11-A 名前空間コンテナ

11-A-1 名前空間コンテナとは

名前空間コンテナ（name space container：以降「nsコンテナ」と記載）はchrootとよく似ていますが、以下の点で、更に機能が強化されています。

・完全に仮想化されたファイルシステム階層構造を持つ
・init（sysytemd）によるブートシーケンスから始まるプロセス階層構造を持つ
・IPCサブシステムを持つ
・ホスト名とドメイン名を持つ

名前空間コンテナはカーネルが提供するPID名前空間によるコンテナごとのPIDの隔離、ユーザ名前空間によるコンテナごとのUID、GIDの隔離等を行います。この故に、名前空間コンテナと呼ばれています。名前空間コンテナはPodman、Docker、Kubernetesが利用するコンテナのイメージ/ランタイムとは仕様が異ります。

コンテナはsystemd-nspawnコマンドにより起動します。したがって、コンテナを利用するためにはsystemd-nspawnコマンドを含むsystemd-containerパッケージが必要です。

11-A-2 nsコンテナのインストールと設定

nsコンテナのイメージは**/var/lib/machines/{コンテナ名}**のディレクトリの下にインストールします。

以下はコンテナ名をc8-nsとした場合のCentOS 8のインストールの例です。

nsコンテナのインストール(ホストがCentOS 8の場合)

```
[...]# dnf -y --releasever=8 --installroot=/var/lib/machines/c8-ns --disablerepo='*'
--enablerepo=BaseOS  install systemd passwd dnf centos-release vim-minimal iproute
iputils
…（途中省略）…
トランザクションの概要
================================================================================
=====================
インストール  192 パッケージ

ダウンロードサイズの合計: 108 M
インストール済みのサイズ: 700 M
パッケージのダウンロード:
(1/192): basesystem-11-5.el8.noarch.rpm
55 kB/s |  10 kB     00:00
(2/192): acl-2.2.53-1.el8.x86_64.rpm
245 kB/s |  81 kB     00:00
(3/192): audit-libs-3.0-0.17.20191104git1c2f876.el8.x86_64.rpm
                  290 kB/s | 116 kB     00:00
```

```
…（途中省略）…
完了しました!
```

以下はコンテナ名をc8-nsとした場合のStream 8のインストールの例です 。「--enablerepo」で指定するリポジトリは「baseos」とします。

nsコンテナのインストール(ホストがStream 8の場合)

```
[...]# dnf -y --releasever=8 --installroot=/var/lib/machines/c8-ns --disablerepo='*'
--enablerepo=baseos install systemd passwd dnf centos-release vim-minimal iproute iputils
…（実行結果省略）…
```

確認と設定

```
[...]# du -sh /var/lib/machines/c8-ns
552M    /var/lib/machines/c8-ns  ←❶
[...]# setenforce 0  ←❷
[...]# chroot /var/lib/machines/c8-ns
bash-4.4# passwd  ←コンテナのrootのパスワードを設定
ユーザー root のパスワードを変更。
新しいパスワード:
新しいパスワードを再入力してください:
passwd: すべての認証トークンが正しく更新できました。
bash-4.4# head -1 /etc/shadow
root:$6$VWinybh …（途中省略）… :18631:0:99999:7:::  ←パスワードが設定された
bash-4.4# exit
[...]# setenforce 1  ←selinuxをEnforcingに戻す

❶コンテナのルートファイルシステムのサイズは約552MB(インストールするバージョンによってサイズは多少異なります)
❷selinuxをPermissiveにして/var/lib/machines/以下の/etc/shadowへの操作を許可
```

Enforcingでコンテナの/etc/shadowへの操作を許可する手順は「15-6 SELinux」の「ポリシーの変更」の項を参照してください。

インストールしたnsコンテナは**systemd-nspawn**コマンドにより起動します。systemd-nspawnコマンドの主なオプションは以下の通りです。

表11-A-1　**主なオプション**

オプション	説明
-M、--machine=	コンテナのマシン名を設定
-D、--directory=	ルートファイルシステムとして使用するディレクトリを指定
-b、--boot	自動的にinitプログラムを探して、PID=1として起動する
--private-network	コンテナネットワークをホストネットワークから分離する
-n、--network-veth	ホストとコンテナ間で一対の仮想イーサネットリンク(veth)を作成する ホスト側の名前はve-{コンテナ名}、コンテナ側の名前はhost0となる このオプションを指定すると「--private-network」も自動的に指定される

nsコンテナを起動し、rootでログイン

```
[...]# systemd-nspawn -M c8-ns -bD /var/lib/machines/c8-ns
Spawning container c8-ns on /var/lib/machines/c8-ns.
Press ^] three times within 1s to kill container.
systemd 239 running in system mode. (+PAM +AUDIT +SELINUX +IMA -APPARMOR +SMACK +SYSVINIT
 +UTMP +LIBCRYPTSETUP +GCRYPT +GNUTLS +ACL +XZ +LZ4 +SECCOMP +BLKID +ELFUTILS +KMOD +IDN2
 -IDN +PCRE2 default-hierarchy=legacy)
Detected virtualization systemd-nspawn.
Detected architecture x86-64.

Welcome to CentOS Linux 8 (Core)!

Failed to install release agent, ignoring: No such file or directory
[  OK  ] Created slice system-getty.slice.
[  OK  ] Listening on initctl Compatibility Named Pipe.
…（途中省略）…
[  OK  ] Reached target Graphical Interface.
         Starting Update UTMP about System Runlevel Changes...
[  OK  ] Started Update UTMP about System Runlevel Changes.

c8-ns login: root  ←rootでログイン
Password:          ←パスワードを入力
Last login: Thu Nov 19 23:26:41 on console
[root@c8-ns ~]#  ←コンテナにログインすると、プロンプトに含まれるホスト名は
                   起動時に-Mオプションで指定した「c8-ns」となる
```

上記実行結果の最後では、nsコンテナが正常に起動し、nsコンテナにログインできています。

11-A-3 コンテナやVM(仮想マシン)の管理

machinectlコマンドにより、コンテナやVM(仮想マシン)の状態を確認できます。

machinectlコマンドはVMとコンテナのレジストレーション・マネージャであるsystemd-machinedを介して、コンテナおよびVMの状態表示、起動、停止ができます。

systemd-machinedについては、「4-6 systemctlコマンドによるサービスの管理」を参照してください。

machinectlコマンド

machinectl [オプション] コマンド [コンテナ/VM名]

表11-A-2　**主なコマンド**

コマンド	説明
show	指定したコンテナ/VMの情報を表示
login	指定したコンテナ/VMにログイン
start	指定したコンテナ/VMを起動
stop	指定したコンテナ/VMを停止

別ターミナルを開き、machinectlコマンドで確認します。

ホスト側でmachinectlコマンドにより確認

```
[...]# machinectl
MACHINE CLASS     SERVICE        OS VERSION ADDRESSES
c8-ns   container systemd-nspawn -  -       -           ←コンテナ(コンテナ名：c8-ns)のエントリ

1 machines listed.

[...]# machinectl -a  ←❶
MACHINE CLASS     SERVICE        OS     VERSION ADDRESSES
.host   host      -              centos 8       192.168.122.231…  ←ホストのエントリ
c8-ns   container systemd-nspawn -      -       -                 ←コンテナのエントリ

2 machines listed.  ←❷

[...]# machinectl show c8-ns  ←コンテナc8-nsの詳細情報を表示
Name=c8-ns
Id=192c87429df34177896bfec616399cac
Timestamp=Sat 2020-12-26 20:19:48 JST
TimestampMonotonic=982729560268
Service=systemd-nspawn
Unit=machine-c8.scope
Leader=378053
Class=container
RootDirectory=/var/lib/machines/c8-ns
State=running
```

❶「-a」オプションを指定すると、.(ドット)で始まる名前のエントリも表示される
❷ホスト1台とコンテナ1台により、2台のマシンのエントリが表示されている

machinectlコマンドを使用すると、ホスト側でコンテナの管理ができます。
以下はコンテナとVMの表示、コンテナへのログイン、コンテナの停止と起動の例です。

selinuxをPermissiveにして/var/lib/machines/以下のファイルへの操作を許可しています。

machinectlによるコンテナの管理

```
[...]# machinectl
MACHINE                  CLASS     SERVICE        OS VERSION ADDRESSES
c8-ns                    container systemd-nspawn -  -       -       ←❶
c8-ns2                   container systemd-nspawn -  -       -       ←❷
qemu-11-stream8-20200629 vm        libvirt-qemu   -  -       -       ←❸
qemu-9-c8.2-min          vm        libvirt-qemu   -  -       -       ←❹

4 machines listed.

[...]# setenforce 0  ←selinuxをPermissiveに設定
[...]# machinectl login c8-ns  ←コンテナ「c8-ns」にログイン
Connected to machine c8-ns. Press ^] three times within 1s to exit session.

CentOS Linux 8
Kernel 4.18.0-193.el8.x86_64 on an x86_64

c8-ns login: root
Password:
[root@c8-ns ~]#  ←コンテナにログイン中(あるいはログアウトの時)に
                   「^]]]」(Ctrl+]を3回)を入力すると、コンテナの
                   ターミナルから切り離されて、ホストに戻る
Connection to machine c8-ns terminated.
[...]# machinectl
MACHINE                  CLASS     SERVICE        OS VERSION ADDRESSES
c8-ns                    container systemd-nspawn -  -       -
… (以降省略) …

[...]# machinectl stop c8-ns  ←コンテナ「c8-ns」を停止
[...]# systemd-nspawn -M c8-ns -bD /var/lib/machines/c8-ns  ←コンテナ「c8-ns」を起動
[...]# setenforce 1  ←selinuxをEnforcingに戻す(別のターミナルで実行)
```

❶コンテナ「c8-ns」が稼動中
❷コンテナ「c8-ns2」が稼動中
❸KVMゲスト「stream8-20200629」がqemu上で稼働中
❹KVMゲスト「c8.2-min」がqemu上で稼働中

以下の例では、nsコンテナにログインしているターミナルに戻り、nsコンテナにアプリケーションをインストールします。
nsコンテナにhttpd(Apache Webサーバ)をインストールして起動します。

httpdをインストールして起動し、動作確認

```
[root@c8-ns ~]# dnf install httpd
[root@c8-ns ~]# apachectl start
[root@c8-ns ~]# ps -ef | grep httpd
root         130       1  0 17:30 ?        00:00:00 /usr/sbin/httpd -DFOREGROUND
apache       131     130  0 17:30 ?        00:00:00 /usr/sbin/httpd -DFOREGROUND
apache       132     130  0 17:30 ?        00:00:00 /usr/sbin/httpd -DFOREGROUND
apache       133     130  0 17:30 ?        00:00:00 /usr/sbin/httpd -DFOREGROUND
apache       134     130  0 17:30 ?        00:00:00 /usr/sbin/httpd -DFOREGROUND
[root@c8-ns ~]# cat > /var/www/html/index.html
Hello! This is c8-ns.
^D

[root@c8-ns ~]# curl http://localhost
Hello! This is c8-ns.
```

nsコンテナはデフォルトの設定ではホストとネットワークを共有します。以下は、nsコンテナ上で稼働しているhttpdの状態をホスト上で確認する例です。

nsコンテナ上で稼働しているhttpdの状態をホスト上で確認

```
[root@centos8 ~]# nmap localhost -p 80
Starting Nmap 7.70 ( https://nmap.org ) at 2020-12-28 12:40 JST
Nmap scan report for localhost (127.0.0.1)
Host is up (0.000044s latency).
Other addresses for localhost (not scanned): ::1

PORT   STATE SERVICE
80/tcp open  http  ←nsコンテナのhttpdにホストのポート80/tcpとしてアクセスできる

Nmap done: 1 IP address (1 host up) scanned in 0.25 seconds
[root@centos8 ~]# ps -ef | grep httpd
root       13881   13715  0 17:30 ?        00:00:00 /usr/sbin/httpd -DFOREGROUND
apache     13882   13881  0 17:30 ?        00:00:00 /usr/sbin/httpd -DFOREGROUND
apache     13883   13881  0 17:30 ?        00:00:00 /usr/sbin/httpd -DFOREGROUND
```

```
apache     13884   13881  0 17:30 ?        00:00:00 /usr/sbin/httpd -DFOREGROUND
apache     13885   13881  0 17:30 ?        00:00:00 /usr/sbin/httpd -DFOREGROUND
[root@centos8 ~]# curl http://localhost
Hello! This is c8-ns.
```

ホスト上で表示されるhttpdは、nsコンテナ「c8-ns」で稼働しているプロセスです。ただし、コンテナ上とホスト上ではプロセスIDは異なります。

11-A-4 nsコンテナのネットワークをホストのネットワークから分離して起動

systemd-nspawnに「-n」オプション（--network-veth）を指定してコンテナを起動することにより、コンテナのネットワークをホストのネットワークから分離することができます。

この場合、コンテナから外部ネットワークに出て行くためには**snat**（masquerade）を、外部ネットワークからコンテナのサービスにアクセスするためには**dnat**によるポートのマッピングが必要になります。

本項では、コンテナでの設定とホストでの設定の例を紹介します。

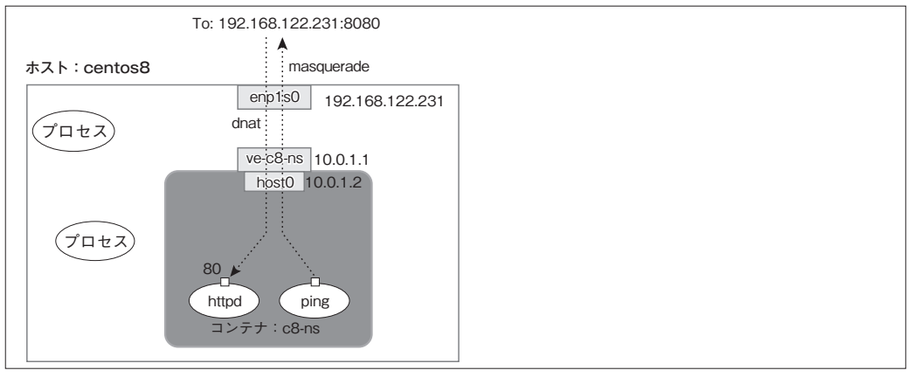
ホスト側（ホスト名：centos8）では以下の設定を行います。

- ネットワークI/FとIPアドレス：enp1s0、192.168.122.231/24
- forwarding（net.ipv4.ip_forward）を有効（1）に設定
- enp1s0にmasqueradeを設定
- dnatにより、192.168.122.231:8080 → 10.0.1.2:80のマッピングを設定
- veth I/FとIPアドレス：ve-c8-ns、10.0.1.1/24

コンテナ側（ホスト名：c8-ns）では以下の設定を行います。

- ネットワークI/FとIPアドレス：host0、10.0.1.2/24
- デフォルトゲートウェイを10.0.1.1に設定

図11-A-1 ホストネットワークとコンテナネットワークを分離した場合の設定例



以下の例では、テーブル（例：ip nat）とチェイン（例：POSTROUTING、PREROUTING）の作成手順を含みます。既にテーブルとチェインが出来ている場合はルールだけを作成します。ルールの作成時にはルールを適用したパケットの個数を確認するためにパラメータcounterを指定していますが、これは必須ではありません。

ホスト側の設定① パケットフォワードとnftablesの設定

```
[root@centos8 ~]# sysctl -w net.ipv4.ip_forward = 1
[root@centos8 ~]# nft create table ip nat
[root@centos8 ~]# nft create chain ip nat POSTROUTING {type nat hook postrouting priority
srcnat\; policy accept\;}
[root@centos8 ~]# nft create chain ip nat PREROUTING {type nat hook prerouting priority
dstnat\; policy accept\;}
[root@centos8 ~]# nft add rule ip nat POSTROUTING counter oifname "enp1s0" masquerade
[root@centos8 ~]# nft add rule ip nat PREROUTING counter ip daddr 192.168.122.231 tcp
dport 8080 dnat to 10.0.1.2:80
[root@centos8 ~]# nft -a list table ip nat
table ip nat { # handle 6
    chain POSTROUTING { # handle 1
        type nat hook postrouting priority srcnat; policy accept;
        counter packets 0 bytes 0 oifname "enp1s0" masquerade # handle 4
    }
    chain PREROUTING { # handle 5
        type nat hook prerouting priority dstnat; policy accept;
        counter packets 0 bytes 0 ip daddr 192.168.122.231 tcp dport 8080 dnat to
10.0.1.2:80 # handle 8
    }
}
```

systemd-nspawnに「-n」オプションを追加してコンテナを起動し、ホスト側で作成される仮想I/F（ve-c8-ns）をupしてIPアドレスを割当てます。

ホスト側の設定② ve-c8-nsの設定

```
[root@centos8 ~]# systemd-nspawn -M c8-ns -bD /var/lib/machines/c8-ns -n
↑コンテナの起動（-nオプションを指定）
…（途中省略）…
```

この後、別ターミナルを開いて、以下の手順を実行します。

ホスト側の設定③ ve-8-c8-nsの設定

```
[root@centos8 ~]# ip addr show dev ve-c8-ns
10: ve-c8-ns@if2: <BROADCAST,MULTICAST> …（以降省略）…
[root@centos8 ~]# ip addr add 10.0.1.1/24 dev ve-c8-ns
[root@centos8 ~]# ip link set dev ve-c8-ns up
[root@centos8 ~]# ip addr show dev ve-c8-ns
5: ve-c8-ns2@if2: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group
default qlen 1000
    link/ether 8a:e8:3d:48:f6:99 brd ff:ff:ff:ff:ff:ff link-netnsid 0
    inet 10.0.1.1/24 scope global ve-c8-ns2
…（以降省略）…
```

以下の実行例では、ホスト側の設定②の実行例のターミナルを使用します。systemd-nspawnに「-n」オプションを追加してコンテナを起動した時、コンテナ側で作成される仮想I/F（host0）をupし、IPアドレスを割当てます。

コンテナ側の設定

```
c8-ns login: root  ←起動したコンテナにrootでログイン
Last login: Sun Jan  3 17:28:05 on console
[root@c8-ns ~]# ip addr show dev host0
2: host0@if4: <BROADCAST,MULTICAST>  …（以降省略）…
[root@c8-ns ~]# ip addr add 10.0.1.2/24 dev host0
[root@c8-ns ~]# ip link set dev host0 up
[root@c8-ns ~]# ip route add default via 10.0.1.1
[root@c8-ns ~]# ping www.google.co.jp  ←pingコマンドで疎通確認
PING www.google.co.jp (172.217.175.227) 56(84) bytes of data.
64 bytes from nrt12s29-in-f3.1e100.net (172.217.175.227): icmp_seq=1 ttl=112 time=122 ms
64 bytes from nrt12s29-in-f3.1e100.net (172.217.175.227): icmp_seq=2 ttl=112 time=122 ms
^C
[root@c8-ns ~]# apachectl start  ←httpdを起動
[root@c8-ns ~]# ps -ef | grep httpd
root         111       1  0 00:24 ?        00:00:00 /usr/sbin/httpd -DFOREGROUND
apache       112     111  0 00:24 ?        00:00:00 /usr/sbin/httpd -DFOREGROUND
apache       113     111  0 00:24 ?        00:00:00 /usr/sbin/httpd -DFOREGROUND
apache       114     111  0 00:24 ?        00:00:00 /usr/sbin/httpd -DFOREGROUND
apache       115     111  0 00:24 ?        00:00:00 /usr/sbin/httpd -DFOREGROUND
```

ホストおよび、コンテナにそれぞれネットワークの設定が完了したので外部ネットワークからホスト内コンテナへアクセスします。

ホスト上でのコマンド実行によるアクセスの場合はローカルプロセスによるアクセスとなり、prerouting hookが適用されません。したがってdnatを使用することができないので注意してください。

新しいターミナルを開き、以下を実行します。

外部ネットワークからホスト内コンテナへのアクセス

```
[root@centos8~]# ip addr show dev wlp0s20f3 | grep "inet "
inet 192.168.43.62/24 brd 192.168.43.255 scope global dynamic noprefixroute wlp0s20f3
     ↑外部ホストのIPアドレスは192.168.43.62
[root@centos8~]# curl http://192.168.122.231:8080  ←コンテナ内のhttpdにアクセス
Hello! This is c8-ns.
```