# Table of Contents

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

**Requirements**............................................................................................4  
  Requirements..............................................................................................4

**Set up Linux for workload management**..................................................6  
  Set up Linux for workload management......................................................6  
  Configure Linux systemd for workload management.................................6  
  Configure Linux systems not running systemd for workload management....11

**Configure workload management**............................................................16  
  Configure workload management.............................................................16  
  Configure workload management on distributed deployments.................26  
  Set access controls for workload managment............................................28

**Allocate and monitor resources**...............................................................29  
  Assign searches to workload pools...........................................................29  
  Monitor workload management..................................................................33
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 be allocated adequate system resources, while less important 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 individual search processes in Splunk Enterprise.

Workload management lets you:

- Reserve system resources for search and indexing processes.
- Prioritize critical search workloads.
- Prevent over-usage of system resources.
- Avoid data-ingestion latency due to heavy search load.
- Create rules to control access to resources based on app or role.

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

For prerequisite Linux configuration requirements, 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

Workload management lets you reserve groups of CPU and memory resources on Linux operating systems, and allocate those resources to search 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.

Workload management concepts and features

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

**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. You can configure systemd to allow splunkd to manage 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 systemd distributions.

**Workload pools**

A workload pool is a specified amount of CPU and memory resources that you can define and allocate to search processes in Splunk Enterprise. Each workload pool reserves a subset of the total amount of CPU and memory available on the system. 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.
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.

Linux preflight checks for workload management require version 7.2.2. 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

To use workload management in Splunk Enterprise, you must set up Linux cgroups on your underlying Linux operating system. How you set up cgroups for workload management depends on whether or not Linux is running under systemd. To determine if your system is running under 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 distributions running under systemd, see Configure systemd distributions.

To configure cgroups on Linux distributions not running systemd, see Configure non-systemd distributions.

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 \texttt{systemd}:

\begin{center}
\includegraphics[width=0.5\textwidth]{systemd_cgroup_hierarchy.png}
\end{center}

For more information, see \texttt{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 \texttt{systemd} to manage \texttt{splunkd} as a service:

- **Configure \texttt{systemd} manually.**
- **Configure \texttt{systemd} using the \texttt{enable boot-start} command.**

Configuring \texttt{systemd} using \texttt{enable boot-start} requires Splunk Enterprise version 7.2.2 or later.

**Permissions requirements for \texttt{systemd}**

\texttt{systemd} has the following permissions requirements:

- Non-root users must have super user permissions to manually configure \texttt{systemd} on Linux.
• Non-root users must have super user permissions to run `start`, `stop`,
  `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 Linux
distribution.

You must use `sudo` to run `systemctl 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 both 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:

   ```ini
   [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 <username>:<username> /sys/fs/cgroup/cpu/system.slice/%n"
ExecStartPost=/bin/bash -c "chown -R <username>:<username> /sys/fs/cgroup/memory/system.slice/%n"

[Install]
WantedBy=multi-user.target

Regarding these lines in the unit file:

ExecStartPost=/bin/bash -c "chown -R <username>:<username> /sys/fs/cgroup/cpu/system.slice/%n"
ExecStartPost=/bin/bash -c "chown -R <username>:<username> /sys/fs/cgroup/memory/system.slice/%n"

If a group does not exist on the system with the name <username>, the splunkd service will not start. To workaround this issue, manually update the unit file with the correct group name.

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, see Systemd unit file properties.

7. Reload the unit file.

```
sudo systemctl daemon-reload
```

8. Start splunkd as a systemd service.

```
sudo systemctl start Splunkd.service
```

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 &lt;USER&gt;:&lt;GROUP of USER&gt;/sys/fs/cgroup/cpu or memory/&lt;SPLUNK_SERVER_NAME&gt;.service</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>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.</td>
</tr>
<tr>
<td>Property</td>
<td>Expected Value</td>
</tr>
<tr>
<td>----------------</td>
<td>------------------------------</td>
</tr>
<tr>
<td></td>
<td>Default is 1024.</td>
</tr>
<tr>
<td>User, Group</td>
<td><code>&lt;Splunk Owner&gt; &lt;Splunk Group&gt;</code></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 `systemctl 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 `systemctl 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:
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:

   ```
   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:

   ```
   # 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
cd /sys/fs/cgroup/memory/splunk/
   echo 2048 > cpu.shares
   echo 14G > memory.limit_in_bytes
   ```

5. Assign physical memory for the splunk cgroup:
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.

You can 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 search pool and a default ingest pool. 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. Create workload pools.
3. Create workload rules.
4. Enable workload management.
5. Check workload management status.

Run preflight checks

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

If all preflight checks pass, then your system is set up correctly and you can 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.

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 <code>Splunkd.service</code> is missing.</td>
</tr>
<tr>
<td></td>
<td>For systemd, the base directory is <code>/sys/fs/cgroup/cpu/system.slice/&lt;unit_file_name&gt;</code>. The unit_file_name is <code>&lt;splunk_server_name&gt;.service</code>. The splunk_server_name must match the Splunk server name in <code>splunk-launch.conf</code>. The default value is <code>Splunkd</code>. See Configure systemd distributions for workload management.</td>
</tr>
<tr>
<td></td>
<td>For non-systemd, the base directory is <code>/sys/fs/cgroup/cpu/splunk</code>. The base directory name must match the <code>workload_pool_base_dir_name</code> defined in <code>workload_pools.conf</code>. The default value is <code>splunk</code>. See Configure non-systemd distributions.</td>
</tr>
<tr>
<td>CPU Splunk base directory permissions</td>
<td>CPU Splunk base directory <code>Splunkd.service</code> requires read and write permissions.</td>
</tr>
<tr>
<td></td>
<td>For systemd, permissions must be set for non-root user in the <code>Splunkd.service</code> unit file. See Configure systemd distributions for workload management.</td>
</tr>
<tr>
<td></td>
<td>For non-systemd, use <code>chown</code> to grant permissions to the splunk base directory. See Configure non-systemd distributions.</td>
</tr>
<tr>
<td>Memory Splunk base directory</td>
<td>Memory Splunk base directory <code>Splunkd.service</code> is missing.</td>
</tr>
<tr>
<td>Name</td>
<td>Mitigation</td>
</tr>
<tr>
<td>------</td>
<td>------------</td>
</tr>
</tbody>
</table>
| present | **For systemd**, the base directory is /sys/fs/cgroup/memory/system.slice/<unit_file_name>.  
See **Configure systemd distributions for workload management**.  

**For non-systemd**, the base directory is /sys/fs/cgroup/memory/splunk.  
See **Configure non-systemd distributions**. |
| Memory Splunk base directory permissions | Memory Splunk base directory `Splunkd.service` requires read and write permissions.  
**For systemd**, permissions must be set for non-root user in the `Splunkd.service` unit file.  
See **Configure systemd distributions for workload management**.  

**For non-systemd**, use `chown` to grant permissions to the splunk base directory.  
See **Configure non-systemd distributions**. |
| Unit file present | The unit file `Splunkd.service` is missing.  
The unit file is located under `/etc/systemd/system` with the name `<splunk_server_name>.service`.  
splunk_server_name is set in splunk-launch.conf.  
See **Configure systemd distributions for workload management**. |
| Delegate property set to true | The **Delegate** property in the unit file must be set to **true**. |
| Splunk launched under systemd | splunkd is running as a **systemd** service.  
In the unit file, the **Restart** property must be set to **always**.  
The **ExecStart** property must include `_internal_launch_under_systemd`.  

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 start to configure workload management.

2. If any preflight checks fail, a page appears showing the 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.

**Create workload pools**

A workload pool is a specified amount of CPU and memory resources that you can define and allocate to search processes.

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

- **Default search pool**
  Searches that are not explicitly mapped to a workload rule are assigned to this pool by default.

- **Default ingest pool**
  Indexing and other non-search processes are assigned to this pool by default.

You can specify if a pool is a default search pool or default ingest pool when you create the workload pool.

**Create a workload pool in Splunk Web**

1. In Splunk Web, click **Settings > Workload Management**.
2. Click **Add Workload Pool**.
3. Configure your new workload pool 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 pool.</td>
</tr>
</tbody>
</table>
Valid characters are alphanumeric and underscore only.

<table>
<thead>
<tr>
<th>Field</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU %</td>
<td>Specify the percentage of the total CPU weight for the cpu control group.</td>
</tr>
<tr>
<td>Memory %</td>
<td>Specify the percentage of the total memory weight for the memory control group.</td>
</tr>
<tr>
<td>Default Search Pool</td>
<td>Toggle the switch to make this pool the default search pool.</td>
</tr>
<tr>
<td>Default Ingest Pool</td>
<td>Toggle the switch to make this pool the default ingest pool.</td>
</tr>
</tbody>
</table>

An individual workload pool cannot be both a default search pool and default ingest pool.

4. Click Submit.

The new workload pool appears in the Workload Management UI.

**Create a workload pool using the CLI**

Run the following CLI command:

```
./splunk add workload-pool <pool_name> [-cpu_weight <group_name> -mem_weight <group_name> -default_pool <true|false> -ingest_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 and edit workload_pools.conf**

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

```
$SPLUNK_HOME/etc/apps/<app_name>/local/workload_pools.conf
```

You can directly edit `workload_pools.conf` to create new pools and modify existing pools.

[generic]
default_pool = pool_1
ingest_pool = pool_2
enabled = 0
[workload_pool:pool_1]
cpu_weight = 20
mem_weight = 40

[workload_pool:pool_2]
cpu_weight = 80
mem_weight = 80

For more information, see workload_pools.conf.

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.

You can only specify a single predicate for each workload rule.

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 single predicate condition to access the workload pool. Enter as &lt;type&gt;=&lt;value&gt;, where valid &lt;type&gt; is &quot;app&quot; or &quot;role&quot;. For example, a workload rule with predicate &quot;app=itsi&quot; maps all searches belonging to the ITSI app to the corresponding workload pool. Similarly, &quot;role=admin&quot; maps all searches for the admin role to the corresponding workload pool.</td>
</tr>
<tr>
<td></td>
<td>Select the workload pool to which this rule applies.</td>
</tr>
</tbody>
</table>
Create a workload rule using the CLI

Run the following CLI command:

./splunk add workload-rule <rule_name> -predicate <type> -workload_pool <pool>

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 and edit workload_rules.conf

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

You can directly edit the workload_rules.conf file to create new rules or modify existing rules.

workload_rules.conf defines both the mappings to workload pools and the order in which rules are evaluated.

```
[workload_rules_order]
rules = my_analyst_rule, my_app_rule
rules_number = 2

[workload_rule:my_app_rule]
predicate = app=search
workload_pool = my_app_pool

[workload_rule:my_analyst_rule]
predicate = role=analyst
workload_pool = my_analyst_pool
```

For more information, 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:
Here is an example of the output from the command:

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

Workload Management Status:
    Enabled: 1
    Supported: 1
    Ingest Pool: pool_5
    Default Pool: pool_2
    Error:
    Pending: 1

Workload Pools:
    pool_2:
        CPU Group: /sys/fs/cgroup/cpu/splunk/pool_2
        Memory Group: /sys/fs/cgroup/memory/splunk/pool_2
        CPU Weight: 20.00
        Memory Weight: 20.00
    pool_4:
        CPU Group: /sys/fs/cgroup/cpu/splunk/pool_4
        Memory Group: /sys/fs/cgroup/memory/splunk/pool_4
        CPU Weight: 35.00
        Memory Weight: 35.00
    pool_5:
        CPU Group: /sys/fs/cgroup/cpu/splunk/pool_5
        Memory Group: /sys/fs/cgroup/memory/splunk/pool_5
        CPU Weight: 30.00
        Memory Weight: 30.00
    pool_6:
        CPU Group: /sys/fs/cgroup/cpu/splunk/pool_6
        Memory Group: /sys/fs/cgroup/memory/splunk/pool_6
        CPU Weight: 15.00
        Memory Weight: 15.00

Workload Rules:
    rule_2:
        Order: 1
        Predicate: role=super_user
        Workload Pool: pool_2

    rule_3:
        Order: 2
        Predicate: role=analyst
```
Workload Pool: pool_2

rule_5:
  Order: 3
  Predicate : app=splunk_instrumentation
  Workload Pool: pool_4

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.</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.</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 the ad-hoc search’s context, such as app or role. 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:

```
searchTotalEventsCount 10
searchTotalEstimatedEventsCount None
id 6f37692e970c9493a25...5b
statusStates Job
is new
workload_pool pool_4
AdditionalInfo timeline
```

**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>
```

**Search concurrency considerations in workload management**

It's important that you consider concurrency-related 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 details and system resource usage.

To view workload management dashboards:

1. In Splunk Web, click **Settings > Monitoring Console**.
2. Click **Resource Usage > Resource Usage: Instance > Workload Management**.
   The Workload Management dashboard page opens.

**Workload management status dashboard**

The workload management status dashboard shows information about your deployment, including whether workload management is supported and enabled on individual Linux instances. It also displays error messages and workload pool configuration details.
CPU and memory usage dashboards

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.
**CPU and memory overflow and limits**

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.

If a search exceeds the maximum memory allocated to its workload pool, it is considered a hard limit, and the pool does not borrow memory from other pools. Instead, memory swaps to disk. If you run out of swap space, splunkd kills the search process.