NAME

lvmsystemid — LVM system ID

DESCRIPTION

The **lvm**(8) system ID restricts Volume Group (VG) access to one host. This is useful when a VG is placed on shared storage devices, or when local devices are visible to both host and guest operating systems. In cases like these, a VG can be visible to multiple hosts at once, and some mechanism is needed to protect it from being used by more than one host at a time.

A VG's system ID identifies one host as the VG owner. The host with a matching system ID can use the VG and its LVs, while LVM on other hosts will ignore it. This protects the VG from being accidentally used from other hosts.

The system ID is a string that uniquely identifies a host. It can be configured as a custom value, or it can be assigned automatically by LVM using some unique identifier already available on the host, e.g. machine-id or uname.

When a new VG is created, the system ID of the local host is recorded in the VG metadata. The creating host then owns the new VG, and LVM on other hosts will ignore it. When an existing, exported VG is imported (vgimport), the system ID of the local host is saved in the VG metadata, and the importing host owns the VG.

A VG without a system ID can be used by LVM on any host where the VG's devices are visible. When system IDs are not used, device filters should be configured on all hosts to exclude the VG's devices from all but one host.

A **foreign VG** is a VG seen by a host with an unmatching system ID, i.e. the system ID in the VG metadata does not match the system ID configured on the host. If the host has no system ID, and the VG does, the VG is foreign and LVM will ignore it. If the VG has no system ID, access is unrestricted, and LVM can access it from any host, whether the host has a system ID or not.

Changes to a host's system ID and a VG's system ID can be made in limited circumstances (see vgexport and vgimport). Improper changes can result in a host losing access to its VG, or a VG being accidentally damaged by access from an unintended host. Even limited changes to the VG system ID may not be perfectly reflected across hosts. A more coherent view of shared storage requires an inter-host locking system to coordinate access.

Valid system ID characters are the same as valid VG name characters. If a system ID contains invalid characters, those characters are omitted and remaining characters are used. If a system ID is longer than the maximum name length, the characters up to the maximum length are used. The maximum length of a system ID is 128 characters.

Print the system ID of a VG to check if it is set:

vgs −o systemid *VG*

Print the system ID of the local host to check if it is configured:

lvm systemid

Limitations and warnings

To benefit fully from system ID, all hosts should have a system ID configured, and all VGs should have a system ID set. Without any method to restrict access, e.g. system ID or device filters, a VG that is visible to multiple hosts can be accidentally damaged or destroyed.

- A VG without a system ID can be used without restriction from any host where it is visible, even from hosts that have a system ID.

- Many VGs will not have a system ID set because LVM has not enabled it by default, and even when enabled, many VGs were created before the feature was added to LVM or enabled. A system ID can be assigned to these VGs by using vgchange −−systemid (see below).

- Two hosts should not be assigned the same system ID. Doing so defeats the purpose of distinguishing different hosts with this value.

- Orphan PVs (or unused devices) on shared storage are unprotected by the system ID feature. Commands that use these PVs, such as vgcreate or vgextend, are not prevented from performing conflicting operations and corrupting the PVs. See the **orphans** section for more information.

- The system ID does not protect devices in a VG from programs other than LVM.

- A host using an old LVM version (without the system ID feature) will not recognize a system ID set in VGs. The old LVM can read a VG with a system ID, but is prevented from writing to the VG (or its LVs). The system ID feature changes the write mode of a VG, making it appear read-only to previous versions of LVM.

  This also means that if a host downgrades to the old LVM version, it would lose access to any VGs it had created with a system ID. To avoid this, the system ID should be removed from local VGs before downgrading LVM to a version without the system ID feature.

Types of VG access

A local VG is meant to be used by a single host.

A shared or clustered VG is meant to be used by multiple hosts.

These can be further distinguished as:

Unrestricted: A local VG that has no system ID. This VG type is unprotected and accessible to any host.

Owned: A local VG that has a system ID set, as viewed from the host with a matching system ID (the owner). This VG type is acessible to the host.

Foreign: A local VG that has a system ID set, as viewed from any host with an unmatching system ID (or no system ID). It is owned by another host. This VG type is not accessible to the host.

Exported: A local VG that has been exported with vgexport and has no system ID. This VG type can only be accessed by vgimport which will change it to owned.

Shared: A shared or "lockd" VG has the lock_type set and has no system ID. A shared VG is meant to be used on shared storage from multiple hosts, and is only accessible to hosts using lvmlockd. Applicable only if LVM is compiled with lvmlockd support.

Clustered: A clustered or "clvm" VG has the clustered flag set and has no system ID. A clustered VG is meant to be used on shared storage from multiple hosts, and is only accessible to hosts using clvmd. Applicable only if LVM is compiled with clvm support.

Host system ID configuration

A host's own system ID can be defined in a number of ways. lvm.conf global/system_id_source defines the method LVM will use to find the local system ID:

none

LVM will not use a system ID. LVM is allowed to access VGs without a system ID, and will create new VGs without a system ID. An undefined system_id_source is equivalent to none.

lvm.conf
```
global {
  system_id_source = "none"
}
```

machineid

The content of /etc/machine-id is used as the system ID if available. See **machine-id**(5) and **systemd-machine−id-setup**(1) to check if machine-id is available on the host.

lvm.conf
```
global {
  system_id_source = "machineid"
}
```

uname

The string utsname.nodename from **uname**(2) is used as the system ID. A uname beginning with "localhost" is ignored and equivalent to none.

lvm.conf
```
global {
  system_id_source = "uname"
}
```

lvmlocal

The system ID is defined in lvmlocal.conf local/system_id.

lvm.conf
```
global {
  system_id_source = "lvmlocal"
}
```

lvmlocal.conf
```
local {
  system_id = "example_name"
}
```

file

The system ID is defined in a file specified by lvm.conf global/system_id_file.

lvm.conf
```
global {
   system_id_source = "file"
   system_id_file = "/path/to/file"
}
```

Changing system_id_source will likely cause the system ID of the host to change, which will prevent the host from using VGs that it previously used (see extra_system_ids below to handle this.)

If a system_id_source other than none fails to produce a system ID value, it is the equivalent of having none. The host will be allowed to access VGs with no system ID, but will not be allowed to access VGs with a system ID set.

Overriding system ID

In some cases, it may be necessary for a host to access VGs with different system IDs, e.g. if a host's system ID changes, and it wants to use VGs that it created with its old system ID. To allow a host to access VGs with other system IDs, those other system IDs can be listed in lvmlocal.conf local/extra_system_ids.

lvmlocal.conf
```
local {
   extra_system_ids = [ "my_other_name" ]
}
```

A safer option may be configuring the extra values as needed on the command line as:
−−config 'local/extra_system_ids=["*id***"]'**

vgcreate

In vgcreate, the host running the command assigns its own system ID to the new VG. To override this and set another system ID:

vgcreate −−systemid *SystemID VG PVs*

Overriding the host's system ID makes it possible for a host to create a VG that it may not be able to use. Another host with a system ID matching the one specified may not recognize the new VG without manually rescanning devices.

If the −−systemid argument is an empty string (""), the VG is created with no system ID, making it accessible to other hosts (see warnings above.)

report/display

The system ID of a VG is displayed with the "systemid" reporting option.

Report/display commands ignore foreign VGs by default. To report foreign VGs, the −−foreign option can be used. This causes the VGs to be read from disk.

vgs −−foreign −o +systemid

When a host with no system ID sees foreign VGs, it warns about them as they are skipped. The host should be assigned a system ID, after which standard reporting commands will silently ignore foreign VGs.

vgexport/vgimport

vgexport clears the VG system ID when exporting the VG.

vgimport sets the VG system ID to the system ID of the host doing the import.

vgchange

A host can change the system ID of its own VGs, but the command requires confirmation because the host may lose access to the VG being changed:

vgchange −−systemid *SystemID VG*

The system ID can be removed from a VG by specifying an empty string ("") as the new system ID. This makes the VG accessible to other hosts (see warnings above.)

A host cannot directly change the system ID of a foreign VG.

To move a VG from one host to another, vgexport and vgimport should be used.

To forcibly gain ownership of a foreign VG, a host can temporarily add the foreign system ID to its extra_system_ids list, and change the system ID of the foreign VG to its own. See Overriding system ID above.

shared VGs

A shared VG has no system ID set, allowing multiple hosts to use it via lvmlockd. Changing a VG to shared will clear the existing system ID. Applicable only if LVM is compiled with lvmlockd support.

clustered VGs

A clustered/clvm VG has no system ID set, allowing multiple hosts to use it via clvmd. Changing a VG to clustered will clear the existing system ID. Changing a VG to not clustered will set the system ID to the host running the vgchange command.

creation_host

In vgcreate, the VG metadata field creation_host is set by default to the host's uname. The creation_host cannot be changed, and is not used to control access. When system_id_source is "uname", the system_id and creation_host fields will be the same.

orphans

Orphan PVs are unused devices; they are not currently used in any VG. Because of this, they are not protected by a system ID, and any host can use them. Coordination of changes to orphan PVs is beyond the scope of system ID. The same is true of any block device that is not a PV.

SEE ALSO

vgcreate(8), **vgchange**(8), **vgimport**(8), **vgexport**(8), **vgs**(8), **lvmlockd**(8), **lvm.conf**(5), **machine-id**(5), **uname**(2)