

# LinuC レベル2 オンラインハンズオンセミナー

～Web編(nginx/Apache/Squid)～

2021/5/23 (Sun) 13:00-15:00

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エンジニアリスト

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。



Linus Torvaldsさん(Linux開発者)



Igor Sysoevさん(nginx開発者)



Alexei Vladishevさん(Zabbix開発者)  
© LPI-Japan / EDUCA all rights reserved.

早くリアル  
イベントが  
出来るよう  
になるとい  
いですね



# ゼウス・エンタープライズからのお知らせ

ZEUS  
enterprise

## 提供するITサービス

### リナックス・ネットワークグループ



リナックスに特化したエンジニアとヘルプデスクの常駐型支援サービスを行っています。研修は、エンジニアの育成に特化したLPI-Japanのアカデミック認定校「Zeus Linux Training Center / Zeus Network Training Center」で行います。資格取得の支援、新しい知識の習得に努め、つねに高水準のサービスの提供をめざしています。

コンピュータシステムの企画・開発・保守  
請負業務  
特定労働者派遣  
・ヘルプデスク  
・エンジニア



### アプリケーショングループ

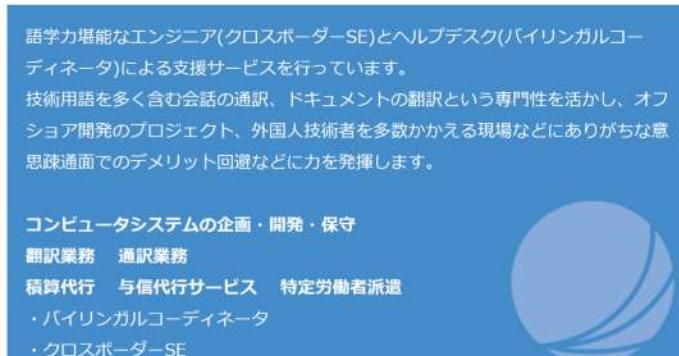


Webアプリケーションやお客様のシステムなど、プログラミング言語を用いた開発支援サービスを行っています。  
また、「失敗しないお店選び!」をモットーとしたグルメサイト「Gourmet True」を2020年1月にリリース。これを皮切りに様々なインターネットサービスも開発しています。



アプリケーション開発  
システム開発

### クロスボーダーSE&ITソリューショングループ



語学力堪能なエンジニア(クロスボーダーSE)とヘルプデスク(バイリンガルコーディネーター)による支援サービスを行っています。  
技術用語を多く含む会話の通訳、ドキュメントの翻訳という専門性を活かし、オフショア開発のプロジェクト、外国人技術者を多数かかえる現場などにありがちな意思疎通でのデメリット回避などに力を発揮します。

コンピュータシステムの企画・開発・保守  
翻訳業務 通訳業務  
積算代行 与信代行サービス 特定労働者派遣  
・バイリンガルコーディネーター  
・クロスボーダーSE



### Zeus Linux Training Center / Zeus Network Training Center

当社で培われた技術者向け研修カリキュラムを、一般のお客様や企業様向けに提供することを目的としたITキャリアスクール「Zeus Linux Training Center / Zeus Network Training Center」を運営しています。  
当スクールはLPI-Japanのアカデミック認定校です。未経験者からプロの技術者養成まで、「現場ですぐに活躍できる人材」の育成をめざした研修は、技術者を求める企業からも高い評価をいただいている。

教育事業 (BtoB, BtoC)  
IT・英会話・マナー  
社員教育  
試験会場

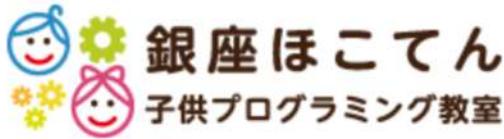


<https://www.zeus-enterprise.co.jp/solution/service>

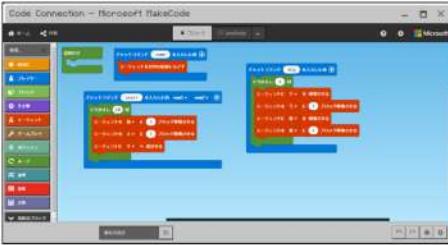




# ゼウス・エンタープライズからのお知らせ



## 提供するITサービス

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！
<b>COURSE</b> ～コース紹介～	
<b>Minecraftコース</b>	
<b>大人気ゲーム！Minecraftを使って楽しくプログラミング！</b>	
「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	

### 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・姉妹一緒に  
お申し込みで  
**10% OFF!**



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



# アジェンダ

1. LinuC Level2試験とは
  - 試験概要と特徴
2. 技術解説
  - サーバ構築のためのLinuxインストール
  - Webサーバ(nginx/Apache)
  - リバースプロキシサーバ(nginx)
  - プロキシサーバ(Squid)
3. ハンズオン
4. Appendix
5. Q&A



# 本日のゴール

- サーバ構築のために必要な知識を知る・理解する
- サーバ(Web/リバースプロキシ/プロキシ)を構築できるようになる
- LinuC受験対策を知る



# ハンズオンにおける注意事項

- PC(WindowsまたはMacなど)でWebブラウザを操作できる環境をご用意ください。
- 今回のセミナーでは、Katacoda／クラウドサービスを利用します。Katacodaでユーザ登録を済ませておいてください。Twitter、LinkedIn、Google、Githubのソーシャルログインが可能です。

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

The screenshot shows the Katacoda platform interface. At the top, there's a header with the O'Reilly Katacoda logo, navigation links for 'LEARN' and 'CREATE', and buttons for 'TRY O'REILLY' and 'LOG IN'. Below the header, a profile card for 'Submattnesk' is displayed, featuring a photo of a person making a peace sign, the name 'Submattnesk, LinuCエヴァンジェリスト', the handle '@submattnesk', and a brief bio mentioning Linux, nginx, python, Deep Learning, AI, and automation. To the right of the profile card is a 'Share Your Success' section with 'in Share' and 'Twitter Share' buttons. At the bottom left, a course card for 'LinuC Level2 Server Exercises' is shown, indicating '3 SCENARIOS' and 'For Studying Server Building', with a large orange 'Start Course' button at the bottom.



## ■ 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

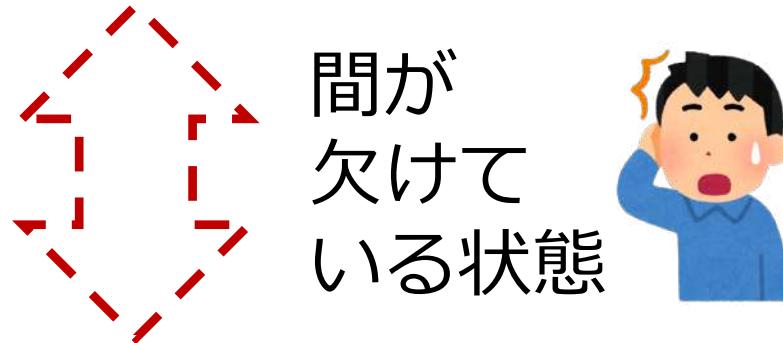
今となっては使わない技術やコマンドの削除、アップデート、新領域の取り込み

### ✓ Linuxの範疇だけにとどまらない領域までカバー

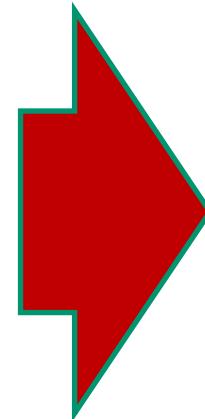
セキュリティや監視など、ITエンジニアであれば必須の領域もカバー



AWSなどの  
パブリッククラウドを  
活用するための技術



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**





# LinuC Level2試験とは

## 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

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

## 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/learn>



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。

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



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、よかったです登録してください！

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過去セミナーの動画

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

The screenshot shows the LPI-Japan YouTube channel page. At the top, there's a banner with the text "open your NEXT future" and logos for LinuC, OSS-DB Silver/Gold, HTML5 Professional Certification, Apache Cloud Stack Certification Exam, and OPCel. Below the banner, the channel name "LPI-Japan" is displayed with a red circular icon. The channel has 921 subscribers. A "チャンネル登録" (Subscribe) button is visible. The main content area shows a grid of video thumbnails for various seminars, each with a title, duration, and date. Below the grid, there are sections for "HTML5 技術解説無料セミナー" and "OSS-DB 技術解説無料セミナー". On the right side of the page, there's a QR code.

セミナー名	日付	概要	再生回数	登録者数
ハードディスクのレイアウトとパーティション (パーティションの構成とデータ)	[2021/01/23]	LinuC-1 技術解説セミナー「ハードウェア」	1:10:33	LPI-Japan 62回視聴・1日前
重要なシステムサービス (システムの構造)	[2021/1/17]	LinuC-1 技術解説セミナー「重要なシステム...」	54:08	LPI-Japan 136回視聴・6日前
システムの保守と運用管理 (リソースの使用状況の把握、監視ツール)	[2020/12/19]	LinuC-2 技術解説セミナー「システムの保...」	1:02:57	LPI-Japan 301回視聴・1か月前
ファイルシステムとストレージ管理 (ファイルシステムマウント、LVMなどの解説とデモ)	[2020/12/19]	LinuC-2 技術解説セミナー「ファイルシス...」	1:12:26	LPI-Japan 311回視聴・1か月前
GNUとUnixのコマンド (コマンド、パイプ、正規表現などの解説とデモ)	[2020/11/29]	LinuC-1 技術解説セミナー「GNUとUnixの...」	59:20	LPI-Japan 385回視聴・1か月前
アカウント管理とセキュリティ管理業務の実	[2020/11/07]	LinuC-1 技術解説セミナー「アカウント管...」	1:13:43	LPI-Japan 418回視聴・2か月前
オフライン・ストレージ系API (Web StorageやIndexed Database APIの実動を確認しよう)	[2020/07/12]	HTML5-1技術解説セミナー「オフライン・ストレージ系API概要」	1:00:11	LPI-Japan 646回視聴・6か月前
トランザクションの概念／SQLコマンド (開発／SQL)	[2020/09/17]	OSS-DB Exam Silver 技術解説無料セミナー「トランザクションの概念／SQLコマンド」	1:15:02	OSS-DB Exam Silver 技術解説無料セミナー「トランザクションの概念／SQLコマンド」
VACUUM,ANALYZEの目的と使い方 (運用管理)	[2020/09/05]	OSS-DB Exam Silver 技術解説セミナー「VACUUM,ANALYZEの目的と使い方」	1:07:22	OSS-DB Exam Silver 技術解説セミナー「VACUUM,ANALYZEの目的と使い方」
バックアップ方法 (運用管理)	[2020/07/19]	OSS-DB Silver 技術解説セミナー「バックアップ方法」	1:17:12	OSS-DB Silver 技術解説セミナー「バックアップ方法」



## 学習の具体的な進め方(2~3か月程度)

試験範囲の確認(LinuC HP)



LinuC認定教材の購入・1週目読込



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！



LinuC認定教材 2週目読込



問題集やメルマガサンプル問題で理解力確認

※理解不足箇所の洗い出し



LinuC認定教材 3週目

※弱点の補強



受験申込



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



## 2.09.1 Apache HTTPサーバーの設定と管理

重要度 3

### 概要

Apache HTTP サーバーのインストールと設定ができる。これには、サーバーの負荷と性能の監視、クライアントからのユーザアクセスの制限、モジュールとしてのスクリプト言語をサポートする設定、およびクライアントユーザの認証設定も含まれる。また、サーバーのオプション設定でリソースの使用を制限することも含まれる。仮想ホストを使用するようApache HTTP サーバーを設定し、ファイルへのアクセスをカスタマイズできる。

### 詳細

Apache HTTP サーバー の設定ファイル、用語、ユーティリティ  
httpd, apache2

httpd.conf, mod\_auth\_basic, mod\_authz\_host  
apachectl, apache2ctl

Apache HTTP サーバーのログファイルの設定と内容

アクセスログとエラーログ

アクセス制限の方法とファイル

.htaccess, AuthUserFile, AuthGroupFile

クライアントユーザを認証するファイルとユーティリティ

htpasswd

最大リクエスト数、最小/最大サーバー数およびクライアント数の設定

Apache HTTP サーバー における仮想ホストの実装

ファイルへのアクセスをカスタマイズするために、Apache HTTP サーバーの設定ファイルで Redirect 文を使用する。



## 2.09.3 nginxの設定と管理

重要度 3

### 概要

リバースプロキシサーバーであるnginxのインストールおよび設定ができる。これには、HTTPサーバーとしてのnginx の設定が含まれる。

### 詳細

nginx の設定と管理

/etc/nginx/, nginx

nginx のSSL設定

ssl, ssl\_certificate, ssl\_certificate\_key, ssl\_ciphers, ssl\_protocols

リバースプロキシサーバーとしての設定

proxy\_pass, proxy\_http\_version, proxy\_set\_header

nginx でリダイレクトを行う



## 2.09.4 Squidの設定と管理

重要度 2

### 概要

プロキシサーバーのインストールと設定ができる。これには、アクセスポリシー、認証、リソースの利用方法も含まれる。

### 詳細

Squid 3.xの設定ファイル、用語、ユーティリティ

  squid.conf, squidclient

アクセス制限の方法

  http\_access

クライアントユーザの認証方法

Squid設定ファイルにおけるACLのレイアウトと内容

  acl



## サーバ構築のためのLinuxインストール



学習用途でサーバを構築する場合、  
構築手順・設定方法・動作確認などが出来ればよく、  
物理マシン/仮想マシン(VM)で最低限のスペックが用意できればよい。

例えば、  
CPU : 2Cores  
メモリー : 2048MB  
ストレージ : 20GB  
NIC : 1つ





しかし、業務でサーバ構築をする場合、考慮すべき点があります。

## ①利用するソフトウェアが必要とするハードウェア要件

### ハードウェア設定の例

以下の表に、様々なハードウェア設定の例を示します：

名前	プラットフォーム	CPU/メモリ	データベース	監視するホスト数
小規模	Ubuntu Linux	PII 350MHz 256MB	SQLite	20
中規模	Ubuntu Linux 64 bit	AMD Athlon 3200 + 2GB	MySQL InnoDB	500
大規模	Ubuntu Linux 64 bit	Intel Dual Core 6400 4GB	RAID10 MySQL InnoDBまたはPostgreSQL	1000以上
非常に大規模	RedHat Enterprise	Intel Xeon 2xCPU 8GB	Fast RAID10 MySQL InnoDBまたはPostgreSQL	10000以上

<https://www.zabbix.com/documentation/2.2/jp/manual/installation/requirements>



## ②利用するソフトウェアが必要とするソフトウェア要件

データベース管理システム

ソフトウェア	バージョン	コメント
MySQL	5.0.3以降	MySQLをZabbixバックエンドデータベースとして使用する場合に必要です。InnoDBエンジンが必要です。
Oracle	10g以降	OracleをZabbixバックエンドデータベースとして使用する場合に必要です。
PostgreSQL	8.1以降	PostgreSQL をZabbixのバックエンドデータベースとして使用する場合に必要です。パフォーマンスがある場合は、少なくとも PostgreSQL 8.3で使用することをお奨めします。
SQLite	3.3.5以降	SQLiteをZabbixバックエンドデータベースとして使用する場合に必要です。
IBM DB2	9.7以降	IBM DB2をZabbixバックエンドデータベースとして使用する場合に必要です。

### フロントエンド

Zabbix フロントエンドを動作させるには、次のソフトウェアが必要です：

ソフトウェア	バージョン	コメント
Apache	1.3.12以降	
PHP	5.3.0以降	PHP v7はサポートしていません。

### PHPエクステンション:

gd	2.0以降	PHP GD エクステンションには、PNG形式の画像 ( <code>--with-png-dir</code> )、JPEG形式の画像 ( <code>--with-jpeg-dir</code> ) およびFreeType2 ( <code>--with-freetype-dir</code> ) のサポートが必要です。
bcmath		php-bcmath ( <code>--enable-bcmath</code> )
ctype		php-ctype ( <code>--enable-ctype</code> )
libXML	2.6.15以降	ディストリビュータから別パッケージとして提供されている場合はphp-xml または php5-dom
xmlreader		ディストリビュータから別パッケージとして提供されている場合はphp-xmlreader
xmlwriter		ディストリビュータから別パッケージとして提供されている場合はphp-xmlwriter
session		ディストリビュータから別パッケージとして提供されている場合はphp-session
sockets		php-net-socket ( <code>--enable-sockets</code> ) ユーザースクリプトのサポートが必要です。
mbstring		php-mbstring ( <code>--enable-mbstring</code> )
gettext		php-gettext ( <code>--with-gettext</code> ) 翻訳機能を動作させるために必要です。
ibm_db2		IBM DB2をZabbixバックエンドデータベースとして使用する場合に必要です。
mysql		MySQLをZabbixバックエンドデータベースとして使用する場合に必要です。
oci8		OracleをZabbixバックエンドデータベースとして使用する場合に必要です。
pgsql		PostgreSQLをZabbixバックエンドデータベースとして使用する場合に必要です。
sqlite3		SQLiteをZabbixバックエンドデータベースとして使用する場合に必要です。

[Back to top](#)



### ③パーティション構成

/var

ログファイルなどサイズが肥大化するファイルが配置されるので、  
ログローテーション(世代管理など)・最大ファイルサイズなど管理する必要ある。  
パーティション容量計算・バックアップ管理も必要。  
独立したパーティションとする。

LVM(Logical Volume Manager)を利用するなどして、  
容量の増加に備える。

CentOS7でLVMを構成する

<https://www.opensourcetech.tokyo/entry/20201119/1605758811>

LVM(Logical Volume Manager)でディスク容量を増加させる on CentOS8

<https://www.opensourcetech.tokyo/entry/20200724/1595552312>



## ④冗長構成

ストレージやNICを冗長し、障害に備える。

CentOS7でNICのチーミング(インストール時)を設定する

<https://www.opensourcetech.tokyo/entry/20200604/1591282780>

DRBD9/Pacemaker1.1/Corocync2/PCSによるストレージ冗長構成構築 on CentOS7

<https://www.opensourcetech.tokyo/entry/20200527/1590505446>



## ⑤その他

電源管理(冗長)

空調管理

監視/運用管理・障害対応

拠点冗長(オンプレ/クラウド)

アカウント管理・権限管理

セキュリティ

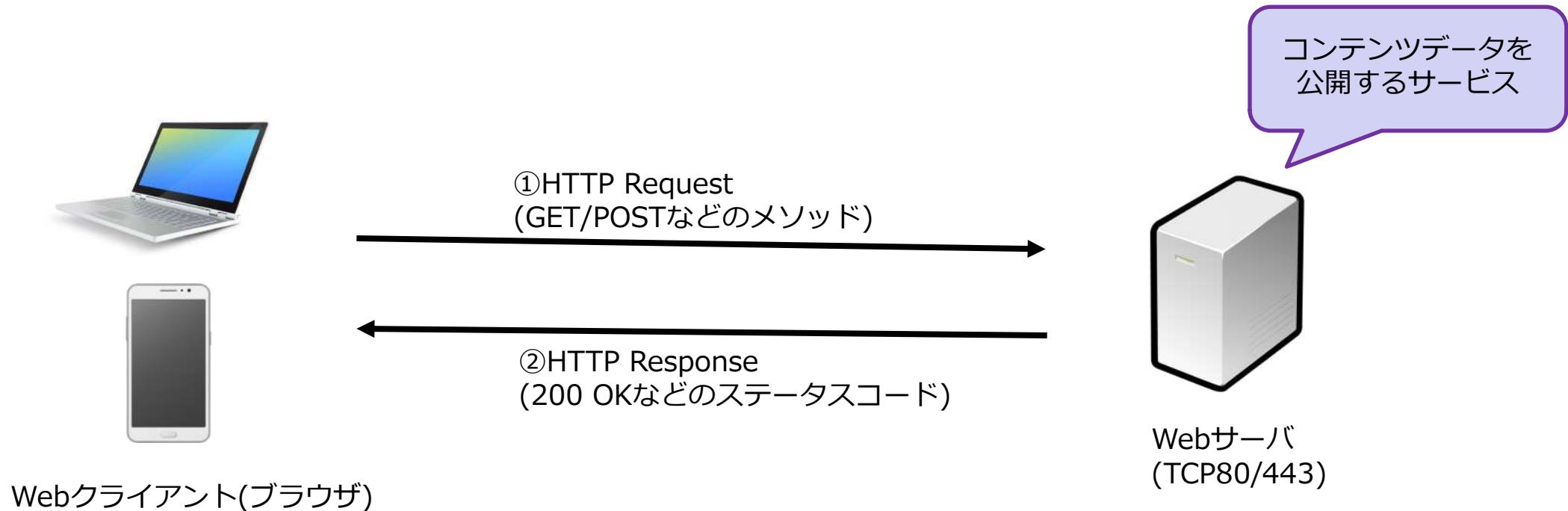
and more



## Webサーバ(nginx/Apache)



## Web(http/https)の通信





## HTTPリクエストメソッド

メソッド	意味
GET	データ要求
POST	クライアントからデータ送信
HEAD	HTTPヘッダを要求
PUT	サーバ上へファイルを配置
DELETE	サーバ上のファイルを削除

<https://www.w3.org/Protocols/rfc2616/rfc2616-sec9.html>



## HTTPステータスコード

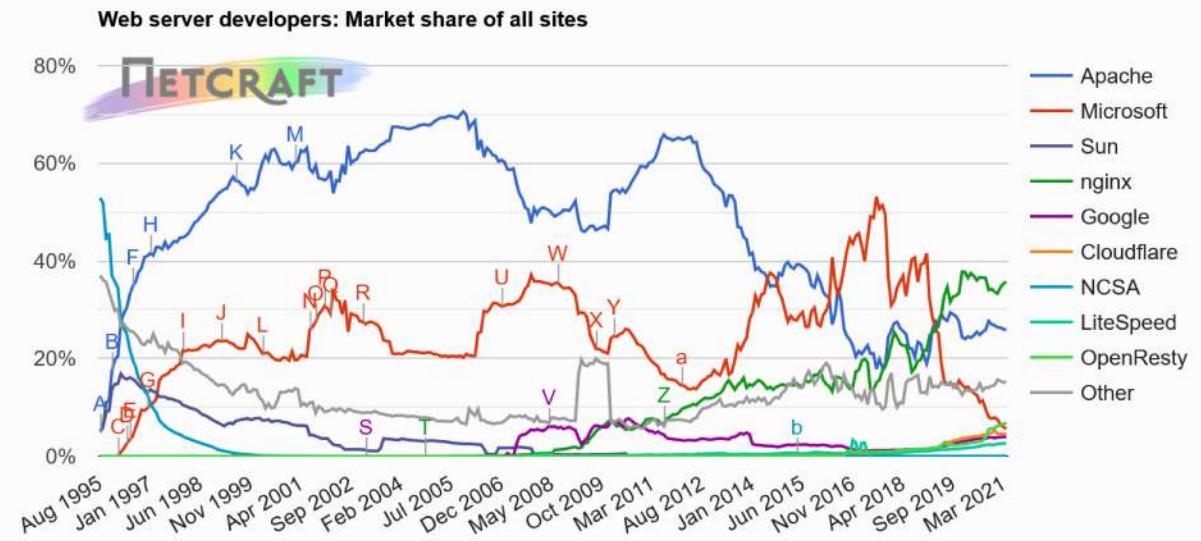
ステータスコード	意味	例
100番台	情報レスポンス	100 Continue など
200番台	成功レスポンス	200 OK など
300番台	リダイレクションメッセージ	301 Moved Permanently 304 Not Modified など
400番台	クライアントエラーレスpons	401 Unauthorized 403 Forbidden 404 Not Found など
500番台	サーバエラーレスpons	500 Internal Server Error 502 Bad Gateway 503 Service Unavailable など

<https://www.w3.org/Protocols/rfc2616/rfc2616-sec10.html>

構築後のエラー原因  
を判断するのに重要



## WebサイトにおけるWebサーバシェア(2021/3)



Developer	March 2021	Percent	April 2021	Percent	Change
nginx	419,637,923	35.34%	432,167,302	35.65%	0.32
Apache	308,509,042	25.98%	313,948,741	25.90%	-0.08
OpenResty	77,819,490	6.55%	81,935,391	6.76%	0.21
Microsoft	70,826,342	5.96%	67,182,740	5.54%	-0.42

<https://news.netcraft.com/archives/category/web-server-survey/>



## nginx

Igor Sysoevさんが開発

2004/10/4 0.1.0 release

2021/5/16現在 1.19.10が最新バージョン

Webサーバの他、リバースプロキシ・メールプロキシなどとしても利用可能



Igor Sysoevさん(nginx開発者)

<http://nginx.org/>

## nginx news

2021-04-20 [nginx-1.20.0](#) stable version has been released, incorporating new features and bug fixes from the 1.19.x mainline branch — including [OCSP validation](#) of client SSL certificates, the [ssl\\_reject\\_handshake](#) and [ssl\\_conf\\_command](#) directives, simplified and improved handling of HTTP/2 connections with the [lingering\\_close](#), [keepalive\\_timeout](#), and [keepalive\\_requests](#) directives, the [keepalive\\_time](#) directive, stricter handling of upstream server responses, [cookie\\_flags](#) handling, cache clearing based on the [minimum amount of free space](#), PROXY protocol support [from clients](#) and [to backend servers](#) in the mail proxy, [proxying SMTP authentication](#), the [set](#) directive in the stream module, and more.

2021-04-13 [nginx-1.19.10](#) mainline version has been released.

2021-03-30 [nginx-1.19.9](#) mainline version has been released.

2021-03-30 [njs-0.5.3](#) version has been [released](#), featuring the `js_var` directive for [http](#) and [stream](#).

2021-03-25 [unit-1.23.0](#) version has been [released](#), featuring SNI support and a number of bugfixes.

NGINX

[english](#)  
[русский](#)

[news](#)  
[2020](#)  
[2019](#)  
[2018](#)  
[2017](#)  
[2016](#)  
[2015](#)  
[2014](#)  
[2013](#)  
[2012](#)  
[2011](#)  
[2010](#)  
[2009](#)  
[about](#)



# Apache

Apacheソフトウェア財団支援のもと、コミュニティベースで開発



**APACHE**  
HTTP SERVER PROJECT

**APACHECON** September 21-23 2021 [www.apachecon.com](http://www.apachecon.com)

**The Number One HTTP Server On The Internet**

The Apache HTTP Server Project is an effort to develop and maintain an open-source HTTP server for modern operating systems including UNIX and Windows. The goal of this project is to provide a secure, efficient and extensible server that provides HTTP services in sync with the current HTTP standards.

The Apache HTTP Server ("httpd") was launched in 1995 and it has been the most popular web server on the Internet since April 1996. It has celebrated its 25th birthday as a project in February 2020.

The Apache HTTP Server is a project of [The Apache Software Foundation](#).

**Apache httpd 2.4.46 Released** 2020-08-07

The Apache Software Foundation and the Apache HTTP Server Project are pleased to [announce](#) the release of version 2.4.46 of the Apache HTTP Server ("httpd").

This latest release from the 2.4.x stable branch represents the best available version of Apache HTTP Server.

Apache HTTP Server version 2.4.43 or newer is required in order to operate a TLS 1.3 web server with OpenSSL 1.1.1.

[Download](#) | [ChangeLog for 2.4.46](#) | [Complete ChangeLog for 2.4](#) | [New Features in httpd 2.4](#)

**Apache httpd 2.2 End-of-Life** 2018-01-01

As previously announced, the Apache HTTP Server Project has discontinued all development and patch review of the 2.2.x series of releases.

The Apache HTTP Server Project had long committed to provide maintenance releases of the 2.2.x flavor through June of 2017. The final release 2.2.34 was published in July 2017, and no further evaluation of bug reports or security risks will be considered or published for 2.2.x releases.

<https://httpd.apache.org/>



## ソフトウェア(nginx/Apache/Squid)のインストールから起動まで

1. インストール(yum・aptなどパッケージ管理・ソースコードなど)
2. 設定ファイル編集
3. コンテンツファイル配置
4. 起動

※iptables・firewalld・ufwなどのファイアウォール/SELinuxなどのセキュリティモジュールは別途実施。



nginxの設定(nginx.conf)

```
user nginx;      ... プロセス起動ユーザー
worker_processes 1;    ... 起動するワーカプロセス数(CPUコア数と同様など)

error_log /var/log/nginx/error.log warn;    ... エラーログ
pid      /var/run/nginx.pid;    ... プロセスIDを書き込むファイル

events {
    worker_connections 1024;    ... ワーカープロセスが同時オープン可能なコネクション数
}

http {
    include      /etc/nginx/mime.types;
    default_type application/octet-stream;
    log_format main '$remote_addr - $remote_user [$time_local] "$request"'
                    '$status $body_bytes_sent "$http_referer"'
                    '"$http_user_agent" "$http_x_forwarded_for"';    ... ログフォーマット
    access_log /var/log/nginx/access.log main;    ... アクセスログ
    sendfile on;
    keepalive_timeout 65;
    include /etc/nginx/conf.d/*.conf;    ... 外部参照設定ファイル指定
}
```



nginxの設定(/etc/nginx/conf.d/default.conf)

```
server {  
    listen     80;  ··· 公開ポート番号  
    listen  [::]:80;  
    server_name www.testweb.local;  ··· サーバ名  
    location / {  
        root   /usr/share/nginx/html;  ··· ルートディレクトリ(コンテンツファイル格納場所)  
        index  index.html index.htm;  ··· クライアントへ提供するコンテンツファイル名  
    }  
    error_page 500 502 503 504 /50x.html;  
    location = /50x.html {  
        root   /usr/share/nginx/html;  
    }  
}
```



## Apacheの設定(httdp.conf)

```
ServerRoot "/etc/httpd"  
Listen 80    ··· 公開するポート番号  
Include conf.modules.d/*.conf  ··· モジュールに関する設定ファイル格納場所
```

```
User apache  
Group apache  
ServerAdmin root@localhost  
DocumentRoot "/var/www/html"  ··· ルートディレクトリ(コンテンツファイル格納場所)
```

```
<Directory "/var/www/html">  
    AllowOverride None  
</Directory>  
  
<IfModule dir_module>  
    DirectoryIndex index.html  ··· クライアントへ提供するコンテンツファイル名  
</IfModule>
```

```
ErrorLog "logs/error_log"  ··· エラーログ  
LogLevel warn
```

```
<IfModule log_config_module>  
    LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-Agent}i\"" combined  ··· ログフォーマット定義  
    CustomLog "logs/access_log" combined  ··· アクセスログ  
</IfModule>
```

```
IncludeOptional conf.d/*.conf  ··· 外部参照設定ファイル指定
```



## トラブルシュートに便利なツールなど

### パケットキャプチャソフト

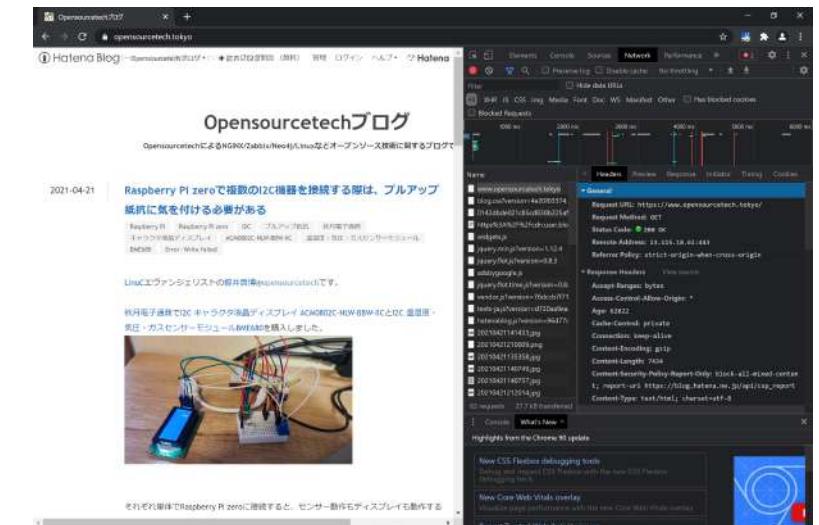
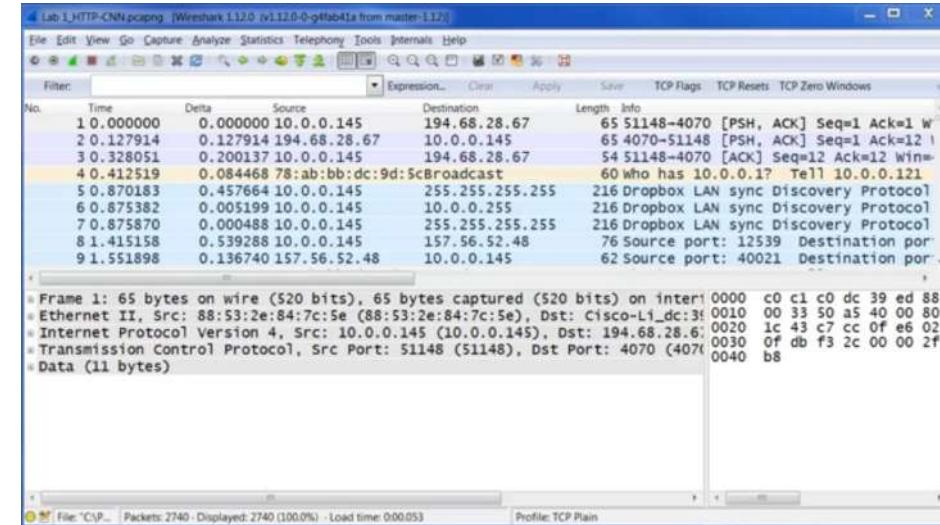
- Wireshark(<https://www.wireshark.org/>)
- tcpdump(<https://linuxjm.osdn.jp/html/tcpdump/man1/tcpdump.1.html>) など

### HTTPヘッダ確認ツール

- chromeデベロッパーツール
- LiveHTTPHeaders など

### ポートスキャンツール

- nmap など

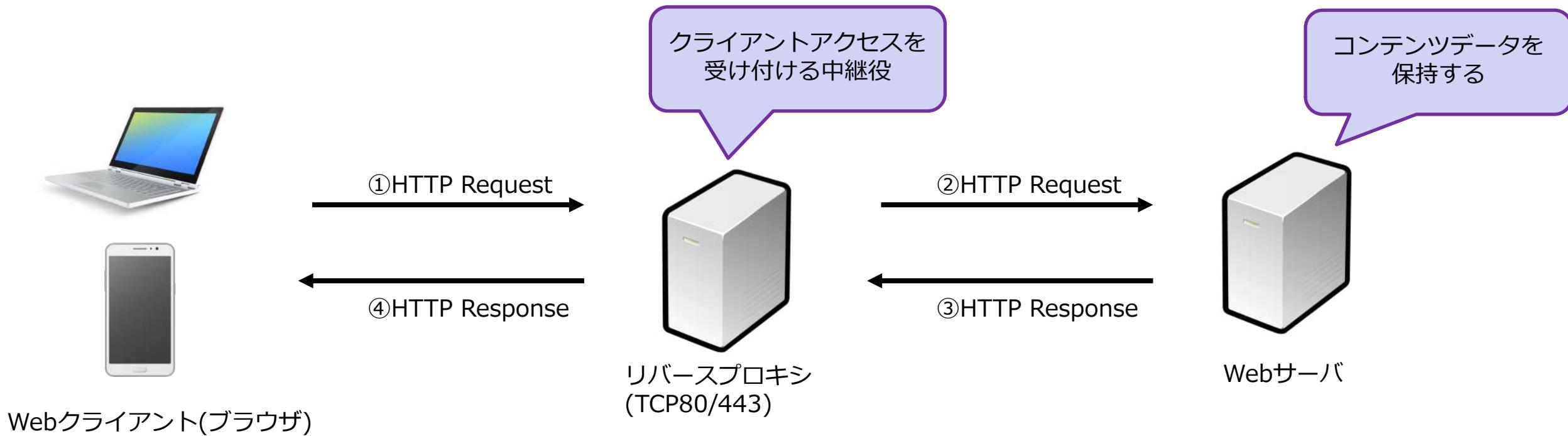




## リバースプロキシサーバ(nginx)

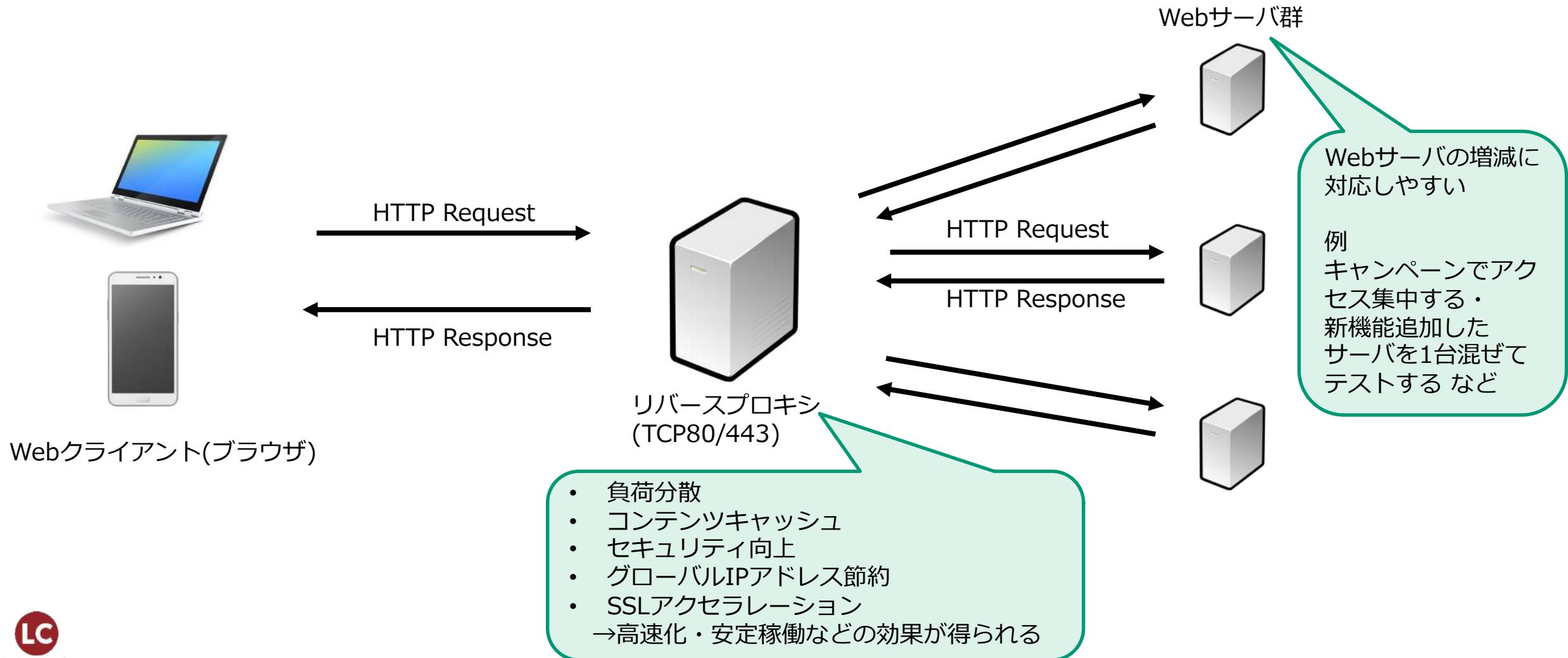


## リバースプロキシを使ったWeb通信





## リバースプロキシを使うメリット





nginxの設定(nginx.conf) ※リバースプロキシ部分のみ抜粋

```
upstream apache {    ··· Webサーバ群(バックエンド)に関する設定
    server server1:80;
    server server2:80;
    server server3:80;
}
```

server { ··· リバースプロキシ(フロントエンド)に関する設定

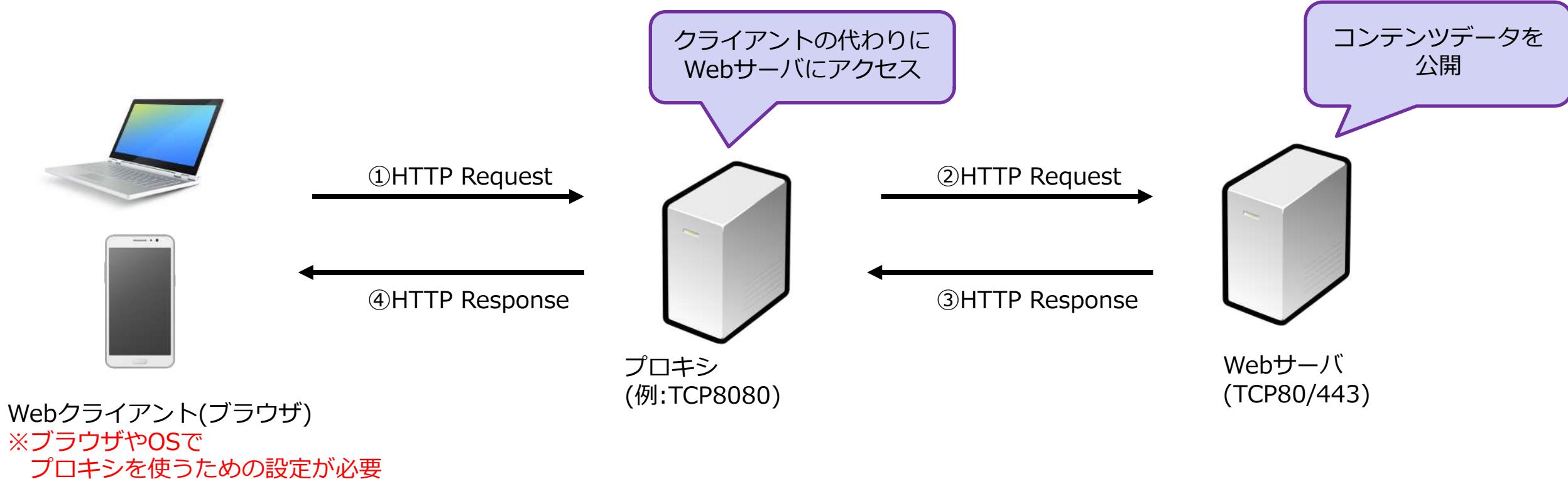
```
listen 80;
location / {
    proxy_pass http://apache;
    proxy_http_version 1.1;
    proxy_set_header Host $host:$server_port;
    proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;          ··· WebサーバのアクセスログにクライアントのIPアドレスを残す
    proxy_set_header X-Forwarded-Proto http;
}
}
```



## プロキシサーバ(Squid)

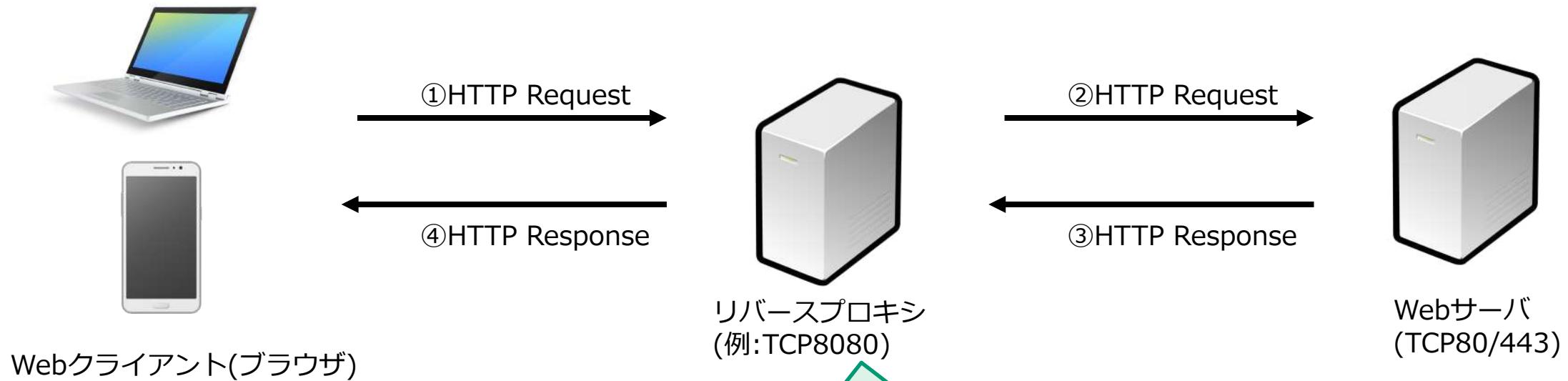


## プロキシを使ったWeb通信





## プロキシを使うメリット



- アクセス制限
- アクセスログ監視
- コンテンツキャッシング  
→通信高速化・セキュリティ向上などの効果が得られる



## Squid

Web通信最適化のため、プロキシサーバ・コンテンツキャッシュとして利用される

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



squidの設定(squid.conf) ※一部抜粋

```
acl localnet src 10.0.0.0/8    # RFC1918 possible internal network
acl localnet src 172.16.0.0/12   # RFC1918 possible internal network
acl localnet src 192.168.0.0/16  # RFC1918 possible internal network
acl localnet src fc00::/7       # RFC 4193 local private network range
acl localnet src fe80::/10      # RFC 4291 link-local (directly plugged) machines
· · · "acl acl名 通信対象" という形式で条件を定義
```

```
acl SSL_ports port 443
acl Safe_ports port 80          # http
acl Safe_ports port 443         # https
acl CONNECT method CONNECT
```

```
http_access deny !Safe_ports
http_access deny CONNECT !SSL_ports
http_access allow localnet
http_access allow localhost
http_access deny all
```

· · · aclで定義された内容の通信許可(allow)/通信拒否(deny)を設定

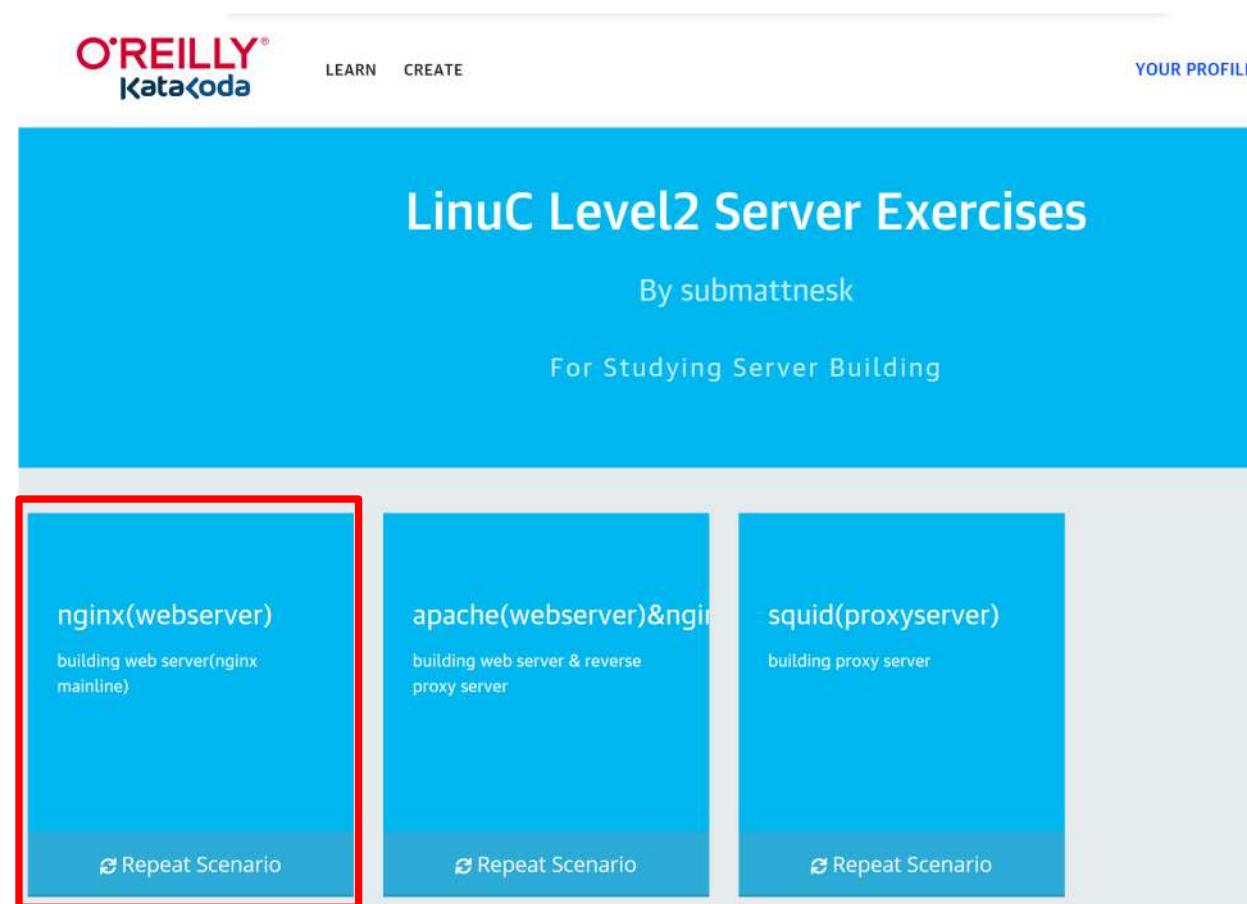
http\_port 8080 · · · プロキシサーバで使用するポート番号



# ハンズオン



まずは、  
nginxによるWebサーバ構築にTry♪  
<https://www.katacoda.com/submattnesk/courses/courses>



The screenshot shows the Katacoda platform interface. At the top, there's a navigation bar with the O'REILLY Katacoda logo, 'LEARN CREATE' buttons, and a 'YOUR PROFILE' link. The main title is 'LinuC Level2 Server Exercises' by 'submattnesk' for 'For Studying Server Building'. Below the title, three scenarios are listed in cards:

- nginx(webserver)**  
building web server(nginx mainline)  
[Repeat Scenario](#)
- apache(webserver)&nginx**  
building web server & reverse proxy server  
[Repeat Scenario](#)
- squid(proxyserver)**  
building proxy server  
[Repeat Scenario](#)

A red box highlights the first scenario, 'nginx(webserver)'. The Katacoda logo is visible in the bottom left corner.



続いて、  
nginxによるリバースプロキシサーバ &  
ApacheによるWebサーバ構築にTry♪  
<https://www.katacoda.com/submattnesk/courses/courses>

The screenshot shows the Katacoda interface for the LinuC Level2 Server Exercises. At the top, the O'Reilly Katacoda logo, 'LEARN CREATE', and 'YOUR PROFILE' are visible. The main title is 'LinuC Level2 Server Exercises' by 'submattnesk' for 'For Studying Server Building'. Below the title, three scenarios are listed: 'nginx(webserver)' (building web server(nginx mainline)), 'apache(webserver)&nginx' (building web server & reverse proxy server), and 'squid(proxyserver)' (building proxy server). The 'apache(webserver)&nginx' scenario is highlighted with a red border. Each scenario has a 'Repeat Scenario' button at the bottom.

Scenario	Description
nginx(webserver)	building web server(nginx mainline)
apache(webserver)&nginx	building web server & reverse proxy server
squid(proxyserver)	building proxy server



最後は、  
Squidによるプロキシサーバ構築にTry♪  
<https://www.katacoda.com/submattnesk/courses/courses>

The screenshot shows a web interface for Katacoda exercises. At the top, there's a navigation bar with the O'REILLY Katacoda logo, 'LEARN CREATE' buttons, and a 'YOUR PROFILE' link. The main title is 'LinuC Level2 Server Exercises' by 'submattnesk' for 'For Studying Server Building'. Below the title, three scenarios are listed in cards:

- nginx(webserver)**: building web server(nginx mainline). [Repeat Scenario](#)
- apache(webserver)&nginx**: building web server & reverse proxy server. [Repeat Scenario](#)
- squid(proxyserver)**: building proxy server. [Repeat Scenario](#)

The 'squid(proxyserver)' card is highlighted with a red border.



# Appendix



# Appendix

## 参考リンクなど

nginx 1.7.10(リバースプロキシ)&Apache(Webサーバ)を使用したロードバランシング(負荷分散)

[https://www.opensourcetech.tokyo/entry/2015/02/13/Nginx1\\_7\\_10%28%E3%83%AA%E3%83%90%E3%83%BC%E3%82%B9%E3%83%97%E3%83%AD%E3%82%AD%E3%82%B7%29%26Apache%28Web%E3%82%B5%E3%83%BC%E3%83%90%29%E3%82%92%E4%BD%BF%E7%94%A8%E3%81%97%E3%81%9F%E3%83%AD](https://www.opensourcetech.tokyo/entry/2015/02/13/Nginx1_7_10%28%E3%83%AA%E3%83%90%E3%83%BC%E3%82%B9%E3%83%97%E3%83%AD%E3%82%AD%E3%82%B7%29%26Apache%28Web%E3%82%B5%E3%83%BC%E3%83%90%29%E3%82%92%E4%BD%BF%E7%94%A8%E3%81%97%E3%81%9F%E3%83%AD)

nginx 1.7.3 with SSL & SPDY on CentOS7 (on virtualbox)

[https://www.opensourcetech.tokyo/entry/2014/07/30/nginx1\\_7\\_3\\_with\\_SSL%26SPDY\\_on\\_CentOS7\\_%28on\\_virtualbox%29](https://www.opensourcetech.tokyo/entry/2014/07/30/nginx1_7_3_with_SSL%26SPDY_on_CentOS7_%28on_virtualbox%29)

nginx本家サイト

<http://nginx.org/>

Apache本家サイト

<https://httpd.apache.org/>

Squid本家サイト

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

また、ドメインやFQDNを管理するDNSについて知っておくことも重要！



# 本日のまとめ



# 本日のまとめ

- ・ サーバ構築する上で知っておくべきこと
- ・ Webサーバの構築
- ・ リバースプロキシサーバの構築
- ・ プロキシサーバの構築



サーバが構築するのって、  
めっちゃ楽しいやん！

よっしゃ  
LinuC2ゲットするで！！



# Q & A



*Thank you for join today's seminar!*



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

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

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