Namespace Overview

Presented & Created By : Janvi

Namespace in Linux

Namespaces are the feature of the linux kernel that partitions kernel resources such that one set of processes sees one set of resources and another set of processes sees a different set of resources. The feature works by having the same namespace for a group of resources and processes , but those namespaces refer to distinct resources.

Types of Namespaces

1. Process Isolation (PID namespace)
2. Network Interfaces (net namespace)
3. Unix Timesharing System (uts namespace)
4. User namespace
5. Mount (mnt namespace)
6. Interprocess Communication (IPC)
7. CGroups

Let’s now discuss each one of these in detail….

System calls in Namespace

1. clone() - this creates a new process and a new namespace.
   - Process creation and termination methods , fork() and exit() are patched to handle the new namespace clone_new* flags.
2. unshare() - This does not create a process , it creates a new namespace and attaches the current processes to it.
3. setns() - a new system call was added , for joining an existing namespace.

1. Process Isolation (PID Namespace)

- PID namespace isolates the process ID number space , means the processes in different PID namespaces can have same PID.
- Each PID namespace has its own numbering starting at 1 which is unique per process namespace that means that if PID 1 goes away the whole namespace is deleted.
- PID namespace allow containers to provide functionality such as suspending /resuming the set of processes in the container.
- PID namespace requires a kernel that is configured with CONFIG_PID_NS. PID namespaces are nested.
- It uses CLONE_NEWPID flag.

2. Network Namespace (net namespace)

- Isolation of the system resources associated with networking : networking devices , IPv4 , IPv6 protocol stacks , IP routing tables , firewall rules, port numbers , various files under proc sysnet etc.
- In terms of devices , a physical network device can live in exactly one network namespace while a virtual network device provides a pipe-like abstraction that can be used to create tunnels between network namespaces which is done using a bridge.
- Network namespaces requires a kernel that is configured with the CONFIG_NET_NS option.

3. UNIX Timesharing System (uts)

- Isolation of Hostname and domainname.
- Changes made to these two are visible to all other processes in the same UTS namespace , but are not visible to processes in other UTS namespaces.
- UTS namespaces requires a kernel that is configured with the CONFIG_UTS_NS option.
- A process create a new UTS namespace with the CLONE_NEWUTS flag , the hostname and domain of the new UTS namespace are copied from the corresponding values in the caller’s UTS namespace.

4. User Namespace

- Feature to provide privilege isolation and user identification segregation across multiple sets of processes.
- User namespace are nested and each new user namespace is considered to be a child of the user namespace that created it.
- A user namespace contains a mapping table converting user IDs from the container’s point of view to the system’s point of view.
- The user namespace allows a process to have root privileges within the namespace , without giving it that access to processes outside of the namespace.
- User namespace uses the CLONE_NEWUSER flag to create the new parent or new child process.

5. Mount (mnt namespace)

- Mount namespaces provide isolation of the list of mounts seen by the processes in each namespace instance.
- A new mount namespace is created using either clone or unshare(2) with the **CLONE_NEWNS** flag. When a new mount namespace is created, its mount list is initialized as follows:

  * If the namespace is created using clone, the mount list of the child's namespace is a copy of the mount list in the parent process's mount namespace.

  * If the namespace is created using unshare, the mount list of the new namespace is a copy of the mount list in the caller's previous mount namespace.

.

6. Inter Process Communication (IPC)

- IPC namespace provides isolation for the process communication mechanism such as semaphores , message queues , shared memory segments and more.
- The processes inside an IPC namespace can’t see or interact with the IPC resources of the upper namespace.
- It isolates the processes from SysV style inter-process communication.
- When an IPC namespace is destroyed , all IPC objects in the namespace are automatically destroyed.
- IPC namespaces require a kernel configured with CONFIG_IPC_NS option.

7. Control Groups (Cgroups)

- Cgroups provide isolation for resource management solution (handling groups)
- By using cgroups, system administrators gain fine-grained control over allocating, prioritizing, denying, managing, and monitoring system resources. Hardware resources can be appropriately divided up among tasks and users, increasing overall efficiency.
- The cgroup modules are not located in one folder but scattered in the kernel tree .
- When a process creates a new cgroup namespace using clone or unshare with the CLONE_NEWCGROUP flag , its current cgroup directory becomes the cgroup root directories of the new namespace.
-