

# LinuC レベル2 技術解説セミナー

## 主題2.03 : ネットワーク構成

2023/04/23 (Sun) 13:00-14:15

LPI-Japan プラチナスポンサー 株式会社ゼウス・エンタープライズ  
鯨井 貴博 (LinuCエヴァンジェリスト)



# Who are you? (講師紹介)

鯨井貴博

LPI-Japan プラチナスポンサー 株式会社ゼウス・エンタープライズ  
LinuCエヴァンジェリスト

大学時代 Unixの存在を知り、日経Linuxを読み始める。  
2000年にVine Linux 2.0で一度挫折を経験。  
その悔しさを忘れきれず、2007年 他業種からIT業界に転職しLinuxに再チャレンジ。

SE・商用製品サポート・インストラクター・プロジェクト管理などを経験し、現在に至る。  
自分自身が学習で苦労した経験から、初心者を含む受講者に分かりやすい講義を行うように心がけている。

また、興味の向くIT技術・オープンソースソフトウェアなどについて、  
Opensourcetechブログ (<https://www.opensourcetech.tokyo/>) で執筆中。  
実際に自分でやってみる/使ってみる・開発者本人から話を聞いてみることを大切にしています。

Linuxとの  
出会い



Linus Torvaldsさん(Linux開発者)



Igor Sysoevさん(nginx開発者)

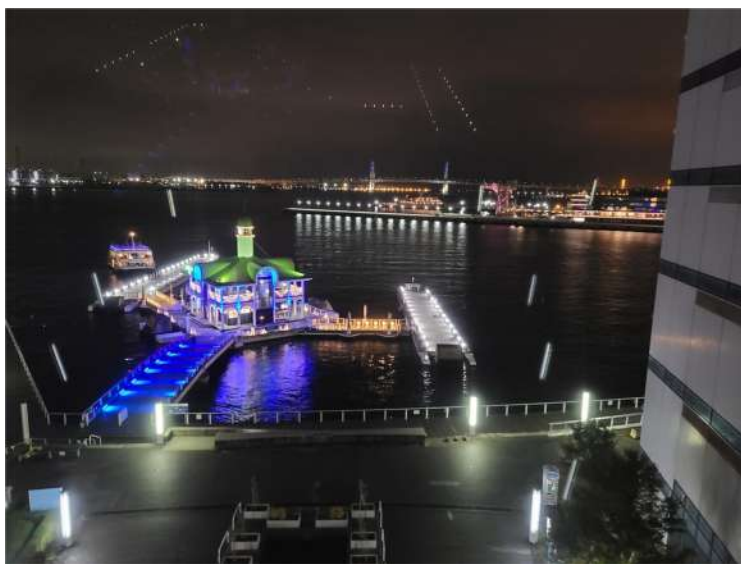


Alexei Vladishevさん(Zabbix開発者)



# Who are you? (講師紹介)

## Open Source Summit Japan 2022 & KubeDay Japan 2022







# Who are you? (講師紹介)

DualStack(IPv4 & IPv6)のkubernetesクラスター構築(v1.26.00・ubuntu22.04)

<https://www.OpensourceTech.tokyo/entry/20230314/1678782139>

MetalLBとService(type:LoadBalancer)による外部クライアントへのサービス公開(kubernetes v1.26.00 on ubuntu22.04)

<https://www.OpensourceTech.tokyo/entry/20230316/1678966960>

NGINX Ingress Controller + Ingressによるサービス公開(kubernetes v1.26)

<https://www.OpensourceTech.tokyo/entry/20230317/1679054672>

nginxコンテナ(Pod)のコンテンツ(index.html)をConfigMapで提供・更新する(kubernetes)

<https://www.OpensourceTech.tokyo/entry/20230319/1679224123>

kubectl topコマンドを実施する(metrics-serverの導入)

<https://www.OpensourceTech.tokyo/entry/20230326/1679832639>

kubernetesにおけるPodへの負荷分散状況の確認(Service/Deployment経由)

<https://www.OpensourceTech.tokyo/entry/20230326/1679837870>

kube-proxyのmodeをiptablesからipvsに変更する

<https://www.OpensourceTech.tokyo/entry/20230327/1679919167>

kubernetes dashboardを使う

<https://www.OpensourceTech.tokyo/entry/20230329/1680093328>

Network Policyによるトラフィック制御(kubernetes)

<https://www.OpensourceTech.tokyo/entry/20230331/1680263776>



最近、kubernetesをよく触ってます！





## 提供するITサービス



### 高水準エンジニアによる支援サービス

#### SES (System Engineering Service)

高い専門スキルを有する  
エンジニア集団だから可能な  
質の高いソリューション

エンジニアの人材不足、ネットワークの構築や保守、システムの開発といったIT分野のニーズに応える支援サービスを行っています。クライアントの悩みや問題に幅広く且つ緻密に対応すべく、1. リナックスとネットワーク技術为基础としたエンジニア、2. ITと英語のスキルを有するバイリンガルエンジニア、3. アプリケーション開発エンジニアという、3分野の専門性に特化した人材でチームを組織。高水準で最適なソリューションを提供します。

### 曖昧さを排除したフェアな人事評価システム

## MyTruth

社員の勤怠データと人事査定を管理して  
公正な社員評価を実現する革新的なシステム

当社が開発した人事査定システムでは、社員の自己申告制によるボトムアップ式の査定を採用。それに基づく評価をポイント化することにより、公正かつ客観的な人事評価を確立します。評価結果のランキング表示によって、社員のパフォーマンスとモチベーションの向上を導き、組織全体のレベルアップを図ります。また、社員の勤怠や賞罰といった労務管理と、社員のポイント評価を一括して管理し、AIによる分析と提案をアウトプットするため、人事・労務担当者の業務負担を大幅に削減して、業務の生産性を高めることが可能です。

### リナックス・ネットワークに強いITスクール

## Zeus Linux Training Center Zeus Network Training Center

未経験者を戦力に育て上げた  
独自のカリキュラムに定評がある  
ITキャリアスクールです

LPI-Japanのアカデミック認定校であるITキャリアスクール「Zeus Linux Training Center / Zeus Network Training Center」を運営し、リナックスとネットワークに強いエンジニアを育成します。当社社員の研修カリキュラムを基にしているため、プロの技術者だけでなく未経験者までも現場ですぐに活躍できる人材に育てます。個人のスキルアップから企業の社員研修まで対応する、幅広いコースを展開しています。

### 技術者のためのSNS サービス

## FIRE SIDER

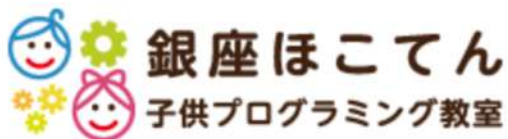
ユーザー間の意見交換と  
企業とのマッチングを提供する  
エンジニアに特化したSNS

人的交流が不足しがちなIT業界において、専門性を持つエンジニアの情報収集を可能にするソーシャルネットワーキングサービスです。ユーザーが話題を投稿するだけでなく、ディスカッションから企画立案・協同制作までのフローを実現する場を提供します。また、ユーザーが個人プロフィールページを作成することにより、AIが企業とのマッチングを最適化します。エンジニアは専門スキルを発信し、企業は採用活動を効率化することができます。

<https://www.zeus-enterprise.co.jp/solution/service/>







## 小学生向けプログラミング教室

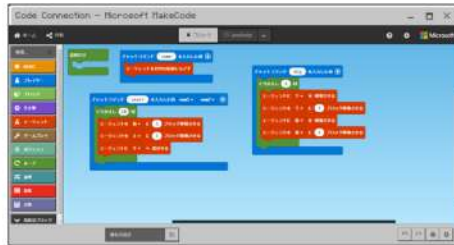
### COURSE ～コース紹介～

#### Minecraftコース

##### 大人気ゲーム！Minecraftを使って楽しくプログラミング！

「Minecraft MakeCode」では、通常のMinecraftとは違い、エージェントという小さなロボットをプログラムによって操ることで、プレイヤーの代わりに様々な作業をととても短い時間で行わせることができます！プログラミング的手法を使い、Minecraftの世界を自由に作り上げましょう！

→各コマ、集合型レッスン定員12名・オンラインレッスン定員3名



※ 保護者同伴可能

#### 会場

銀座ほこてん子供プログラミング教室  
〒104-0061 東京都中央区銀座5丁目8-20 銀座コア8階

#### 対象年齢

小学校3年～6年

#### 講習時間

10:30～11:30/12:00～13:00/13:30～14:30/  
15:00～16:00/16:30～17:30

#### 講習曜日

毎日

#### 持ち物

筆記用具

#### 入学金

ありません

#### 月謝

6,000円～（税込） / 月2回～（1コマ60分）

#### 教材費

Minecraftのライセンス代：3300円（税込）  
/ テキスト代：2530円（税込）

#### 無料体験

好評受付中！必要機材は全てお貸しします！

### オンラインレッスン対応!!



当スクールでは通常のレッスンをオンラインでもご受講いただけます！  
インターネット環境とPCをお持ちでしたら、Zoomを使用し  
オンラインコースでもプログラミングを学べます！

### キャンペーン

兄弟・姉妹一緒の  
お申し込みで **10%**OFF!  
全員月謝がずっと

<https://www.it-training.tokyo/kids/index.html>





## 貸し会議室



<使用例>



スクール



会議



発表会



説明会



オンライン配信

WEB会議・  
ウェビナー  
など



<https://ginza168.tokyo/>



[https://www.zeus-enterprise.co.jp/ikebukuro\\_office/](https://www.zeus-enterprise.co.jp/ikebukuro_office/)



1. LinuCについて  
試験概要と特徴
2. 技術解説  
主題2.03：ネットワーク構成
  - 2.03.1 基本的なネットワーク構成
  - 2.03.2 高度なネットワーク構成
  - 2.03.3 ネットワークの問題解決
3. Appendix
4. お知らせ
5. Q&A





- Linuxの基本的なネットワーク設定を理解する
- ルーティングテーブルの操作を理解する
- ネットワーク関連のツールの使用方法を理解する
- ネットワーク関連のトラブルシュート方法を理解する



# LinuC について



## ■LinuCとは

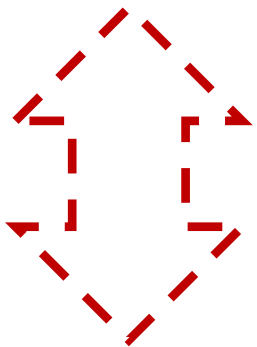
クラウド時代の即戦力エンジニアであることを証明するLinux技術者認定

- ✓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





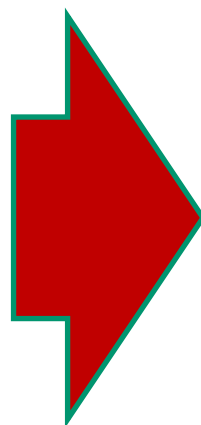
AWSなどの  
パブリッククラウドを  
活用するための技術



間が  
欠けて  
いる状態



オンプレミスの  
サーバーサイドLinux技術



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

【今まで／その他】

LC LinuC Version10.0





## 101試験

- 1.01 : Linuxのインストールと仮想マシン・コンテナの利用
  - 1.01.1 Linuxのインストール、起動、接続、切断と停止
  - 1.01.2 仮想マシン・コンテナの概念と利用
  - 1.01.3 ブートプロセスとsystemd
  - 1.01.4 プロセスの生成、監視、終了
  - 1.01.5 デスクトップ環境の利用
- 1.02 : ファイル・ディレクトリの操作と管理
  - 1.02.1 ファイルの所有者とパーミッション
  - 1.02.2 基本的なファイル管理の実行
  - 1.02.3 ハードリンクとシンボリックリンク
  - 1.02.4 ファイルの配置と検索
- 1.03 : GNUとUnixのコマンド
  - 1.03.1 コマンドラインの操作
  - 1.03.2 フィルタを使ったテキストストリームの処理
  - 1.03.3 ストリーム、パイプ、リダイレクトの使用
  - 1.03.4 正規表現を使用したテキストファイルの検索
  - 1.03.5 エディタを使った基本的なファイル編集の実行
- 1.04 : リポジトリとパッケージ管理
  - 1.04.1 apt コマンドによるパッケージ管理
  - 1.04.2 Debianパッケージ管理
  - 1.04.3 yumコマンドによるパッケージ管理
  - 1.04.4 RPMパッケージ管理
- 1.05 : ハードウェア、ディスク、パーティション、ファイルシステム
  - 1.05.1 ハードウェアの基礎知識と設定
  - 1.05.2 ハードディスクのレイアウトとパーティション
  - 1.05.3 ファイルシステムの作成と管理、マウント

<https://linuc.org/linuc1/range/101.html>  
<https://linuc.org/linuc1/range/102.html>

## 102試験

- 1.06 : シェルおよびスクリプト
  - 1.06.1 シェル環境のカスタマイズ
  - 1.06.2 シェルスクリプト
- 1.07 : ネットワークの基礎
  - 1.07.1 インターネットプロトコルの基礎
  - 1.07.2 基本的なネットワーク構成
  - 1.07.3 基本的なネットワークの問題解決
  - 1.07.4 クライアント側のDNS設定
- 1.08 : システム管理
  - 1.08.1 アカウント管理
  - 1.08.2 ジョブスケジューリング
  - 1.08.3 ローカライゼーションと国際化
- 1.09 : 重要なシステムサービス
  - 1.09.1 システム時刻の管理
  - 1.09.2 システムのログ
  - 1.09.3 メール配送エージェント(MTA)の基本
- 1.10 : セキュリティ
  - 1.10.1 セキュリティ管理業務の実施
  - 1.10.2 ホストのセキュリティ設定
  - 1.10.3 暗号化によるデータの保護
  - 1.10.4 クラウドセキュリティの基礎
- 1.11 : オープンソースの文化
  - 1.11.1 オープンソースの概念とライセンス
  - 1.11.2 オープンソースのコミュニティとエコシステム



## 201試験

- 2.01 : システムの起動とLinuxカーネル
  - 2.01.1 ブートプロセスとGRUB
  - 2.01.2 システム起動のカスタマイズ
  - 2.01.3 Linux カーネルの構成要素
  - 2.01.4 Linuxカーネルのコンパイル
  - 2.01.5 カーネル実行時における管理とトラブルシューティング
- 2.02 : ファイルシステムとストレージ管理
  - 2.02.1 ファイルシステムの設定とマウント
  - 2.02.2 ファイルシステムの管理
  - 2.02.3 論理ボリュームマネージャの設定と管理
- 2.03 : ネットワーク構成
  - 2.03.1 基本的なネットワーク構成
  - 2.03.2 高度なネットワーク構成
  - 2.03.3 ネットワークの問題解決
- 2.04 : システムの保守と運用管理
  - 2.04.1 makeによるソースコードからのビルドとインストール
  - 2.04.2 バックアップとリストア
  - 2.04.3 ユーザへの通知
  - 2.04.4 リソース使用状況の把握
  - 2.04.5 死活監視、リソース監視、運用監視ツール
  - 2.04.6 システム構成ツール
- 2.05 : 仮想化サーバー
  - 2.05.1 仮想マシンの仕組みとKVM
  - 2.05.2 仮想マシンの作成と管理
- 2.06 : コンテナ
  - 2.06.1 コンテナの仕組み
  - 2.06.2 Dockerコンテナとコンテナイメージの管理

## 202試験

- 2.07 : ネットワーククライアントの管理
  - 2.07.1 DHCPサーバーの設定と管理
  - 2.07.2 PAM認証
  - 2.07.3 LDAPクライアントの利用方法
  - 2.07.4 OpenLDAPサーバーの設定
- 2.08 : ドメインネームサーバー
  - 2.08.1 BINDの設定と管理
  - 2.08.2 ゾーン情報の管理
  - 2.08.3 セキュアなDNSサーバーの実現
- 2.09 : HTTPサーバーとプロキシサーバー
  - 2.09.1 Apache HTTPサーバーの設定と管理
  - 2.09.2 OpenSSLとHTTPSの設定
  - 2.09.3 nginxの設定と管理
  - 2.09.4 Squidの設定と管理
- 2.10 : 電子メールサービス
  - 2.10.1 Postfixの設定と管理
  - 2.10.2 Dovecotの設定と管理
- 2.11 : ファイル共有サービス
  - 2.11.1 Sambaの設定と管理
  - 2.11.2 NFSサーバーの設定と管理
- 2.12 : システムのセキュリティ
  - 2.12.1 iptables や firewalld によるパケットフィルタリング
  - 2.12.2 OpenSSH サーバーの設定と管理
  - 2.12.3 OpenVPNの設定と管理
  - 2.12.4 セキュリティ業務
- 2.13 : システムアーキテクチャ
  - 2.13.1 高可用システムの実現方式
  - 2.13.2 キャパシティプランニングとスケーラビリティの確保
  - 2.13.3 クラウドサービス上のシステム構成
  - 2.13.4 典型的なシステムアーキテクチャ





# LinuC 学習のコツ

- ①出題範囲の内容について調べてみる  
公式ドキュメント・技術書など
- ②実際に操作してみる  
これが大事！
- ③学習の補助教材などを利用する
  - ・メールマガジン
  - ・標準教科書
  - ・過去のセミナー資料詳細は、<https://lpi.or.jp/learning/>



LPI-Japanが開発した  
大人気の教科書で  
Linuxを効率的に学ぶ

- Linux 標準教科書
- Linux サーバー標準教科書
- 高信頼システム構築標準教科書
- Linux セキュリティ標準教科書
- Linux システム管理標準教科書

Linux初心者のための入門編と  
中級者向けのネットワーク編の  
Linux解説コラム

- Linux 道場入門編
- Linux 道場ネットワーク編
- Linux 道場 Linux学習環境構築編

Linux豆知識  
Linuxを学習する上で出てくる素朴な疑問や  
便利なテクニックなどを紹介しています。



## メールマガジンでコツコツと

### 学習に役立つメールマガジン

LPI-Japanでは、試験レベルごとの例題解説など、  
学習に役立つメールマガジンを無料でお届けしています。

LPI-Japan LinuC通信  
「レベル2・レベル3  
を受けてみよう！」で  
サンプル問題作ってる  
ので、よかったら登録  
してください！

### 過去のメールマガジンの 例題解説をまとめています。

LPI-Japanでは、試験レベルごとの例題解説など、  
学習に役立つメールマガジンを無料でお届けしています。



## 人気の技術解説無料 セミナーも活用して

LPI-Japanでは、『LinuCレベル1～新出題範囲における  
受験準備とポイント解説』など、レベル別の  
技術解説無料セミナーを開催しています。  
学習の仕方で迷ったら是非足を運んでみてください。  
他の受験者の方と意見交換もでき、モチベーションもあがります！

過去のセミナー資料のダウンロードはこちら④



## ④過去セミナーの動画

<https://www.youtube.com/user/LPIJapan>

検索

open your NEXT future

LinuC OSS-DB Silver/Gold HTMLS Professional Certification OPCEL

頼られるための、頼れる資格

Linux技術者認定「LinuC」

LPI-Japan

@LPIJapan チャンネル登録者数 3890人 79本の動画

LPI-Japanチャンネルは、LinuC (Linux) やOSS-DB (PostgreSQL) 、HT... >

登録済み

ホーム 動画 再生リスト コミュニティ チャンネル 概要

人気の動画 ▶ すべて再生

テーマ	LinuCレベル	Version	技術解説無料セミナー
仮想マシン・コンテナの概念と利用	レベル1	Version10.0	1:07:35
システムの起動とLinuxカーネル (ブートプロセスとGRUB、カーネルの構成要素、ほか)	レベル2	Version10.0	1:06:36
ブートプロセスとsystemd	レベル1	Version10.0	1:15:40
シェルスクリプトとジョブスケジューリング (シェルスクリプトの基本構文、cronの使い方、ほか)	レベル1	Version10.0	1:15:44
ネットワークの基礎 (プロトコル、ネットワーク構成、ほか)	レベル1	Version10.0	1:25:44

仮想マシン・コンテナの概念と利用 (Linux学習) 9775 回視聴・2年前

システムの起動とLinuxカーネル (Linux学習) 8487 回視聴・2年前

ブートプロセスとsystemd (Linux学習) 8087 回視聴・2年前

シェルスクリプトとスケジューリング (Linux学習) 5822 回視聴・2年前

ネットワークの基礎 (Linux学習) 5757 回視聴・2年前





学習の具体的な進め方(2~3か月程度)

学習開始(0%)

試験範囲の確認(LinuxC HP)

●月頃、受験しようという目標を立てる



LinuxC認定教材の購入・1週目読込 <https://lpi.or.jp/linuc1/book.shtml>  
<https://lpi.or.jp/linuc2/book.shtml>



LinuxC認定教材・Webサイトを参考に、実機操作(サーバ構築やコマンド操作)を試す  
※操作やトラブルシュートで力が身に付く！

1ヶ月



LinuxC認定教材 2週目読込

●月●日頃、受験しようと決める



問題集やメルマガサンプル問題で理解力確認 <https://linuc.org/study/samples/>  
※理解不足箇所の洗い出し



2ヶ月

LinuxC認定教材 3週目

※弱点の補強



仕上げ  
(80%)

受験申込

受験日の変更も可能なので安心



問題を8~9割以上、正解となるまで繰り返し解く  
苦手な部分を重点的に復習



受験

- ・ 受験まで継続して学習すること
- ・ 繰り返し学習し、理解度/問題正解率を高めた状態で受験すること

2.5ヶ月  
完全に理解した！  
(100%)





## 主題2.03：ネットワーク構成

試験としての重要度(出題割合)は高くないが、  
実践では特にトラブルシュートで力を発揮する内容なので  
しっかりと理解しておきたい！



## 2.03.1 基本的なネットワーク構成

重要度 3

### 概要

ネットワークデバイスを設定し、有線または無線のローカルネットワークと広域ネットワークに接続できる。

### 詳細

Ethernetネットワークインターフェイスを設定および操作する。デフォルトルートの設定を含む。

ip, ifconfig, route, arp, nmcli

無線ネットワークを構成する。

iw, iwconfig, iwlist



## 2.03.2 高度なネットワーク構成

重要度 3

### 概要

複数のサブネットへの経路設定ができる。これにはルータ機能の設定も含まれる。  
ネットワークの状態を監視できる。  
ネットワークのデバイス、通信状態などを分析できる。

### 詳細

ルーティングテーブルを操作するユーティリティ

ip, route

IPフォワードを設定してルータ機能を実装する。

/etc/sysctl.conf, sysctl

ネットワークデバイスの状態を分析するユーティリティ

ip, ifconfig

TCP/IPの通信状態やトラフィックを監視および分析するユーティリティ

ping, ping6, netcat(nc, ncat), tcpdump, nmap, ss, netstat





## 2.03.3 ネットワークの問題解決

重要度 3

### 概要

一般的なネットワーク設定に関する問題を特定して解決できる。  
これには、基本的な設定ファイルの位置とコマンドに関する知識も含まれる。

### 詳細

ネットワークの設定に関する情報を取得する。

hostname, /etc/hostname, /etc/hosts, /etc/resolv.conf, nmcli, ip

ネットワークの通信経路の問題を特定して解決する。

traceroute, traceroute6, ip, route, mtr

ハードウェアの認識と利用に関する情報を取得する。

dmesg, /var/log/syslogおよび/var/log/messagesなどのシステムのログファイルおよび systemd のジャーナル  
システムの初期化ファイルとその内容 (systemd)

NetworkManagerおよびそれがネットワーク設定に及ぼす影響について知っている。

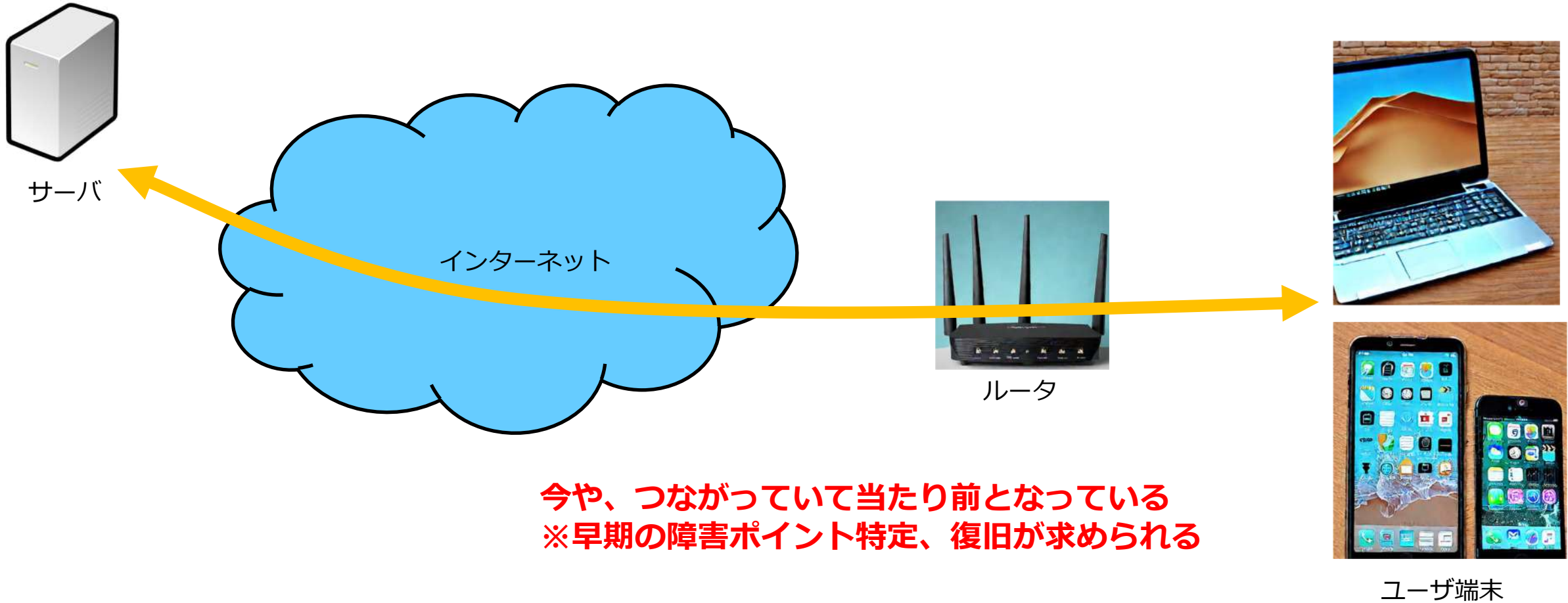
/etc/network/, /etc/sysconfig/network-scripts/



はじめに



## ネットワークの重要性





## トラブルシューティングの進め方(OSI参照モデル)

階層	階層名	取り扱う情報など
Layer7	アプリケーション層	アプリケーションデータ
Layer6	プレゼンテーション層	
Layer5	セッション層	
Layer4	トランスポート層	ポート番号(TCP/UDPなど)
Layer3	ネットワーク層	IPアドレス
Layer2	データリンク層	MACアドレス(ARP)
Layer1	物理層	物理メディア



トラブルシューティングは  
下の階層から！





## IPアドレスの種類

- IPv4(32bits、10進数で標記)  
0.0.0.0 ~ 255.255.255.255の範囲

プライベート : 10.0.0.0 ~ 10.255.255.255(/8)  
172.16.0.0 ~ 172.31.255.255(/12)  
192.168.0.0 ~ 192.168.255.255(/16)  
※RFC1918 <https://www.rfc-editor.org/rfc/rfc1918>

- IPv6(128bits、16進数で表記)  
0000.0000.0000.0000.0000.0000.0000.0000 ~ ffff.ffff.ffff.ffff.ffff.ffff.ffff.ffffの範囲

グローバルユニキャストアドレス

リンクローカルユニキャストアドレス ※ルータを超えない組織内で利用

ユニークローカルユニキャストアドレス ※同一LAN内のみで利用

fe80:1::a00:27ff:fe16:5fac/64

: 1 → : 0001 の"0"が省略されている  
: : → 連続する"0"が省略されている

IPv4(or IPv6)のみを使った構成を  
SingleStack、

IPv4/IPv6の両方を使った構成を  
DualStackと呼ぶことがある。



## MACアドレス(48bits 、 16進数で表記)

08:00:27:70:df:8a

前半24bits(08:00:27)は、ベンダーコード

<https://gist.github.com/aallan/b4bb86db86079509e6159810ae9bd3e4>

```
List of MAC addresses with vendors identities
mac-vendor.txt
1 000000 Officially Xerox
2 000001 SuperLAN-2U
3 000002 BBN (was internal usage only, no longer used)
4 000003 XEROX CORPORATION
5 000004 XEROX CORPORATION
6 000005 XEROX CORPORATION
7 000006 XEROX CORPORATION
8 000007 XEROX CORPORATION
9 000008 XEROX CORPORATION
10 000009 powerpipes?
11 00000A OMRON TATEISI ELECTRONICS CO.
12 00000B MATRIX CORPORATION
13 00000C Cisco
14 00000D FIBRONICS LTD.
15 00000E Fujitsu
16 00000F Next
17 000010 Hughes
18 000011 Tektrnix
19 000012 INFORMATION TECHNOLOGY LIMITED
20 000013 Camex
21 000014 Netronix
22 000015 Datapoint Corporation
23 000016 DU PONT PIXEL SYSTEMS
24 000017 Oracle
25 000018 Webster Computer Corporation Appletalk/Ethernet Gateway
26 000019 APPLIED DYNAMICS INTERNATIONAL
27 00001A AMD
```

後半24bits(70:df:8a)は、ベンダー内で重複がない一意



## 2.03.1 基本的なネットワーク構成



## 2.03.1 基本的なネットワーク構成

重要度 3

### 概要

ネットワークデバイスを設定し、有線または無線のローカルネットワークと広域ネットワークに接続できる。

### 詳細

Ethernetネットワークインターフェイスを設定および操作する。デフォルトルートの設定を含む。

ip, ifconfig, route, arp, nmcli

無線ネットワークを構成する。

iw, iwconfig, iwlist





## IPアドレスの確認

```
ubuntu@ubuntu-server:~$ ip addr show
```

```
.  
. .  
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default  
qlen 1000  
    link/ether 08:00:27:16:5f:ac brd ff:ff:ff:ff:ff:ff  
    inet 192.168.1.111/24 metric 100 brd 192.168.1.255 scope global dynamic enp0s3  
        valid_lft 6284sec preferred_lft 6284sec  
    inet6 240f:32:57b8:1:a00:27ff:fe16:5fac/64 scope global dynamic mngtmpaddr noprefixroute  
        valid_lft 298sec preferred_lft 298sec  
    inet6 fe80::a00:27ff:fe16:5fac/64 scope link  
        valid_lft forever preferred_lft forever
```

```
[root@localhost ~]# ifconfig
```

```
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500  
    inet 192.168.1.140 netmask 255.255.255.0 broadcast 192.168.1.255  
    inet6 240f:32:57b8:1:5e06:7ace:4d2e:3456 prefixlen 64 scopeid 0x0<global>  
    inet6 fe80::81ce:f1ca:2a63:fbf3 prefixlen 64 scopeid 0x20<link>  
    ether 08:00:27:70:df:8a txqueuelen 1000 (Ethernet)  
    RX packets 232 bytes 19508 (19.0 KiB)  
    RX errors 0 dropped 0 overruns 0 frame 0  
    TX packets 142 bytes 15387 (15.0 KiB)  
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

※利用するディストリビューションによって、  
デフォルトで用意されているコマンドが異なる

※IPv6については、  
デフォルトでは無効化されていることがある



## IPアドレスの確認

```
[root@localhost ~]# nmcli device show
GENERAL.DEVICE:           enp0s3
GENERAL.TYPE:             ethernet
GENERAL.HWADDR:           08:00:27:70:DF:8A
GENERAL.MTU:              1500
GENERAL.STATE:            100 (接続済み)
GENERAL.CONNECTION:       enp0s3
GENERAL.CON-PATH:         /org/freedesktop/NetworkManager/ActiveConnection/1
WIRED-PROPERTIES.CARRIER: オン
IP4.ADDRESS[1]:           192.168.1.140/24
IP4.GATEWAY:              192.168.1.1
IP4.ROUTE[1]:             dst = 0.0.0.0/0, nh = 192.168.1.1, mt = 100
IP4.ROUTE[2]:             dst = 192.168.1.0/24, nh = 0.0.0.0, mt = 100
IP4.DNS[1]:               8.8.8.8
IP4.DNS[2]:               8.8.4.4
IP6.ADDRESS[1]:           240f:32:57b8:1:5e06:7ace:4d2e:3456/64
IP6.ADDRESS[2]:           fe80::81ce:f1ca:2a63:fbf3/64
IP6.GATEWAY:              fe80::6ee4:daff:fedd:7400
IP6.ROUTE[1]:             dst = 240f:32:57b8:1::/64, nh = ::, mt = 100
IP6.ROUTE[2]:             dst = ::/0, nh = fe80::6ee4:daff:fedd:7400, mt = 100
IP6.ROUTE[3]:             dst = ff00::/8, nh = ::, mt = 256, table=255
IP6.ROUTE[4]:             dst = fe80::/64, nh = ::, mt = 100.
.
.
```

IPアドレス以外にも、

- MTU
  - DNS名
  - ルーティング情報
  - インターフェイスの状態
- などが読み取れる



## IPアドレスの一時的な適用

```
[root@localhost ~]# ifconfig enp0s3 192.168.1.140 netmask 255.255.255.0
```

```
[root@localhost ~]# ip addr add 192.168.1.140/255.255.255.0 dev enp0s3
```



## IPアドレスの恒久的な適用

```
[root@localhost ~]# nmcli c modify enp0s3 ipv4.method manual ipv4.addresses "192.168.1.140/24"
```

```
[root@localhost ~]# vi /etc/sysconfig/network-scripts/ifcfg-enp0s3
```

```
[root@localhost ~]# cat /etc/sysconfig/network-scripts/ifcfg-enp0s3
```

```
TYPE=Ethernet
```

```
PROXY_METHOD=none
```

```
BROWSER_ONLY=no
```

```
BOOTPROTO=none . . . . "static"と記載してもOK、DHCPを使う場合"dhcp"と記載する
```

```
.
```

```
.
```

```
.
```

```
NAME=enp0s3
```

```
UUID=22e3c1a4-ff0e-4c3e-bacd-d75d719e367a
```

```
DEVICE=enp0s3
```

```
ONBOOT=yes . . . . OS起動時に利用するかどうか
```

```
IPADDR=192.168.1.140 . . . . IPv4のアドレス指定
```

```
PREFIX=24 . . . . プレフィックス(サブネットマスク)指定
```

```
GATEWAY=192.168.1.1 . . . . ゲートウェイ指定
```

```
DNS1=8.8.8.8 . . . . 優先DNSサーバ指定
```

```
DNS2=8.8.4.4 . . . . セカンダリDNSサーバ指定
```

```
[root@localhost ~]# systemctl restart network
```





## IPアドレスの恒久的な適用(ubuntu)

How to Configure Networking on Ubuntu with Netplan

<https://vitux.com/how-to-configure-networking-with-netplan-on-ubuntu/>

<https://netplan.readthedocs.io/en/stable/tutorials/>

```
tin@ubuntu: /  
File Edit View Search Terminal Help  
GNU nano 2.9.3 /etc/netplan/01-network-manager-all.yaml  
  
# Let NetworkManager manage all devices on this system  
network:  
  version: 2  
  renderer: NetworkManager  
  ethernets:  
    ens33:  
      dhcp4: no  
      addresses: [192.168.72.150/24]  
      gateway4: 192.168.72.2  
      nameservers:  
        addresses: [8.8.8.8,8.8.4.4]  
  
^G Get Help  ^O Write Out  ^W Where Is  ^K Cut Text  ^J Justify  
^X Exit      ^R Read File  ^\ Replace   ^U Uncut Text ^T To Spell
```

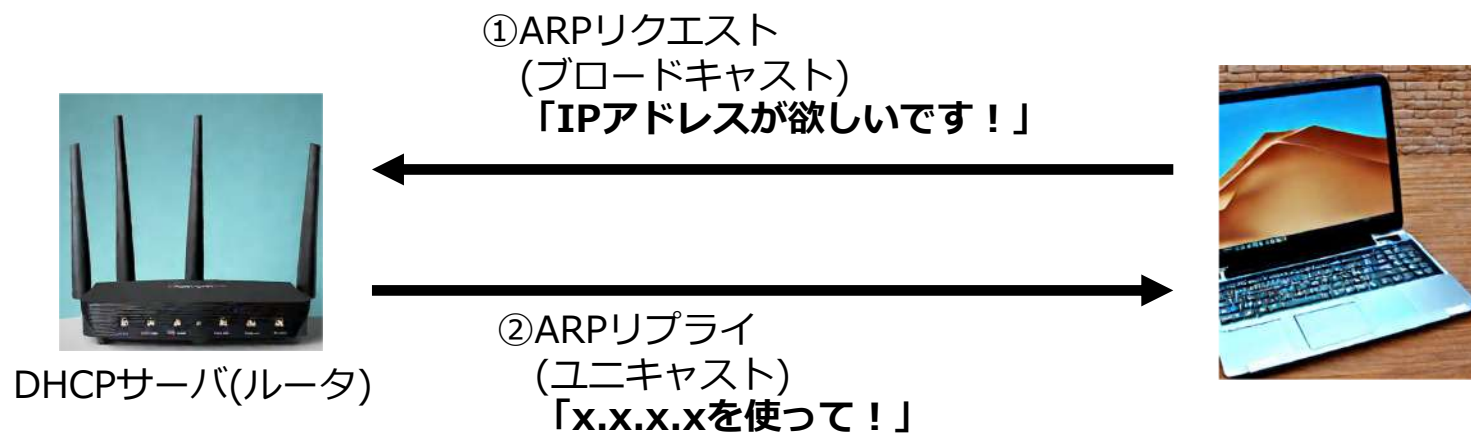


## ARPの確認

```
ubuntu@ubuntuserver:~$ arp -a  
? (192.168.1.123) at 7c:b2:7d:de:c2:f9 [ether] on enp0s3  
? (192.168.1.113) at 16:27:f5:0a:2d:01 [ether] on enp0s3  
_gateway (192.168.1.1) at 74:da:88:e6:b3:f8 [ether] on enp0s3.
```

関連付けられている  
IPアドレスとMACアドレスの一覧を表示

## ARPのやりとり





## ルーティング情報の参照

```
[root@localhost ~]# route
```

Kernel IP routing table

Destination	Gateway	Genmask	Flags	Metric	Ref	Use	Iface
default	gateway	0.0.0.0	UG	100	0	0	enp0s3
192.168.1.0	0.0.0.0	255.255.255.0	U	100	0	0	enp0s3

```
[root@localhost ~]# ip route show
```

default via 192.168.1.1 dev enp0s3 proto static metric 100

192.168.1.0/24 dev enp0s3 proto kernel scope link src 192.168.1.140 metric 100

```
[root@localhost ~]# nmcli c show enp0s3
```

connection.id: enp0s3

.  
. .  
.

ipv4.method: manual

ipv4.dns: 8.8.8.8,8.8.4.4

ipv4.dns-search: --

ipv4.dns-options: ""

ipv4.dns-priority: 0

ipv4.addresses: 192.168.1.140/24

ipv4.gateway: 192.168.1.1



## ルーティング情報の設定

```
[root@localhost ~]# route add default gw 192.168.1.1
```

```
[root@localhost ~]# cat /etc/sysconfig/network-scripts/ifcfg-enp0s3
```

```
.  
.
.
```

GATEWAY=192.168.1.1 . . . . ゲートウェイ指定

DNS1=8.8.8.8 . . . . 優先DNSサーバ指定

DNS2=8.8.4.4 . . . . セカンダリDNSサーバ指定

```
[root@localhost ~]# systemctl restart network
```

```
[root@localhost ~]# nmcli connection modify enp0s3 ipv4.gateway "192.168.1.1"
```



## Linuxに無線LAN設定

### iwコマンド

- 無線LAN設定の確認や操作をする  
<https://linux.die.net/man/8/iw>

### iwlistコマンド

- 無線LANデバイスで検知した無線LAN情報の表示など  
<https://linux.die.net/man/8/iwlist>

### iwconfigコマンド

- 無線LAN設定をする

<https://linux.die.net/man/8/iwconfig>

### Ubuntu 22.04でのwi-fi設定(netplan)

<https://www.opensourcetech.tokyo/entry/20230218/1676723684>

```
ubuntu@kvm2ubuntu:/etc/netplan$ cat 00-installer-config-wifi.yaml
# This is the network config written by 'subiquity'

network:

  wifis:

    wlx00029e4bbb7f:

      access-points:

        myaccesspoint:

          password: "myappassword"

      addresses:

        - 192.168.100.12/24

      gateway4: 192.168.100.1

      nameservers:

        addresses:

          - 8.8.8.8

          - 8.8.4.4

          - 192.168.100.1

      search: []
```





## 2.03.2 高度なネットワーク構成



## 2.03.2 高度なネットワーク構成

重要度 3

### 概要

複数のサブネットへの経路設定ができる。これにはルータ機能の設定も含まれる。  
ネットワークの状態を監視できる。  
ネットワークのデバイス、通信状態などを分析できる。

### 詳細

ルーティングテーブルを操作するユーティリティ

ip, route

IPフォワードを設定してルータ機能を実装する。

/etc/sysctl.conf, sysctl

ネットワークデバイスの状態を分析するユーティリティ

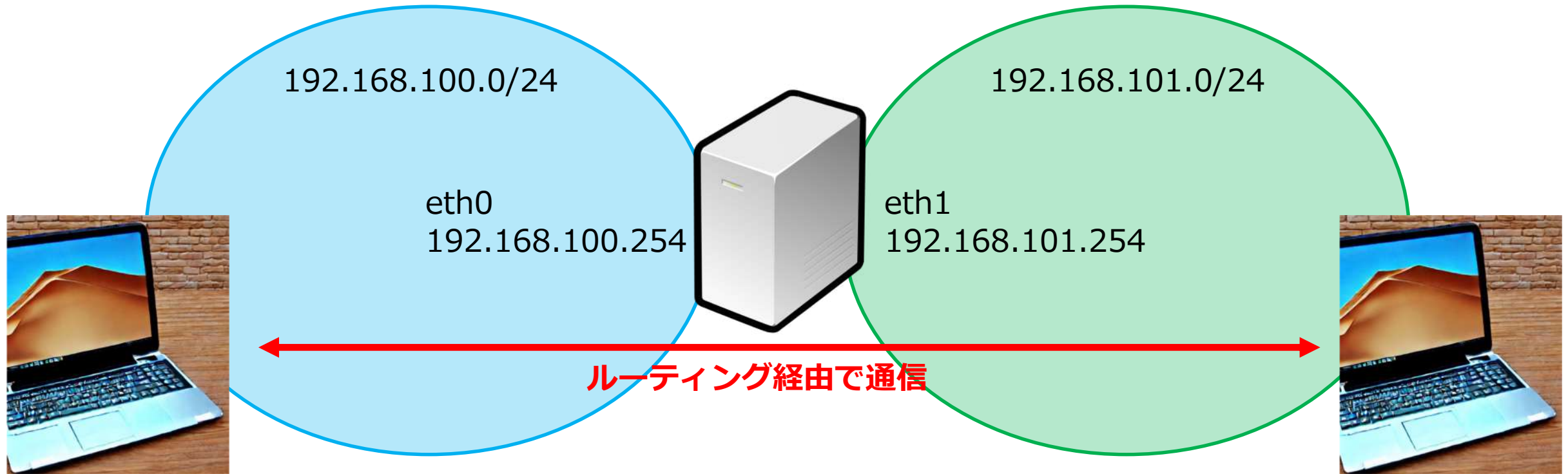
ip, ifconfig

TCP/IPの通信状態やトラフィックを監視および分析するユーティリティ

ping, ping6, netcat(nc, ncat), tcpdump, nmap, ss, netstat



## Linuxをルータとして動作させる





/etc/sysctl.conf, sysctl = カーネルパラメータの表示や変更をする

```
kubeuser@master01:~$ cat /etc/sysctl.d/k8s.conf
```

```
net.ipv4.ip_forward = 1  
net.ipv6.conf.all.forwarding = 1
```

```
kubeuser@master01:~$ sudo sysctl --system
```

```
.  
. .  
. .
```

```
* Applying /etc/sysctl.d/k8s.conf ...
```

```
net.bridge.bridge-nf-call-ip6tables = 1  
net.bridge.bridge-nf-call-iptables = 1  
net.ipv4.ip_forward = 1  
net.ipv6.conf.all.forwarding = 1  
• Applying /etc/sysctl.conf ...
```

```
kubeuser@master01:~$ sudo sysctl -a
```

```
.  
. .  
. .
```

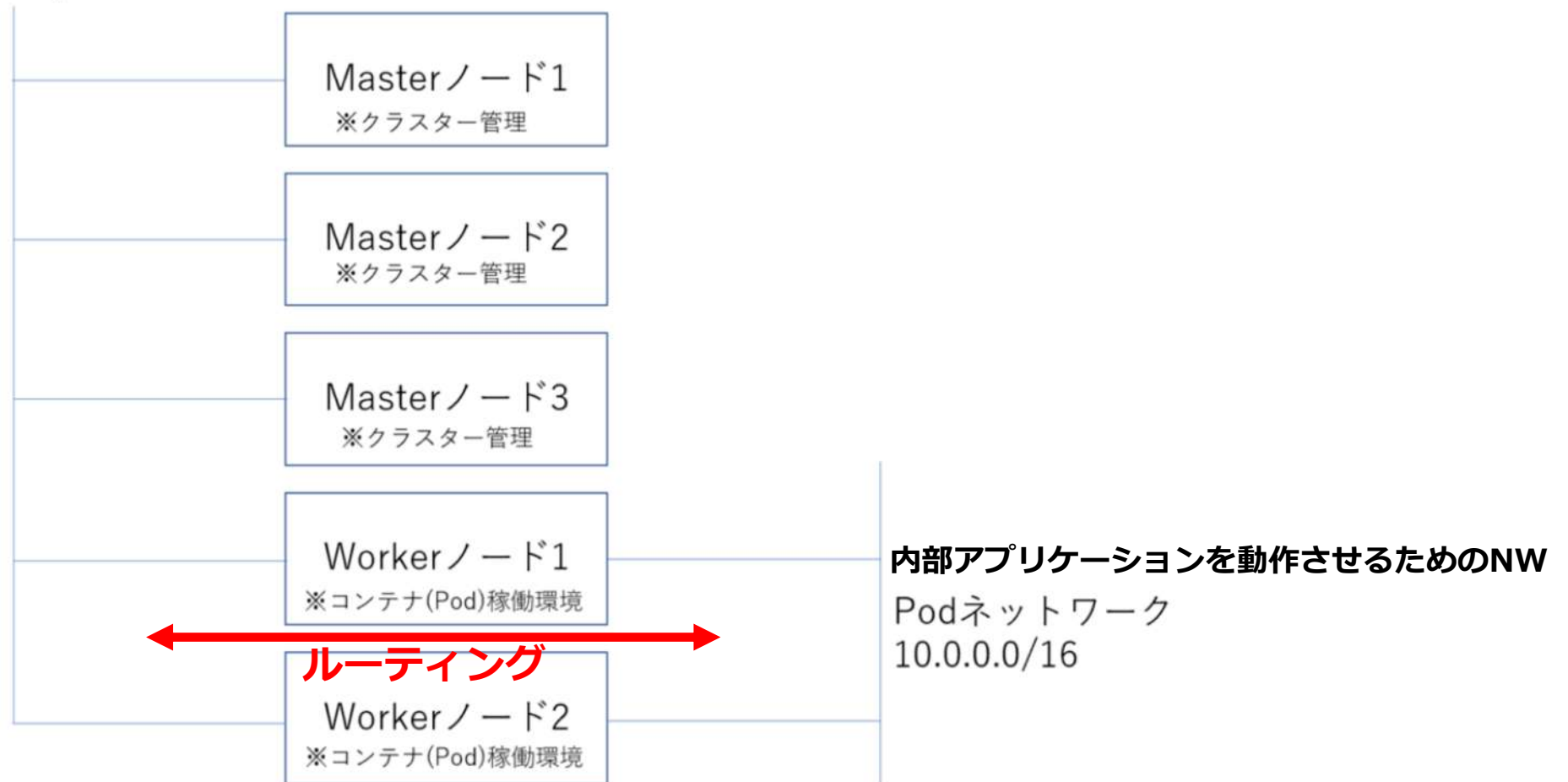
```
net.ipv4.ip_forward = 1
```

デフォルトでは無効(0)となっているので、有効化(1)に変更する必要がある



kubernetesなどの構築では必要

192.168.1.0/24 複数のPCを連携させるためのNW







## ping/ping6

```
kubeuser@master01:~$ ping -c 4 8.8.8.8
```

```
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
```

```
64 bytes from 8.8.8.8: icmp_seq=1 ttl=57 time=4.46 ms
```

```
64 bytes from 8.8.8.8: icmp_seq=2 ttl=57 time=4.67 ms
```

```
64 bytes from 8.8.8.8: icmp_seq=3 ttl=57 time=4.60 ms
```

```
64 bytes from 8.8.8.8: icmp_seq=4 ttl=57 time=4.32 ms
```

```
--- 8.8.8.8 ping statistics ---
```

```
4 packets transmitted, 4 received, 0% packet loss, time 3004ms
```

```
rtt min/avg/max/mdev = 4.324/4.513/4.672/0.133 ms
```

```
kubeuser@master01:~$ ping6 -c 4 2a05:d014:edb:5704::6
```

```
PING 2a05:d014:edb:5704::6(2a05:d014:edb:5704::6) 56 data bytes
```

```
64 bytes from 2a05:d014:edb:5704::6: icmp_seq=1 ttl=43 time=248 ms
```

```
64 bytes from 2a05:d014:edb:5704::6: icmp_seq=2 ttl=43 time=248 ms
```

```
64 bytes from 2a05:d014:edb:5704::6: icmp_seq=3 ttl=43 time=249 ms
```

```
64 bytes from 2a05:d014:edb:5704::6: icmp_seq=4 ttl=43 time=249 ms
```

```
--- 2a05:d014:edb:5704::6 ping statistics ---
```

```
4 packets transmitted, 4 received, 0% packet loss, time 3000ms
```

```
rtt min/avg/max/mdev = 247.655/248.354/249.284/0.621 ms
```



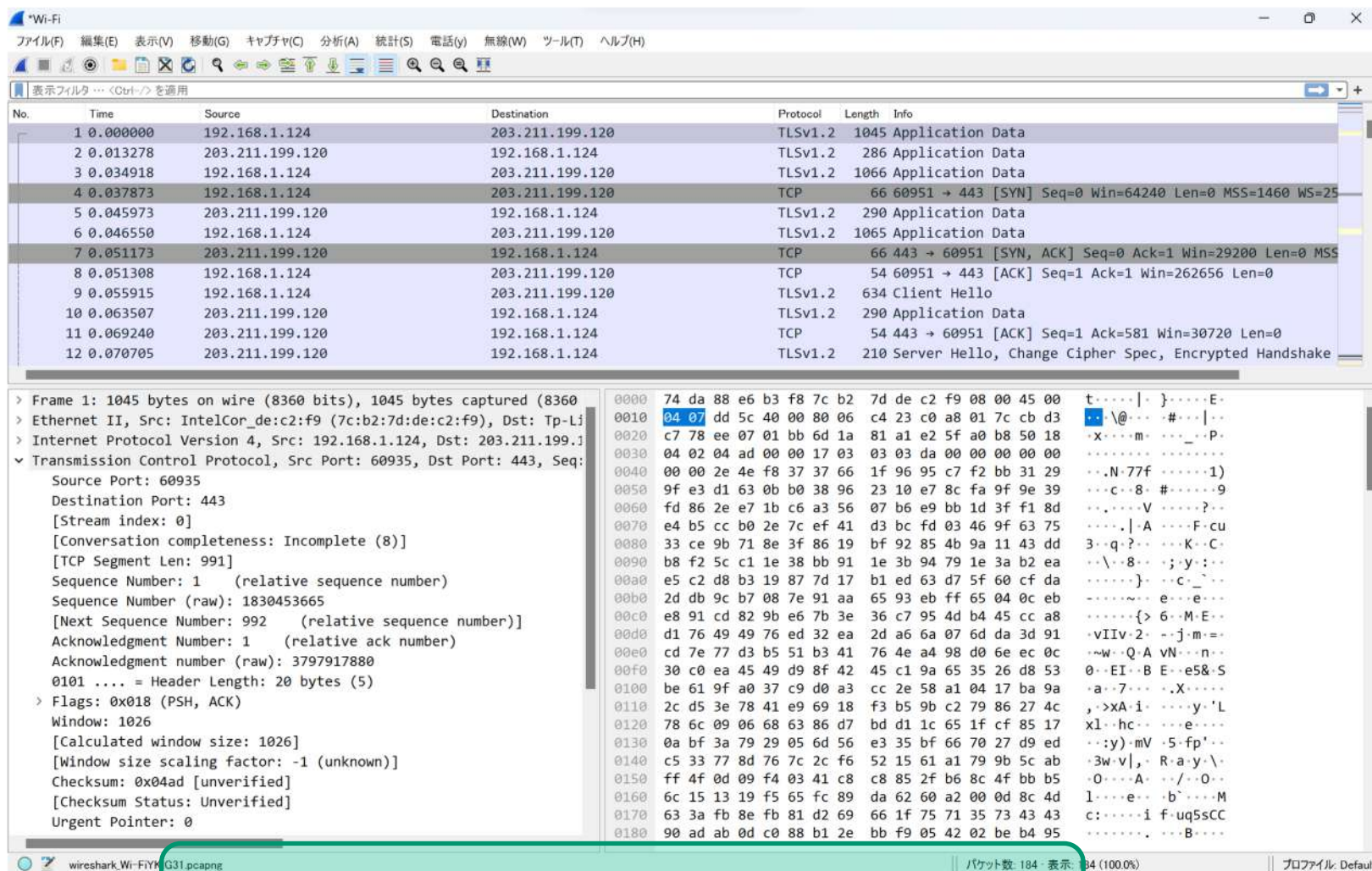
tcpdump = パケットキャプチャコマンド

```
ubuntu@ubuntuserver:~$ sudo tcpdump -i enp0s3 icmp
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on enp0s3, link-type EN10MB (Ethernet), snapshot length 262144 bytes
05:44:10.792774 IP 192.168.1.140 > ubuntuserver2: ICMP echo request, id 1550, seq 1, length 64
05:44:10.792828 IP ubuntuserver2 > 192.168.1.140: ICMP echo reply, id 1550, seq 1, length 64
05:44:11.794747 IP 192.168.1.140 > ubuntuserver2: ICMP echo request, id 1550, seq 2, length 64
05:44:11.794780 IP ubuntuserver2 > 192.168.1.140: ICMP echo reply, id 1550, seq 2, length 64
^C
4 packets captured
4 packets received by filter
0 packets dropped by kernel
```

フィルターを活用しないと大量のログが出力されるので注意  
解析には通信内容に関する知識(例: ICMPやhttpのやりとりやアプリケーション通信が  
どのようなものか、TCP/UDP・IP・ARPなど)が必須



wireshark = GUIのネットワークキャプチャツール



tcpdumpを利用するのと同様の知識などは必須



nmap = 対象IPアドレスで公開されているポートの一覧確認など

```
ubuntu@ubuntuserver:~$ nmap 192.168.1.140
```

```
Starting Nmap 7.80 ( https://nmap.org ) at 2023-04-17 05:40 UTC
```

```
Nmap scan report for 192.168.1.140
```

```
Host is up (0.0040s latency).
```

```
Not shown: 984 closed ports
```

```
PORT      STATE SERVICE
```

```
21/tcp    open  ftp
```

```
22/tcp    open  ssh
```

```
25/tcp    open  smtp
```

```
53/tcp    open  domain
```

```
80/tcp    open  http
```

```
110/tcp   open  pop3
```

```
111/tcp   open  rpcbind
```

```
143/tcp   open  imap
```

```
465/tcp   open  smtps
```

```
587/tcp   open  submission
```

```
993/tcp   open  imaps
```

```
995/tcp   open  pop3s
```

```
2030/tcp  open  device2
```

```
3306/tcp  open  mysql
```

```
8009/tcp  open  ajp13
```

```
8080/tcp  open  http-proxy
```

同一NW内で通信可能なIPアドレス一覧検索など便利  
システムに侵入した攻撃者がサーバ情報を調べることに使ったり



ss/netstat = 現在通信が行われているエントリーを表示

```
[root@localhost ~]# ss -tanp
```

State	Recv-Q	Send-Q	Local Address:Port	Peer Address:Port
.	.	.	.	.
LISTEN	0	100	*:465	*:*
users:((	"master",pid=1278,fd=22))			
ESTAB	0	0	192.168.1.140:22	192.168.1.124:60596
users:((	"sshd",pid=1387,fd=3))			
LISTEN	0	10	:::1:53	:::*
users:((	"named",pid=937,fd=23))			
LISTEN	0	9	:::21	:::*
users:((	"pure-ftpd",pid=922,fd=5))			
LISTEN	0	128	:::22	:::*
users:((	"sshd",pid=866,fd=4))			
LISTEN	0	100	:::25	:::*
users:((	"master",pid=1278,fd=14))			
.	.	.	.	.
users:((	"httpd",pid=947,fd=4),("httpd",pid=946,fd=4),("httpd",pid=945,fd=4),("httpd",pid=944,fd=4))			
LISTEN	0	100	:::465	:::*
users:((	"master",pid=1278,fd=23))			



## 2.03.3 ネットワークの問題解決





## 2.03.3 ネットワークの問題解決

重要度 3

### 概要

一般的なネットワーク設定に関する問題を特定して解決できる。  
これには、基本的な設定ファイルの位置とコマンドに関する知識も含まれる。

### 詳細

ネットワークの設定に関する情報を取得する。

hostname, /etc/hostname, /etc/hosts, /etc/resolv.conf, nmcli, ip

ネットワークの通信経路の問題を特定して解決する。

traceroute, traceroute6, ip, route, mtr

ハードウェアの認識と利用に関する情報を取得する。

dmesg, /var/log/syslogおよび/var/log/messagesなどのシステムのログファイルおよび systemd のジャーナル  
システムの初期化ファイルとその内容 (systemd)

NetworkManagerおよびそれがネットワーク設定に及ぼす影響について知っている。

/etc/network/, /etc/sysconfig/network-scripts/



hostname、 /etc/hostname = 自身のホスト名を設定、表示

```
ubuntu@ubuntuserver:~$ man hostname
```

```
HOSTNAME(1)                                Linux Programmer's Manual
HOSTNAME(1)
```

## NAME

hostname - show or set the system's host name  
domainname - show or set the system's NIS/YP domain name  
ypdomainname - show or set the system's NIS/YP domain name  
nisdomainname - show or set the system's NIS/YP domain name  
dnsdomainname - show the system's DNS domain name

```
ubuntu@ubuntuserver:~$ hostname  
ubuntuserver
```

```
ubuntu@ubuntuserver:~$ cat /etc/hostname  
ubuntuserver
```



/etc/hosts = IPアドレスとホスト名の組み合わせを記載し、静的な名前解決に使われる

```
kubeuser@master01:~$ cat /etc/hosts
```

```
127.0.0.1 localhost
127.0.1.1 master01
```

```
192.168.1.41 master01
192.168.1.45 worker01
192.168.1.46 worker02
```

```
240f:32:57b8:1:5054:ff:fe8e:5428 master01
240f:32:57b8:1:5054:ff:fe93:acfc worker01
240f:32:57b8:1:5054:ff:fe9e:4f00 worker02
```

```
fe80::5054:ff:fe8e:5428 master01
fe80::5054:ff:fe93:acfc worker01
fe80::5054:ff:fe9e:4f00 worker02
```

```
# The following lines are desirable for IPv6 capable hosts
::1    ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
```

“IPアドレス ホスト名”  
という記載方法



/etc/resolv.conf = 名前解決に使うDNSサーバを記載する

```
ubuntu@ubuntuserver:~$ cat /etc/resolv.conf
# This is /run/systemd/resolve/stub-resolv.conf managed by man:systemd-resolved(8).
# Do not edit.
#
# This file might be symlinked as /etc/resolv.conf. If you're looking at
# /etc/resolv.conf and seeing this text, you have followed the symlink.
#
# This is a dynamic resolv.conf file for connecting local clients to the
# internal DNS stub resolver of systemd-resolved. This file lists all
# configured search domains.
#
# Run "resolvectl status" to see details about the uplink DNS servers
# currently in use.
#
# Third party programs should typically not access this file directly, but only
# through the symlink at /etc/resolv.conf. To manage man:resolv.conf(5) in a
# different way, replace this symlink by a static file or a different symlink.
#
# See man:systemd-resolved.service(8) for details about the supported modes of
# operation for /etc/resolv.conf.
```

その他、  
**nslookup/dig/host**など  
名前解決に利用するコマンドも併せて覚えておくとよりよい

**nameserver 127.0.0.53**

options edns0 trust-ad  
search



traceroute/traceroute6 = 宛先に到達するまでのルータとその到達性を確認する

```
ubuntu@ubuntuserver:~$ traceroute 8.8.8.8
traceroute to 8.8.8.8 (8.8.8.8), 30 hops max, 60 byte packets
 1 _gateway (192.168.1.1) 2.790 ms 3.779 ms 4.229 ms
 2 192.168.0.1 (192.168.0.1) 8.498 ms 8.418 ms 9.704 ms
 .
 .
 .
 8 * * *
 9 dns.google (8.8.8.8) 7.632 ms 7.627 ms 7.621 ms
```

ルーティング経路の  
どこでNGとなっているか  
確認できる

```
kubeuser@master01:~$ traceroute6 2001:4860:4860::8888
traceroute to 2001:4860:4860::8888 (2001:4860:4860::8888), 30 hops max, 80 byte packets
 1 240f:32:57b8:1:6ee4:daff:fedd:7400 (240f:32:57b8:1:6ee4:daff:fedd:7400) 0.835 ms 0.860 ms
 0.858 ms
 2 2001:268:de13::62 (2001:268:de13::62) 4.504 ms 4.386 ms 4.362 ms
 .
 .
 .
 7 2404:6800:800a::1 (2404:6800:800a::1) 4.016 ms 2404:6800:8102::1 (2404:6800:8102::1)
 4.617 ms 2404:6800:80c3::1 (2404:6800:80c3::1) 5.099 ms
 8 dns.google (2001:4860:4860::8888) 3.956 ms 3.872 ms 3.793 ms
```



dmesg = Linux起動時のメッセージを表示

```
ubuntu@ubuntuserver:~$ sudo dmesg
```

```
[sudo] password for ubuntu:
```

```
[ 0.000000] Linux version 5.15.0-69-generic (buildd@lcy02-amd64-080) (gcc (Ubuntu 11.3.0-1ubuntu1~22.04) 11.3.0, GNU ld (GNU Binutils for Ubuntu) 2.38) #76-Ubuntu SMP Fri Mar 17 17:19:29 UTC 2023 (Ubuntu 5.15.0-69.76-generic 5.15.87)
```

```
[ 0.000000] Command line: BOOT_IMAGE=/vmlinuz-5.15.0-69-generic root=/dev/mapper/ubuntu--vg-ubuntu--lv ro
```

```
.
```

```
.
```

```
.
```

```
[ 4.229262] e1000: Intel(R) PRO/1000 Network Driver
```

```
[ 4.229964] e1000: Copyright (c) 1999-2006 Intel Corporation.
```

```
.
```

```
.
```

```
.
```

```
[ 5.577426] e1000 0000:00:03.0 eth0: (PCI:33MHz:32-bit) 08:00:27:16:5f:ac
```

```
[ 5.577458] e1000 0000:00:03.0 eth0: Intel(R) PRO/1000 Network Connection
```

```
[ 5.579397] e1000 0000:00:03.0 enp0s3: renamed from eth0
```

```
.
```

```
.
```

```
.
```

```
[ 9.902019] e1000: enp0s3 NIC Link is Up 1000 Mbps Full Duplex, Flow Control: RX
```

```
[ 9.913040] IPv6: ADDRCONF(NETDEV_CHANGE): enp0s3: link becomes ready
```

```
.
```

```
.
```

左記のように  
正常であればNIC関連のログが  
出力されているはず





## /var/log/messagesなどのシステムのログファイル

```
[root@localhost ~]# tail -f /var/log/messages
```

```
Apr 17 14:26:02 localhost server: 情報: Server startup in 15122 ms
```

```
Apr 17 14:26:07 localhost dhclient[911]: XMT: Info-Request on enp0s3, interval 27510ms.
```

```
Apr 17 14:26:22 localhost NetworkManager[605]: <warn> [1681709182.3445] dhcp6 (enp0s3):  
request timed out
```

```
Apr 17 14:26:22 localhost NetworkManager[605]: <info> [1681709182.3446] dhcp6 (enp0s3): state  
changed unknown -> timeout
```

```
Apr 17 14:26:22 localhost NetworkManager[605]: <info> [1681709182.3483] dhcp6 (enp0s3):  
canceled DHCP transaction, DHCP client pid 911
```

```
Apr 17 14:26:22 localhost NetworkManager[605]: <info> [1681709182.3484] dhcp6 (enp0s3): state  
changed timeout -> done
```

```
Apr 17 14:26:46 localhost chronyd[625]: Selected source 162.159.200.123
```

```
Apr 17 14:26:48 localhost systemd: Created slice User Slice of root.
```

```
Apr 17 14:26:48 localhost systemd: Started Session 1 of user root.
```

```
Apr 17 14:26:48 localhost systemd-logind: New session 1 of user root.
```

問題のあるプログラムがちゃんと起動しているかなどを  
“tail -f”(リアルタイムでログに吐き出される内容を確認できる)を  
使って確認できる。

各プログラムの詳細ログは、/var/logなどにあるログファイルを参照する。



## systemd のジャーナル

```
[root@localhost ~]# journalctl -u sshd
-- Logs begin at 月 2023-04-17 14:25:23 JST, end at 月 2023-04-17 14:30:01 JST. --
4月 17 14:25:34 localhost.localdomain systemd[1]: Starting OpenSSH server daemon...
4月 17 14:25:34 localhost.localdomain sshd[866]: Server listening on 0.0.0.0 port 22.
4月 17 14:25:34 localhost.localdomain sshd[866]: Server listening on :: port 22.
4月 17 14:25:34 localhost.localdomain systemd[1]: Started OpenSSH server daemon.
4月 17 14:26:48 localhost.localdomain sshd[1387]: Accepted password for root from 192.168.1.124
port 60596 ssh2
```

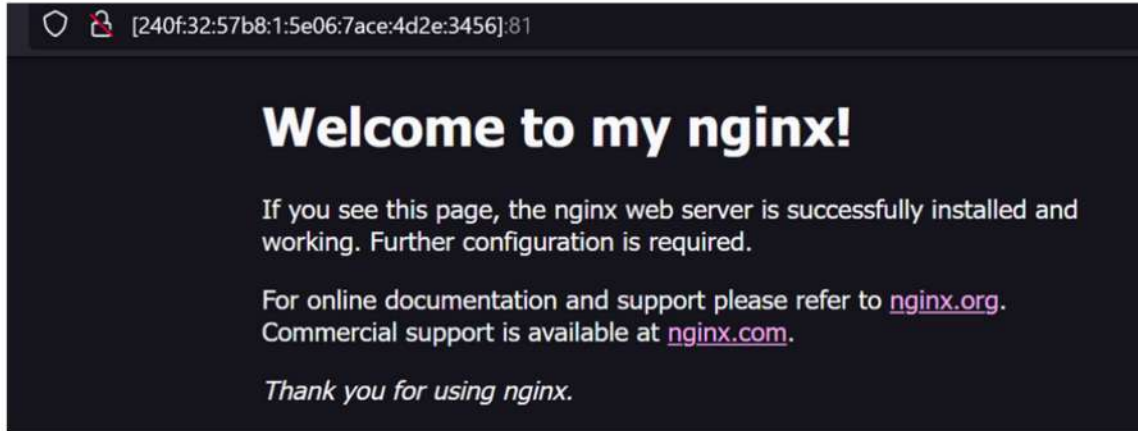
**systemdのログを参照すると、  
プログラムがうまく起動していない原因が分かるかも。**



# Appendix



## IPv6アドレスを使ってWebサイトにアクセスする



firefoxからアクセスしてますが、**`http://[240f:32:57b8:1:5e06:7ace:4d2e:3456]:81`**  
/というようにURLを書きます。

<https://www.opensourcetechnology.com/entry/20230131/1675164136>



## LinuCLレベル2 201試験 主題2.03の例題と解説

[https://linuc.org/study/samples/index/s/2\\_03/](https://linuc.org/study/samples/index/s/2_03/)



 **LinuCL** open your NEXT future  
LPI - JAPAN

資格・試験概要 ▾ 受験案内 受験申込み 学習の

例題と解説 ▾ / LinuCLレベル2 201試験 / 主題2.03

LinuCLレベル2 201試験  
主題2.03の例題と解説

いいね! 0 シェアする ツイート B! 0

前へ 1 2 次へ

### 2.03.2 高度なネットワーク構成 >

LinuCLレベル2 201試験の出題範囲から「2.03.2 高度なネットワーク構成」についての例題を解いてみます。

このテーマは、ネットワークの状態を把握・設定するための内容が含まれます。実践でのサーバーではサービス提供用のNIC、管理用のNICなど複数のインターフェイスを使い分ける必要があるため、このテーマで学んだ内容を活かすことができるのでしっかりと理解しておきましょう。

2023年03月16日



# お知らせ





## Kubernetes(LFS458-JP)の研修、はじめました！

LFS458-JP

# kubernetes 研修

**Kubernetes** クラスタを 構築・管理するためのコースです。  
また、Kubernetesを 管理するために  
必要なスキルを 身に付けることができます。

大規模    どこでも実行    柔軟

おすすめ！

### パッケージプラン

Kubernetesに関する  
知識・技術・CKA取得を  
目的としたプランです。

LFS250-JP

+

LFS458-JP

+

CKA

464,500 円(税込)

＞ お申し込み

Kubernetesに関する  
知識・技術の取得を  
目的としたプランです。

LFS250-JP

+

LFS458-JP

409,500 円(税込)

＞ お申し込み

Kubernetesに関する  
CKA取得を  
目的としたプランです。

LFS458-JP

+

CKA

451,000 円(税込)

＞ お申し込み

<https://www.it-training.tokyo/kubernetes/>





- Linuxの基本的なネットワーク設定を理解する
- ルーティングテーブルの操作を理解する
- ネットワーク関連のツールの使用方法を理解する
- ネットワーク関連のトラブルシュート方法を理解する

**IPv4だけでなく、IPv6についても理解しておく必要がある**

**構築段階だけでなく、トラブルシュート時に非常に重要となってくる**

**レイヤー(1~7)やアプリケーション通信に関する理解も必要**



# Q & A



*Thank you for joining my seminar!*



<https://www.opensourcetek.tokyo/>



[https://twitter.com/matt\\_zeus](https://twitter.com/matt_zeus)



<https://www.facebook.com/takahiro.kujirai.1>

