



## 2.4. OBTAINING INFORMATION ABOUT CONTROL GROUPS

Use the `systemctl` command to list system units and to view their status. Also, the `systemd-cgls` command is provided to view the hierarchy of control groups and `systemd-cgtop` to monitor their resource consumption in real time.

### 2.4.1. Listing Units

Use the following command to list all active units on the system:

```
~]# systemctl list-units
```

The `list-units` option is executed by default, which means that you will receive the same output when you omit this option and execute just:

```
~]$systemctl UNIT LOAD ACTIVE SUB DESCRIPTION abrt-ccpp.service loaded active  
exited Install ABRT coredump hook abrt-oops.service loaded active running ABRT  
kernel log watcher abrt-vmcore.service loaded active exited Harvest vmcores for  
ABRT abrt-xorg.service loaded active running ABRT Xorg log watcher ...
```

The output displayed above contains five columns:

- *UNIT* – the name of the unit that also reflects the unit's position in the cgroup tree. As mentioned in [the section called “Systemd Unit Types”](#), three unit types are relevant for resource control: *slice*, *scope*, and *service*. For a complete list of `systemd`'s unit types, see the chapter called *Managing Services with systemd* in [Red Hat Enterprise Linux 7 System Administrators Guide](#).
- *LOAD* – indicates whether the unit configuration file was properly loaded. If the unit file failed to load, the field contains the state *error* instead of *loaded*. Other unit load states are: *stub*, *merged*, and *masked*.
- *ACTIVE* – the high-level unit activation state, which is a generalization of SUB.
- *SUB* – the low-level unit activation state. The range of possible values depends on the unit type.
- *DESCRIPTION* – the description of the unit's content and functionality.

By default, `systemctl` lists only active units (in terms of high-level activations state in the *ACTIVE* field). Use the `--all` option to see inactive units too. To limit the amount of information in the output list, use the `--type ( -t )` parameter that requires a comma-separated list of unit types such as *service* and *slice*, or unit load states such as *loaded* and *masked*.

### Example 2.8. Using `systemctl list-units`

To view a list of all slices used on the system, type:

```
~]$ systemctl -t slice
```

To list all active masked services, type:

```
~]$ systemctl -t service,masked
```

To list all unit files installed on your system and their status, type:

```
~]$ systemctl list-unit-files
```

## 2.4.2. Viewing the Control Group Hierarchy

The aforementioned listing commands do not go beyond the unit level to show the actual processes running in cgroups. Also, the output of `systemctl` does not show the hierarchy of units. You can achieve both by using the `systemd-cgls` command that groups the running process according to cgroups. To display the whole cgroup hierarchy on your system, type:

```
~]$ systemd-cgls
```

When `systemd-cgls` is issued without parameters, it returns the entire cgroup hierarchy. The highest level of the cgroup tree is formed by slices and can look as follows:

```
├─system | ├─1 /usr/lib/systemd/systemd --switched-root --system --deserialize 20  
| ... | ├─user | ├─user-1000 | | └─ ... | ├─user-2000 | | └─ ... | ... | └─machine  
├─machine-1000 | └─ ... ...
```

Note that machine slice is present only if you are running a virtual machine or a container. For more information on the cgroup tree, see [the section called “Systemd Unit Types”](#).

To reduce the output of `systemd-cgls`, and to view a specified part of the hierarchy, execute:

```
~]$ systemd-cgls name
```

Replace *name* with a name of the resource controller you want to inspect.

As an alternative, use the `systemctl status` command to display detailed information about a system unit. A cgroup subtree is a part of the output of this command.

```
~]$ systemctl name
```

To learn more about `systemctl status`, see the chapter called *Managing Services with systemd* in [Red Hat Enterprise Linux 7 System Administrators Guide](#).

## Example 2.9. Viewing the Control Group Hierarchy

To see a cgroup tree of the `memory` resource controller, execute:

```
~]$ systemd-cgls memory memory: └─ 1 /usr/lib/systemd/systemd --switched-root --
system --deserialize 23 └─ 475 /usr/lib/systemd/systemd-journald ...
```

The output of the above command lists the services that interact with the selected controller. A different approach is to view a part of the cgroup tree for a certain service, slice, or scope unit:

```
~]# systemctl status httpd.service httpd.service - The Apache HTTP Server
Loaded: loaded (/usr/lib/systemd/system/httpd.service; enabled) Active: active
(running) since Sun 2014-03-23 08:01:14 MDT; 33min ago Process: 3385
ExecReload=/usr/sbin/httpd $OPTIONS -k graceful (code=exited, status=0/SUCCESS)
Main PID: 1205 (httpd) Status: "Total requests: 0; Current requests/sec: 0;
Current traffic: 0 B/sec" CGroup: /system.slice/httpd.service └─1205
/usr/sbin/httpd -DFOREGROUND └─3387 /usr/sbin/httpd -DFOREGROUND └─3388
/usr/sbin/httpd -DFOREGROUND └─3389 /usr/sbin/httpd -DFOREGROUND └─3390
/usr/sbin/httpd -DFOREGROUND └─3391 /usr/sbin/httpd -DFOREGROUND ...
```

Besides the aforementioned tools, **systemd** also provides the `machinectl` command dedicated to monitoring Linux containers.

### 2.4.3. Viewing Resource Controllers

The aforementioned `systemctl` commands enable monitoring the higher-level unit hierarchy, but do not show which resource controllers in Linux kernel are actually used by which processes. This information is stored in dedicated process files, to view it, type as `root` :

```
~]# cat proc/PID/cgroup
```

Where *PID* stands for the ID of the process you wish to examine. By default, the list is the same for all units started by **systemd**, since it automatically mounts all default controllers. See the following example:

```
~]# cat proc/27/cgroup 10:hugetlb:/ 9:perf_event:/ 8:blkio:/ 7:net_cls:/  
6:freezer:/ 5:devices:/ 4:memory:/ 3:cpuacct,cpu:/ 2:cpuset:/ 1:name=systemd:/
```

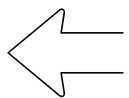
By examining this file, you can determine if the process has been placed in the correct cgroups as defined by the systemd unit file specifications.

## 2.4.4. Monitoring Resource Consumption

The `systemd-cgls` command provides a static snapshot of the cgroup hierarchy. To see a dynamic account of currently running cgroups ordered by their resource usage (CPU, Memory, and IO), use:

```
~]# systemd-cgtop
```

The behavior, provided statistics, and control options of `systemd-cgtop` are akin of those of the `top` utility. See `systemd-cgtop (1)` manual page for more information.



[2.3.2. Modifying Unit Files](#)

[2.5. Additional Resources](#)

