CentOS 7 を用いたサーバー管理

○野崎公隆

工学系技術支援室　情報通信技術系

概要

サーバー用 OS として用いられている CentOS は、2014 年に新しいバージョンがリリースされた。バージョンが 6 から 7 へと上がり、システム管理ツールが一新された。それにともない、従来の init やランレベルといった仕組みの廃止、NetworkManager による NIC の設定、各種デーモンやサービスの起動・停止、ファイアウォールの設定手順など、管理方法も変化している。また、コンテナ管理ツールの Docker が利用できるようになった。CentOS は全く新しく生まれ変わった 7 を採用する時期に差し掛かっている。

そこで、CentOS7 のポイントとなる基本的な操作方法や手順を研修を通して習得し、今後の通常業務でのサーバーの構築、運用、保守に役立てることを研修の目的とした。

1. 研修で使用した機器

サーバー3 台を使用した。

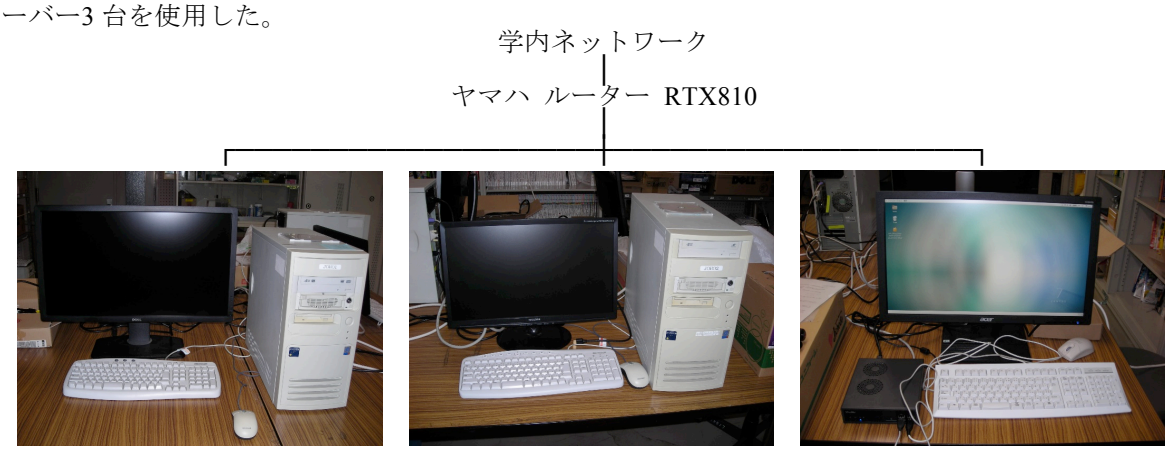

図 1. 研修用システムの接続構成

2. ランレベルの廃止とターゲットの導入

CentOS6 のランレベルと CentOS7 のターゲットの対応は表 1 の通りで、CentOS7 では systemctl コマンドで切り替えるようになった。

multi-user.target ターゲットを設定する場合は以下のコマンドを実行してから再起動する。

```
# systemctl　set-default　multi-user.target
```

すぐに反映させる場合は以下のコマンドを使用する。

```
# systemctl　isolate　multi-user.target
```

表 1. ランレベルとターゲットの対応

	CentOS6 のランレベル	CentOS7 のターゲット
マルチユーザモード	3	multi-user.target

グラフィカルターゲット	5	graphical.target

3. サービス（デーモン）の起動・終了

httpd の起動・終了などは、CentOS6 と CentOS7 で表 2 のように変わり、CentOS7 では systemctl を使う。

表 2. サービス管理コマンドの例

	CentOS 6	CentOS 7
サービス一覧	chkconfig --list	systemctl --type service
サービス自動起動 ON	chkconfig httpd on	systemctl enable httpd
サービス自動起動 OFF	chkconfig httpd off	systemctl disable httpd
サービス起動	service httpd start	systemctl start httpd
サービス停止	service httpd stop	systemctl stop httpd
サービス強制終了	kill -9 <httpd の PID>	systemctl kill --signal=9 httpd
サービス再起動	service httpd restart	systemctl restart httpd
サービス状態確認	service httpd status	systemctl status httpd
サービス設定リロード	service httpd reload	systemctl reload httpd

4. ログの管理

CentOS6 では、rsyslogd デーモンでログを収集していたが、CentOS7 では journald デーモンを利用し表 3 のように使う。

表 3. ログ管理コマンドの例

	CentOS 6	CentOS 7
ログを表示	less /var/log/messages	journalctl
起動時のログを表示	dmesg	journalctl –k

また、CentOS 7 ではシステムを再起動すると journal ログが削除されてしまうので、これを防ぐために journal ログを永続化する必要があり、次のように設定ファイル修正する。

vi /etc/systemd/journald.conf

[Journal]

#Storage=auto → Storage=persistent

5. ネットワーク管理ツール

CentOS 6 までは /etc/sysconfig/network-scripts 配下の ifcfg-eth*ファイルを直接編集することで設定を変更していたが、CentOS 7 では nmcli コマンドを使用する。

最初にインターフェースのデバイス名の一覧を確認する。

nmcli device show → eno1, enp1s0 などのデバイスの詳細が表示

ここで IP アドレスを 192.168.100.11 に変更するには

nmcli connection modify eno1 ipv4.addresses ”192.168.100.11/24”

nmcli connection down eno1 && nmcli connection up eno1

よく利用されるコマンドを表 4 に抜粋する。

表 4. ネットワーク管理コマンドの例

	CentOS 6	CentOS 7
IP アドレス、MAC アドレス	ifconfig -a	ip a

ルーティングテーブル	route -n	ip route
ARP テーブル	arp -a	ip neigh

6. Docker を用いた仮想サーバーの構築

Docker とは仮想化のためのソフトウェアである。Docker を使うことで一つのホスト OS 上で複数のコンテナを簡単に動作させることができる。コンテナは OS とアプリケーションをひとまとめにした実行環境で、コンテナ型の仮想化は KVM などのハイパーバイザー型に比べて軽いのが特徴である。研修ではコンテナを使用してサーバーを構築した。サーバー構築の手順は、1. yum で docker をインストール、2. Docker のイメージファイルをリポジトリからダウンロード、3. docker run のコマンドでコンテナを作成、4. コンテナ上にアプリケーションをインストール、となる。

図 2 は、コンテナに Apache をインストールし Web サーバーを構築した後で、ホスト OS と別のサーバーからブラウザを通して、コンテナが提供しているコンテンツを閲覧したときのスクリーンショットである。コンテナが Web サービスを提供しているのがわかる。

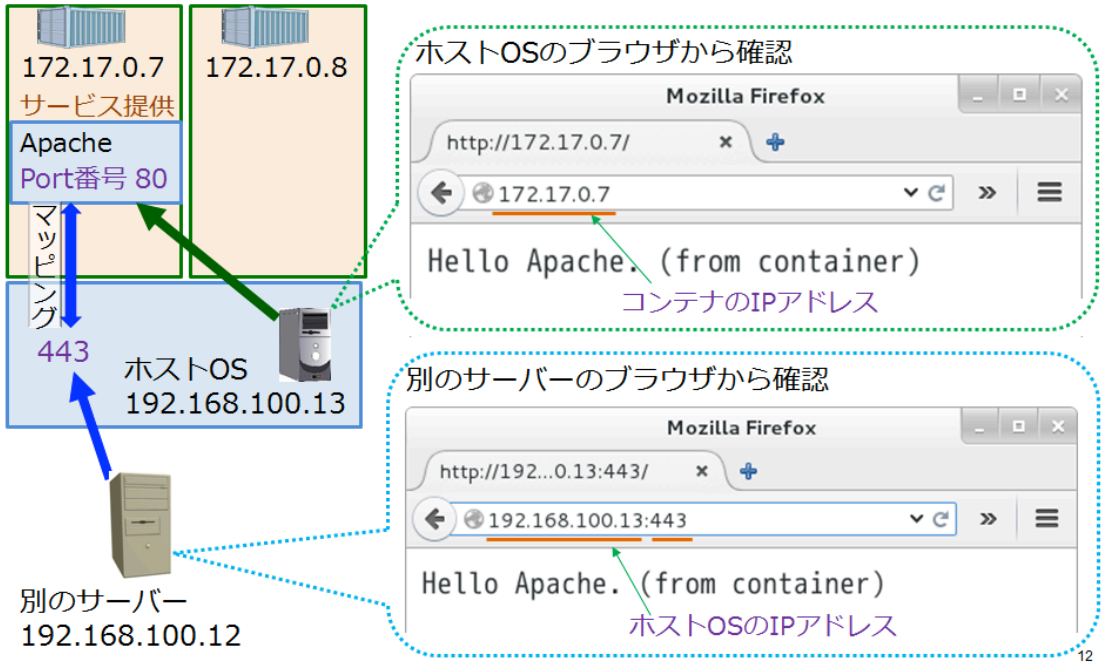

図 2. コンテナによる Web サービスの提供

7. cgroup によるハードウェア管理

cgroup とは Control Group の略で、プロセスに対して CPU、メモリなどの資源を動的に割り当てることが可能となる仕組みである。プロバイダや通信事業者が実施しているように、ユーザーの通信速度を動的に制限することができる。

研修では、通信速度に制限をかけた状態で 30MB のファイルを別サーバーへ FTP で転送し、それに掛かる時間を計測した。制限速度を一度目の 10 倍に設定すると、転送時間は 1/10 に短縮された。

・通信速度を 1MB/秒に設定した場合　→　30 秒掛かっている

```
[root@ken02 ~]# tc class change dev eth0 parent 1: classid 1:2 htb rate 1mbps
[root@ken02 ~]# cgexec --sticky -g net_cls:test01 ./scp.sh
root@192.168.100.13's password:
testfile                                      100%   30MB   1.0MB/s   00:30
```

図 3. ファイル転送の結果（1MB/秒）

・通信速度を 10MB/秒に設定した場合　→　3 秒に短縮された

```
[root@ken02 ~]# tc class change dev eth0 parent 1: classid 1:2 htb rate 10mbps
[root@ken02 ~]# cgexec --sticky -g net_cls:test01 ./scp.sh
root@192.168.100.13's password:
testfile                                      100%   30MB  10.0MB/s   00:03
```

図 4. ファイル転送の結果（10MB/秒）

8. firewalld のセキュリティ機能

CentOS6 での iptables に代わって、CentOS7 では firewalld が搭載された。

図 5 のような ssh のアクセス制限をする場合、下記のように①で 192.168.100.12 からの ssh を許可して、②で 13 からの ssh を許可し、③でその他の IP アドレスからのアクセスを拒否する。

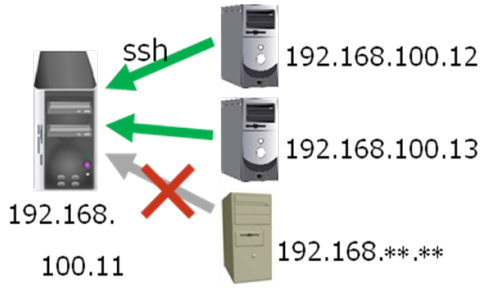

図 5. ssh のアクセス制限

```
# firewall-cmd　--permanent --direct --add-rule ipv4 filter INPUT　1　¥
 -m conntrack --ctstate NEW -m tcp -p tcp --dport 22 -s 192.168.100.12　　-j ACCEPT　　←①

# firewall-cmd　--permanent --direct --add-rule ipv4 filter INPUT　2　¥
 -m conntrack --ctstate NEW -m tcp -p tcp --dport 22 -s 192.168.100.13　　-j ACCEPT　　←②

# firewall-cmd　--permanent --direct --add-rule ipv4 filter INPUT　3 -m conntrack ¥
 --ctstate NEW -m tcp -p tcp --dport 22 -j DROP　　←③

# firewall-cmd　--reload　　←設定を反映させる
```

9. まとめ

・CentOS 6 のサポートの完全更新期限は 2017 年、メンテナンス更新期限は 2020 年なので CentOS 7 への移行は必須となる。

・CentOS 7 では、サービスの ON・OFF の切り替え、ファイアウォールの設定方法、ネットワークの管理方法などの基本的な使い方が変更されたが、慣れれば自然と身に付く。

・Docker は、軽くて簡単に複数の仮想環境を得られるのが便利。ただ、どのような目的で使うのかを考える必要があり、長期的に安定した運用ができるかは不明である。

・cgroup について、自宅のプロバイダがヘビーユーザーに対して帯域制限をかける仕組みの基礎がわかった。

参考文献

[1]　インプレス社　CentOS7 実践ガイド　著者 古賀政純　2015 年 3 月発行