

# Linux研修【ハンズオン】 ネットワークの基礎を学ぼう！ / ネットワークインターフェイスとDNSの設定

LinuC取得を目指す人を応援する情報サイト「リナスク」主宰

LinuCエバンジェリスト

水澤泰敬

## ■水澤泰敬（みずさわやすたか）

文系大学を卒業後、日経新聞関連企業、商社などへの転職を重ねたのち、フリーランスを経て有限会社を設立。10数年にわたり数々の情報まとめサイトの運営に携わる。サイト運営時に取材したキュレーションサイトへの転職、事業譲渡によるTSUTAYA関連企業への転籍を経て再度フリーランスとなる。2020年、日本のエンジニア育成に何らかのカタチで携わりたいと思い、エンジニアスクールで企業研修講師を務めながら約半年の準備期間を経てLinux技術者認定資格取得を目指す人を応援する情報サイト「リナスク」を立ち上げ。趣味はギターとテレビゲーム。

### セミナー歴

- 2021/1/30 OSC 2021 Online/Osaka 「**LinuCの学習にも使える、ブラウザだけで使えるクラウド環境紹介ハンズオン**」（LPI-Japanスポンサー枠）
- 2021/3/26 LPI-Japan技術解説セミナー「**Linux研修【ハンズオン】コンテナを体験してみよう！**」
- 2021/7/31 OSC 2021 Online/Kyoto 「**LinuCの学習にも使える、ブラウザで動かすクラウドLinuxハンズオン**」（LPI-Japanスポンサー枠）



Linux技術者認定資格取得を目指す人を応援する情報サイト

<https://linuc.spa-miz.com/>



## 今回のテーマ

### ネットワークの基礎を学ぼう！ / ネットワークインターフェイスとDNSの設定

- ハンズオンを通じて、ネットワーク関連コマンドを実行し、ネットワークの基礎に関する理解を深める
- LinuCレベル1の出題範囲となっている「ネットワーク基礎」の学習に役立てる

## ■主題・内容

- 主題1.07：ネットワークの基礎
- 副題1.07.2 基本的なネットワーク構成
- 副題1.07.3 基本的なネットワークの問題解決
- 副題1.07.4 クライアント側のDNS設定

## ■ポイント

- ネットワークインターフェイスの設定とその確認
- ホストの基本的なTCP/IP設定、デフォルトルートの設定の確認
- ネットワークに関する問題の原因調査方法
- リモートDNSサーバーへの問い合わせ方法
- ホスト間のSSH接続

## ■参加者の想定スキルレベル

- LinuCレベル1の取得を目指している方

## ■ゴール

- ホスト上の設定を参照、変更、確認できる
- ネットワークに関する問題解決ができる
- コマンドを使用してネットワークの調査ができる
- DNSクライアントの設定ができる

ネットワークはLinuCレベル2でも大きく取りあげられており、サーバ構築においてもネットワークに関する理解は極めて重要です

## ■LinuCとは

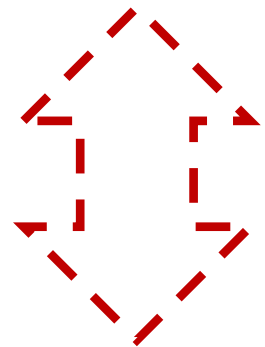
クラウド時代の即戦力エンジニアであることを証明するLinux技術者認定

- ✓現場で「今」求められている新しい技術要素に対応
  - オンプレミス／仮想化・コンテナを問わず様々な環境下でのサーバー構築
  - 他社とのコラボレーションの前提となるオープンソースへの理解
  - システムの多様化に対応できるアーキテクチャへの知見
- ✓全面的に見直した「今」身につけておくべき技術範囲を網羅
 

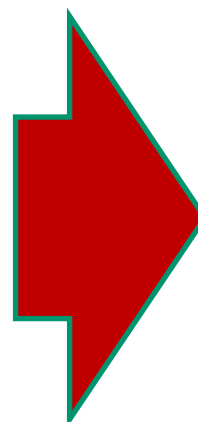
今となっては使わない技術やコマンドの削除、アップデート、新領域の取り込み
- ✓Linuxの範疇だけにとどまらない領域までカバー
 

セキュリティや監視など、ITエンジニアであれば必須の領域もカバー

AWSなどの  
パブリッククラウドを  
活用するための技術



間が  
欠けて  
いる状態



AWSなどの  
パブリッククラウドを  
活用するための技術

仮想マシン/コンテナ技術、  
クラウドセキュリティ、  
アーキテクチャ、ほか

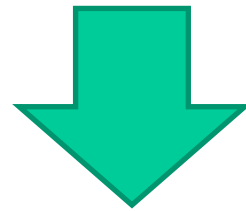
オンプレミスの  
サーバーサイドLinux技術

オンプレミスの  
サーバーサイドLinux技術

【今まで/その他】



「LinuCレベル1 / レベル2」の認定を取得できるIT技術者であれば、どのような分野のIT技術者にもつながる技術を持っていることを証明できる



LinuCを学習することで、IT技術者として身に付けておきたい重要かつ基礎的なスキルを獲得できる

## ■ 実機

- Windows PCなどにLinux OSを直接導入
- 外付HDD/SDDに導入して外付から起動すればWindowsを消さなくて済む

## ■ 仮想環境

- パソコンのホストOSはそのままにLinux仮想環境を導入
- Virtual BoxやVagrantなどを利用

## ■ クラウド型環境

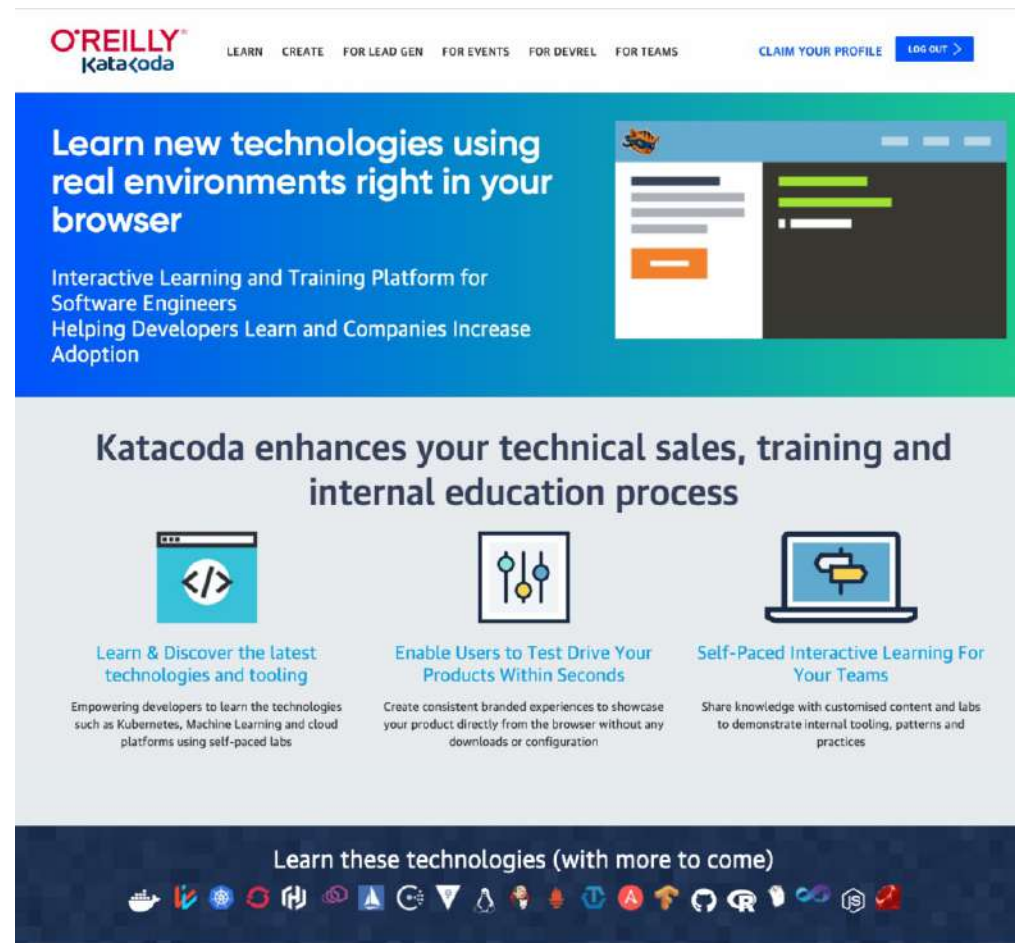
- AWSやGCPの仮想マシンでLinux環境を構築（無料枠もあるが基本的には有料）
- Webブラウザで使えるクラウド型学習環境サービス（無料） → 今回のハンズオンで使用

LinuCを取得するなら、実機、仮想、クラウド、それぞれを可能な限り試してみよう





- コンピュータ関連書籍を出版している米オライリー社が運営するWebブラウザベースのオンライン学習サービス
- テーマ毎に様々なシナリオが用意されており、トータルで300を超えるシナリオを無料提供
- サービス自体が仮想化技術で構築されている



<https://www.katacoda.com/>

## メリット

- CentOSとUbuntu両方ともに自由度の高いインスタンスが利用可能
- LinuxインスタンスにあらかじめDockerが導入済み
- コンテナ技術の学習コンテンツが豊富に用意されている
- 無料（2021/7現在）



**LinuC バージョン10.0 の学習に最適**

## デメリット

- 時間制限があり、インスタンスを継続して利用できない
- 極めて素のLinuxの動作に近いがなんらかの制限はある
  - ターミナルでexitを入力するとインスタンスが終了してしまう
  - Dockerでデタッチ（Ctrl+PのあとにCtrl+Q）するとインスタンスが終了してしまうなど

<p>34 SCENARIOS</p> <h3>Docker, Container Runtimes, Builders and Registries</h3> <p>Learn how to use Docker, Podman, Firecracker and other container runtimes.</p> <p>Start Course</p>	<p>17 SCENARIOS</p> <h3>Kubernetes Introduction</h3> <p>Get started using Kubernetes</p> <p>Start Course</p>	<p>8 SCENARIOS</p> <h3>Extending Kubernetes</h3> <p>Learn how to extend Kubernetes</p> <p>Start Course</p>	<p>39 SCENARIOS</p> <h3>Machine Learning</h3> <p>Learn Machine Learning with Tensorflow, Kubernetes and related topics</p> <p>Start Course</p>
<p>7 SCENARIOS</p> <h3>Linux &amp; Operating Systems</h3> <p>Learn the foundations of Linux and Bash and how they apply to Ubuntu, CentOS, RHEL and CoreOS.</p> <p>Start Course</p>	<p>30 SCENARIOS</p> <h3>Continuous Integration and Continuous Delivery</h3> <p>Learn Continuous Integration and Delivery using Jenkins and related topics</p> <p>Start Course</p>	<p>50 SCENARIOS</p> <h3>Orchestration &amp; Application Definition</h3> <p>Learn how to deploy applications using orchestrations like Kubernetes and Nomad and definitions such as Helm and Operators</p> <p>Start Course</p>	<p>40 SCENARIOS</p> <h3>Observability &amp; Analysis</h3> <p>Learn how to monitor and observe with Prometheus, OpenTracing and related tooling</p> <p>Start Course</p>
<p>27 SCENARIOS</p> <h3>Service Proxy, Discovery &amp; Mesh</h3> <p>Learn how to use Service Meshes including Istio, Consul, Envoy Proxy and Linkerd</p> <p>Start Course</p>	<p>3 SCENARIOS</p> <h3>Networking</h3> <p>This set of labs explains how to deploy Weave Network and Weave Scope</p> <p>Start Course</p>	<p>20 SCENARIOS</p> <h3>Cloud Native Storage</h3> <p>Learn how to persistent data and how to use products from StorageOS, Portworx and more.</p> <p>Start Course</p>	<p>11 SCENARIOS</p> <h3>Cloud Native Security &amp; Compliance</h3> <p>Learn how to secure Cloud Native deployments such as Docker and Kubernetes Security, Secrets Management and Authentication</p> <p>Start Course</p>

<p>4 SCENARIOS</p> <h3>Learn Ubuntu</h3> <p>Start Course</p>	<p>1 SCENARIOS</p> <h3>Learn CentOS</h3> <p>Start Course</p>	<p>3 SCENARIOS</p> <h3>Learn CoreOS</h3> <p>Start Course</p>
--	--	--

<h3>Deploy Microk8s on Ubuntu</h3> <p>Learn how to deploy Kubernetes workloads locally using Microk8s</p> <p>Start Scenario</p>	<h3>Ubuntu 16.04 Playground</h3> <p>Use Ubuntu in a sandboxed playground environment</p> <p>Start Scenario</p>	<h3>Ubuntu 18.04 Playground</h3> <p>Use Ubuntu in a sandboxed playground environment</p> <p>Start Scenario</p>	<h3>Ubuntu 19.04 Playground</h3> <p>Use Ubuntu in a sandboxed playground environment</p> <p>Start Scenario</p>
<h3>Ubuntu 20.04 Playground</h3> <p>Use Ubuntu in a sandboxed playground environment</p> <p>Start Scenario</p>	<h3>Playground</h3> <p>Experiment with a clean Ubuntu environment</p> <p>Explore Playground</p>	<h3>Your Content Here</h3> <p>Add your own content to Katacoda and share your experience or product with the community.</p> <p>Create Content</p>	

- Ubuntu : 16.04, 18.04, 19.04, 20.04
- CentOS : 7
- Fedora : 30

学習用インスタンスは画面左の解説を読みながら、画面右のコンソールで操作します。

O'REILLY Katacoda KATACODA OVERVIEW & SOLUTIONS CLAIM YOUR PROFILE LOG OUT >

## What is a container?

Step 1 of 6 >

### Processes

Containers are just normal Linux Processes with additional configuration applied. Launch the following Redis container so we can see what is happening under the covers.

```
docker run -d --name=db redis:alpine
```

The Docker container launches a process called `redis-server`. From the host, we can view all the processes running, including those started by Docker.

```
ps aux | grep redis-server
```

Docker can help us identify information about the process including the PID (Process ID) and PPID (Parent Process ID) via `docker top db`.

Who is the PPID? Use `ps aux | grep <ppid>` to find the parent process. Likely to be `Containerd`.

The command `pstree` will list all of the sub processes. See the Docker process tree using `pstree -c -p -A $(pgrep dockerd)`.

As you can see, from the viewpoint of Linux, these are standard processes and have the same properties as other processes on our system.

### Process Directory

Linux is just a series of magic files and contents, this makes it fun to explore and navigate to see

Terminal +

Your Interactive Bash Terminal. A safe place to learn and execute commands.

```
$
$
```

プレイグラウンドは解説がありません。画面右のコンソールで自分の好きなように操作します。

O'REILLY Katacoda KATACODA OVERVIEW & SOLUTIONS CLAIM YOUR PROFILE LOG OUT >

## CentOS Playground

### Hello World

Interested in writing your own scenarios and demos? Visit [www.katacoda.com/teach](http://www.katacoda.com/teach)

CONTINUE

Terminal +

Your Interactive Bash Terminal. A safe place to learn and execute commands.

```
[root@532654cb42d9 ~]#
```

## ■主題1.07：ネットワークの基礎（重要度合計：14）

- 60問中14問出題される（約23%、4分の1）

## ■副題1.07.1：インターネットプロトコルの基礎（重要度：4）

- TCP/IPネットワークの基礎を適切に理解していることを示すことができる。
  - ネットワークマスクとCIDR表記法
    - サブネット化
  - プライベートとパブリックのドット区切り形式のIPアドレスの違い
  - 一般的なTCPおよびUDPのポート
    - 22/SSH, 25/SMTP, 53/DNS, 80/HTTP, 123/NTP, 443/HTTPS
  - UDP、TCP、およびICMPの違いや主な機能
  - IPv4とIPv6の主な違い
  - IPv6の基本的な機能

## ■副題1.07.2 基本的なネットワーク構成（重要度：4）

- ホスト上の設定を参照、変更、確認できる
  - ネットワークインターフェイスの設定を手作業および自動で行う
    - **/etc/hostname**
    - **ifconfig, ifup, ifdown, nmcli**
    - **ip addr**
  - ホストの基本的なTCP/IP設定
    - **/etc/hosts, /etc/nsswitch.conf, ping**
  - デフォルトルートの設定
    - **ip route, route**

## ■副題1.07.3 基本的なネットワークの問題解決（重要度：4）

- ホスト上のネットワークに関する問題解決ができる
  - ネットワークに関する問題の原因を調査する
  - **host, dig, ping, ping6, netcat(nc, ncat), netstat, ss**
  - **traceroute, traceroute6, tracepath, tracepath6**
  - 必要に応じてネットワークインターフェースの追加、起動、停止、再起動、削除、および再設定する
  - **ifconfig, ifup, ifdown, hostname**
  - ルーティングテーブルを変更、参照、設定し、不適切なデフォルトルート設定を手作業で訂正する
  - **ip route, route**

## ■副題1.07.4 クライアント側のDNS設定（重要度：2）

- DNSクライアントの設定ができる。
  - リモートDNSサーバーに問い合わせる
  - **host, dig**
  - ローカルの名前解決の設定によりリモートDNSサーバーを使用する
  - **/etc/resolv.conf**
  - 名前解決の実行順序を変更する
  - **getent, /etc/nsswitch.conf, /etc/hosts**



## ■TCP/IP

- インターネットにおける標準プロトコル（4つの層でできている）
- TCP : 伝送内容を管理  
IP : データを小分け（パケット）にして送信
- UDP : スピードを優先させたいときに使用（音声通話、ストリーミング向け）

OSI	TCP/IP	主なプロトコル
5-7	アプリケーション層	HTTP,SMTP,DNS,SSHなど
4	トランスポート層	TCP,UDPなど
3	インターネット層	IP,ICMPなど
1-2	ネットワークインターフェース層(リンク層)	ARP,PPPなど

# 1.07.1 : インターネットプロトコルの基礎

- **IPアドレス** : ネットワークに接続された機器を識別するために割り当てられる番号

11000000.10101000.00000000.00000001 (2進数表記)

192.168.0.1 (10進数表記)

- **ネットマスク** : ネットワークがどこまでなのかを (2進数表記の1を使って) 表す

11000000.10101000.00000000.00000001 = 192.168.0.1

11111111.11111111.11111111.00000000 = 255.255.255.0

- **サブネット (マスク)** : ネットマスクによってネットワークを分割する

11000000.10101000.00000000.00000001 = 192.168.0.1

11111111.11111111.11111111.11000000 = 255.255.255.192

- **CIDR** : ネットマスクでネットワークを分割する方式の表記手法

IPアドレスの後ろに/を表記し、続けてサブネットマスクのビット数を表記

192.168.0.1/24を4つのサブネットに分割(256÷4=64)

192.168.0.1/26, 192.168.0.64/26, 192.168.0.128/26, 192.168.0.192/26

## ■ ウェルノウポート → /etc/servicesファイルで確認可能

ポート番号	プロトコル	トランスポートプロトコル
20	FTP(データ用)	TCP
21	FTP(制御用)	TCP
<b>22</b>	<b>SSH</b>	TCP
23	Telnet	TCP
<b>25</b>	<b>SMTP</b>	TCP
<b>53</b>	<b>DNS</b>	<b>UDP, TCP(ゾーン転送)</b>
<b>80</b>	<b>HTTP</b>	TCP
110	POP3	TCP
<b>123</b>	<b>NTP</b>	TCP
143	IMAP	TCP
<b>443</b>	<b>HTTPS</b>	TCP
465	SMTPS	TCP
993	IMAPS	TCP
995	POPS	TCP

### ■ホスト名の確認

- **/etc/hostname**ファイル
  - ホスト名が格納されているファイル
  - \$ cat /etc/hostname
- **hostname**コマンド
  - ホスト名を設定・変更する
  - \$ hostname (現在のホスト名を表示)
  - # hostname xxx.yyy.zzz (引数をホスト名として設定。再起動で元に戻る)
- **/etc/hosts**ファイル
  - ホスト名とIPアドレスの対応
  - \$ cat /etc/hosts
- **getent**コマンド
  - ネームサービスのデータベースを表示
  - \$ getent hosts (=cat /etc/hosts)
  - \$ getent hosts [ホスト名/IPアドレス]
  - \$ getent passwd
- **/etc/nsswitch.conf**ファイル
  - 名前解決の順番を指定
  - \$ cat /etc/nsswitch.conf
- **hostnamectl**コマンド(試験範囲外)
  - ホスト名を管理
  - 引数無しでホスト名と関連情報を表示

## ■ ネットワークの設定

NetworkManagerサービスで提供されるnmcli/nmtuiコマンドを使用

### • nmcliコマンド

- # nmcli general {status|hostname|hostname ホスト名}
- # nmcli networking {on|off|connectivity [check]}
- # nmcli radio {wifi [on|of]|wwan [on|off]|all [on|off]}
- # nmcli connection {show [-active]|modify インターフェース パラメータ|{up|down} ID}
- # nmcli device {status|show インターフェース|modify インターフェース パラメータ  
|{connect|disconnect} インターフェース|delete インターフェース|monitor インターフェース  
|wifi list|wifi connect SSID など}

### • nmtuiコマンド

- # nmtui

※katacodaではUbuntu 18.04 以上のPlaygroundで実行可能。CentOS Playgroundではインストールが必要。

## ■ ネットワークの状況確認

### • ip コマンド

ネットワークインターフェースやルーティングテーブルを管理

- ip 操作対象 [サブコマンド]

IPアドレスの確認

- \$ ip addr show
- \$ ip addr
- \$ ip a

ルーティングテーブルの確認

- \$ ip route show
- \$ ip route
- \$ ip r

データリンク層の確認

- \$ ip link show
- \$ ip link
- \$ ip l

IPアドレスの設定

- # ip addr add 192.168.0.xxx/24 dev enp0s3

デフォルトゲートウェイの設定

- # ip route add default via 192.168.0.xxx

## ■ ネットワークの状況確認

### • ifconfig コマンド

ネットワークインターフェースの設定や管理

- ifconfig [ネットワークインターフェース] [パラメータ]

引数無しでネットワーク情報の表示

- \$ ifconfig

IPアドレスとサブネットマスクの設定

- ifconfig eth0 192.168.0.xxx netmask 255.255.255.0

### • ifup/ifdown コマンド

- # ifup [ネットワークインターフェース]
- # ifdown [ネットワークインターフェース]

### • route コマンド

ルーティングテーブルの確認

- \$ route
- \$ netstat -r と同じ

ルーティングテーブルの編集

- # route add
- # route del -net 192.168.xxx.xxx netmask 255.255.255.0 dev xxx

### • ping コマンド (ICMP)

- ネットワークの疎通を確認
- ping [オプション] ホスト名または IP アドレス
- \$ ping -c 5 [www.lpi.org](http://www.lpi.org)

## ■ ネットワークの状況を表示

### • netstatコマンド

- TCP/UDPポート、ソケットの状態を表示
- netstat [オプション]

オプション	内容
-a	全てのソケット情報
-i	ネットワークインターフェースの状態
-l	待ち受けているポート
-n	アドレスやポートを数値で表示
-p	PIDとプロセス名
-r	ルーティングテーブルを表示
-t	tcpポートを表示
-u	udpポートを表示
-4	ipv4を表示
-6	ipv6を表示

### • ssコマンド

- TCP/UDPポート、ソケットの状態を表示
- ss [オプション]

オプション	内容
-a	全てのソケット情報
-l	待ち受けているポート
-n	アドレスやポートを数値で表示
-p	PIDとプロセス名
-t	tcpポートを表示
-u	udpポートを表示
-4	ipv4を表示
-6	ipv6を表示



## ■ net-toolsパッケージとiprouteパッケージの比較

net-tools	iproute2
ifconfig -a	ip addr show
ifconfig インターフェイス {down up}	ip link set インターフェイス {down up}
ifconfig インターフェイス IPアドレス netmask ネットマスク値	ip addr add IPアドレス(/CIDR値) dev インターフェイス
ifconfig インターフェイス mtu mtu値	ip link set インターフェイス mtu mtu値
netstat -r	ip route show
route	ip route show
route {add del} default gw ゲートウェイアドレス	ip route {add del} default via ゲートウェイアドレス
route {add del} {-net -host} 宛先 netmask gw ゲートウェイアドレス インターフェイス	ip route {add del} 宛先 via ゲートウェイアドレス
arp -a	ip neigh
netstat [-a,-n,-t,-u,-n]	ss [-a,-n,-t,-u,-n]

※昨今はnet-toolsよりもiprouteパッケージの利用が推奨されているが、試験ではどちらも出題され同じ設定内容を選択する問題などが多い

## ■ ネットワークの状況を表示

- **traceroute** コマンド (ICMP)

- ネットワークの経路を表示
- traceroute ホスト名またはIPアドレス

- **tracpath** コマンド

- ネットワークの経路を表示
- tracpath ホスト名またはIPアドレス[/ポート番号]

- **hostname** コマンド

- ホスト名の設定・変更
- hostname ホスト名

- **nmap** コマンド

- オープンしているポートの状況を表示
- nmap ホスト名

- **nc(netcat)** コマンド

- TCP/UDPを使ったネットワーク通信を行う
- オープンしているポートの状況を表示
- nc [オプション] [ホスト名/IPアドレス] [ポート番号]

## ■ リモートDNSサーバーに問い合わせ

### • hostコマンド

- host [オプション] ホスト名/ドメイン名/IPアドレス[DNSサーバ]

#### - オプション

- -t : タイプ : 検索タイプを指定

- -C : SOAレコードを表示

#### - 検索タイプ

- a : Aレコード → IPv4アドレス

- ns : NSレコード → ネームサーバ

- mx : MXレコード → メールサーバ

- soa : SOAレコード → ゾーンの権威情報

## ■ リモートDNSサーバーに問い合わせ

### • digコマンド

- dig [オプション] [@DNSサーバ] ホスト名/ドメイン名/IPアドレス [検索タイプ]

#### - オプション

- -x : 逆引きで問い合わせ

#### - 検索タイプ

- a : Aレコード→IP v 4アドレス
- ns : NSレコード→ネームサーバ
- mx : MXレコード→メールサーバ
- soa : SOAレコード→ゾーンの権威情報

## ■問い合わせるDNSサーバーの指定

- /etc/resolv.confファイル
  - domain : 自分が所属するドメイン名を指定 (ドットがない)
  - search : 所属ドメイン名に付加するドメイン名を指定
  - nameserver : 問合せするDNSサーバーのIPアドレスを指定 (最大3つ)

## ■名前解決の実行順序を変更

- /etc/nsswitch.confファイル
  - grep hosts /etc/nsswitch.conf
  - hosts:           files dns mymachines (左から右に順に参照する)

- ハンズオンで実行したネットワーク関連コマンド、確認した関連ファイルの内容
  - ホスト名の確認 (/etc/hostname,hostname,/etc/hosts,getent,/etc/nsswitch.conf)
  - ネットワークの設定 (nmcli,nmtui)
  - ネットワークの状況確認 (ip,ifconfig,route,ping)
  - ネットワークの状況表示 (netstat,ss,traceroute,tracepath,nmap,nc)
  - リモートDNSサーバーに問い合わせ (host,dig)
  - 問い合わせるDNSサーバーの指定と名前解決の実行順序を変更 (/etc/resolv.conf,/etc/switcg.conf)
  
- クラウド型の学習環境を使うと、数ステップでLinuxを操作できる。思い立った時にすぐに使える。ネットワークコマンドの実行や各種サーバ構築など、踏み込んだ学習も可能だが、時間的な制約があるため、継続性が要求される内容の学習は難しい。使い捨て感覚で利用する方が精神衛生上も良い。メリットとデメリットをしっかりと把握した上で、実機環境と使い分けてカジュアルに活用したい。
  
- クラウドの次は、実機や仮想環境でさらに突っ込んだ検証をオススメ。