LPI-JAPAN

LinuC レベル2
技術解説セミナー

～ LinuC レベル2で学ぶサーバ構築! ～

ゼウス・エンタープライズ
LinuCエバンジェリスト 鯨井貴博

2018年4月15日(日) 13:30～16:30
@AP浜松町

Who are you?

[簡単なプロフィール]

- 前職：建設業
- LinuxやNetwork、セキュリティ講師
- 最近は、Juniper / Junosもやってます
- Opensourcetechブログ（http://opensourcetech.hatenablog.jp/）書いてます。
  よかったら、みてください！

[保有資格]

- LinuC、LPIC、HTML5プロフェッショナル、
- ACCEL（Apache CloudStack技術者認定資格 by LPI-JAPAN）
- 情報セキュリティスペシャリスト / ネットワークスペシャリスト
- 応用情報処理技術者 / 基本情報処理技術者
- 情報セキュリティマネジメント
- MCP70-640 Microfoft Windows Server 2008 Active Directory
- ITIL Foundation
- CCNA、MOS

とか いっぱい持ってますw

Who are you?

LPI-Japan発行のメルマガ「LinuC Level2・Level3を受けてみよう！」でサンプル問題作ってます！

メルマガ登録は、以下から。
http://www.lpi.or.jp/mail/#mail02

解答と解説

答えは 「services」 です。

sssdとは、識別サービスと認証サービスの管理を行うデーモンとなり、キャッシュを使用しそれらサービスの負荷軽減などを行います。

sssdの設定ファイルはsssd.confであり、[sssd]などの各セクションごとに「パラメター = 値」という形式で設定します。

パラメータservicesでは、nss、pam、sudo、autofs、sshなどがサポートされており、設定するサービスをカンマで区切って表記します。

sssd.confについて
https://github.com/kujiraitakahiro/MailMagazine/blob/master/man_sssd.conf

■例題解説提供者
LPI-Japanアカデミック認定校　Zeus IT Camp　鯨井 貴博 氏（登録インストラクター）

Agenda

LinuCとは？

技術解説

- 主題208：HTTPサービス
  - ✓Apache
  - ✓Squid
  - ✓nginx
- 主題207：ドメインネームサービス
  - ✓BIND
- 主題212：システムのセキュリティ
  - ✓iptables（firewalld）
  - ✓システムの起動（SystemV・systemd）

Appendix

- Linuxルータの作り方
- 実務で活かせるLinuC2で学ぶサーバ構築 ユースケース
- 資格取得後のスキル向上方法

LinuCとは？

ところで LinuC とは？

LinuC（リナック）

■正式資格名称
和名：Linux技術者認定資格
英名：Linux Professional Certification

■正式試験名称
和名：Linux技術者認定試験
英名：Linux Professional Certification Exam

■略称
LinuC
読み：リナック

■資格ロゴ

Linux技術者認定試験「LinuC（リナック）」とは

１）試験範囲：

Phase 1: LPICと同じ。試験問題はLPICと異なる。

→　教育コースや受験対策本の変更は不必要。

Phase 2: LPICと試験範囲も異なる予定。

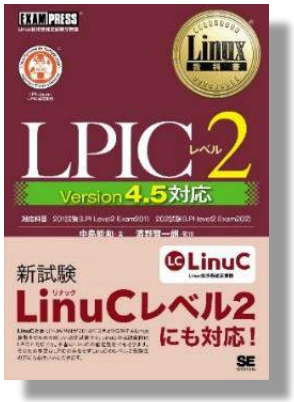
（リリース日は約１年後を予定）

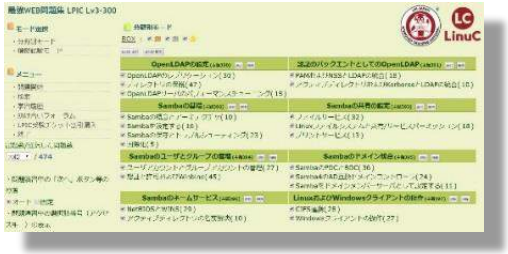
２）LPI-JapanがLPI-Japanの理事企業、日本のLinux専門家ののサポートを得て、日本の市場に最適化した試験を開発。

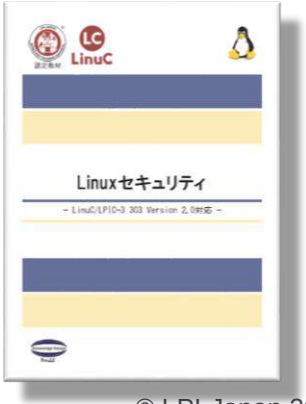
３）外国語での提供も予定。

アジア地域を中心に、試験問題の一部を其々の国のニーズに合わせた試験問題に変更予定。

LinuC対応の認定教材の一例







「LinuC（リナック）」開発の背景

日本の市場に最適化した試験を提供

1）日本のIT企業、受験者が期待する試験及びLPI-Japanの信頼性の確保。

2）日本の市場が求める技術分野に即した試験範囲にできる。

3）新しい技術分野への対応をLPI-Japanが提供する認定試験の体系で考えることができる。

4）翻訳による判りにくさは無く、ベータ試験も日本語で提供できる。

5）受験者の方々に付加価値の高いサービスを提供できる。

LinuC（リナック）の試験体系と資格認定

■ LPIC と同じ試験体系、試験範囲を採用。

Standard → Specialist

※）どの試験から受験しても良い。
※）下位レベルの認定を取得していないと上位レベルの認定は取得できない。

LinuC レベル1 認定（LinuC-1）

2試験合格で認定取得

- 102試験
- 101試験

Linuxシステムの構築・運用・管理

実務で必要なLinuxの操作とシステム管理が行えるエンジニア

LinuC レベル2 認定（LinuC-2）

2試験合格で認定取得

- 202試験
- 201試験

Linuxサーバやネットワークを含むシステムの構築・運用・保守

Linuxのシステムデザイン、ネットワーク構築において、企画、導入、維持、トラブルシューティング、キャパシティプランニングができるエンジニア

LinuC レベル3 認定（LinuC-3）

いずれか1試験合格で認定取得

- 304試験（仮想化&高可用性）
- 303試験（セキュリティ）
- 300試験（混在環境）

各分野の最高技術レベルの専門家

304 Virtualization & High Availability
303 Security
300 Mixed Environment

LinuC（リナック）の試験概要

■ 再受験ポリシー以外は、LPIC と同じ各種条件や方式を採用。

	LinuCレベル1	LinuCレベル2	LinuCレベル3		
認定名	LinuC-1	LinuC-2	LinuC-3 Mixed Environment	LinuC-3 Security	LinuC-3 Virtualization & High Availability
試験名	101試験 102試験	201試験 202試験	300試験	303試験	304試験
呼称/略称	LinuCレベル1	LinuCレベル2	LinuCレベル3 混在環境	LinuCレベル3 セキュリティ	LinuCレベル3 仮想化&高可用性
受験前提条件	特になし（どの試験から受験しても良い）				
認定の条件	LinuC-1の「101と102」に合格すること	「有意なLinuC-1」を保有し、LinuC-2の「201と202」に合格すること	「有意なLinuC-2」を保有し、LinuC-3の300に合格すること	「有意なLinuC-2」を保有し、LinuC-3の303に合格すること	「有意なLinuC-2」を保有し、LinuC-3の304に合格すること
受験費用	15,000円（税別、1試験あたり）		30,000円（税別、1試験あたり）		
試験実施方式	CBT（コンピュータベーストテスティング）またはPBT（ペーパーベーストテスティング）				
CBT会場	ピアソンVUEテストセンター（全国約200カ所）				
問題数	各試験「60問」				
所用時間	90分				
有意性の期限	有効期限は無いが、有意性の期限として5年。				
再受験ポリシー	次スライドでご説明。				

LinuC（リナック）の再受験ポリシー

■ 高信頼性確保により、分かりやすく再チャレンジや、試験改訂に合わせた資格更新をしやすい再受験ポリシーを実現。

	LinuC	LPIC
不合格時	2回目以降： 7日目以降(土日含む)	2回目： 7日目以降(土日含む) 3回目以降： 30日以降
合格時	なし	2年以降

「ITスキル標準（ITSS）」のキャリアフレームワークにおける位置づけ

■LinuC（リナック）の位置づけはLPICと同じ。

共通キャリア・フレームワーク	ITスペシャリスト						アプリケーションスペシャリスト		ソフトウェアディベロップメント			カスタマーサービス			ITサービスマネジメント			
専門分野レベル	プラットフォーム	ネットワーク	データベース	アプリケーション共通基礎	システム管理	セキュリティ	業務システム	業務パッケージ	基本ソフト	ミドルソフト	応用ソフト	ハードウェア	ソフトウェア	ファシリティマネジメント	運用管理	システム管理	オペレーション	サービスデスク
レベル3	LinuCレベル3														LinuCレベル3			
レベル2	LinuCレベル2														LinuCレベル2			
レベル1	LinuCレベル1														LinuCレベル1			

※特定非営利活動法人スキル標準ユーザ協会へマップへの反映を申請済み

有意なLPICをお持ちの方へ

LPIC認定者様向け

LinuC認定取得優待プログラム

2018年8月31日まで

2018年8月31日までは、
LPICの認定履歴などを
LinuCへ引き継ぐことが可能！

LPI-Japanの受験者ページに
ログインするだけ！！

LinuCレベル2の試験範囲（201）

- ✓ **主題200：キャパシティプランニング**
  - ✓ 200.1 リソースの使用率の測定とトラブルシュート 重要度 6
  - ✓ 200.2 将来のリソース需要を予測する 重要度 2
- ✓ **主題201：Linuxカーネル**
  - ✓ 201.1 カーネルの構成要素 重要度 2
  - ✓ 201.2 Linuxカーネルのコンパイル 重要度 3
  - ✓ 201.3 カーネル実行時における管理とトラブルシュート 重要度 4
- ✓ **主題202：システムの起動**
  - ✓ 202.1 システムの起動をカスタマイズする 重要度 3
  - ✓ 202.2 システムのリカバリ 重要度 4
  - ✓ 202.3 その他のブートローダ 重要度 2
- ✓ **主題203：ファイルシステムとデバイス**
  - ✓ 203.1 Linuxファイルシステムを操作する 重要度 4
  - ✓ 203.2 Linuxファイルシステムの保守 重要度 3
  - ✓ 203.3 ファイルシステムを作成してオプションを構成する 重要度 2
- ✓ **主題204：高度なストレージ管理**
  - ✓ 204.1 RAIDを構成する 重要度 3
  - ✓ 204.2 記憶装置へのアクセス方法を調整する重要度 2
  - ✓ 204.3 論理ボリュームマネージャ 重要度 3
- ✓ **主題205：ネットワーク構成**
  - ✓ 205.1 基本的なネットワーク構成 重要度 3
  - ✓ 205.2 高度なネットワーク構成 重要度 4
  - ✓ 205.3 ネットワークの問題を解決する 重要度 4
- ✓ **主題206：システムの保守**
  - ✓ 206.1 ソースからプログラムをmakeしてインストールする 重要度 2
  - ✓ 206.2 バックアップ操作 重要度 3
  - ✓ 206.3 システム関連の問題をユーザに通知する 重要度 1

LinuCレベル2の試験範囲（202）

- ✓ **主題207：ドメインエームサーバ**
  - ✓ 207.1 DNSサーバの基本的な設定 重要度 3
  - ✓ 207.2 DNSゾーン作成と保守 重要度 3
  - ✓ 207.3 DNSサーバを保護する 重要度 2
- ✓ **主題208：HTTPサービス**
  - ✓ 208.1 Apacheの基本的な設定 重要度 4
  - ✓ 208.2 HTTPS向けのApacheの設定 重要度 3
  - ✓ 208.3 キャッシュプロキシとしてのSquidの実装 重要度 2
  - ✓ 208.4 WebサーバおよびリバースプロキシとしてのNginx実装 重要度 2
- ✓ **主題209：ファイル共有**
  - ✓ 209.1 Sambaサーバの設定 重要度 5
  - ✓ 209.2 NFSサーバの設定 重要度 3
- ✓ **主題210：ネットワーククライアントの管理**
  - ✓ 210.1 DHCPの設定 重要度 2
  - ✓ 210.2 PAM認証 重要度 3
  - ✓ 210.3 LDAPクライアントの利用方法 重要度 2
  - ✓ 210.4 OpenLDAPサーバの設定 重要度 4
- ✓ **主題211：電子メールサービス**
  - ✓ 211.1 電子メールサーバの使用 重要度 4
  - ✓ 211.2 電子メール配信を管理する 重要度 2
  - ✓ 211.3 メールボックスアクセスを管理する 重要度 2
- ✓ **主題212：システムのセキュリティ**
  - ✓ 212.1 ルータを構成する 重要度 3
  - ✓ 212.2 FTPサーバの保護 重要度 2
  - ✓ 212.3 セキュアシェル（SSH） 重要度 4
  - ✓ 212.4 セキュリティ業務 重要度 3
  - ✓ 212.5 OpenVPN 重要度 2

学習のポイント1

重要度

実機
操作

受験日
設定

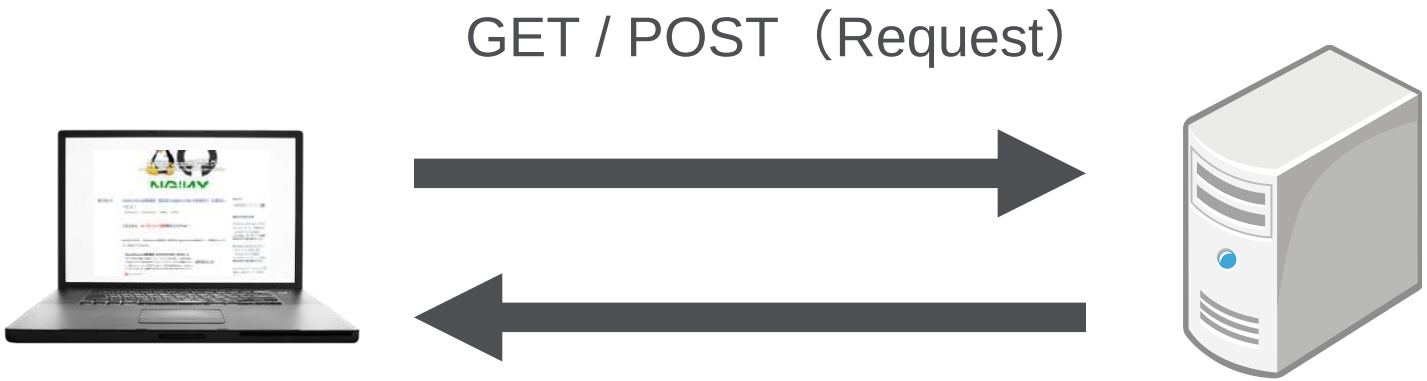
学習のポイント2

- 試験内容の把握
  - ✓ https://linuc.org/linuc2/range/201.html
  - ✓ https://linuc.org/linuc2/range/202.html
- 試験対策方法の把握
  - ✓ http://lpi.or.jp/learning/
- 受験までのタスクを数値化
  - ✓ 問題集を5周する
  - ✓ 問題集の正答率を90%以上になるまでやる　など

主題208：HTTPサービス

HTTPサービスとは？

HTTPサービス

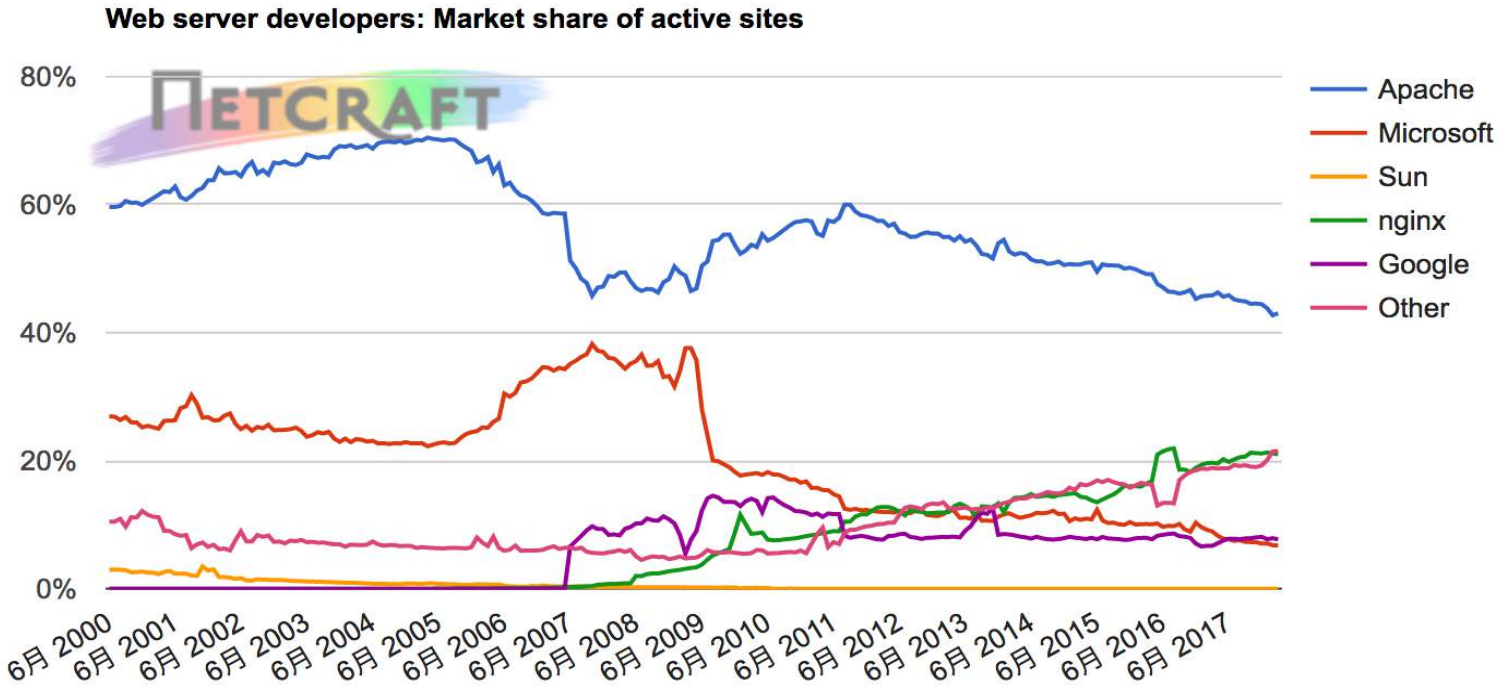


Apacheとは？

Apacheとは？

APACHE HTTP SERVER PROJECT

https://httpd.apache.org/



https://news.netcraft.com/archives/2018/03/27/march-2018-web-server-survey.html

Apacheとは？

CentOSなどのRPM系では **httpd**、
UbuntuなどDebian系では **apache2**という名前のパッケージです。

※パッケージ名だけでなく、設定ファイル名やそのパスも変わるので、
　注意が必要です。

Apacheの構成・設定

Apacheの構成（RPM系の場合）

/etc/httpd/conf/httpd.conf・・・・メイン設定ファイル
/etc/httpd/conf.d/*.conf・・・・includeされる設定ファイル
ssl.conf・・・・SSL関連設定

/var/www/html/*.htmlなど・・・・コンテンツファイル
/etc/httpd/modules/*.so・・・・モジュール（追加機能）
.htaccess・・・・ディレクトリ毎に異なる設定を適用するファイル

※Debian系では、apache2.confがメイン設定ファイルとなる。

Apacheの設定（ディレクティブ）

Listen・・・・サーバの待ち受けポート（default:80）
DocumentRoot・・・・ドキュメントルート（/var/www/htmlなど）
DirectoryIndex・・・・デフォルトの公開ファイル名（index.html）
Include・・・・メイン設定ファイルとは別で参照される設定ファイル
User・・・・Apacheを動作させるユーザー名
Group・・・・Apacheを動作させるグループ名
Options・・・・オプション設定
CustomeLog・・・・アクセスログファイル指定
ErrorLog・・・・エラーログファイル指定
<Directory />～</Direcotry>・・・・特定ディレクトリ設定

https://httpd.apache.org/docs/2.4/ja/mod/directives.html

Apacheの構築

Apacheの構築1（シンプルなWebサーバ）

パッケージからのインストール
yum install httpd
※Debian系では、apt-get（apt） install apache2など

ソースからのインストール
依存関係などを解決した上で、makeなどを使用する。

設定の構文チェック
apachectl（apachectl2）configtest

起動
systemctl start httpd
（/etc/init.d/httpd start、service httpd start、apachectl start）

Apacheの構築2（https化）

パッケージからのインストール
yum install mod_ssl

再起動
systemctl restart httpd
（/etc/init.d/httpd restart、service httpd restart、apachectl restart）

※上記で、デフォルトのlocalhost.crt（公開鍵）・
localhost.key（秘密鍵）を使ったhttps通信ができるようになる

Apacheの構築3（basic認証）

パスワードファイル作成

htpasswd –b –c /etc/httpd/conf/.htpasswd username password

※二人目以降のユーザ作成は、-cオプション不要

httpd.confに追加

```
<Directory "/var/www/html/basic/">
        AuthUserFile    /etc/httpd/conf/.htpasswd
        AuthName        ”Please enter username and password"
        AuthType        Basic
        Require         valid-user
</Directory>
```

指定ディレクトリやコンテンツファイルを配置

Apacheの動作確認

Apacheの動作確認

起動に関しては、
/var/log/messages
systemctl status httpd
netstat（ss）

```
[root@localhost ~]# systemctl status httpd
● httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled; vendor prese
t: disabled)
   Active: active (running) since 月 2018-04-02 18:11:50 JST; 7s ago
     Docs: man:httpd(8)
           man:apachectl(8)
 Main PID: 1870 (httpd)
   Status: "Processing requests..."
   CGroup: /system.slice/httpd.service
           ├─1870 /usr/sbin/httpd -DFOREGROUND
           ├─1871 /usr/sbin/httpd -DFOREGROUND
           ├─1872 /usr/sbin/httpd -DFOREGROUND
           ├─1873 /usr/sbin/httpd -DFOREGROUND
           ├─1874 /usr/sbin/httpd -DFOREGROUND
           └─1875 /usr/sbin/httpd -DFOREGROUND

 4月 02 18:11:50 localhost.localdomain systemd[1]: Starting The Apache HTTP ...
 4月 02 18:11:50 localhost.localdomain httpd[1870]: AH00558: httpd: Could no...
 4月 02 18:11:50 localhost.localdomain systemd[1]: Started The Apache HTTP S...
Hint: Some lines were ellipsized, use -l to show in full.
```

Apacheの動作確認

Apacheへのアクセスに関しては、
/var/log/http/access.log
/var/log/http/error.log

tail –fで見ると、リアルタイムでログチェックできて便利

```
[root@localhost ~]# iptables -F
[root@localhost ~]#
[root@localhost ~]# tail -f /var/log/httpd/access_log
::1 - - [02/Apr/2018:18:19:05 +0900] "OPTIONS * HTTP/1.0" 200 - "-" "Apache/2.4.
6 (CentOS) OpenSSL/1.0.2k-fips (internal dummy connection)"
192.168.11.2 - - [02/Apr/2018:18:19:09 +0900] "GET / HTTP/1.1" 403 4897 "-" "Moz
illa/5.0 (Macintosh; Intel Mac OS X 10.12; rv:59.0) Gecko/20100101 Firefox/59.0"
192.168.11.2 - - [02/Apr/2018:18:19:09 +0900] "GET /images/apache_pb.gif HTTP/1.
1" 304 - "http://127.0.0.1:2280/" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10.12;
 rv:59.0) Gecko/20100101 Firefox/59.0"
```

Apacheのトラブルシュート

Apacheのトラブルシュート

起動しない場合
起動させた時の出力や/var/log/messageのログなどをヒントにする

コンテンツファイルが参照できない
セキュリティ設定やパーミッション、
アクセスログ（/var/log/httpd/access.logやerror.log）をヒントにする
少し高度な方法としては、パケットキャプチャ確認もある

HTTPのスターテスコード

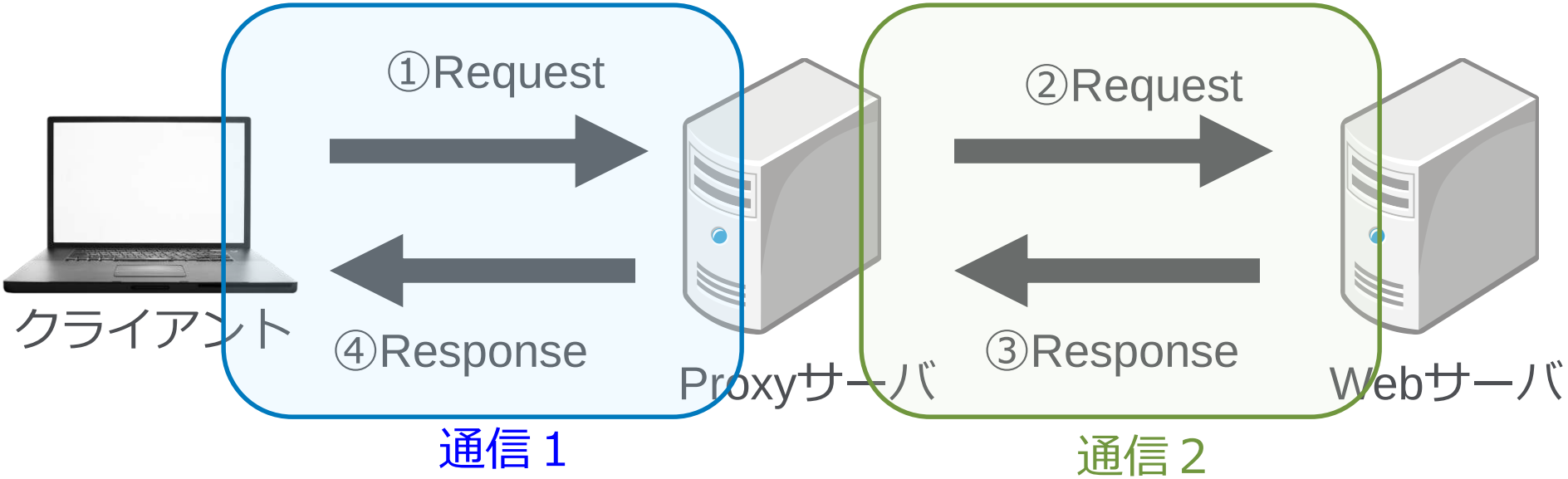
200番台・・・アクセス成功（200 OK）
300番台・・・リダクレクト（301 Moved Permanently / 304 Not Modified）
400番台・・・クライアントエラー（401 Unauthorized / 404 Not Found）
500番台・・・サーバエラー（500 Internal Server Error）

HTTPのmethod

GET・・・URIで指定した情報を要求する際に使用される
POST・・・フォームのデータを送る際に使用される
HEAD・・・ヘッダーのみを取得
CONNECT・・・httpsのプロキシ通信などの際に使用される
OPTIONS・・・サーバがサポートしているメドッドなどの確認

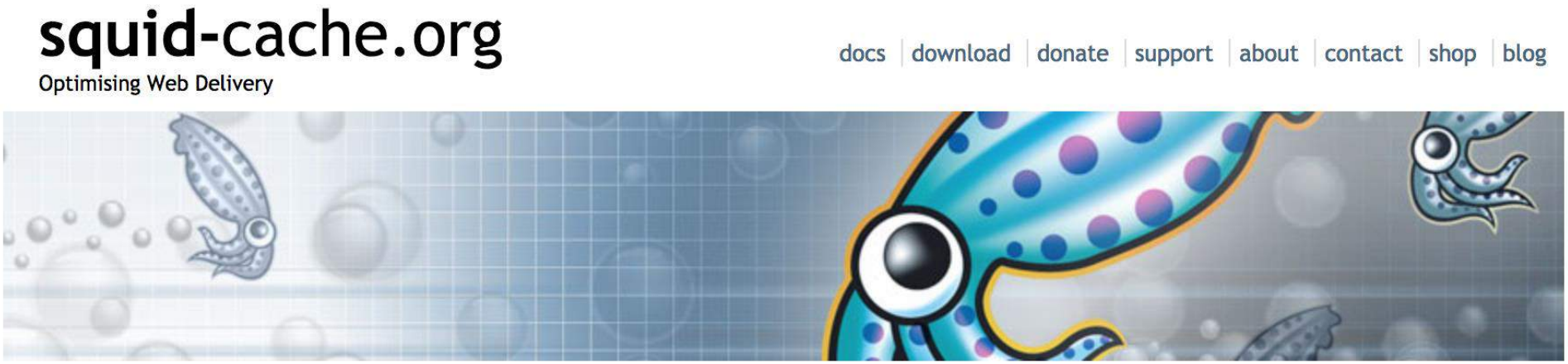
Proxyとは？

Proxy（代理応答）



Squidとは？

Squidとは？



http://www.squid-cache.org/

代理応答・ウェブのフィルタリング・ウェブへのアクセス履歴などで使われる

Squidの構成・設定

Squidの構成

/etc/squid/squid.conf・・・・メイン設定ファイル

Squidの設定1（ディレクティブ）

http_port・・・・サーバの待ち受けポート（default:3128）
cache_dir・・・・キャッシュディレクトリ設定
　cach_dir ufs /var/spool/squid 100 16 254
　（数字は左から、ディスクキャッシュ容量・一次ディレクトリ数・
　　二次ディレクトリ数を示す。単位は、MB）

Squidの設定2（ディレクティブ）

acl・・・・ネットワークやポートのリスト

例：acl SSL_ports port 443

（TCP443は、SSL_portsという名前にする）

acl localnet src 192.168.0.0/16

（送信元ネットワーク192.168.0.0/16は、localnetという名前にする）

http_access・・・・リストに基づいてProxyによる代理応答を許可（allow）/ 拒否（deny）

例：http_access allow localnet

http_access deny !Safe_ports

Squidの構築

Squidの構築

パッケージからのインストール

yum install squid

起動

systemctl start squid

（/etc/init.d/squid start、service squid start）

Squidの動作確認

Squidの動作確認

起動に関しては、

/var/log/messages

systemctl status squid

netstat（ss）

```
[root@localhost ~]# ss -tan
State      Recv-Q Send-Q Local Address:Port                Peer Address:Port

LISTEN     0      128          *:22                        *:*
LISTEN     0      100    127.0.0.1:25                        *:*

ESTAB      0      36     192.168.11.4:22                192.168.11.2:58412

LISTEN     0      128         :::80                       :::*
LISTEN     0      128         :::22                       :::*
LISTEN     0      128         :::3128                     :::*
LISTEN     0      100        ::1:25                       :::*
LISTEN     0      128         :::443                      :::*
```

Squidの動作確認

Apacheへのアクセスに関しては、
/var/log/squid/access.log
/var/log/squid/error.log

tail –fで見ると、リアルタイムでログチェックできて便利

```
[root@localhost ~]# tail -f /var/log/squid/access.log
1522663539.072     51 192.168.11.2 TCP_TUNNEL/200 0 CONNECT www.google.com:443 -
 HIER_DIRECT/172.217.161.68 -
1522663539.541     61 192.168.11.2 TCP_MISS/301 3857 GET http://yahoo.co.jp/ - H
IER_DIRECT/182.22.59.229 text/html
1522663539.882     85 192.168.11.2 TCP_TUNNEL/200 17019 CONNECT s.yimg.jp:443 -
HIER_DIRECT/183.79.250.123 -
1522663539.884     85 192.168.11.2 TCP_TUNNEL/200 9203 CONNECT s.yimg.jp:443 - H
IER_DIRECT/183.79.250.123 -
1522663539.898     95 192.168.11.2 TCP_TUNNEL/200 5749 CONNECT s.yimg.jp:443 - H
IER_DIRECT/183.79.250.123 -
1522663539.908     96 192.168.11.2 TCP_TUNNEL/200 8270 CONNECT s.yimg.jp:443 - H
IER_DIRECT/183.79.250.123 -
```

Squidのトラブルシュートや注意点

Squidのトラブルシュートや注意点

起動しない場合
起動させた時の出力や/var/log/messageのログなどをヒントにする

Proxyへアクセス出来ない
セキュリティ設定やパーミッション、
アクセスログ（/var/log/httpd/access.logやerror.log）をヒントにする
また、クライアントのWebブラウザーやOS機能などのProxy設定を確認する
少し高度な方法としては、パケットキャプチャ確認もある

Squid停止時は、Cacheしたコンテンツの処理などに時間のかかるケースがあるので注意が必要

nginxとは？

nginxとは？

Introducing NGINX 1.13.10 with gRPC support.
Learn more

nginx news

2018-03-23 unit-0.7 beta version has been released with Ruby module.

2018-03-20 nginx-1.13.10 mainline version has been released, featuring the gRPC proxy module.

2018-02-20 nginx-1.13.9 mainline version has been released.

2018-02-09 unit-0.6 beta version has been released with Perl module and advanced process management.

2018-01-15 unit-0.4 beta version has been released with regression fixes.

2017-12-29 unit-0.3 beta version has been released with HTTP keep-alive support, latency optimizations, Python and Go improvements, and more.

NGINX

english
русский

news
2017
2016
2015
2014
2013
2012
2011
2010
2009

nttp://nttp://nginx.org/

Igor Sysoevさんが2000年頃に作成したソフトウェア
C10K問題に対応

nginxの特徴

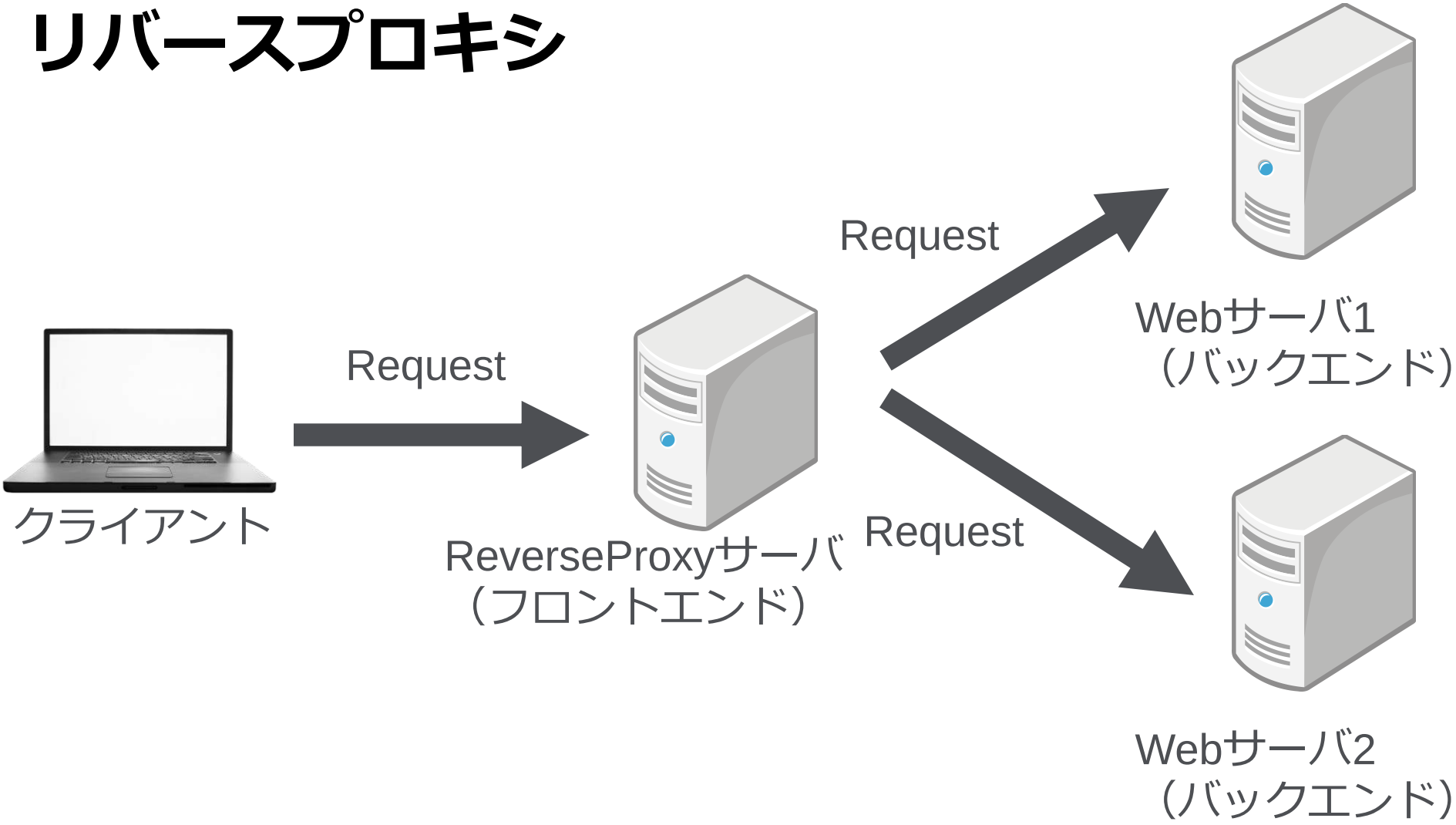
以下の動作をさせることができる

- Webサーバ（Apacheと同様）
- リバースプロキシ
- メールプロキシ

※余談ですが、最近はunitというアプリケーションサーバもリリースされている

リバースプロキシとは？

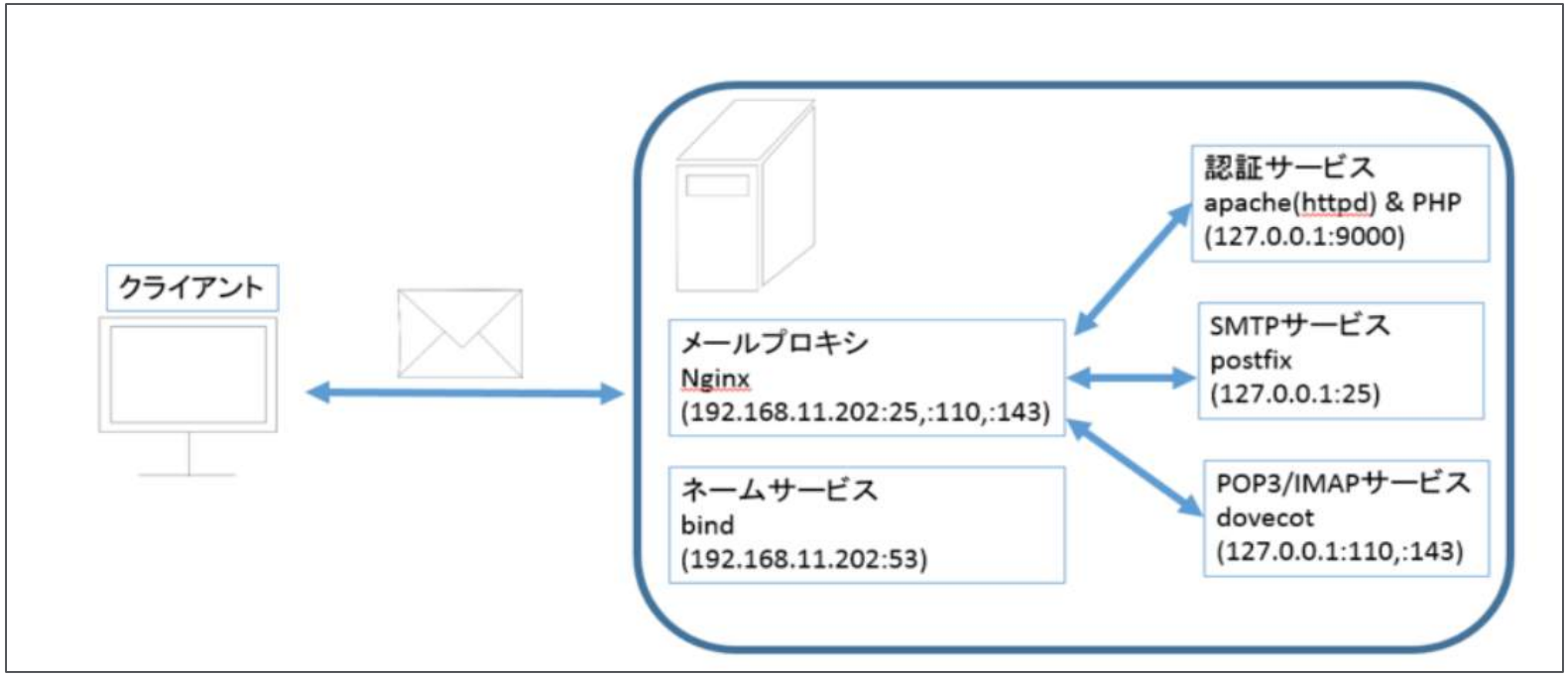
リバースプロキシ



メールプロキシとは？

メールプロキシ

興味のある方は、
「opensourcetech メールプロキシ」で検索してください。
nginx / BIND / Apache / Postfix / dovecot / PHPを使った
方法を紹介してます



nginxの構成・設定（リバースプロキシ）

nginxの構成

/etc/nginx/nginx.conf・・・・メイン設定ファイル
/etc/nginx/conf.d/*.conf・・・・includeされる設定ファイル
/usr/share/nginx/html/*.htmlなど・・・・コンテンツファイル

nginxの設定（ブロック）

nginxでは、設定（ディレクティブ）は決められたブロック配下に記述する

events・・・・プロセス制御やログなどに関する設定
server・・・・使用するポートやサーバ名などの設定
http・・・・httpに関する設定
mail・・・・メールに関する設定

nginxの設定（ディレクティブ）

listen・・・・使用するポート番号
server_name・・・・サーバ名
location・・・・URIパス毎の設定
root・・・・ドキュメントルート
index・・・・デフォルトの公開ファイル名
user・・・・nginx起動時のユーザ
worker_processes・・・・使用するCPUコア数
worker_connections・・・・1workerプロセス毎の最大同時接続数
access_log / error_log・・・・ログファイルの場所など
log_format・・・・ログフォーマット
pid・・・・プロセスID
include・・・・nginx.conf以外に参照するファイルの場所
http://nginx.org/en/docs/dirindex.html

nginxの設定例

```
upstream apache {
        server 192.168.2.205:8080;
        server 192.168.2.206:8080;
}
server {
        listen 80;
        location / {
                proxy_pass http://apache;
        proxy_http_version 1.1;
        proxy_set_header X-Forwarded-For
$proxy_add_x_forwarded_for;
        proxy_set_header Host $host:$server_port;
        }
}
```

バックエンドサーバの指定

バックエンドサーバに渡すヘッダー情報

nginxの設定例（opensourcetech.hatenablog.jp）

nginx reverse proxy構築の参考
「nginxによるリバースプロキシ(reverse proxy)構築 on CentOS6.5」

nginx SSLサイト構築の参考
「簡単に nginx でhttpsを実施する方法」

nginx SSLクライアント認証の参考
「Nginx 1.7.8におけるSSLクライアント認証 & 2014年のNginxを振り返る」

nginx モジュールの使用・ソースからインストールの参考
「Nginx 1.7.10 with GeoIP moduleによるアクセスログ管理」
※その他、モジュールも色々試してます

nginxの構築

nginxの構築

パッケージからのインストール
yum install nginx

ソースからのインストール
※デフォルト無効になっているモジュールを使用する場合など
依存関係などを解決した上で、makeなどを使用する。
http://nginx.org/en/docs/configure.html

起動
systemctl start nginx
（/etc/init.d/nginx start、service nginx start）

nginxの動作確認

nginxの動作確認

起動に関しては、

/var/log/messages

systemctl status nginx

netstat（ss）

```
[[root@localhost nginx]# systemctl status nginx
● nginx.service - nginx - high performance web server
   Loaded: loaded (/usr/lib/systemd/system/nginx.service; disabled; vendor prese
t: disabled)
   Active: active (running) since 月 2018-04-02 20:04:35 JST; 2s ago
     Docs: http://nginx.org/en/docs/
  Process: 7916 ExecStart=/usr/sbin/nginx -c /etc/nginx/nginx.conf (code=exited,
 status=0/SUCCESS)
 Main PID: 7917 (nginx)
   CGroup: /system.slice/nginx.service
           ├─7917 nginx: master process /usr/sbin/nginx -c /etc/nginx/nginx.c...
           └─7918 nginx: worker process

 4月 02 20:04:35 localhost.localdomain systemd[1]: Starting nginx - high per...
 4月 02 20:04:35 localhost.localdomain systemd[1]: PID file /var/run/nginx.p...
 4月 02 20:04:35 localhost.localdomain systemd[1]: Started nginx - high perf...
Hint: Some lines were ellipsized, use -l to show in full.
```

nginxの動作確認

Apacheへのアクセスに関しては、
/var/log/nginx/access.log
/var/log/nginx/error.log

```
[root@localhost nginx]# tail -f /var/log/nginx/access.log
192.168.11.2 - - [02/Apr/2018:20:05:54 +0900] "GET / HTTP/1.1" 200 612 "-" "Mozi
lla/5.0 (Macintosh; Intel Mac OS X 10.12; rv:59.0) Gecko/20100101 Firefox/59.0"
"_"
```

```
[root@localhost nginx]# tail -f /var/log/nginx/error.log
2018/04/02 20:03:03 [emerg] 7805#7805: bind() to 0.0.0.0:80 failed (98: Address
already in use)
2018/04/02 20:03:03 [emerg] 7805#7805: bind() to 0.0.0.0:80 failed (98: Address
already in use)
2018/04/02 20:03:03 [emerg] 7805#7805: bind() to 0.0.0.0:80 failed (98: Address
already in use)
2018/04/02 20:03:03 [emerg] 7805#7805: still could not bind()
```

nginxのトラブルシュート

nginxのトラブルシュート

起動しない場合
起動させた時の出力や/var/log/messageのログなどをヒントにする
他のWebサーバソフトがTCP80などを使用していないか確認

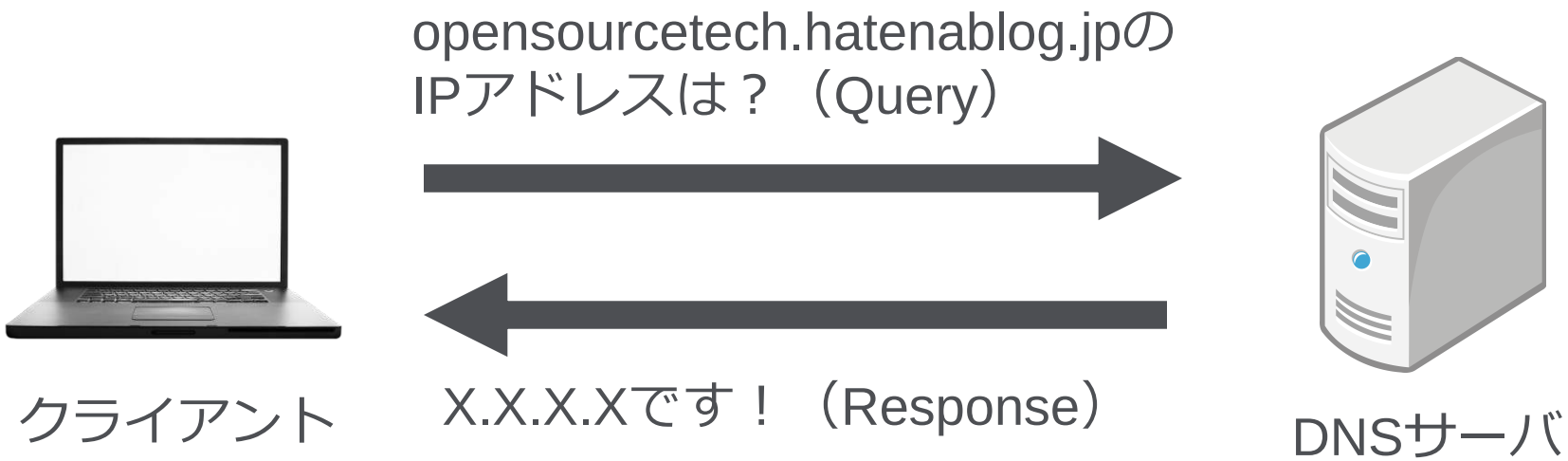
nginxへアクセス出来ない
セキュリティ設定やパーミッション、
アクセスログ（/var/log/httpd/access.logやerror.log）をヒントにする
少し高度な方法としては、パケットキャプチャ確認もある

nginx自体（フロントエンド）の問題なのか、転送されるWebサーバ（バックエンド）などの問題なのかを切り分ける必要がある

主題207：ドメインネームサービス

DNS（ドメインネームサービス）とは？

DNS（ドメインネームサービス）



BINDとは？

BINDとは？

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

BINDの構成・設定

BINDの構成

/etc/named内
named.conf・・・・メイン設定ファイル
rfc1912.zones・・・・localhost用正逆引き設定ファイル

/var/named内
****.zones・・・・使用ドメイン用正引きゾーンファイル
****.rev・・・・使用ドメイン用逆引きゾーンファイル
named.ca・・・・ルートDNSゾーンファイル
　※named.rootという場合もある
named.localhost・・・・localhost用正引きゾーンファイル
named.loopback・・・・localhost用逆引きゾーンファイル
named.empty・・・・ゾーンファイルの雛形

BINDの設定（ステートメント）

options・・・・BINDの動作に関連するグローバル設定
logging・・・・ログ関連の設定
zone・・・・ゾーン情報関連の設定
include・・・・外部設定ファイルの参照関連設定

BINDの設定（オプション）

listen-on port・・・・IPv4の待ち受けIPアドレスやポート番号
listen-on-v6 port・・・・IPv6の待ち受けIPアドレスやポート番号
directory・・・・ゾーンファイルなどを配置するディレクトリ
allow-query・・・・Queryを受け付ける対象
recursion・・・・再帰的問い合わせをするかどうか
pid-file・・・・PIDファイルの指定

```
zone “管理ドメイン" IN {
        type [master/slave/hintのいずれか];
        file ”ゾーンファイル名";
};
```

BINDの設定（DNSレコード）

SOAレコード・・・・管理ドメインに関する情報
Aレコード・・・・IPv4正引き用
AAAAレコード・・・・IPv6正引き用
MXレコード・・・・メールサーバ（Mail eXchange）用
NSレコード・・・・ネームサーバ（DNS）用
CNAMEレコード・・・・別名定義用
PTRレコード・・・・逆引き用
TXTレコード・・・・テキスト登録用

BINDの設定（ゾーンファイル）

正引き

```
$TTL 3H
@       IN SOA  ns.test.local. root.test.local. (
                                2018041501      ; serial
                                        1D      ; refresh
                                        1H      ; retry
                                        1W      ; expire
                                        3H )    ; minimum
        NS      ns.test.local.
        MX 10   mail.test.local.
ns      A       192.168.11.4
www     A       192.168.11.4
ftp     A       192.168.11.4
mail    A       192.168.11.4
```

逆引き

```
$TTL 3H
@       IN SOA  ns.test.local. root.test.local. (
                                2018041501      ; serial
                                        1D      ; refresh
                                        1H      ; retry
                                        1W      ; expire
                                        3H )    ; minimum
        NS      ns.test.local.
        MX 10   mail.test.local.
4       PTR     ns.test.local.
4       PTR     www.test.local.
4       PTR     ftp.test.local.
4       PTR     mail.test.local.
```

BINDの構築

BINDの構築

パッケージからのインストール

yum install bind

設定ファイルの構文やゾーンのチェック

named-checkconf

named-checkzone

起動

systemctl start named

（/etc/init.d/named start、service named start）

BINDの動作確認

BINDの動作確認

起動に関しては、

/var/log/messages

systemctl status bind

host/nslookup/dig

```
[[root@localhost ~]# dig opensourcetech.hatenablog.jp

; <<>> DiG 9.9.4-RedHat-9.9.4-51.el7_4.2 <<>> opensourcetech.hatenablog.jp
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 55898
;; flags: qr rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 512
;; QUESTION SECTION:
;opensourcetech.hatenablog.jp.  IN      A

;; ANSWER SECTION:
opensourcetech.hatenablog.jp. 59 IN     A       13.115.18.61
opensourcetech.hatenablog.jp. 59 IN     A       13.230.115.161

;; Query time: 100 msec
;; SERVER: 8.8.8.8#53(8.8.8.8)
;; WHEN: 火  4月 03 03:04:11 JST 2018
;; MSG SIZE  rcvd: 89
```

BINDのトラブルシュート

BINDのトラブルシュート

起動しない場合
起動させた時の出力や/var/log/messageのログなどをヒントにする

正引き・逆引きが出来ない
セキュリティ設定やパーミッション、
アクセスログ（/var/log/httpd/access.logやerror.log）をヒントにする

クライアント側のネットワーク設定などが、BINDに向いているかなども留意する必要がある

BINDのセキュリティ1

chroot
/var/namedディレクトリを、**/var/named/chroot**/var/namedとして見せるセキュリティ対策
bind-chrootパッケージの導入で実装できる

allow-query
Queryを受け付ける対象を制限する

バージョン情報を隠す
named.conf内で以下を記述

```
options {
        version "latest";
};
```

BINDのセキュリティ2

頻繁に脆弱性情報、それに対するアップデートがアナウンスされるので、常にチェックしておく

DNS サーバ BIND の脆弱性対策について(CVE-2017-3145)

最終更新日：2018年1月18日

※追記・改訂すべき情報がある場合には、その都度このページを更新する予定です。

概要

DNS サーバの BIND に、遠隔からの攻撃によって異常終了し、サービス不能 (DoS) 状態となる脆弱性が存在します。

脆弱性を悪用した攻撃はまだ確認されていませんが、今後攻撃が発生する可能性があるため至急 DNS サーバ管理者はアップデートを適用して下さい。

影響を受けるバージョン

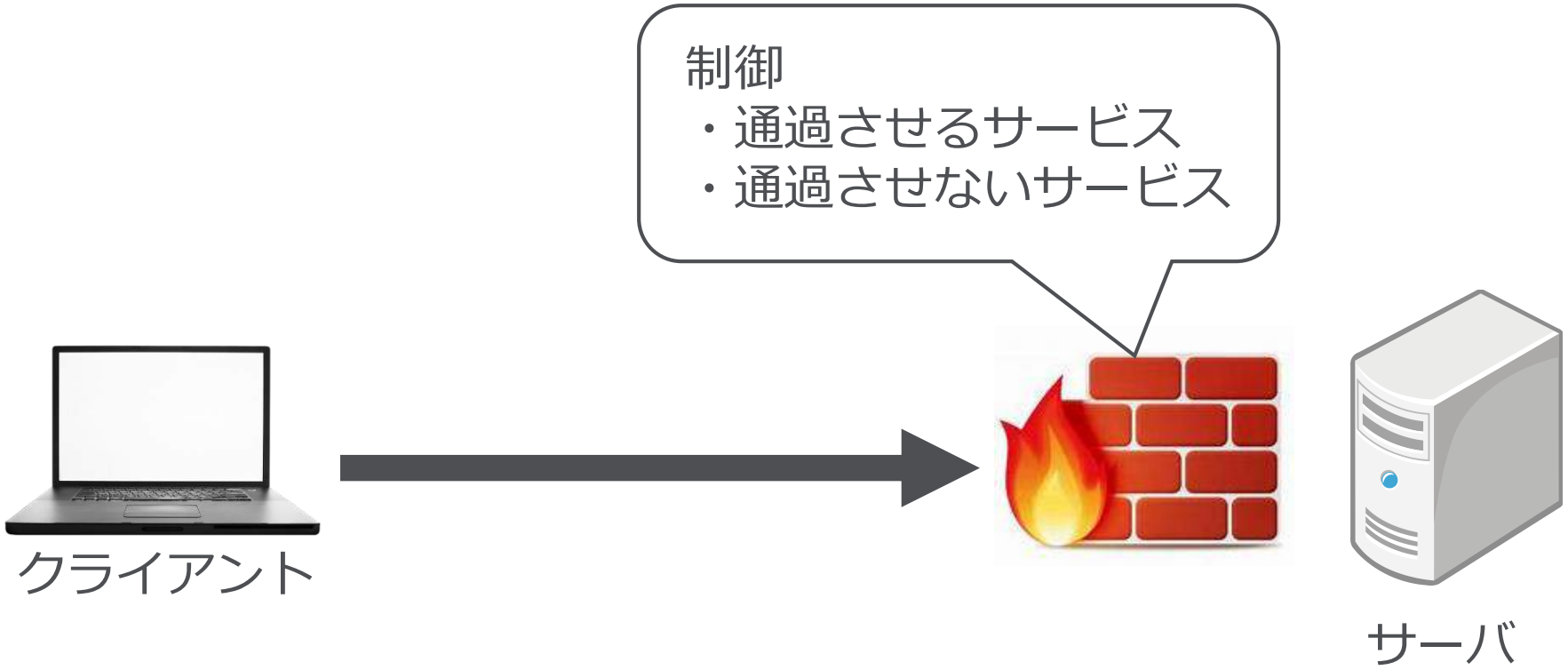
- BIND 9.0.0 から 9.8.x までのバージョン
- BIND 9.9.0 から 9.9.11 までのバージョン
- BIND 9.10.0 から 9.10.6 までのバージョン
- BIND 9.11.0 から 9.11.2 までのバージョン
- BIND 9.9.3-S1 から 9.9.11-S1 までのバージョン
- BIND 9.10.5-S1 から 9.10.6-S1 までのバージョン
- BIND 9.12.0a1 から 9.12.0rc1 までのバージョン

https://www.ipa.go.jp/security/ciadr/vul/20180118-bind.html

主題212：システムのセキュリティ

iptablesとは？

iptables（Firewallみたいなサービス）



※CentOS7などでは、firewalldというプログラムになっている
opensourcetech.hatenablog.jp「シンプルなfirewalldの使い方」

iptablesの設定

iptablesの構成

/etc/sysconfig/itpables

```
# Firewall configuration written by system-config-firewall
# Manual customization of this file is not recommended.
*filter
:INPUT ACCEPT [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
-A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
-A INPUT -p icmp -j ACCEPT
-A INPUT -i lo -j ACCEPT
-A INPUT -m state --state NEW -m tcp -p tcp --dport 22 -j ACCEPT
-A INPUT -j REJECT --reject-with icmp-host-prohibited
-A FORWARD -j REJECT --reject-with icmp-host-prohibited
COMMIT
```

iptablesの動作確認

iptablesの動作確認

iptables -L

```
[root@localhost ~]# iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
ACCEPT     all  --  anywhere             anywhere             state RELATED,ESTABLISHED
ACCEPT     icmp --  anywhere             anywhere
ACCEPT     all  --  anywhere             anywhere
ACCEPT     tcp  --  anywhere             anywhere             state NEW tcp dpt:ssh
REJECT     all  --  anywhere             anywhere             reject-with icmp-host-prohibited

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination
REJECT     all  --  anywhere             anywhere             reject-with icmp-host-prohibited

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
```

iptablesの設定変更

iptablesの設定変更

ルールのクリア
iptables –F

ルールのバックアップ
iptables-save > /etc/sysconfig/iptables.backup

ルールのリストア
iptables-restore < /etc/sysconfig/iptables.backup

ルールの追加
iptables –A INPUT –p tcp –-dport 80 –j ACCEPT（既存ルールの最後）
iptables –I INPUT -p tcp –-dport 80 –j ACCEPT（ルール順指定）

主題212：システムの起動

SystemV その1

起動
/etc/init.d/*** start
service *** start

停止
/etc/init.d/*** stop

状態確認
/etc/init.d/*** status

SystemV その2

自動起動設定
chkconfig –list（一覧確認）
chkconfig *** on（自動起動有効化）
chkconfig *** off（自動起動無効化）
chkconfig –level 35 on（ランレベル3と5の場合に自動起動有効化）

systemd その1

起動
systemctl start ***
（systemctl start ***.serviceでも同様）

停止
systemctl stop ***

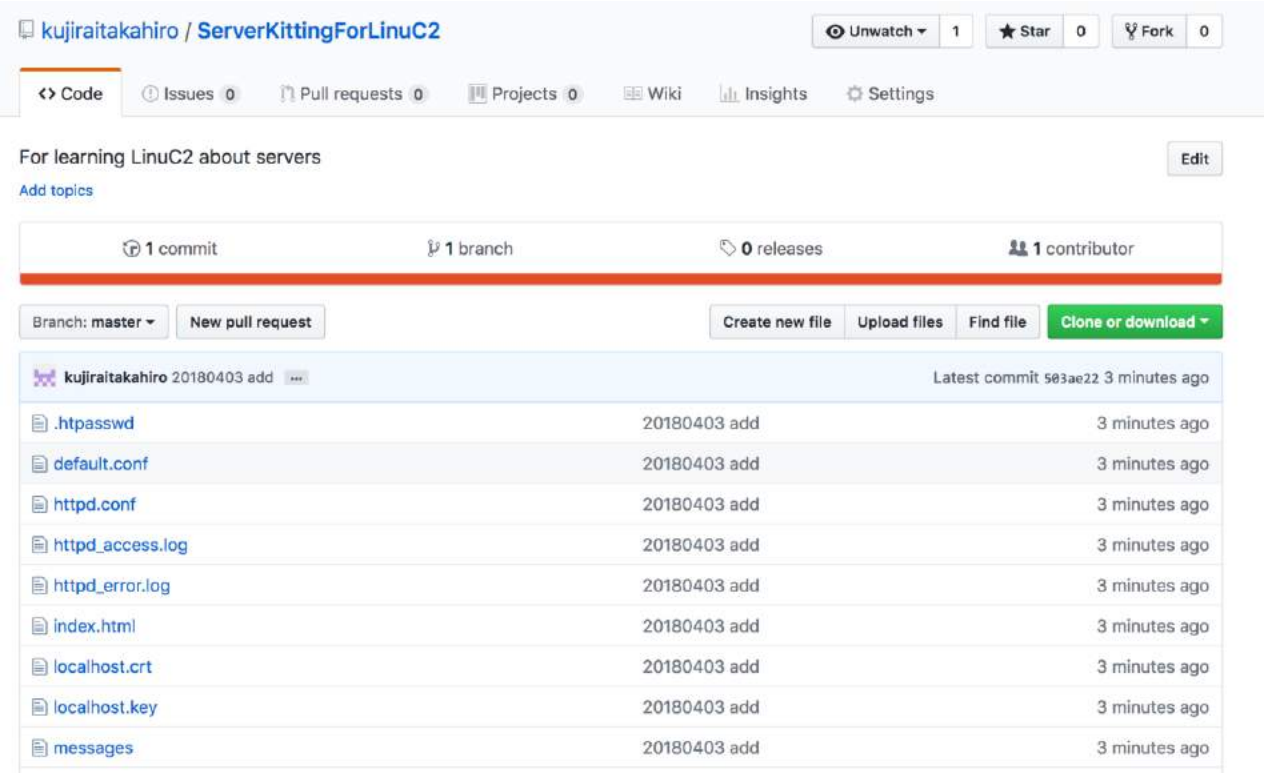
状態確認
systemctl status ***

systemd その2

自動起動設定
systemctl is-enabled ***（確認）
systemctl list-unit-files --type=service（一覧確認）
systemctl enable ***（自動起動有効化）
systemctl disable ***（自動起動無効化）

本日使用した設定ファイルなどは、
Githubへアップしてあります
**　※お好きにどうぞ！**

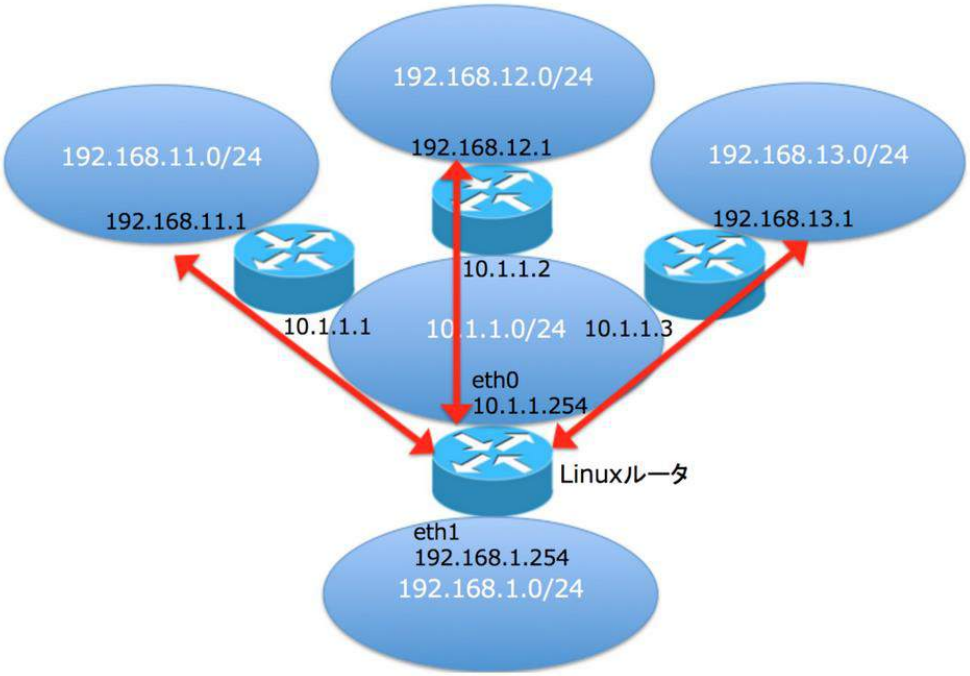
https://github.com/kujiraitakahiro/ServerKittingForLinuC2



Appendix

「簡単にLinuxルータを作成する方法」（opensourcetech.hatenablog.jp）

```
echo 1 > /proc/sys/net/ipv4/ip_forward
/etc/sysctl.conf
 net.ipv4.ip_forward = 1
```



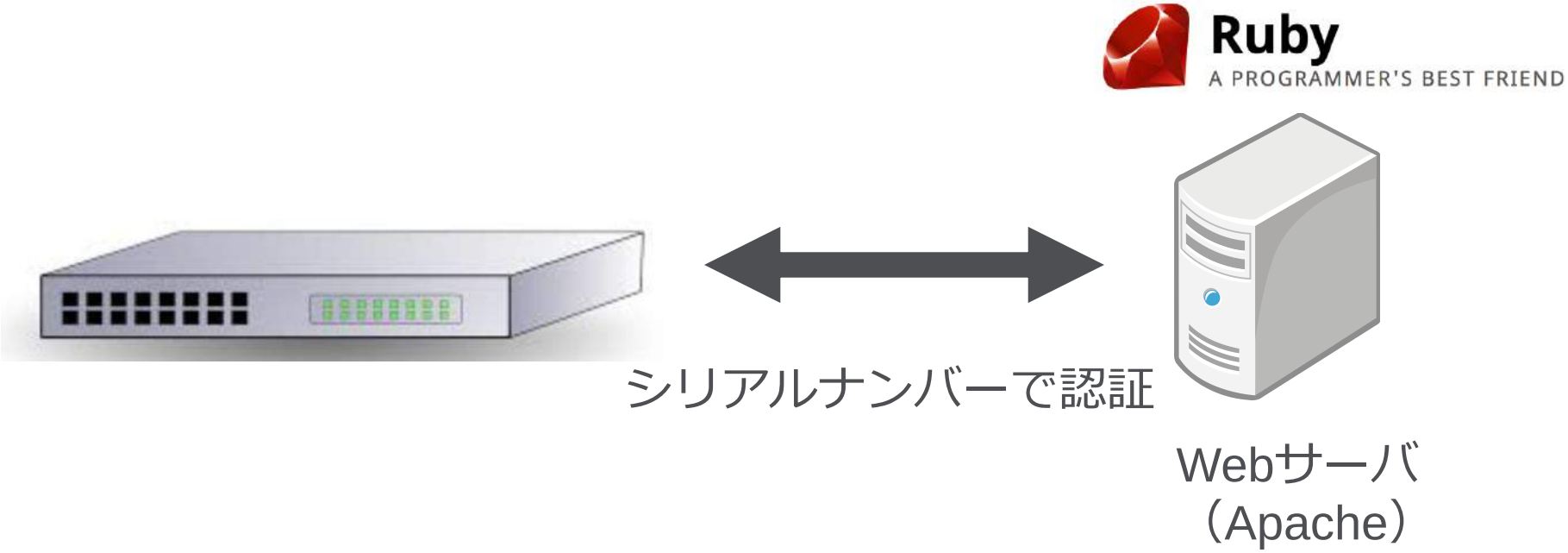
実務で活かせたLinuC2で学ぶサーバ構築ユースケース

なぜ、LinuC2に出てくるサーバ構築が役に立つのか？

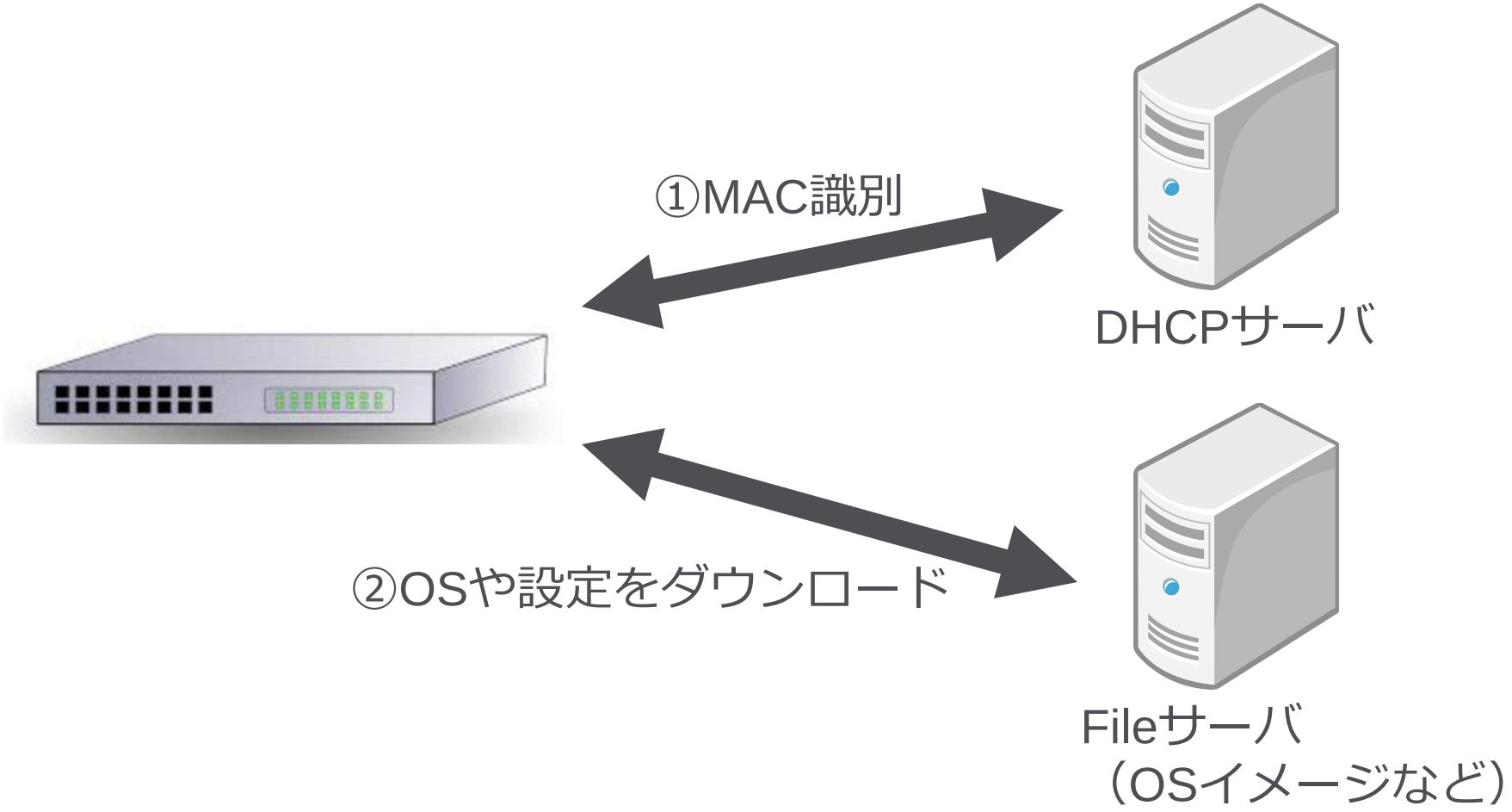
WindowsなどでGUIから操作出来るサーバソフトもあるが、
xampp（web、DBなど）
Tftpd32（syslog）
結局、サーバ構築方法が分からないと理解して操作出来ない。
また、カスタマイズ設定などが出来ない。

ニッチなサーバを作らないといけないとなると、
簡単に使えるGUI操作出来るソフトは、途端になくなる。

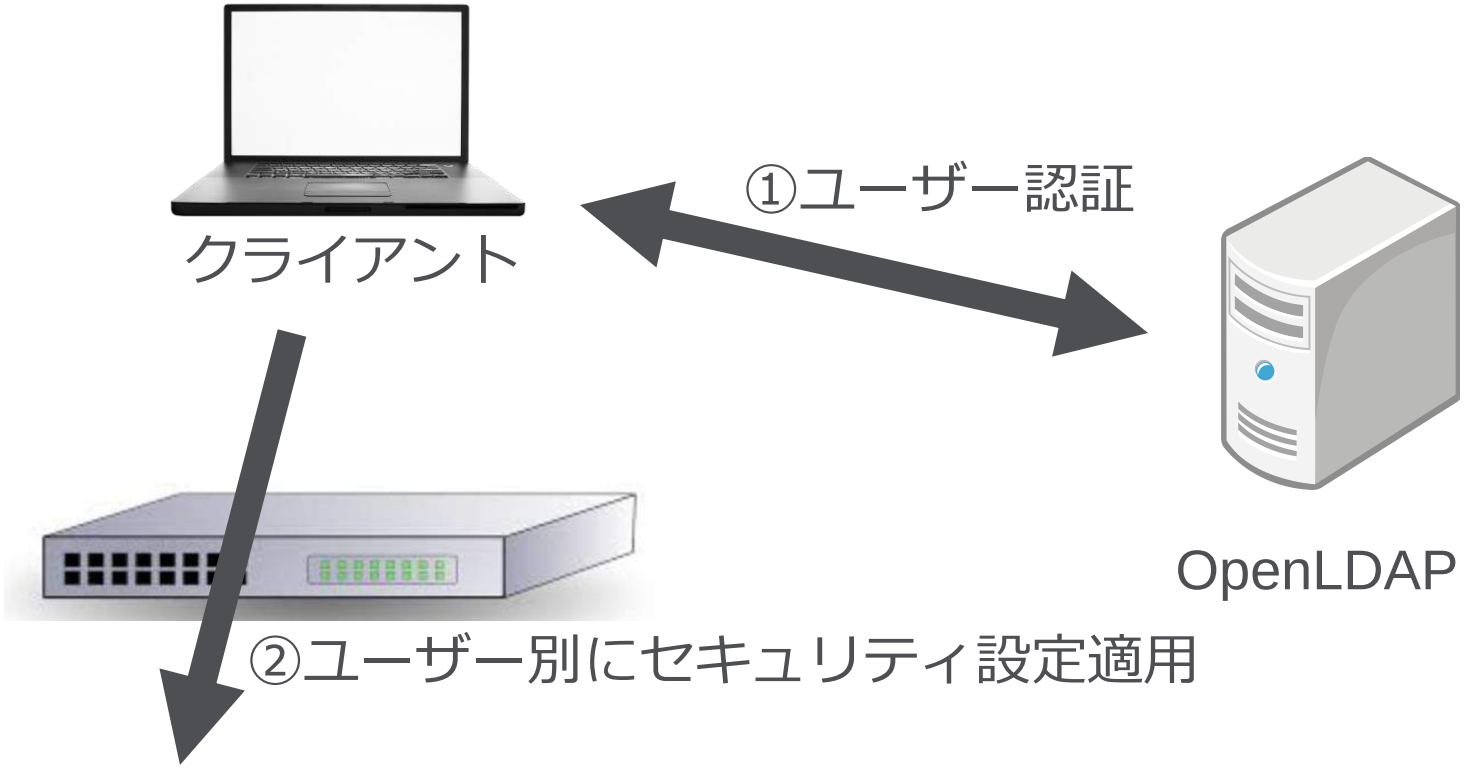
Apache & rubyでNW機器認証したい！



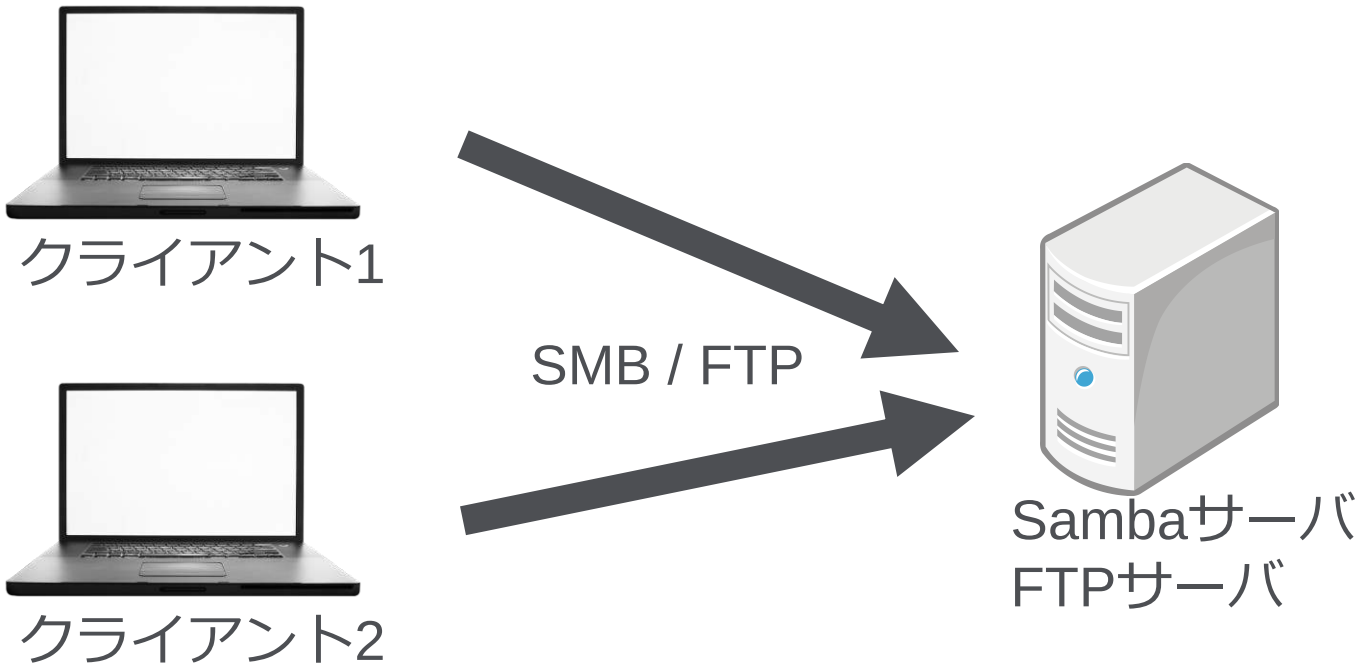
dhcpdでNW機器識別して、nextサーバへアクセスさせたい！



OpenLDAPでユーザ認証させて、NW機器の設定と連携させたい！



samba/ftpサーバで、社内で簡単にファイル共有したい！



資格取得後のスキル向上方法

1.業務でLinuxを選択出来るチャンスがあれば、選択する
　→業務課することで習熟度がかなり高まり、
　　経験としてアピール出来る

2.勉強会やITイベントなどに参加してみる
　→効率的に情報が入る、視野が広がる、トレンドがわかる

3.情報発信する（SNSなどで良い）
　→適度なプレッシャーがあり、
　　正確な情報ソースなどを把握する習慣が身につく

4.セミナーなど発表するチャンスがあれば、立候補する
　→資料作りや、それに関する検証・調査など本気で取り組める

みなさまのLinuC合格、
そして、また再会できることを
楽しみにしています！

Thank you !