



Docker под капотом

Спикер: Елизавета Михно

Namespaces

- Pid
- NET
- MOUNT
- UTS
- IPC

СлѢДМ Docker

slurm.io

СТЕРА Docker

slurm.io

Option	Result
<code>memory=inf, memory-swap=inf</code> (default)	There is no memory limit in container. The container can use as much memory as needed.
<code>memory=L<inf, memory-swap=inf</code>	(specify memory and set memory-swap as <code>-1</code>) The container is not allowed to use more than <code>L</code> bytes of memory, but can use as much swap as is needed (if the host supports swap memory).
<code>memory=L<inf, memory-swap=2*L</code>	(specify memory without memory-swap) The container is not allowed to use more than <code>L</code> bytes of memory, swap <i>plus</i> memory usage is double of that.
<code>memory=L<inf, memory-swap=S<inf, L<=S</code>	(specify both memory and memory-swap) The container is not allowed to use more than <code>L</code> bytes of memory, swap <i>plus</i> memory usage is limited by <code>S</code> .

CPU

- `--cpuset-cpus`
- `--cpu-period`
- `--cpu-quota`



- `--privileged`
- `--cap-drop`

Capability Key	Capability Description
SETCAP	Modify process capabilities.
MKNOD	Create special files using mknod(2).
AUDIT_WRITE	Write records to kernel auditing log.
CHOWN	Make arbitrary changes to file UIDs and GIDs (see chown(2)).
NET_RAW	Use RAW and PACKET sockets.
DAC_OVERRIDE	Bypass file read, write, and execute permission checks.
FOwner	Bypass permission checks on operations that normally require the file system UID of the process to match the UID of the file.
FSETID	Don't clear set-user-ID and set-group-ID permission bits when a file is modified.
KILL	Bypass permission checks for sending signals.
SETGID	Make arbitrary manipulations of process GIDs and supplementary GID list.
SETUID	Make arbitrary manipulations of process UIDs.
NET_BIND_SERVICE	Bind a socket to internet domain privileged ports (port numbers less than 1024).
SYS_CHROOT	Use chroot(2), change root directory.
SETFCAP	Set file capabilities.

Судьба Docker

- | Capability Key | Capability Description |
|------------------------------|---|
| <code>SYS_MODULE</code> | Load and unload kernel modules. |
| <code>SYS_RAWIO</code> | Perform I/O port operations (<code>iopl(2)</code> and <code>ioperm(2)</code>). |
| <code>SYS_PACCT</code> | Use <code>acct(2)</code> , switch process accounting on or off. |
| <code>SYS_ADMIN</code> | Perform a range of system administration operations. |
| <code>SYS_NICE</code> | Raise process nice value (<code>nice(2)</code> , <code>setpriority(2)</code>) and change the nice value for arbitrary processes. |
| <code>SYS_RESOURCE</code> | Override resource Limits. |
| <code>SYS_TIME</code> | Set system clock (<code>settimeofday(2)</code> , <code>stime(2)</code> , <code>adjtimex(2)</code>); set real-time (hardware) clock. |
| <code>SYS_TTY_CONFIG</code> | Use <code>vhangup(2)</code> ; employ various privileged <code>ioctl(2)</code> operations on virtual terminals. |
| <code>AUDIT_CONTROL</code> | Enable and disable kernel auditing; change auditing filter rules; retrieve auditing status and filtering rules. |
| <code>MAC_ADMIN</code> | Allow MAC configuration or state changes. Implemented for the Smack LSM. |
| <code>MAC_OVERRIDE</code> | Override Mandatory Access Control (MAC). Implemented for the Smack Linux Security Module (LSM). |
| <code>NET_ADMIN</code> | Perform various network-related operations. |
| <code>SYSLOG</code> | Perform privileged <code>syslog(2)</code> operations. |
| <code>DAC_READ_SEARCH</code> | Bypass file read permission checks and directory read and execute permission checks. |
| <code>LINUX_IMMUTABLE</code> | Set the <code>FS_APPEND_FL</code> and <code>FS_IMMUTABLE_FL</code> i-node flags. |
| <code>NET_BROADCAST</code> | Make socket broadcast, and listen to multicasts. |
| <code>IPC_LOCK</code> | Lock memory (<code>mlock(2)</code> , <code>mlockall(2)</code> , <code>mmap(2)</code> , <code>shmct(2)</code>). |
| <code>IPC_OWNER</code> | Bypass permission checks for operations on System V IPC objects. |
| <code>SYS_PTRACE</code> | Trace arbitrary processes using <code>ptrace(2)</code> . |
| <code>SYS_BOOT</code> | Use <code>reboot(2)</code> and <code>kexec_load(2)</code> , reboot and load a new kernel for later execution. |
| <code>LEASE</code> | Establish leases on arbitrary files (see <code>fcntl(2)</code>). |
| <code>WAKE_ALARM</code> | Trigger something that will wake up the system. |
| <code>BLOCK_SUSPEND</code> | Employ features that can block system suspend. |

СЛЁРМ



slurm.io



Southbridge



southbridge.io