk Enterprise is an abstraction of the underlying functionality of Linux cgroups.

Before you can configure workload management in Splunk Enterprise, you must set up cgroups on your Linux operating system. For more information, see *Set up Linux for workload management*.

**systemd**

systemd is a system startup and service manager for Linux operating systems that organizes processes under cgroups. *systemd* uses instructions for a daemon specified in a unit configuration file. You can configure this unit file to run *splunkd* as a *systemd* service.
For information on how to configure `systemd` for workload management, see Configure Linux systemd for workload management.
Requirements

Requirements

Workload management has the following requirements and limitations.

Splunk Enterprise version requirements

Workload management requires Splunk Enterprise version 7.2.0 or later.

The following workload management features require the specified Splunk Enterprise version:

- Automated Linux preflight checks for workload management require version 7.2.2. or later.
- Workload pool categories (search, ingest, and misc) for workload management requires version 7.3.0 or later.
- Workload rules for workload pool monitoring and actions requires version 8.0.0 or later.

Operating system requirements

Workload management is currently supported in Splunk Enterprise on Linux operating systems only.

For more information, see Supported operating systems.

Linux operating system requirements

Linux kernel

Workload management requires Linux kernel version 2.6.25 or later.

Cgroups version

Workload management requires cgroups version 1.0.
Systemd version

Workload management supports systemd version 219 or later.

Systemd is not a mandatory requirement, but if it is running on your Linux instance, it must be version 219 or later.

Supported Linux distributions

Splunk Enterprise supports workload management on these Linux distributions:

- RHEL 6 and 7
- CentOS 6 and 7
- Ubuntu 10.04 LTS and later
- SUSE 11 and 12

Configure cgroups for splunkd

Before you can configure and enable workload management in Splunk Enterprise, you must set up the underlying Linux operating system to allow splunkd to manage cgroups. See Set up Linux for workload management.
Set up Linux for workload management

Before you can configure workload management in Splunk Enterprise, you must set up cgroups on your underlying Linux operating system. How you set up cgroups for workload management depends on whether your Linux system is running systemd. To determine if your system is running systemd, see Is Linux running systemd?

Is Linux running systemd?

Use one of the following options to determine if your Linux distribution is running systemd.

- Run the `systemctl` command to check for a systemd version number.
  ```bash
  $ systemctl --version
  systemd 219
  +PAM +AUDIT +SELINUX +IMA +APPARMOR +SMACK +SYSVINIT +UTMP
  +LIBCRYPTSETUP +GCrypt +GNUTLS +ACL +XZ -LZ4 +SECCOMP +BLKID
  +ELFUTILS +KMOD -IDN
  Workload management supports systemd version 219 or later. See Requirements.
  ```

- Check for a systemd process ID. If the output shows PID=1, then you are running systemd. For example:
  ```bash
  $ pidof systemd
  1
  ```

To configure cgroups on Linux systems running systemd, see Configure systemd distributions for workload management.

To configure cgroups on Linux systems not running systemd, see Configure non-systemd distributions for workload management.

Configure Linux systemd for workload management

Before you can configure workload management on Linux distributions running systemd, you must configure systemd to manage splunkd as a service by creating a unit file that defines a cgroup hierarchy.
The following diagram illustrates the cgroup hierarchy under `systemd`:

![Systemd Cgroup Hierarchy Diagram](image)

For more information, see `cgroups`.

You must configure cpu and memory cgroups for workload management on all search heads and indexers.

**Configure systemd to manage splunkd as a service**

There are two ways to configure `systemd` to manage `splunkd` as a service:

- Configure `systemd` manually.
- Configure `systemd` using the `splunk enable boot-start` command.

Configuring `systemd` using `splunk enable boot-start` requires Splunk Enterprise version 7.2.2 or later.

**System requirements**

- To run `splunkd` as a `systemd` service requires one of the following supported Linux distributions:
  - RHEL 6 and 7
  - CentOS 6 and 7
  - Ubuntu 16.04 LTS and later
To enable workload management in Splunk Enterprise under systemd requires systemd version 219 or higher. For more information, see Linux operating system requirements in the Workload Management manual.

**Permissions requirements for systemd**

systemd has the following permissions requirements:

- Non-root users must have super user permissions to manually configure systemd on Linux.
- Non-root users must have super user permissions to run start, stop, and restart commands under systemd.

For instructions on how to create a new user with super user permissions, see your Linux documentation. The specific steps can vary depending on the specific Linux distribution.

You must use sudo to run splunk start\|stop\|restart. If you do not use sudo, you must authenticate. For example:

```bash
==== AUTHENTICATING FOR org.freedesktop.systemd1.manage-units ====
Authentication is required to manage system services or units.
Multiple identities can be used for authentication:
  1.  <username_1>
  2.  <username_2>
Choose identity to authenticate as (1-2): 2
Password:
==== AUTHENTICATION COMPLETE ====
```

**Configure systemd manually**

Follow these steps to configure systemd to manage splunkd as a service:

1. Confirm that your Linux machine is running systemd. See Is Linux running systemd?.
2. Before you create, delete, or modify the systemd unit file, you must stop splunkd.

   ```bash
   $SPLUNK_HOME/bin/splunk stop
   ```
3. If you enabled Splunk software to start at boot using enable boot-start, run disable boot-start to remove the splunk init script from /etc/init.d and its symbolic links.

   ```bash
   sudo $SPLUNK_HOME/bin/splunk disable boot-start
   ```
4. Open the $SPLUNK_HOME/etc/splunk-launch.conf file and note the value of SPLUNK_SERVER_NAME. The default value is Splunkd.

5. In the /etc/systemd/system directory, create a unit file named <SPLUNK_SERVER_NAME>.service, such as Splunkd.service.

You can change the SPLUNK_SERVER_NAME to any name you choose by directly editing the splunk-launch.conf file.

6. Add the following content to the <SPLUNK_SERVER_NAME>.service unit file:

   [Unit]
   After=network.target

   [Service]
   Type=simple
   Restart=always
   ExecStart=/home/<username>/splunk/bin/splunk
   _internal_launch_under_systemd
   LimitNOFILE=65536
   SuccessExitStatus=51 52
   RestartPreventExitStatus=51
   RestartForceExitStatus=52
   KillMode=mixed
   KillSignal=SIGINT
   TimeoutStopSec=10min
   User=<username>
   Delegate=true
   MemoryLimit=100G
   CPUShares=1024
   PermissionsStartOnly=true
   ExecStartPost=/bin/bash -c "chown -R <userid>:<groupid> /sys/fs/cgroup/cpu/system.slice/%n"
   ExecStartPost=/bin/bash -c "chown -R <userid>:<groupid> /sys/fs/cgroup/memory/system.slice/%n"

   [Install]
   WantedBy=multi-user.target

The following unit file properties are set specifically for Splunk workload management:
   Type=simple
   Restart=always
   Delegate=true

Do not change these values unless you are familiar with systemd or receive guidance from Splunk support.

Do not use the following unit file properties. These properties can cause splunkd to fail on restart.
   RemainAfterExit=yes
   ExecStop
For more information on unit file configuration settings, see Systemd unit file properties.

7. Reload the unit file.

    sudo systemctl daemon-reload

8. Start splunkd.

    sudo $SPLUNK_HOME/bin/splunk start

   This starts splunkd as a systemd service.

Under systemd, splunk start|stop|restart commands are mapped to systemctl start|stop|restart commands.

9. Verify that splunkd is running as a systemd service:

    sudo systemctl status <SPLUNK_SERVER_NAME>.service

When you create the splunkd service, systemd creates corresponding CPU and Memory cgroups in these locations:

    CPU: /sys/fs/cgroup/cpu/system.slice/<SPLUNK_SERVER_NAME>.service
    Memory: /sys/fs/cgroup/memory/system.slice/<SPLUNK_SERVER_NAME>.service

10. For distributed deployments, repeat steps 1-9 on all search heads and indexers.

### systemd unit file properties

The following table lists the unit file properties you must specify to run splunkd as a service under systemd:

<table>
<thead>
<tr>
<th>Property</th>
<th>Expected Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Restart</td>
<td>always</td>
</tr>
<tr>
<td>Type</td>
<td>simple</td>
</tr>
<tr>
<td>ExecStart</td>
<td>$SPLUNK_HOME/bin/splunk _internal_launch_under_systemd</td>
</tr>
<tr>
<td>ExecStartPost</td>
<td>chown -R USER:&lt;GROUP of User&gt;/sys/fs/cgroup/&lt;cpu or memory&gt;/system.slice/%n&quot;</td>
</tr>
<tr>
<td>Delegate</td>
<td>True</td>
</tr>
<tr>
<td>SuccessExitStatus</td>
<td>51 52</td>
</tr>
<tr>
<td>RestartPreventExitStatus</td>
<td>51</td>
</tr>
<tr>
<td>RestartForceExitStatus</td>
<td>52</td>
</tr>
<tr>
<td>Property</td>
<td>Expected Value</td>
</tr>
<tr>
<td>-----------------------</td>
<td>-----------------------------------------------------</td>
</tr>
<tr>
<td>RemainAfterExit</td>
<td>no (default)</td>
</tr>
<tr>
<td>MemoryLimit</td>
<td>Example: 12G</td>
</tr>
<tr>
<td>CPUShares</td>
<td>Example: 8192 (Allowed range is 2 to 262144. Default is 1024.)</td>
</tr>
<tr>
<td>User, Group</td>
<td>&lt;Splunk Owner&gt; &lt;Splunk Group&gt;</td>
</tr>
</tbody>
</table>

For more information on systemd unit file properties, see Service unit configuration.

Manage clusters under systemd

When managing an indexer cluster under systemd:

You must use the `sudo` command to start, stop, and restart the cluster master or individual peer nodes using `splunk start|stop|restart` commands. You do not need `sudo` to perform a rolling restart using the `splunk rolling-restart cluster-peers` command, or to take a peer offline using the `splunk offline` command.

When managing a search head cluster under systemd:

You must use the `sudo` command to start, stop, and restart cluster members using `splunk start|stop|restart` commands. You do not need `sudo` to perform a rolling restart using the `splunk rolling-restart shcluster-members` command, or to remove a cluster member using the `splunk remove shcluster-members` command.

Next step

After you set up cgroups on your Linux operating system, you can configure workload management in Splunk Enterprise. See Configure workload management.

Configure Linux systems not running systemd for workload management

Before you can configure workload management on Linux systems not running systemd, you must create a cgroup hierarchy in which `splunkd` and other system
processes run in their own cgroups.

The following diagram illustrates the cgroup hierarchy on Linux systems not running systemd:

For more information, see cgroups.

**Configure cgroups on non-systemd distributions**

There are two ways to configure cgroups for workload management on Linux systems not running systemd:

- Configure cgroups using the `cgconfig` service.
- Configure cgroups using filesystem operations.

You must configure cpu and memory cgroups for workload management on all search heads and indexers.

**Configure cgroups using cgconfig**

cgconfig is supported on RHEL 7.0 and earlier.

To configure cgroups for workload management using cgconfig:

1. Check that `/sys/fs` is mounted. If it is not mounted, mount the tmpfs in-memory filesystem under `/sys/fs/cgroup` as follows:

   ```bash
   sudo mount -t tmpfs -o size=10M tmpfs /sys/fs
   ```
Splunk Enterprise examines the `/proc/mounts` file to determine whether your Linux machine can support workload management.

2. Create the `cgconfig.conf` file under `/etc` and add the following contents:

```plaintext
# Copyright IBM Corporation. 2007
#
# Authors: Balbir Singh <balbir@linux.vnet.ibm.com>
# This program is free software; you can redistribute it and/or
# modify it
# under the terms of version 2.1 of the GNU Lesser General Public License
# as published by the Free Software Foundation.
#
# This program is distributed in the hope that it would be
# useful, but
# WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
#
# See man cgconfig.conf for further details.
#
# By default, mount all controllers to /cgroup/<controller>

mount {
    cpuset = /sys/fs/cgroup/cpuset;
    cpu = /sys/fs/cgroup/cpu;
    cpuacct = /sys/fs/cgroup/cpuacct;
    memory = /sys/fs/cgroup/memory;
    devices = /sys/fs/cgroup/devices;
    freezer = /sys/fs/cgroup/freezer;
    net_cls = /sys/fs/cgroup/net_cls;
    blkio = /sys/fs/cgroup/blkio;
}

group splunk {
    perm {
        admin {
            uid = "splunk";
            gid = "splunk";
        }
        task {
            uid = "splunk";
            gid = "splunk";
        }
    }
    cpu {
        cpu.shares = "2048";
    }
    memory {
        memory.limit_in_bytes = "3G";
    }
}
```
3. Restart the cgconfig service.

```
# service cgconfig restart
Stopping cgconfig service:     [  OK  ]
Starting cgconfig service:     [  OK  ]
```

This creates the splunk cgroup.

The cgroup name must match the value of workload_pool_base_dir_name defined in workload_pools.conf. The default value is splunk.

**Configure cgroups using filesystem operations**

To configure cgroups, the Linux admin, logged in as root user, must create the cgroups and assign the splunk user permissions to manage the cgroups, as follows:

1. In the workload_pools.conf. file, set workload_pool_base_dir_name to the root cgroup to be used by splunk. For example:

   ```
   [general]
   workload_pool_base_dir_name = splunk
   ```

   Or, send a POST request:

   ```
   workloads/config/set-base-dirname -workload_pool_base_dir_name <base_dir_name>
   ```

   For endpoint details, see workloads/config/set-base-dirname in the Splunk Enterprise REST API Reference Manual.

2. Create cpu and memory cgroups:

   ```
   sudo mkdir /sys/fs/cgroup/cpu/<workload_pool_base_dir_name>
   sudo mkdir /sys/fs/cgroup/memory/<workload_pool_base_dir_name>
   ```

3. Assign the splunk user permissions to manage the respective cgroups:

   ```
   sudo chown -R ${USER} /sys/fs/cgroup/cpu/<workload_pool_base_dir_name>
   sudo chown -R ${USER} /sys/fs/cgroup/memory/<workload_pool_base_dir_name>
   ```

   If you have specified a user as SPLUNK_OS_USER in splunk-launch.conf, you must specify the same user as {USER} in the command. For more information see splunk-launch.conf in the Splunk Enterprise "Admin Manual".

4. Assign CPU shares for splunk cgroup:

   ```
   cd /sys/fs/cgroup/cpu/splunk
   ```
echo 2048 > cpu.shares

5. Assign physical memory for the splunk cgroup:

cd /sys/fs/cgroup/memory/splunk/
echo 14G > memory.limit_in_bytes

Next step

After you set up cgroups on your Linux operating system, you can configure workload management in Splunk Enterprise. See Configure workload management.
Configure workload management

Before you can configure workload management in Splunk Enterprise, you must set up Linux cgroups on your underlying Linux operating system. For instructions, see Set up Linux for workload management.

This topic discusses how to configure workload management on a single instance. For information on how to configure workload management on distributed deployments, see Configure workload management on distributed deployments.

Before you can enable workload management, you must create a default pool in the search category and a default pool in the ingest category. You can optionally create workload rules to control access to workload pools at any time. You can configure workload management using Splunk Web, CLI, or REST.

Follow these steps to configure workload management on a single instance:

1. Run preflight checks.
2. Configure workload categories.
3. Create workload pools.
4. Create workload rules.
5. Enable workload management.
6. Check workload management status.

Run preflight checks

When you open workload management in Splunk Web, a set of system checks runs automatically to determine if your underlying Linux operating system is set up properly for workload management.

If all preflight checks pass, this means your system is set up correctly and you can proceed to configure workload management. If any preflight checks fail, review the error messages to identify the Linux configuration issues you must fix before you can configure workload management.

You can optionally run preflight checks manually using the CLI or REST API.
Workload management preflight checks reflect the status of the local instance only.

Workload management runs the following preflight checks:

<table>
<thead>
<tr>
<th>Name</th>
<th>Mitigation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operating system</td>
<td>Operating system must be Linux. Workload management is not currently supported on Windows OS.</td>
</tr>
<tr>
<td>Cgroup Version</td>
<td>Cgroup must be version 1. Workload management does not support pre-cgroup or cgroup version 2 Linux kernels.</td>
</tr>
<tr>
<td>CPU Splunk base directory present</td>
<td>CPU Splunk base directory Splunkd.service is missing.</td>
</tr>
<tr>
<td></td>
<td>For systemd, the base directory is /sys/fs/cgroup/cpu/system.slice/&lt;unit_file_name&gt;. The unit_file_name is &lt;SPLUNK_SERVER_NAME&gt;.service.</td>
</tr>
<tr>
<td></td>
<td>The SPLUNK_SERVER_NAME must match the Splunk server name in splunk-launch.conf. The default value is Splunkd.</td>
</tr>
<tr>
<td></td>
<td>See Configure systemd distributions for workload management.</td>
</tr>
<tr>
<td></td>
<td>For non-systemd, the base directory is /sys/fs/cgroup/cpu/splunk.</td>
</tr>
<tr>
<td></td>
<td>The base directory name must match the workload_pool_base_dir_name defined in workload_pools.conf. The default value is splunk.</td>
</tr>
<tr>
<td></td>
<td>See Configure non-systemd distributions.</td>
</tr>
<tr>
<td>CPU Splunk base directory permissions</td>
<td>CPU Splunk base directory Splunkd.service requires read and write permissions.</td>
</tr>
<tr>
<td></td>
<td>For systemd, permissions must be set for non-root user in the Splunkd.service unit file.</td>
</tr>
<tr>
<td></td>
<td>See Configure systemd distributions for workload management.</td>
</tr>
<tr>
<td></td>
<td>For non-systemd, use chown to grant permissions to the splunk base directory.</td>
</tr>
<tr>
<td></td>
<td>See Configure non-systemd distributions.</td>
</tr>
<tr>
<td>Name</td>
<td>Mitigation</td>
</tr>
<tr>
<td>-------------------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Memory Splunk base directory present</td>
<td>Memory Splunk base directory <code>Splunkd.service</code> is missing. For systemd, the base directory is <code>/sys/fs/cgroup/memory/system.slice/&lt;unit_file_name&gt;</code>.</td>
</tr>
<tr>
<td></td>
<td>See Configure systemd distributions for workload management. For non-systemd, the base directory is <code>/sys/fs/cgroup/memory/splunk</code>. See Configure non-systemd distributions.</td>
</tr>
<tr>
<td>Memory Splunk base directory permissions</td>
<td>Memory Splunk base directory <code>Splunkd.service</code> requires read and write permissions. For systemd, permissions must be set for non-root user in the <code>Splunkd.service</code> unit file.</td>
</tr>
<tr>
<td></td>
<td>See Configure systemd distributions for workload management. For non-systemd, use <code>chown</code> to grant permissions to the splunk base directory. See Configure non-systemd distributions.</td>
</tr>
<tr>
<td>Unit file present</td>
<td>The unit file <code>Splunkd.service</code> is missing. The unit file is located under <code>/etc/systemd/system</code> with the name <code>&lt;SPLUNK_SERVER_NAME&gt;.service</code>. SPLUNK_SERVER_NAME is set in <code>splunk-launch.conf</code>. See Configure systemd distributions for workload management.</td>
</tr>
<tr>
<td>Delegate property set to true</td>
<td>The Delegate property in the unit file must be set to true.</td>
</tr>
<tr>
<td>Splunk launched under systemd</td>
<td><code>splunkd</code> is running as a systemd service. In the unit file, the Restart property must be set to always. The ExecStart property must include <code>_internal_launch_under_systemd</code>.</td>
</tr>
</tbody>
</table>

For more information on unit file properties, see systemd unit file properties.

For more information on how to set up Linux for workload management, see Set up Linux for workload management.
Run preflight checks in Splunk Web

1. Click Settings > Workload Management. The Linux preflight checks run automatically. If all preflight checks pass, the workload management UI opens, and you can proceed to configure workload management.
2. If one or more preflight checks fail, a page appears showing the overall results of the preflight checks. Review the error messages and fix the specified Linux configuration issues.

3. Click Rerun preflight checks.

Run preflight checks using the CLI

To run preflight checks for workload management using the CLI:

1. Log into your Linux machine.
2. Run the following CLI command.

```
./splunk check workload-config
```

Here is an example of the output from this command:

Workload Management Preflight Checks failed. Fix the following issues:
- CPU Splunk base directory Splunkd.service requires read and write permissions.
- CPU Splunk base directory Splunkd.service is missing.
- The 'Delegate' property in the unit file must be set to 'true'. Restart Splunk then rerun preflight checks.
- In the unit file, the 'Restart' property must be set to 'always'. The 'ExecStart' property must include '_internal_launch_under_systemd'. Make sure the up-to-date unit
file is loaded.
Memory Splunk base directory Splunkd.service requires read
and write permissions.
Memory Splunk base directory Splunkd.service is missing.
Unit file Splunkd.service is missing. Restart Splunk then
rerun preflight checks.

**Run preflight checks using REST**

Send a GET request to:

```
workloads/config/preflight-checks
```

For endpoint details, see workloads/config/preflight-checks in the REST API Reference Manual.

**Configure workload categories**

Workload categories determine the total amount of system cpu and memory resources available for workload pools running specific process types in Splunk Enterprise. For example, the search category determines the total amount of resources available to all workload pools running search processes. When you create a workload pool you must assign it to a workload category.

Workload management provides the following three workload categories:

- **search**: Scheduled searches and ad hoc searches, accelerated reports and data models.
- **ingest**: Indexing and other splunkd processes, including process runner, KV store, app server, and introspection.
- **misc**: Scripted inputs and modular inputs only.

Each workload category has its own cpu and memory resource allocation. And each workload pool within a category is assigned a fraction of the total cpu and a percentage of the total memory allocated to that category. You can modify the resource allocation for a category to ensure that sufficient resources are available to workload pools running high-priority processes.

You can edit workload categories using Splunk Web, CLI, or REST.

**Edit workload categories using Splunk Web**

To edit the resource allocation for a workload category in Splunk Web:
1. In Splunk Web, click **Settings > Workload Management**.
The workload management UI opens.
2. Click the **All Categories** tile.
3. Click **Edit** under the specific category.
4. Specify the resource allocation:

<table>
<thead>
<tr>
<th>Field</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU Weight</td>
<td>Specify the total CPU weight available for pools in this category.</td>
</tr>
<tr>
<td>Memory Limit %</td>
<td>Specify the maximum percentage of Memory available for pools in this category.</td>
</tr>
</tbody>
</table>

5. Click **Submit**.

The percentage of CPU allocated to a category is a ratio of the total CPU weight across all categories. When you change the CPU weight for one category the CPU allocated to all other categories and all workload pools updates to reflect the change.

For more information, see [Resource allocation in workload management](#).

**Edit workload categories using the CLI**

To edit a workload category, run the following CLI command:

```bash
./splunk edit workload-category <category> [-cpu_weight <number> -mem_weight <number>]
```

where `<category>` is search, ingest, or misc.

To list workload categories:

```bash
./splunk list workload-category
```

**Edit workload categories using REST**

Send a POST request to:

```
workloads/categories
```

For endpoint details, see `workloads/categories` in the **REST API Reference Manual**.
Create workload pools

A workload pool is a specified amount of CPU and memory resources that you can define and allocate to processes in Splunk Enterprise, including search, indexing, and other splunkd processes.

To configure workload management, you must create, at a minimum, these two workload pools:

- **Default pool in the search category**: Searches that are not explicitly mapped to a workload rule are assigned to this pool by default.
- **Default pool in the ingest category**: Indexing and other non-search processes are assigned to this pool by default.

You can optionally create a default pool in the misc category. Scripted and modular inputs run in this pool by default. If you do not create a default pool in the misc category, scripted and modular inputs run in the default pool in the ingest category.

You can only create one workload pool in the ingest and misc categories.

**Create a workload pool in Splunk Web**

1. In Splunk Web, click **Settings > Workload Management**.
2. Click **Add Workload Pool**.
3. Specify the following fields:

<table>
<thead>
<tr>
<th>Field</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pool Category</td>
<td>Select a workload category based on the type of process the pool will run (search, ingest, or misc). See Configure workload categories.</td>
</tr>
<tr>
<td>Name</td>
<td>Specify the name of the workload pool. Valid characters are alphanumeric and underscore only.</td>
</tr>
<tr>
<td>CPU Weight</td>
<td>The fraction of total available CPU for this pool.</td>
</tr>
<tr>
<td>Memory Limit %</td>
<td>The maximum percentage of total available memory for this pool.</td>
</tr>
<tr>
<td>Default Pool</td>
<td>Toggle the switch to make this pool the default pool for the selected category.</td>
</tr>
</tbody>
</table>

4. Click **Submit**.
The workload pool appears in the Workload Management UI.
For more information, see Resource allocation in workload management.

**Create a workload pool using the CLI**

Run the following CLI command:

```
./splunk add workload-pool <pool_name> [-category <search/ingest/misc> -cpu_weight <number> -mem_weight <number> -default_category_pool <true|false>]
```

**Create a workload pool using REST**

Send a POST request to:

```
workloads/pools
```

For endpoint details, see workloads/pools in the *REST API Reference Manual*.

**View workload_pools.conf**

Do not place `workload_pools.conf` files in more than a single app context. Having identical `workload_pools.conf` stanzas in multiple app contexts can cause workload management enable/disable functions to fail and cause other issues.

When you create a workload pool, the configuration is stored in

```
$SPLUNK_HOME/etc/apps/<app_name>/local/workload_pools.conf
```
workload_pools.conf defines the cpu and memory resource allocation for workload categories (search, ingest, and misc) and the individual workload pools created under those categories. For example:

```
[general]
default_pool = pool_1
ingest_pool = pool_3
enabled = 0

[workload_category:search]
cpu_weight = 70
mem_weight = 70

[workload_category:ingest]
cpu_weight = 20
mem_weight = 20

[workload_category:misc]
cpu_weight = 10
mem_weight = 10

[workload_pool:pool_1]
cpu_weight = 70
mem_weight = 70
category = search
default_category_pool = 1

[workload_pool:pool_2]
cpu_weight = 30
mem_weight = 30
category = search
default_category_pool = 0

[workload_pool:pool_3]
cpu_weight = 100
mem_weight = 100
category = ingest
default_category_pool = 1

[workload_pool:pool_4]
cpu_weight = 100
mem_weight = 100
category = misc
default_category_pool = 1
```

For more information workload pool settings, see workload_pools.conf.
Delete workload pools

You can delete any workload pool under a category, except for the default category pool. If you try to delete the default category pool an error message appears. You can delete workload pools using Splunk Web, CLI, REST, or editing workload_pools.conf.

To delete a workload pool using the CLI:

```
./splunk remove workload-pool <pool_name>
```

You cannot delete a workload pool while a process is running in that pool. Any pool you delete that has an active process running in it will not be deleted until after workload_pools.conf reloads or splunkd restarts.

Create workload rules

Workload rules provide a policy-based method for assigning searches to workload pools. Each rule specifies a predicate condition that must match before you can assign searches to the designated pool. You can use workload rules to ensure that high-priority searches have access to adequate resources while low-priority searches are restricted.

Workload rules are evaluated in the order that you create them. If the predicate condition defined in a rule does not match, the next rule in order is evaluated. If there is no match with any rule, the search is assigned to the default search pool. In this way, workload rules let you prioritize the assignment of system resources based on conditions that you define.

Create a workload rule in Splunk Web

1. In Splunk Web, click Settings > Workload Management.
2. Click Add Workload Rule.
3. Configure your new workload rule by defining the following fields:

<table>
<thead>
<tr>
<th>Field</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Specify the name of the workload rule.</td>
</tr>
<tr>
<td>Predicate</td>
<td>Specify a predicate condition to access the workload pool. The format is a logical expression where &lt;type&gt;=&lt;value&gt; with optional AND, OR, NOT, (). The valid &lt;type&gt; are &quot;app&quot;, &quot;role&quot;, &quot;index&quot;, or &quot;user&quot;. For example, &quot;app=search AND role=power&quot; maps all searches belonging to both the search app and the power role to the corresponding workload pool.</td>
</tr>
<tr>
<td>Field</td>
<td>Action</td>
</tr>
<tr>
<td>---------------------</td>
<td>------------------------------------------------------------------------</td>
</tr>
<tr>
<td>For more information on predicates, see workload_rules.conf.</td>
<td></td>
</tr>
<tr>
<td>Workload Pool</td>
<td>Select the workload pool to which this rule applies.</td>
</tr>
</tbody>
</table>

4. Click **Submit**.

**Create a workload rule using the CLI**

Run the following CLI command:

```bash
./splunk add workload-rule <rule_name> -predicate <predicate> -workload_pool <pool_name>
```

where <predicate> is a logical expression in the format `<type>=<value>` with optional AND, NOT, OR, (). For example:

```bash
./splunk add workload-rule "my_role_rule" -predicate "app=search AND (NOT index=_internal)" -workload_pool "pool_a"
```

**Create a workload rule using REST**

Send a POST request to:

```
workloads/rules
```

For endpoint details, see workloads/rules in the **REST API Reference Manual**.

**View workload_rules.conf**

When you create a workload rule, the configuration is stored in
$SPLUNK_HOME/etc/apps/<app_name>/local/workload_rules.conf.

workload_rules.conf defines both the mapping of rules to workload pools and the order in which rules are evaluated. For example:

```
[workload_rules_order]
rules = new_rule_2,new_rule,SearchAppRule,DMC_rule

[workload_rule:SearchAppRule]
predicate = app=search
workload_pool = app_search

[workload_rule:DMC_rule]
predicate = app=splunk_monitoring_console
```
workload_pool = app_dmc

[workload_rule:new_rule]
predicate = app=search AND (NOT index=_internal)
workload_pool = pool_1

[workload_rule:new_rule_2]
predicate = NOT role=power OR user=admin
workload_pool = pool_2

For more information on workload rules settings, see workload_rules.conf.

Enable or disable workload management

After you create your workload pools and rules, you must enable workload management. When you initiate a request to enable workload management, a series of health checks run in the background to validate both the workload management configuration and the underlying Linux system configuration. If these health checks fail, you cannot enable workload management and a failure message appears.

For more information on Linux configuration requirements, see Set up Linux for workload management.

Enable or disable workload management in Splunk Web

1. In Splunk Web, click Settings > Workload Management.
2. Toggle the switch to Enabled.

   This applies any pending configuration changes and enables workload management.

   To disable workload management, toggle the switch to "Disabled".

Enable or disable workload management using the CLI

To enable or disable workload management, run the following CLI command:

```
./splunk <enable|disable> workload-management
```

Enable or disable workload management using REST

You can enable or disable workload management using REST. For endpoint details, see workloads/config/enable or workloads/config/disable in the REST API Reference Manual
Check workload management status

You can view the current active configuration of workload management using the CLI or REST. Output shows configuration details of all workload pools and rules, and whether workload management is supported and enabled on the instance.

**Check workload management status using the CLI**

Run the following CLI command:

```bash
./splunk show workload-management-status
```

Here is an example of the output from the command:

```
Workload Management Status:
   Enabled: 1
   Supported: 1
   Error:

Workload Category: ingest

    CPU Group:
    /sys/fs/cgroup/cpu/system.slice/Splunkd.service/ingest
    Memory Group:
    /sys/fs/cgroup/memory/system.slice/Splunkd.service/ingest
    CPU Weight: 25
    Memory Weight: 25
    Default Category Pool: pool_2

    pool_2:
    CPU Group:
    /sys/fs/cgroup/cpu/system.slice/Splunkd.service/ingest/pool_2
    Memory Group:
    /sys/fs/cgroup/memory/system.slice/Splunkd.service/ingest/pool_2
    CPU Weight: 100
    Memory Weight: 100

Workload Category: search

    CPU Group:
    /sys/fs/cgroup/cpu/system.slice/Splunkd.service/search
    Memory Group:
    /sys/fs/cgroup/memory/system.slice/Splunkd.service/search
    CPU Weight: 75
    Memory Weight: 75
    Default Category Pool: pool_1

    pool_1:
    CPU Group:
    /sys/fs/cgroup/cpu/system.slice/Splunkd.service/search/pool_1

31
Memory Group:
/sys/fs/cgroup/memory/system.slice/Splunkd.service/search/pool_1
  CPU Weight: 20
  Memory Weight: 20

pool_3:
  CPU Group:
/sys/fs/cgroup/cpu/system.slice/Splunkd.service/search/pool_3
  Memory Group:
/sys/fs/cgroup/memory/system.slice/Splunkd.service/search/pool_3
  CPU Weight: 20
  Memory Weight: 20

Workload Category: misc

  CPU Group:
/sys/fs/cgroup/cpu/system.slice/Splunkd.service/misc
  Memory Group:
/sys/fs/cgroup/memory/system.slice/Splunkd.service/misc
  CPU Weight: 12
  Memory Weight: 12
  Default Category Pool: misc_pool

misc_pool:
  CPU Group:
/sys/fs/cgroup/cpu/system.slice/Splunkd.service/misc/misc_pool
  Memory Group:
/sys/fs/cgroup/memory/system.slice/Splunkd.service/misc/misc_pool
  CPU Weight: 100
  Memory Weight: 100

Workload Rules:

  rule_1:
    Order: 1
    Predicate: app="search"
    Workload Pool: pool_1

  rule_2:
    Order: 2
    Predicate: app="search" AND (NOT index="_internal")
    Workload Pool: pool_3

Check workload management status using REST

To view workload management status information, send a GET request to:

workloads/status

For endpoint details, see workloads/status in the REST API Reference Manual.
Next Step

After you configure workload management, you can allocate resources to individual scheduled and ad-hoc search processes in Splunk Enterprise. For more information, see Assign searches to workload pools.

Configure workload management on distributed deployments

You can use workload management to allocate resources in both non-clustered and clustered distributed search environments.

Configure workload management on non-clustered indexers

To configure workload management on non-clustered indexers, you must first configure and enable workload management on the search head, then copy the enabled `workload_pools.conf` file to all indexers.

You do not need to copy `workload_rules.conf` to indexers. Its functionality applies to search heads only.

To configure and enable workload management on non-clustered indexers:

1. Configure and enable workload management on the search head. See Configure workload management.
2. Copy the enabled `workload_pools.conf` file to all indexers.
3. Reload `workload_pools.conf` on each indexer. For example:

   ```
curl -k -u admin:pass https://<host>:<mPort>/services/configs/services/workloads/pools/_reload
   ```

   Reloading `workload_pools.conf` enables workload management.

Configure workload management on an indexer cluster

To configure workload management on an indexer cluster, you must first configure and enable workload management on the search head, then use the configuration bundle method to push `workload_pools.conf` from the cluster master to peer nodes.

You do not need to push `workload_rules.conf` to the indexer cluster. Its functionality applies to search heads only.
To configure and enable workload management on an indexer cluster:

1. Configure and enable workload management on the search head. See Configure workload management.
2. Copy the enabled `workload_pools.conf` file from the search head to the configuration bundle on the cluster master.
3. Distribute the configuration bundle to all peer nodes. For detailed instructions, see Distribute the configuration bundle. After the bundle push, peer nodes automatically reload the enabled configuration file, which enables workload management.

**Configure workload management on a search head cluster**

To configure workload management on a search head cluster, use configuration replication to replicate workload management configuration files to all search head cluster members.

Both `workload_pools.conf` and `workload_rules.conf` are required on all search heads.

1. On any cluster member, in Splunk Web, click **Settings > Workload Management**.
2. Configure and enable workload management. See Configure workload management.

   The cluster automatically replicates the configuration to all cluster members. This triggers a reload of the enabled configuration files on each cluster member, which enables workload management.

For more information on configuration replication, see Configuration updates the cluster replicates.

You can use the deployer to push workload management configuration files to search head cluster members. However, you cannot use both the deployer bundle push method and the configuration replication method to update cluster configurations. You must choose one method and use only that method. Using both methods can cause workload management to fail. For more information, see Use the deployer to distribute apps and configuration updates.

**Set access controls for workload management**
To view, create, and use workload pools and workload rules, a user’s role must have the appropriate capabilities.

The following capabilities are enabled for role_admin by default:

<table>
<thead>
<tr>
<th>Capability</th>
<th>Permissions granted to role</th>
</tr>
</thead>
<tbody>
<tr>
<td>list_workload_pools</td>
<td>List and view workload pools and workload categories.</td>
</tr>
<tr>
<td>list_workload_rules</td>
<td>List and view workload rules.</td>
</tr>
<tr>
<td>edit_workload_pools</td>
<td>Create and edit workload pools and edit workload categories.</td>
</tr>
<tr>
<td>edit_workload_rules</td>
<td>Create and edit workload rules.</td>
</tr>
<tr>
<td>select_workload_pools</td>
<td>Assign scheduled and ad hoc searches to a workload pool.</td>
</tr>
</tbody>
</table>

The admin role can add the above capabilities to other roles to grant the specified permissions. For example, to allow a power user to assign search jobs to a workload pool, the admin can add the select_workload_pools capability to the power user role.

To add a capability to a role in Splunk Web, see Add and edit roles with Splunk Web.

To add a capability to a role in authorize.conf, see Add and edit roles with authorize.conf.
Allocate and monitor resources

Assign searches to workload pools

Before you can assign searches to workload pools, you must configure and enable workload management. See Configure workload management.

Workload management lets you allocate system resources to individual search processes. To allocate resources to a search, you must assign the search to a workload pool. How you assign a search to a workload pool depends on whether the search is a scheduled search or an ad-hoc search.

Assign a scheduled search to a workload pool

You can assign a scheduled search to a workload pool using Splunk Web, CLI, or REST.

When you assign a scheduled search to a workload pool, the pool information is written to savedsearches.conf. For more information, see savedsearches.conf.spec.

Assign a scheduled search using Splunk Web

To assign a scheduled search to a workload pool using Splunk Web, follow these steps:

1. Click on Settings > Searches, Reports, and Alerts.
2. Find the specific saved search, and click Edit > Advanced Edit.
3. In the Workload Pool field, enter the name of the pool.
4. Click Save.
   The workload pool information is written to local/savedsearches.conf and the scheduled search runs in the specified pool.

Assign a scheduled search using the CLI

To assign a scheduled search to a workload pool, run the following CLI command:

./splunk add saved-search -name <search_name> -workload_pool <pool_name>
Assign a scheduled search using REST

Send a POST request to the saved/searches/{name} endpoint. For example:

curl -k -u admin:pass
https://localhost:8089/services/searches/<search_name> -d
workload_pool=<pool_name>

Assign an ad-hoc search to a workload pool

You can assign an ad-hoc search to a workload pool using Splunk Web, CLI, or REST.

To assign an ad-hoc search to a workload pool, a role must have both the
list_workload_pools and select_workload_pools capabilities. See Set access
controls for workload management.

Assign an ad-hoc search using Splunk Web

1. In the Search bar, enter your ad-hoc search string.
2. Select a workload pool from the menu.
3. Run the search.
   The ad-hoc search job runs in the specified workload pool.

If you select Policy-Based Pool, workload management automatically
assigns the search to a pool based on any defined workload rules. If an
explicit match for the search is not found, workload management assigns
the search to the default pool.

The workload pool menu is only visible to roles that have
list_workload_pools and select_workload_pools capabilities.
4. Click **Job > Inspect Job > Search job properties.**
5. Confirm that the ad-hoc search ran in the specified pool. For example:

![Image of search results with workload pool set to pool_4]

**Assign an ad-hoc search using CLI**

To assign an ad-hoc search, run the following CLI command:

```
./splunk search "index=_internal" -workload_pool=<pool_name>
```

**Assign an ad-hoc search using REST**

Send a POST request to the search/jobs endpoint. For example:

```
curl -k -u admin:pass https://localhost:8089/services/search/jobs -d
search="search index=_internal" -d workload_pool=pool_1
```

**Change the workload pool for a running search**

You can re-assign an actively running search to a different workload pool using Splunk Web or REST. This applies to both scheduled searches and ad-hoc searches.

To change the workload pool for a running search, a role must have the `edit_workload_pools` capability. See Set access controls for workload management.

**Change workload pool using Splunk Web**

1. Click **Activity > Jobs.**
2. For the specific running search, click **Job > Edit Job Settings.**
3. Select a new pool from the Workload Pool menu.
Re-assigning an ad-hoc search on the Search bar triggers a new search process in the new pool. To continue running the same search process in a new pool, re-assign the search via the Job Activity page or REST endpoint.

**Change workload pool using REST**

Send a POST request to the search/jobs/{search_id}/control endpoint. For example:

```bash
curl -k -u admin:pass https://localhost:8089/services/search/jobs/{search_id}/control -d action=setworkloadpool -d workload_pool=<pool_name>
```

**Assign accelerated reports to workload pools**

You can assign any report that qualifies for acceleration to a workload pool.

Assigning an accelerated report to a workload pool with ample cpu and memory resources can help you minimize performance issues that can occur during report acceleration, which can be resource intensive.

You can assign an accelerated report to a workload pool using Splunk Web.

1. Click **Settings > Searches, Reports, and Alerts**.
2. Find the report you want to accelerate and click **Edit > Edit Acceleration**.
3. Select the **Accelerate Report** checkbox.
4. Select the **Summary Range** for the report acceleration.
5. Select a workload pool from the menu.
6. Click **Save**.

The workload pool that you specify in the Edit Acceleration modal is written to the `auto_summarize.workload_pool` setting in `savedsearches.conf`
Assign accelerated data models to workload pools

You can assign an accelerated data model to a workload pool using Splunk Web:

1. Click **Settings > Data models**.
2. Find the data model you want to accelerate and click **Edit > Edit Acceleration**.
3. Select the **Accelerate** checkbox.
4. Select the **Summary Range** for the data model acceleration.
5. Select a workload pool from the menu.
6. Click **Save**.

For more information on accelerated data models, see Accelerate data models in the *Knowledge Manager Manual*.

CPU overflow and memory sharing

If a search exceeds the maximum CPU resources allocated to its workload pool, it is considered a soft limit, and the pool can borrow available CPU resources from other pools.

Similarly, if a search exceeds the maximum memory limit allocated to its pool, the pool can share memory resources from other pools up to the total amount of memory available for all pools in the category.

For more information, see Resource allocation in workload management.

Search concurrency considerations in workload management

It's important to consider search concurrency constraints when you assign searches to workload pools.

*Search concurrency limits in workload management*

Splunk Enterprise enforces concurrent search limits globally. As a result, in the context of resource reservation in workload management, searches are not entirely isolated, and increasing search load in one workload pool can limit the number of searches you can run in other pools.
The following concurrent search quotas can impact search performance in workload management:

Scheduler concurrency limits
This limit determines the maximum number of searches that the scheduler can run concurrently. For detailed information, see How the scheduler determines concurrent search limits.

User/role search quotas
This quota determines the maximum number of historical searches allowable for a specific user/role. These quotas are configured with `srchJobsQuota` and related settings in `authorize.conf`. See `Authorize.conf`.

To minimize search performance issues due to concurrent search limits, make sure adequate search quota is available.

For detailed information on how concurrent search quotas work in a search head cluster environment, see How the cluster handles concurrent search quotas.

**Search priority in workload management**

Search priority in workload management is determined by two main factors:

Search scheduler priority
When the total number of searches reaches the maximum concurrent search limit, the search scheduler runs additional searches in priority order as search quota becomes available. To ensure that important searches are not skipped, you can set a scheduled search to high-priority in the search scheduler. For more information, see Configure the priority of scheduled reports.

Workload rules order
Workload rules control access to resources in workload pools based on app or role. The order of a rule determines which apps or roles, and therefore which searches, have priority access to a workload pool. For more information, see Create workload rules.

To avoid skipped searches and other search concurrency issues due to search priority, make sure to assign high-priority searches to workload pools that provide sufficient resources.
Monitor workload management

The monitoring console includes workload management dashboards that provide insight into various aspects of your workload management deployment, including configuration information, resource usage, and activity details for both single instance and distributed deployments.

To view workload management dashboards:

1. In Splunk Web, click Settings > Monitoring Console.
2. Click Resource Usage > Workload Management.
3. Select from the following dashboard pages:
   • Workload Management Overview
   • Workload Management Activity: Instance
   • Workload Management Activity: Deployment

Workload Management Overview

The Workload management overview page includes these dashboards:

• Workload management status and workload pool configuration
• CPU and memory usage

Workload management status and workload pool configuration

These dashboards shows information about your deployment, including whether workload management is supported and enabled on individual Linux instances. They also displays error messages and workload pool configuration details.
The CPU and memory usage dashboards show resource consumption on a per pool basis. You can use these dashboards to monitor the total amount of resources that assigned search processes are consuming within individual pools.

Monitoring workload pool consumption can help you provision resources efficiently and help you avoid assigning too many searches to a pool, which can impact search performance.
Workload Management Activity: Instance

This set of dashboards let you monitor workload management activity on a per instance basis. Use the snapshot view to monitor current resource usage across all workload pools.

Use the historical view to monitor CPU and memory usage of individual workload pools over a selected time range.

The historical view also displays the top 10 memory-consuming searches in a pool. This dashboard can help you identify searches that are consuming large amounts of CPU and memory resources.
Workload Management Activity: Deployment

This set of dashboards lets you monitor workload management activity across a distributed deployment. Use the snapshot view to monitor the current status of workload pools across multiple instances.
Use the historical view to monitor CPU and memory usage of individual workload pools over a selected time range across your distributed deployment.

<table>
<thead>
<tr>
<th>Select views:</th>
<th>All</th>
<th>Snapshot</th>
<th>Historical</th>
</tr>
</thead>
</table>

**Historical Charts**

- **Time Range:** Last 4 hours
- **Workload Pool:** default_search (search)

**Median Memory Usage**

- **Aggregation:** Median
- **Split by:** Type

![Median Memory Usage Chart](chart.png)
Upgrade workload management

Upgrade workload management

Workload management in Splunk Enterprise 7.3.x introduces a new level in the cgroup hierarchy that facilitates separation of resources into workload categories based on process type. There are three workload categories: search, ingest, and misc.

When you migrate from 7.2.x to 7.3.x, workload management automatically creates workload categories and places your existing workload pools under the appropriate category as follows:

- **search**: All existing search pools are placed under the search category. The existing `default_pool` becomes the `default_category_pool` in the search category.
- **ingest**: The existing `ingest_pool` becomes the `default_category_pool` in the ingest category.
- **misc**: The misc category is created with no pools defined and no resources allocated.

For more information, see Configure workload categories.

On upgrade, workload management implements these additional changes:

- The existing `workload_pools.conf` configuration file is converted to a new format that includes `workload_category` definitions. See `workload_pools.conf`.
- CPU and memory resources are recalculated for workload pools based on the amount of resources allocated to the parent workload category. CPU weights are recalculated as a ratio of the parent category’s value. Memory is recalculated as a percentage of the parent category’s value.

Workload management continues to function without interruption during the migration process.

Before you upgrade to 7.3.x, make sure workload management is enabled in 7.2.x. Or, if it is disabled, make sure you have a valid 7.2.x `workload_pools.conf` configuration. If the 7.2.x configuration is invalid, migration does not occur. See `workload_pools.conf`. 
Upgrade workload management on a non-clustered deployment

To upgrade workload management on a single instance or a non-clustered distributed deployment:

Upgrade each instance separately following the normal procedure for any Splunk Enterprise upgrade, as described in How to upgrade Splunk Enterprise in the Installation Manual.

During upgrade, workload management migrates to the updated feature architecture in 7.3.x and converts the existing `workload_pools.conf` configuration to the new file format.

Upgrade workload management on a search head cluster

To upgrade workload management on a search head cluster:

Upgrade the search head cluster using rolling upgrade. During upgrade, workload management migrates to the new format in 7.3.x and writes the changes to disk on each cluster member. See Use rolling upgrade in the Distributed Search manual.

During rolling upgrade, do not make any edits to the workload management configuration. This can introduce errors into the updated workload management configuration due to ongoing configuration replication on the cluster.

During rolling upgrade, due to ongoing configuration replication on the cluster, the migrated 7.2.x to 7.3.x configuration propagates the newly converted cpu_weight and mem_weight values back to members running the earlier version. While some deprecated stanzas from the 7.2.x format are retained in the 7.3.x format to maintain backwards compatibility, these recalculated cpu_weight and memory_weight values can affect ongoing searches.

Deployer considerations for workload management migration

During rolling upgrade, as part of migration, workload management reads the existing configuration in the app's default directory and writes the updated configuration to the app's local directory on the members. After the migration, if you use the deployer to push new workload management configurations to members using the default push mode, `merge_to_default`, the new configurations that you push to the app's default directory will be overridden by
the migrated configuration in the app's local directory due to configuration file precedence.

To avoid this behavior, push the new workload management configuration to members as follows:

1. Copy the upgraded `workload_pools.conf` file from the app's local directory on any member to the app's local directory on the deployer.
2. Delete the local `workload_pools.conf` file on each cluster member.
3. Set the deployer bundle push mode that fits your use case. See Choose a deployer push mode.
4. Push the configuration bundle from the deployer to the cluster. See Push the configuration bundle.

After migration to 7.3.x, do not push workload management 7.2.x configurations to the members. Pushing 7.2.x configurations can cause workload management to fail. You must push 7.3.x configurations only.

**Upgrade workload management on an indexer cluster**

To upgrade workload management on an indexer cluster:

1. Upgrade the cluster master, following the normal procedure for any Splunk Enterprise upgrade. See How to upgrade Splunk Enterprise in the Installation Manual.
2. Upgrade the search heads connected to the indexer cluster. If you are upgrading a search head cluster, use rolling upgrade.
3. Upgrade the peer nodes. See Upgrade an indexer cluster in Managing indexers and clusters of indexers.

When you upgrade the peer nodes, workload management converts the existing configuration to the new format in 7.3.x, but does not write the configuration to disk. Instead, a message appears in Splunk Web on the search heads alerting you that indexers are out of sync between memory and disk, and that a bundle push is required to update the peer node configuration, as described in steps 4 and 5.
4. Copy the upgraded configuration from any search head to the cluster master.
5. Push the configuration bundle to the peer nodes. See Distribute the configuration bundle.

50
**Search head and indexer version compatibility**

You can run workload management with search heads and cluster master on version 7.3.x and indexers on 7.2.x. However, due to the updated `workload_pools.conf` file format in 7.3.x, indexers running 7.2.x cannot understand the pool information sent from search heads running 7.3.x. In this case, indexers always use the default_pool specified in the 7.2.x version of `workload_pools.conf` to run searches.