LinuC レベル2
技術解説セミナー

DNSの役割を理解しよう！

2022/06/05（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

Linuxとの出会い

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、
Opensourceteh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。


Linus Torvaldsさん(Linux開発者)


Igor Sysoevさん(nginx開発者)


Alexei Vladishevさん(Zabbix開発者)

ゼウス・エンタープライズからのお知らせ

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を基礎としたエンジニア、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荷を大幅に削減して、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

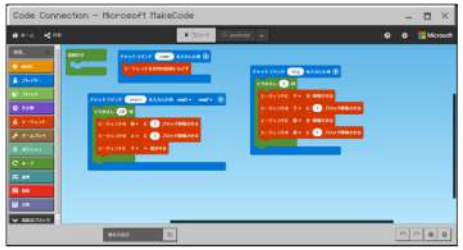
銀座ほこてん
子供プログラミング教室

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 全員月謝がずっと %OFF!

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

ゼウス・エンタープライズからのお知らせ

貸し会議室

＜使用例＞

https://www.zeus-enterprise.co.jp/ikebukuro_office/

https://ginza168.tokyo/

アジェンダ

ZEUS enterprise

1. LinuCについて
   - 試験概要と特徴

2. 技術解説
   DNSの役割を理解しよう!
   主題2.08：ドメインネームサーバー
   - 2.08.1BIND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

3. Appendix

4. Q&A

LinuC

本日のゴール

ZEUS enterprise

- DNSの役割を理解する
- BINDの設定方法を理解する
- BINDのゾーンファイル書式を理解する

LinuC

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

クラウドを活用できるITエンジニアに必須の技術がまとまっている

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

【今まで／その他】

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

LinuC Version10.0

LinuC Level1試験

ZEUS enterprise

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
- 1.01.1Linux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
- 1.04.1apt コマンドによるパッケージ管理
- 1.04.2Debianパッケージ管理
- 1.04.3yumコマンドによるパッケージ管理
- 1.04.4RPM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

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

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

LinuC

LinuC Level2試験

ZEUS enterprise

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

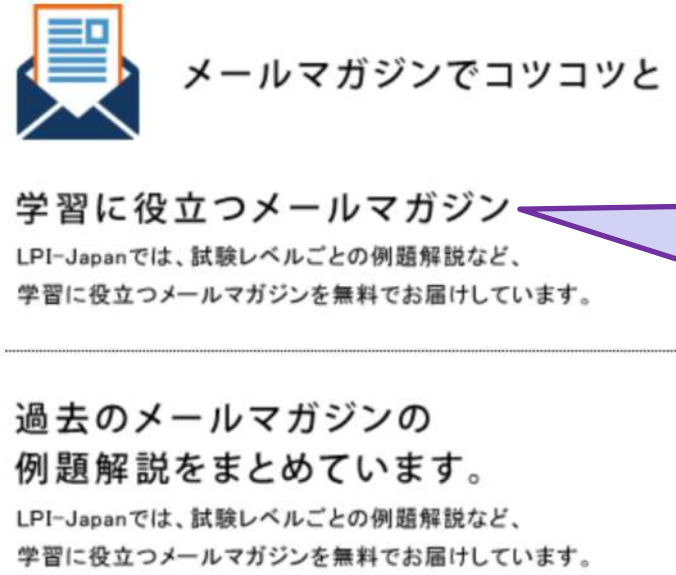
- 2.07：ネットワーククライアントの管理
  - 2.07.1 DHCPサーバーの設定と管理
  - 2.07.2 PAM認証
  - 2.07.3 LDAPクライアントの利用方法
  - 2.07.4 OpenLDAPサーバーの設定
- 2.08：ドメインネームサーバー
  - 2.08.1 BINDの設定と管理
  - 2.08.2 ゾーン情報の管理
  - 2.08.3 セキュアなDNSサーバーの実現
- 2.09：HTTPサーバーとプロキシサーバー
  - 2.09.1 Apache HTTPサーバーの設定と管理
  - 2.09.2 OpenSSLとHTTPSの設定
  - 2.09.3 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

https://linuc.org/linuc2/range/201.html
https://linuc.org/linuc2/range/202.html

LC LinuC

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 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！

④過去セミナーの動画

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

LinuC 学習のコツ

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(LinuC HP)

●月頃、受験しようという目標を立てる

↓

LinuC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

↓

LinuC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

↓

LinuC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

↓

LinuC認定教材 3週目

※弱点の補強

2ヶ月

↓

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

2.5ヶ月

↓

完全に理解した！(100%)

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

DNSの役割を理解しよう！
（主題2.08：ドメインネームサーバー）

2.08.1 BINDの設定と管理

重要度　　3

概要
　権威サーバー、再帰サーバー、キャッシュ専用DNSサーバーとして機能するようにBINDを設定できる。
　これには、稼働中のサーバーを管理すること、ログの設定も含まれる。

詳細
　BIND の設定ファイル、用語、ユーティリティ
　　named.conf, host, dig, nslookup
　BIND の設定ファイルで、BINDゾーンファイルの位置を定義する。
　　named.conf
　変更した設定ファイルおよびゾーンファイルの再読込
　　rndc, named-checkconf
　代替ネームサーバーとしての dnsmasq, Unbound, NSD, PowerDNS について知っている。

2.08.2 ゾーン情報の管理

重要度　2

概要

正引き、逆引きのゾーンファイルおよびルートヒントファイルを作成できる。
これには、レコードに適切な値を設定すること、ホストをゾーンに追加すること、ゾーンをDNSに追加することも含まれる。
また、他のDNSサーバーにゾーンの委任を行うことも含まれる。

詳細

BINDゾーンファイルのレイアウト、内容、ファイル配置
　ゾーンファイルの書式, リソースレコードの書式
逆引きゾーンを含む、ゾーンファイルに新しいホストを追加する際の確認方法
　named-compilezone, named-checkzone

2.08.3 セキュアなDNSサーバーの実現

重要度　2

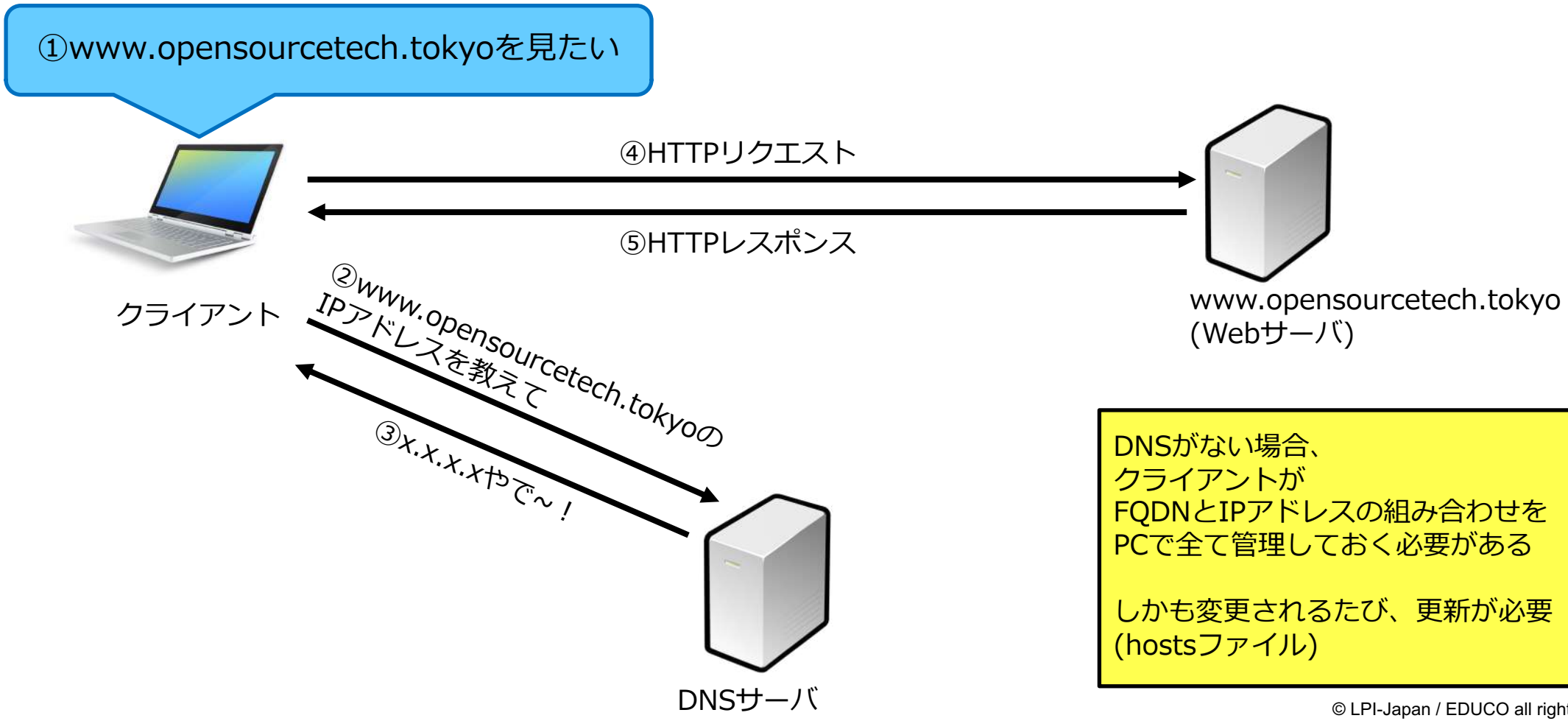
概要
DNSサーバーをroot以外のユーザとしてchroot 環境で実行するよう設定できる。
これには、DNSサーバー間で安全なデータ交換を行うことも含まれる。

詳細
chroot 環境で稼働するようBINDを設定する。
forwarders文を使用してBINDの構成を分割する。
named.conf
DNSSEC および基本的なツールについて知っている。
dnssec-keygen, dnssec-signzone, TSIG(Transaction Signature)
DANE および関連レコードについて知っている。

DNSの役割

Domain Name System

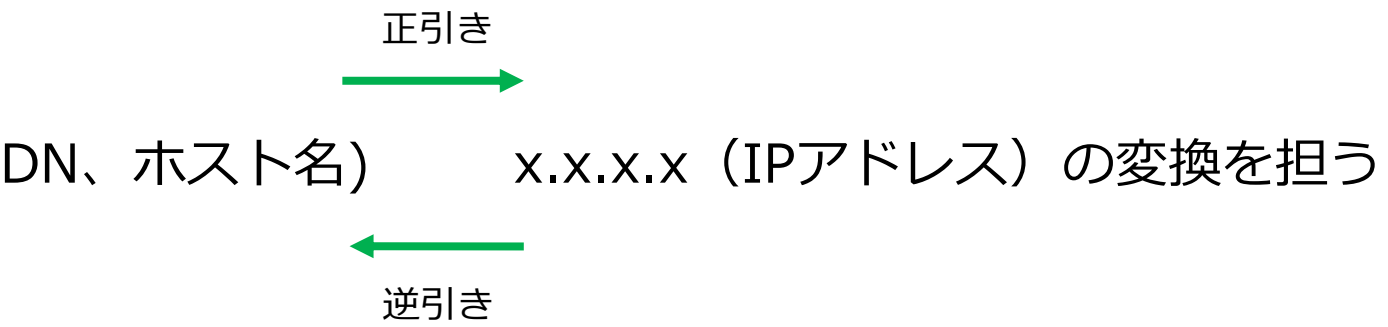
www.opensourcetech.tokyo (FQDN、ホスト名) ⇔ x.x.x.x（IPアドレス）の変換を担う



DNSがない場合、
クライアントが
FQDNとIPアドレスの組み合わせを
PCで全て管理しておく必要がある

しかも変更されるたび、更新が必要
(hostsファイル)

名前解決の種類



※FQDN：*Fully Qualified Domain Name(*完全修飾ドメイン名*)*

hostsファイル（/etc/hosts）

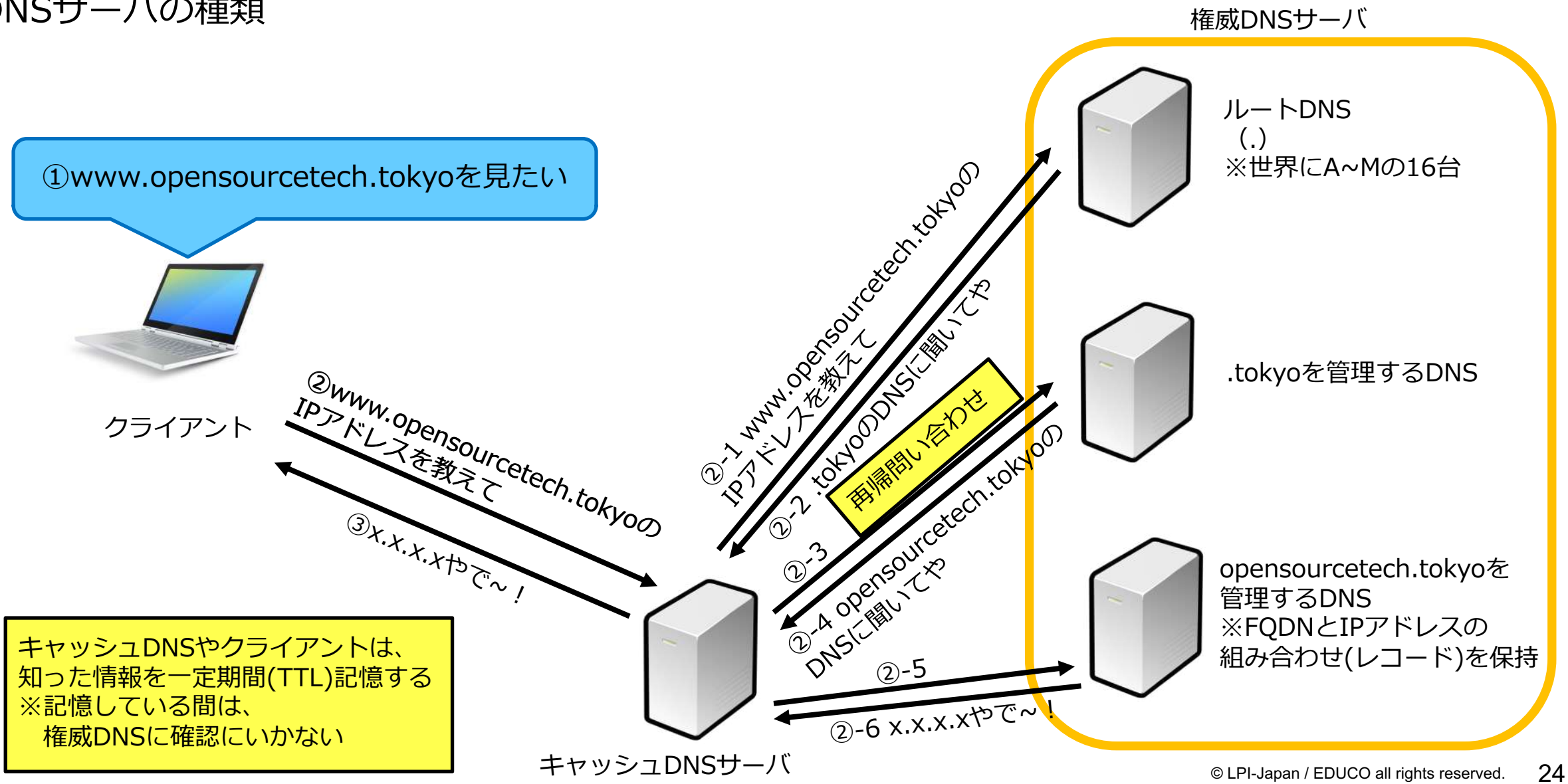
IPアドレスとFQDNの組み合わせを記載するローカルファイル

```
ubuntu@linucserver:~$ cat /etc/hosts
127.0.0.1 localhost
127.0.1.1 linucserver

# 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
```

※Windowsの場合、C:¥Windows¥System32¥drivers¥etc に格納されており、更新には管理者権限が必要

DNSサーバの種類



キャッシュDNSやクライアントは、
知った情報を一定期間(TTL)記憶する
※記憶している間は、
　権威DNSに確認にいかない

BINDのDNSキャッシュ情報参照

ubuntu@linucserver:/var/cache/bind$ sudo rndc dumpdb　→ /var/cache/bindにnamed_dump.dbというファイルに出力している

```
ubuntu@linucserver:/var/cache/bind$ ls
managed-keys.bind  managed-keys.bind.jnl named_dump.db
```

ubuntu@linucserver:/var/cache/bind$ cat named_dump.db → 出力されたキャッシュを確認

```
;; Start view _default
;;; Cache dump of view '_default' (cache _default)
;; using a 604800 second stale ttl$DATE 20220522144043
; secure.                       1123193 IN NS   a.root-servers.net.
                                1123193 IN NS   b.root-servers.net.
                                1123193 IN NS   c.root-servers.net.
                                1123193 IN NS   d.root-servers.net.
                                1123193 IN NS   e.root-servers.net.
                                1123193 IN NS   f.root-servers.net.
                                1123193 IN NS   g.root-servers.net.
                                1123193 IN NS   h.root-servers.net.
                                1123193 IN NS   i.root-servers.net.
                                1123193 IN NS   j.root-servers.net.
                                1123193 IN NS   k.root-servers.net.
                                1123193 IN NS   l.root-servers.net.
                                1123193 IN NS   m.root-servers.net
                                以下省略
```

参考：WindowsクライアントのDNSキャッシュ情報参照

```
C:¥Users>ipconfig /displaydns

Windows IP 構成

    www.google.co.jp
    ----------------------------------------
    レコード名 . . . . . . . : www.google.co.jp
    レコードの種類 . . . . . : 28
    Time To Live  . . . . . .: 293
    データの長さ . . . . . . : 16
    セクション . . . . . . . : 回答
    AAAA レコード. . . . . . : 2404:6800:4004:80c::2003


    www.google.co.jp
    ----------------------------------------
    レコード名 . . . . . . . : www.google.co.jp
    レコードの種類 . . . . . : 1
    Time To Live  . . . . . .: 239
    データの長さ . . . . . . : 4
    セクション . . . . . . . : 回答
    A (ホスト) レコード. . . : 172.217.175.67
```

ルートDNSサーバ

図3 各ルートサーバの運用組織と所在地

https://www.nic.ad.jp/ja/newsletter/No45/0800.html

```
ubuntu@linucserver:~$ cat /usr/share/dns/root.hints
.                       3600000     NS    A.ROOT-SERVERS.NET.
A.ROOT-SERVERS.NET.     3600000     A     198.41.0.4
A.ROOT-SERVERS.NET.     3600000     AAAA
2001:503:ba3e::2:30
;
; FORMERLY NS1.ISI.EDU
;
.                       3600000     NS    B.ROOT-SERVERS.NET.
B.ROOT-SERVERS.NET.     3600000     A     199.9.14.201
B.ROOT-SERVERS.NET.     3600000     AAAA  2001:500:200::b
;
; FORMERLY C.PSI.NET

(省略)

.                       3600000     NS    M.ROOT-SERVERS.NET.
M.ROOT-SERVERS.NET.     3600000     A     202.12.27.33
M.ROOT-SERVERS.NET.     3600000     AAAA  2001:dc3::35
```

BIND

OSSのDNSサーバ

https://www.isc.org/bind/

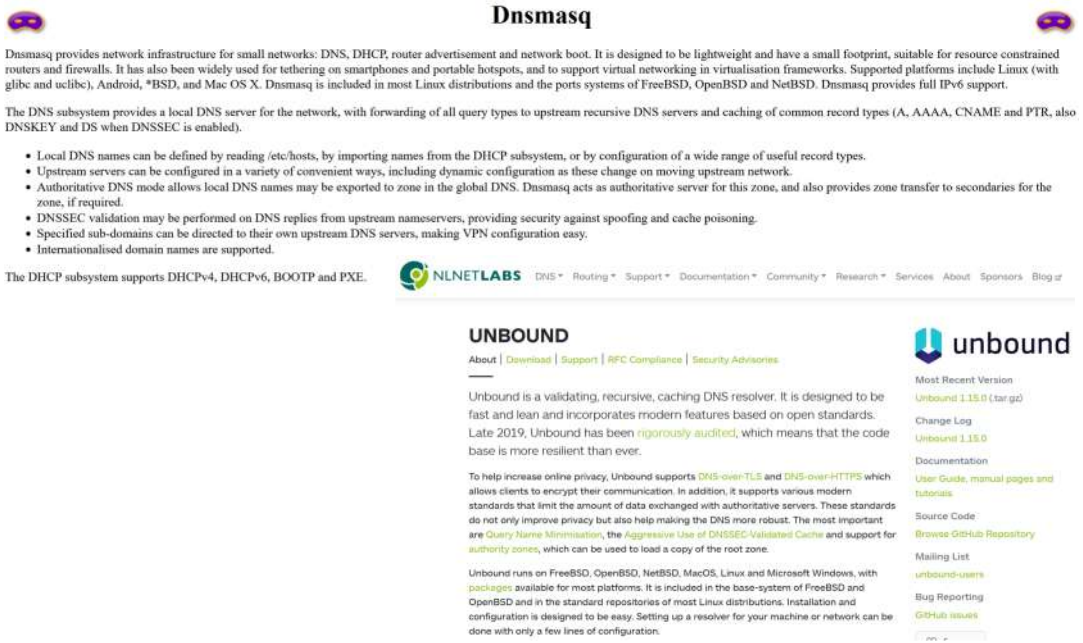
https://jprs.jp/tech/

重要度が高いサーバなので、
脆弱性情報を見逃さないようにするべし

DNSの役割

その他のDNSサーバ

dnsmasq, Unbound, NSD, PowerDNS



Dnsmasq

Dnsmasq provides network infrastructure for small networks: DNS, DHCP, router advertisement and network boot. It is designed to be lightweight and have a small footprint, suitable for resource constrained routers and firewalls. It has also been widely used for tethering on smartphones and portable hotspots, and to support virtual networking in virtualisation frameworks. Supported platforms include Linux (with glibc and uclibc), Android, *BSD, and Mac OS X. Dnsmasq is included in most Linux distributions and the ports systems of FreeBSD, OpenBSD and NetBSD. Dnsmasq provides full IPv6 support.

The DNS subsystem provides a local DNS server for the network, with forwarding of all query types to upstream recursive DNS servers and caching of common record types (A, AAAA, CNAME and PTR, also DNSKEY and DS when DNSSEC is enabled).

- Local DNS names can be defined by reading /etc/hosts, by importing names from the DHCP subsystem, or by configuration of a wide range of useful record types.
- Upstream servers can be configured in a variety of convenient ways, including dynamic configuration as these change on moving upstream network.
- Authoritative DNS mode allows local DNS names may be exported to zone in the global DNS. Dnsmasq acts as authoritative server for this zone, and also provides zone transfer to secondaries for the zone, if required.
- DNSSEC validation may be performed on DNS replies from upstream nameservers, providing security against spoofing and cache poisoning.
- Specified sub-domains can be directed to their own upstream DNS servers, making VPN configuration easy.
- Internationalised domain names are supported.

The DHCP subsystem supports DHCPv4, DHCPv6, BOOTP and PXE.

NLNETLABS DNS Routing Support Documentation Community Research Services About Sponsors Blog

UNBOUND

About | Download | Support | RFC Compliance | Security Advisories

unbound

Most Recent Version

Unbound 1.15.0 (tar.gz)

Change Log

Unbound 1.15.0

Documentation

User Guide, manual pages and tutorials

Source Code

Browse GitHub Repository

Mailing List

unbound-users

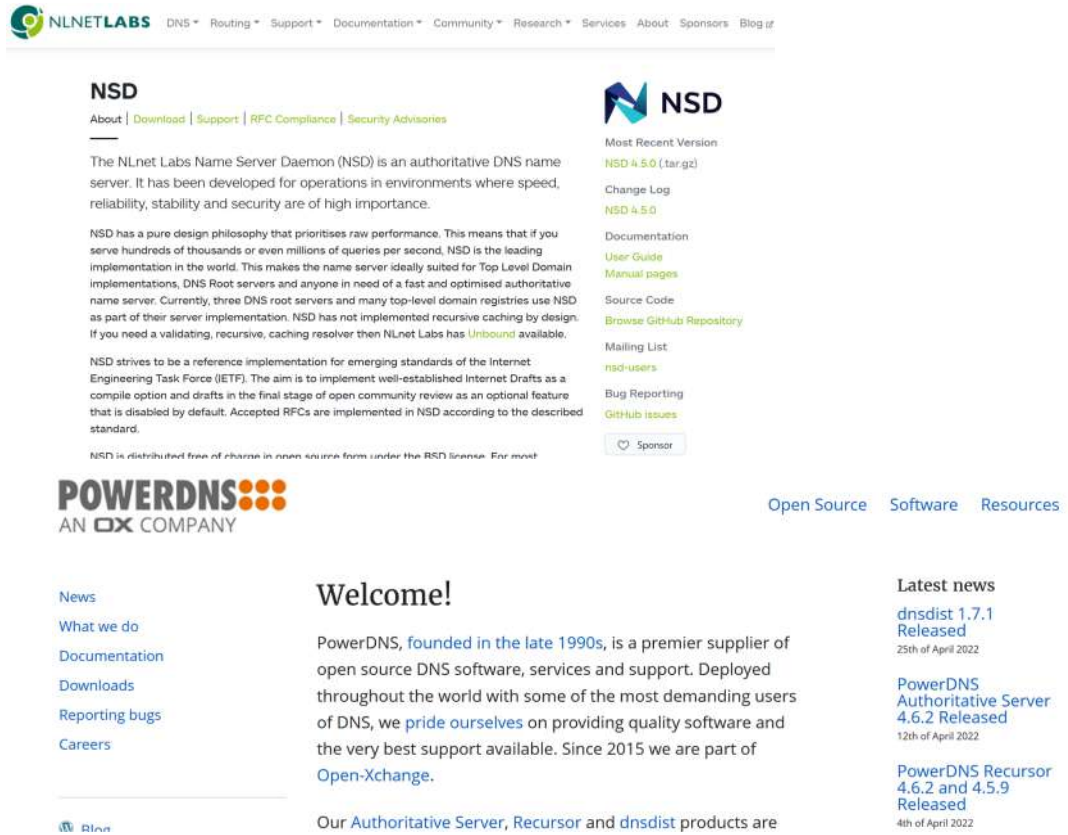
Bug Reporting

GitHub issues

Unbound is a validating, recursive, caching DNS resolver. It is designed to be fast and lean and incorporates modern features based on open standards. Late 2019, Unbound has been rigorously audited, which means that the code base is more resilient than ever.

To help increase online privacy, Unbound supports DNS-over-TLS and DNS-over-HTTPS which allows clients to encrypt their communication. In addition, it supports various modern standards that limit the amount of data exchanged with authoritative servers. These standards do not only improve privacy but also help making the DNS more robust. The most important are Query Name Minimisation, the Aggressive Use of DNSSEC-Validated Cache and support for authority zones, which can be used to load a copy of the root zone.

Unbound runs on FreeBSD, OpenBSD, NetBSD, MacOS, Linux and Microsoft Windows, with packages available for most platforms. It is included in the base-system of FreeBSD and OpenBSD and in the standard repositories of most Linux distributions. Installation and configuration is designed to be easy. Setting up a resolver for your machine or network can be done with only a few lines of configuration.



NLNETLABS DNS Routing Support Documentation Community Research Services About Sponsors Blog

NSD

About | Download | Support | RFC Compliance | Security Advisories

The NLnet Labs Name Server Daemon (NSD) is an authoritative DNS name server. It has been developed for operations in environments where speed, reliability, stability and security are of high importance.

NSD has a pure design philosophy that prioritises raw performance. This means that if you serve hundreds of thousands or even millions of queries per second, NSD is the leading implementation in the world. This makes the name server ideally suited for Top Level Domain implementations, DNS Root servers and anyone in need of a fast and optimised authoritative name server. Currently, three DNS root servers and many top-level domain registries use NSD as part of their server implementation. NSD has not implemented recursive caching by design. If you need a validating, recursive, caching resolver then NLnet Labs has Unbound available.

NSD strives to be a reference implementation for emerging standards of the Internet Engineering Task Force (IETF). The aim is to implement well-established Internet Drafts as a compile option and drafts in the final stage of open community review as an optional feature that is disabled by default. Accepted RFCs are implemented in NSD according to the described standard.

NSD is distributed free of charge in open source form under the BSD license. For most

NSD

Most Recent Version

NSD 4.5.0 (tar.gz)

Change Log

NSD 4.5.0

Documentation

User Guide

Manual pages

Source Code

Browse GitHub Repository

Mailing List

nsd-users

Bug Reporting

GitHub issues

Sponsor

POWERDNS AN OX COMPANY

Open Source Software Resources

News

What we do

Documentation

Downloads

Reporting bugs

Careers

Blog

Welcome!

PowerDNS, founded in the late 1990s, is a premier supplier of open source DNS software, services and support. Deployed throughout the world with some of the most demanding users of DNS, we pride ourselves on providing quality software and the very best support available. Since 2015 we are part of Open-Xchange.

Our Authoritative Server, Recursor and dnsdist products are

Latest news

dnsdist 1.7.1 Released

25th of April 2022

PowerDNS Authoritative Server 4.6.2 Released

12th of April 2022

PowerDNS Recursor 4.6.2 and 4.5.9 Released

4th of April 2022

https://thekelleys.org.uk/dnsmasq/doc.html
https://nlnetlabs.nl/projects/unbound/about/
https://www.nlnetlabs.nl/projects/nsd/about/
https://www.powerdns.com/

BINDの設定

BINDの動作を制御する “named.conf”

レコード情報を保持する “ゾーンファイル” で構成される

```
include "/etc/bind/named.conf.options";
include "/etc/bind/named.conf.local";
include "/etc/bind/named.conf.default-zones";
include "/etc/bind/named.conf.my-zones";

options {
        directory "/var/cache/bind";

        dnssec-validation auto;

        listen-on-v6 { any; };
};

zone "opensourcetech.test" {
        type master;
        file "/etc/bind/test.zone";
};

zone "1.168.192.in-addr.arpa" {
        type master;
        file "/etc/bind/test.rev";
};
```

test.zoneの内容

```
$ORIGIN opensourcetech.test.
$TTL    604800
@       IN      SOA     dns.opensourcetech.test.
root.opensourcetech.test. (
                        2022051501      ; Serial
                        604800          ; Refresh
                         86400          ; Retry
                        2419200         ; Expire
                        604800 )        ; Negative Cache TTL
;
        IN      NS      dns.opensourcetech.test.
        IN      MX 10   mail.opensourcetech.test.
dns     IN      A       192.168.1.247
www     IN      A       192.168.1.247
mail    IN      A       192.168.1.247
ftp     IN      A       192.168.1.247
smb     IN      A       192.168.1.247
```

named.conf

```
include "/etc/bind/named.conf.local";
include “/etc/bind/named.conf.default-zones”;          ・・・・・外部設定ファイルの参照先

options {
        directory “/var/cache/bind”;    ・・・・bindの作業ディレクトリ指定
        recursion yes;                          ・・・・再帰問い合わせの実施
        listen-on-v6 { any; };       ・・・・IPv6通信用のIPアドレス指定
};

zone “.” {                                  ・・・・ルートDNSサーバの情報
        type hint;
        file "/usr/share/dns/root.hints";
};

zone “opensourcetech.test” {    ・・・・ドメイン opensourcetech.test については、権威(master)であり、test.zone にレコード記載あり
        type master;
        file "/etc/bind/test.zone";
};

zone “1.168.192.in-addr.arpa” {    ・・・・ドメイン 192.168.1.0/24については、権威(master)であり、test.rev にレコード記載あり
        type master;
        file "/etc/bind/test.rev";
};
```

設定行は、”;(セミコロン)” を最後に付ける
ディレクティブは、”{” で始まり、”}” で閉じる
192.168.1.0/24の逆引きは、” 1.168.192.in-addr.arpa” と書く
など慣れないと記載ミスをしやすい

forwarders・・・自分で再帰問い合わせを行わず、別のDNSに依頼する

named.conf

```
include "/etc/bind/named.conf.local";
include “/etc/bind/named.conf.default-zones”;

options {
        directory “/var/cache/bind”;
        recursion no;                    ・・・・再帰問い合わせの実施
        forwarders { 192.168.2.254; 192.168.2.253; };
        listen-on-v6 { any; };
};

zone “opensourcetech.test” {
        type master;
        file "/etc/bind/test.zone";
};

zone “1.168.192.in-addr.arpa” {
        type master;
        file "/etc/bind/test.rev";
};
```

named-checkconf

設定ファイル(named.conf)の構文チェックツール

問題ない場合
ubuntu@linucserver:~$ named-checkconf /etc/bind/named.conf
ubuntu@linucserver:~$ 何も表示されない

ミスがある場合 ※設定の最後の”;”を忘れている場合
ubuntu@linucserver:~$ named-checkconf /etc/bind/named.conf
/etc/bind/named.conf:13: missing ';' before end of file

ゾーンファイル(正引き)

root@opensourcetech.testを表す

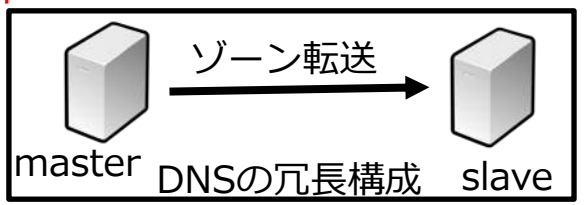
“opensourcetech.test”を意味する

```
$ORIGIN opensourcetech.test.
$TTL    604800      ・・・・問い合わせ時クライアントが取得したレコード情報の有効(キャッシュ)期間 Time To Live
@       IN      SOA     dns.opensourcetech.test. root.opensourcetech.test. (
                        2022051501      ; Serial
                        604800          ; Refresh
                        86400           ; Retry
                        2419200         ; Expire
                        604800 )        ; Negative Cache TTL
;
        IN      NS      dns.opensourcetech.test.
        IN      MX 10   mail.opensourcetech.test.
dns     IN      A       192.168.1.247
www     IN      A       192.168.1.247
mail    IN      A       192.168.1.247
ftp     IN      A       192.168.1.247
smb     IN      A       192.168.1.247
```

無い場合、
自動的に ”opensourcetech.test”を補完する

空白は、”@”が補完される

ゾーンファイルのシリアルナンバー
ゾーン情報更新のチェック間隔
ゾーン転送失敗時の待機時間
ゾーンを無効とみなすまでの時間
ネガティブキャッシュ(問い合わせレコードが存在しない)の生存時間



“www” の後に “opensourcetech.test” が補完される

パラメータや記載方法・レコードの種類など覚えることが多く、
慣れないと記載ミスをしやすい

レコードの種類
- SOA：ゾーンの管理情報
- A：IPv4正引き
- AAAA：IPv6正引き
- NS：ネームサーバ
- MX：メールサーバ ※優先度記載必須
- PTR：逆引き
- TXT：テキスト
- CNAME：別名用

ゾーンファイル(逆引き)

```
$ORIGIN 1.168.192.in-addr.arpa.
$TTL    604800
@       IN      SOA     dns.opensourcetech.test. root.opensourcetech.test. (
                        2022051501      ; Serial
                         604800         ; Refresh
                          86400         ; Retry
                        2419200         ; Expire
                         604800 )       ; Negative Cache TTL
;
        IN      NS      dns.opensourcetech.test.
247     IN      PTR     www.opensourcetech.test.
247     IN      PTR     dns.opensourcetech.test.
247     IN      PTR     mail.opensourcetech.test.
247     IN      PTR     ftp.opensourcetech.test.
247     IN      PTR     smb.opensourcetech.test.
```

“192.168.1.0/24” を意味する

無い場合、
自動的に ”opensourcetech.test”を補完する

IPアドレスのドメイン(1.168.192.in-addr.arpa)に含まれない
第4オクテットを記載

パラメータや記載方法に慣れていないと、
記載ミスをしやすい

named-checkzone

ゾーンファイルの構文に間違いがないかチェックするツール

正引きゾーンファイルのチェック
ubuntu@linucserver:~$ named-checkzone opensourcetech.test /etc/bind/test.zone
zone opensourcetech.test/IN: loaded serial 2022051501
OK

逆引きゾーンファイルのチェック
ubuntu@linucserver:~$ named-checkzone 1.168.192.in-addr.arpa /etc/bind/test.rev
zone 1.168.192.in-addr.arpa/IN: loaded serial 2022051501
OK

設定変更時は、プロセスを再起動や設定ファイル再読み込みが必要

```
ubuntu@linucserver:~$ sudo systemctl start named
ubuntu@linucserver:~$ sudo systemctl reload named
ubuntu@linucserver:~$ sudo systemctl restart named

ubuntu@linucserver:~$ sudo rndc reconfig
ubuntu@linucserver:~$ sudo rndc status
version: BIND 9.16.1-Ubuntu (Stable Release) <id:d497c32>
running on linucserver: Linux x86_64 5.4.0-109-generic #123-Ubuntu SMP Fri Apr 8 09:10:54 UTC 2022
boot time: Fri, 27 May 2022 17:31:45 GMT
last configured: Fri, 27 May 2022 17:31:55 GMT
configuration file: /etc/bind/named.conf
CPUs found: 2
worker threads: 2
UDP listeners per interface: 2
number of zones: 104 (97 automatic)
debug level: 0
xfers running: 0
xfers deferred: 0
soa queries in progress: 0
query logging is OFF
recursive clients: 0/900/1000
tcp clients: 0/150
TCP high-water: 0
server is up and running
```

DNSクライアントツール

nslookup

```
ubuntu@linucserver:/etc/bind$ nslookup
> server 192.168.1.247          ・・・・問い合わせをするDNSサーバ指定
Default server: 192.168.1.247
Address: 192.168.1.247#53

> www.opensourcetech.test          ・・・・正引き確認
Server:         192.168.1.247
Address:        192.168.1.247#53

Name:   www.opensourcetech.test
Address: 192.168.1.247

> 192.168.1.247                          ・・・・逆引き確認
247.1.168.192.in-addr.arpa      name = smb.opensourcetech.test.
247.1.168.192.in-addr.arpa      name = dns.opensourcetech.test.
247.1.168.192.in-addr.arpa      name = www.opensourcetech.test.
247.1.168.192.in-addr.arpa      name = mail.opensourcetech.test.
247.1.168.192.in-addr.arpa      name = ftp.opensourcetech.test.
```

host

```
ubuntu@linucserver:~$ host www.opensourcetech.test 192.168.1.247      ・・・・正引き確認
Using domain server:
Name: 192.168.1.247
Address: 192.168.1.247#53
Aliases:

www.opensourcetech.test has address 192.168.1.247

ubuntu@linucserver:~$ host 192.168.1.247 192.168.1.247        ・・・・逆引き確認
Using domain server:
Name: 192.168.1.247
Address: 192.168.1.247#53
Aliases:

247.1.168.192.in-addr.arpa domain name pointer mail.opensourcetech.test.
247.1.168.192.in-addr.arpa domain name pointer smb.opensourcetech.test.
247.1.168.192.in-addr.arpa domain name pointer www.opensourcetech.test.
247.1.168.192.in-addr.arpa domain name pointer dns.opensourcetech.test.
247.1.168.192.in-addr.arpa domain name pointer ftp.opensourcetech.test.
```

dig(正引き)

```
ubuntu@linucserver:~$ dig @192.168.1.247 www.opensourcetech.test

; <<>> DiG 9.16.1-Ubuntu <<>> @192.168.1.247 www.opensourcetech.test
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 59778
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: fbd650458f83221401000000062910ef8dd82414dcab408d5 (good)
;; QUESTION SECTION:
;www.opensourcetech.test.       IN      A          ・・・・問い合わせ内容

;; ANSWER SECTION:
www.opensourcetech.test. 604800 IN      A       192.168.1.247          ・・・・DNSサーバからの回答

;; Query time: 0 msec                                  ・・・・問い合わせに関する統計情報
;; SERVER: 192.168.1.247#53(192.168.1.247)
;; WHEN: Fri May 27 17:48:40 UTC 2022
;; MSG SIZE  rcvd: 96
```

dig(逆引き)

```
ubuntu@linucserver:~$ dig @192.168.1.247 -x 192.168.1.247

; <<>> DiG 9.16.1-Ubuntu <<>> @192.168.1.247 -x 192.168.1.247
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 42403
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 5, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: cb2b08dddb06dbf70100000062910f2c5c2c3aa1126009c7 (good)
;; QUESTION SECTION:
;247.1.168.192.in-addr.arpa.    IN      PTR

;; ANSWER SECTION:
247.1.168.192.in-addr.arpa. 604800 IN   PTR     smb.opensourcetech.test.
247.1.168.192.in-addr.arpa. 604800 IN   PTR     www.opensourcetech.test.
247.1.168.192.in-addr.arpa. 604800 IN   PTR     ftp.opensourcetech.test.
247.1.168.192.in-addr.arpa. 604800 IN   PTR     dns.opensourcetech.test.
247.1.168.192.in-addr.arpa. 604800 IN   PTR     mail.opensourcetech.test.

;; Query time: 0 msec
;; SERVER: 192.168.1.247#53(192.168.1.247)
;; WHEN: Fri May 27 17:49:32 UTC 2022
;; MSG SIZE  rcvd: 193
```

ZEUS enterprise

Appendix

LinuC

chroot
　”/var/chroot”などのディレクトリを、
　ファイルシステムのトップディレクトリである”/”にみせかける手法
　これにより攻撃者にBIND(named)を乗っ取られた場合に、
　システム全体に影響が及ぶのを防ぐもの
　https://linuc.org/study/knowledge/420/

DANE

DNS-Based Authentication of Named Entitiesの略で、
認証情報をDNSを用いて通信するための仕組み
https://www.nic.ad.jp/ja/basics/terms/dane.html

DNSSEC

DNS問い合わせに対する応答が改ざんなどされていないか検証する仕組み
https://www.nic.ad.jp/ja/newsletter/No43/0800.html

TSIG

Transaction SIGnatureの略で、
DNSのメッセージに対して電子署名を行うことで通信経路上における改ざんを防ぐ仕組み
https://jprs.jp/tech/material/rfc/RFC2845-ja.txt

LinuCレベル2 202試験
主題2.08の例題と解説

https://linuc.org/study/samples/index/s/2_08/

本日のまとめ

ZEUS enterprise

- DNSの役割を理解する
- BINDの設定方法を理解する
- BINDのゾーンファイル書式を理解する

構築には慣れが必要だが、達成感がある！
※特に設定ファイルやゾーンファイル

設定や動作を理解することで、DNSのトラブルシュート力が増す

独自ドメインの取得やドメイン管理の業務に役に立つ

LinuC

ZEUS enterprise

Q & A

LinuC

ZEUS enterprise

Thank you for join today's seminar!

https://www.opensourcetech.tokyo/

https://twitter.com/matt_zeus

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

LC LinuC