

クレジット:

UTokyo Online Education Education コンピュータシステム概論 2018 小林克志

ライセンス:

利用者は、本講義資料を、教育的な目的に限ってページ単位で利用することができます。特に記載のない限り、本講義資料はページ単位でクリエイティブ・コモンズ 表示-非営利-改変禁止 ライセンスの下に提供されています。

<http://creativecommons.org/licenses/by-nc-nd/4.0/>

本講義資料内には、東京大学が第三者より許諾を得て利用している画像等や、各種ライセンスによって提供されている画像等が含まれています。個々の画像等を本講義資料から切り離して利用することはできません。個々の画像等の利用については、それぞれの権利者の定めるところに従ってください。



# コンピュータシステム概論 第8回

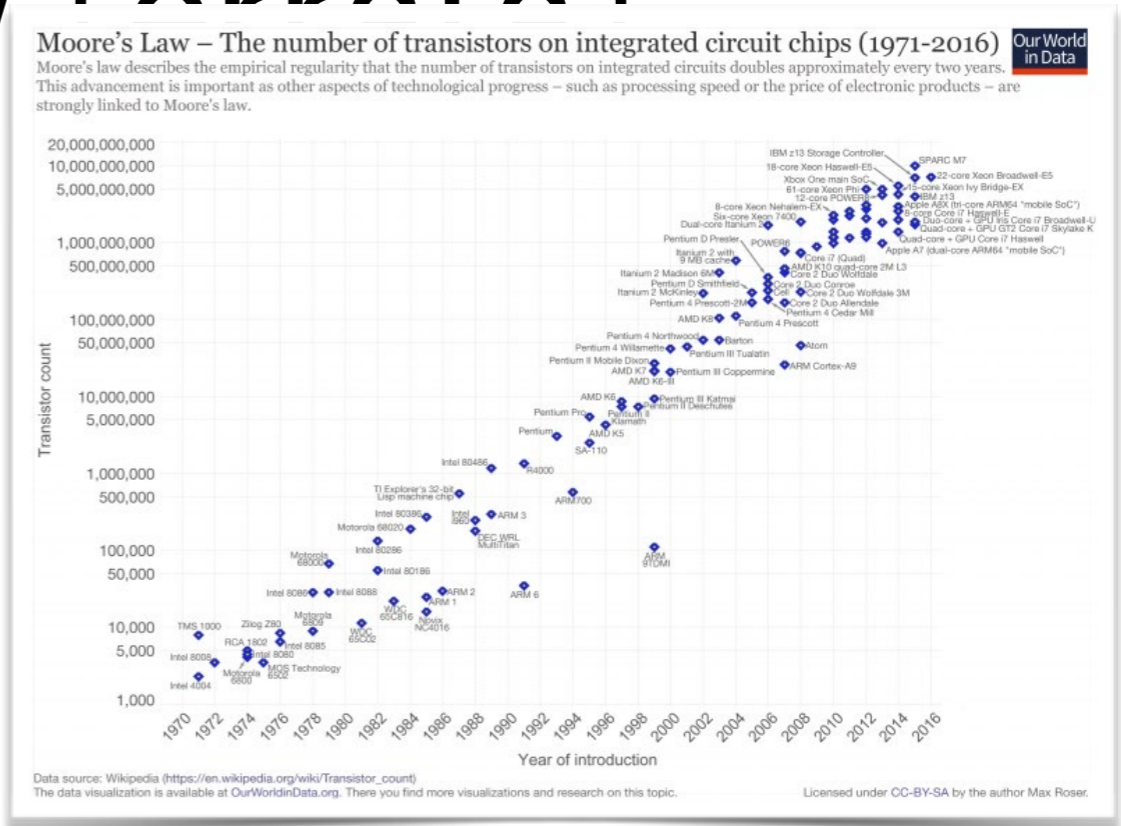
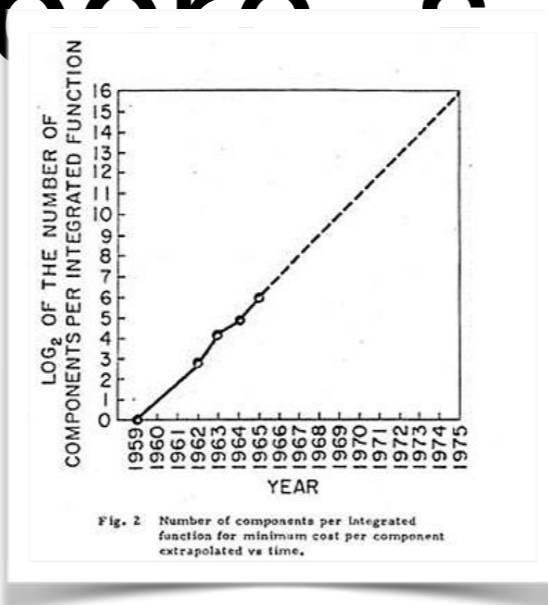
小林克志

- 事務連絡
- 先週の課題、レビュー(振り返り)
- まくら
- ログ管理
- Domain Name System
- Web サービス
  - ・ HTTP / セッション管理
  - ・ 多層アーキテクチャ
- 演習: Wordpress インストール
- データベース

# Moore's law (correct)

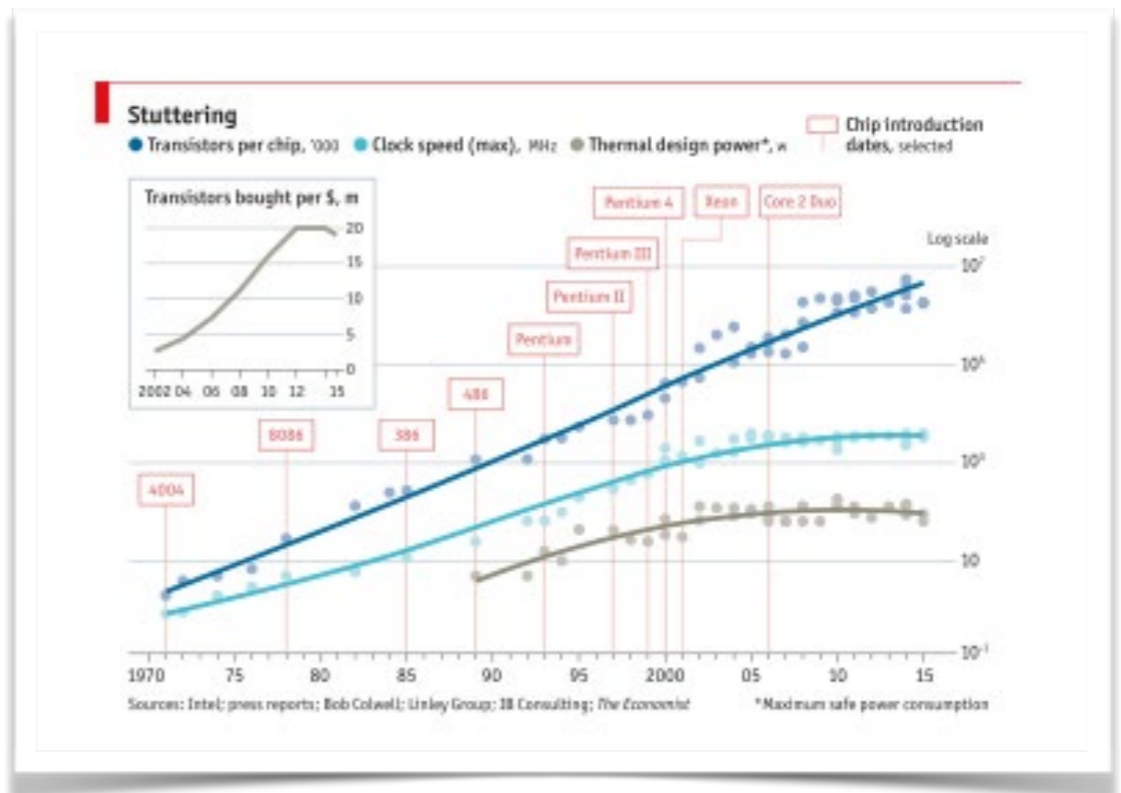
<https://ourworldindata.org/technological-progress/#note-2>, CC BY-SA by the author Max Roster

[http://www.intel.com/pressroom/kits/events/moores\\_law\\_40th/index.htm?iid=tech\\_mooreslaw+body\\_presskit](http://www.intel.com/pressroom/kits/events/moores_law_40th/index.htm?iid=tech_mooreslaw+body_presskit)



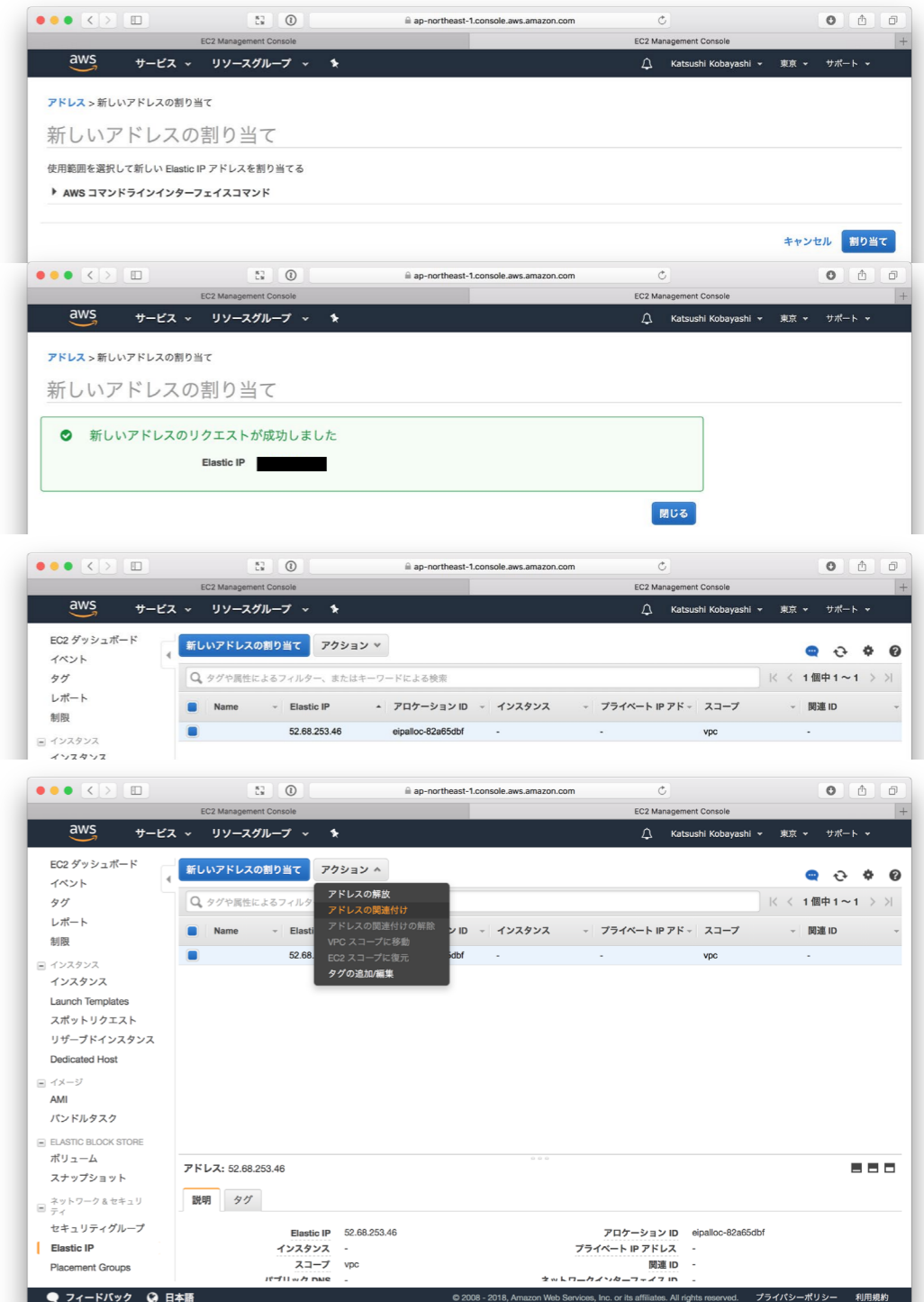
- A doubling every 1.5 year in the number of components per integrated circuit.
- Not covers CPU speed, such as clock rates, single threads performance.

<http://www.economist.com/technology-quarterly/2016-03-12/after-moores-law>



# Elastic IP の取得・関連付け2/3

- アドレス → 新しいアドレスの割り当てで、割り当てをクリック、成功すればページを閉じる。
- ここで、割り当てられた IP アドレスを講義の Web から登録する。(これを出席とする)
- EC2 → ネットワーク&セキュリティ → Elastic IP で、Elastic IP を選択
- アクション → アドレスの関連付けをクリック



# What type of document they signed ?

著作権の都合により  
ここに挿入されていた画像を削除しました

2018年6月12日 米朝首脳会談 写真  
トランプ大統領と金正恩氏の握手

# すべて「標準」

- Standard
  - ISO (International Organization of Standard)
  - IEEE-SA (Institute of Electrical and Electronics Engineers Standards Association)
- Recommendation
  - ITU (International Telecommunication Union)
  - W3C (World Wide Web Consortium)
- RFC(Request For Comments)
  - IETF (Internet Engineering Task Force)

# ログ管理

- コンピュータサービスではアクセスログの管理は必須となる。以下のような技術・管理目的に利用される:
  - サービス品質管理
  - 不正アクセス対策
  - 発信者情報開示請求などへ対応
- 時刻合わせは必須(Network Time Protocol (NTP): RFC5905):
  - 複数のサーバのログの突き合わせ
- 粒度・保存期間はケースバイケース:
  - 短すぎるのも問題だが、長すぎるのもプライバシー侵害のリスクが指摘される。
- ログを別のサーバに送ることもよくある (Syslog : RFC5424)。
  - 多くのサーバログをまとめて監視・管理したい。
  - ログの改ざん・消去リスクを抑える。



# Amazon Linux のログ

- ログの多くは /var/log 以下のファイルに保存される。  
一定の保存期間・容量を超えればファイルを切り替え、すなわち rotate、する。
- Amazon Linux は syslog システムとして rsyslog を利用している。保存先は /etc/rsyslog.conf に定義されており:
  - /var/log/secure : セキュリティ関連のログ、ssh ログイン記録など
  - /var/log/cron : 定期実行デーモン(cron)の実行記録
  - /var/log/maillog : メール配送デーモンのログ
  - /var/log/messages : その他のログ
- rsyslog 以外にも以下のログも重要となる
  - /var/log/dmesg : カーネルログなど
  - /var/httpd/access\_log : httpd (Web サーバ)のアクセスログ
  - /var/httpd/error\_log : httpd (Web サーバ)のエラーログ

# FQDN(Fully Qualified Domain Name)

- ネットワークシステムとユーザのギャップ:
  - 端末(end-system)はIPアドレスで通信先を識別
  - ユーザは名前の方が理解しやすい
    - apple.com, harvard.edu, [whitehouse.gov](http://whitehouse.gov), army.mil
    - [toyota.co.jp](http://toyota.co.jp), [u-tokyo.ac.jp](http://u-tokyo.ac.jp), [kantei.go.jp](http://kantei.go.jp)
- 名前空間の階層化
  - Domain Name System (DNS):IP アドレスと名前の対応を再帰的(Recursive)に問い合わせ
  - 名前管理:レジストリに(階層的に)権限を委譲
    - TLD (Top Level Domain)の割り当ては米商務省から委託を受け IANA がおこなってきたが、2006 年 ICANN に移管
      - ccTLD (Country Code) TLD : .jp, .uk, .cn, .hk, .us
      - gTLD : .com, .edu, .gov, .mil, .org, .net, .int, .arpa...
  - 国際化(非 ASCII のサポート)、gTLD の拡大

# How clients access Web services

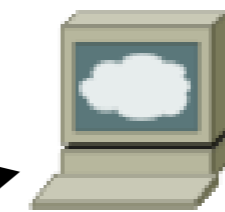
?

(3) Prepare content(s)



DNS (cache) server

www.example.com

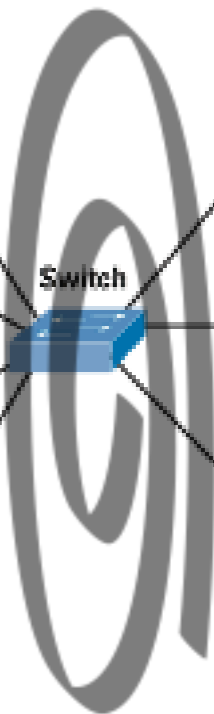


Web Server

Client Requests



Presentation/Web Servers



Application Servers



Application Tier

Presentation or Web Tier

(1) Query/Reply IP address of www.example.com

(2) Request content(s) using the resolved IP.

(4) Reply content(s) to the client

(5) Parse the replied content(s).  
And, do actions, if needed.

(0) Input URL or click a link



PC



Google



www.example.co



リーダー



情報ポータルサイト

Apple

Yahoo!

Google Maps

Popular

ニュース

お役立ち

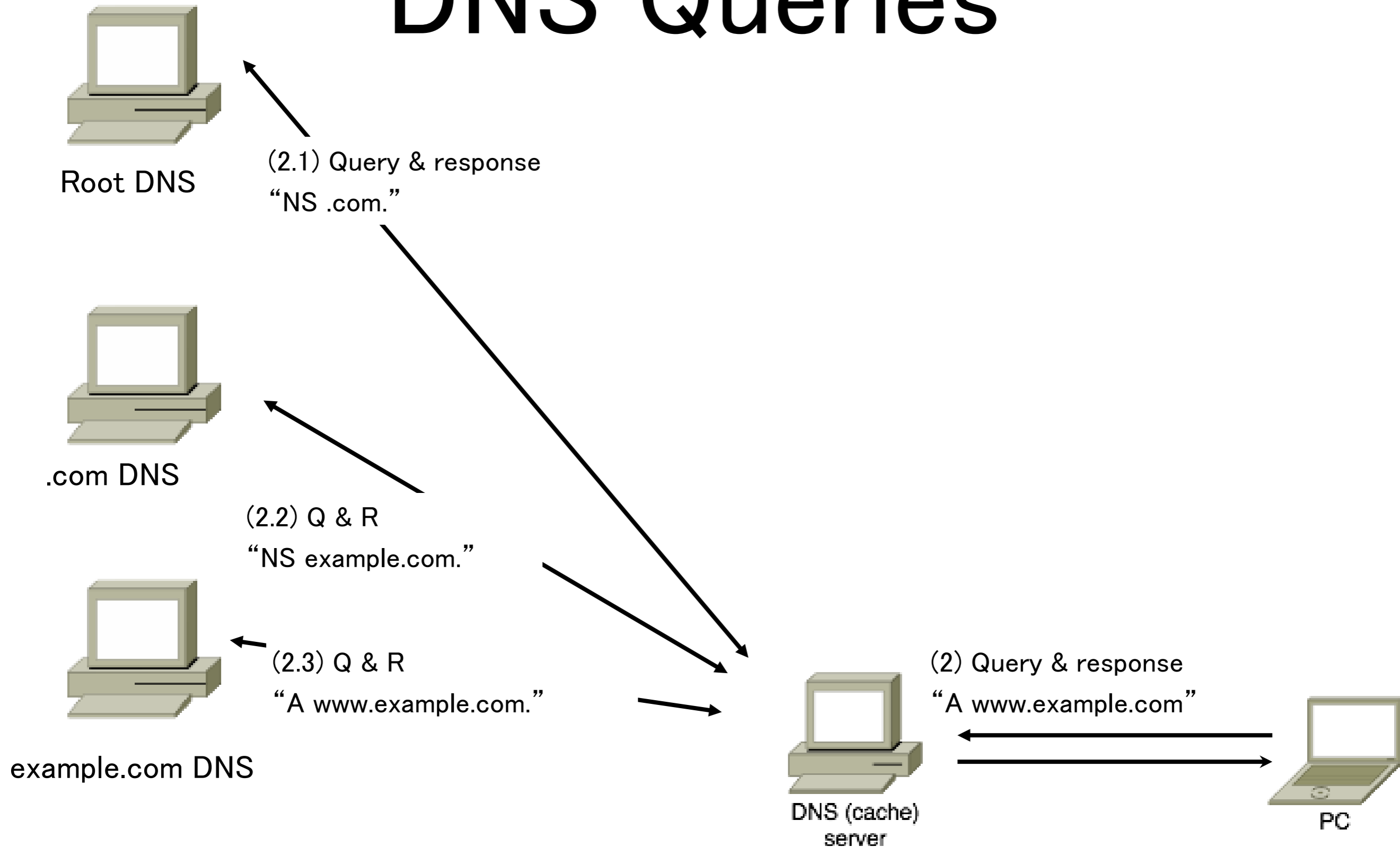
ikob

Wikipedia

Google



# DNS Queries



# 演習：DNS 設定を確認する

- nslookup コマンドで申告した EIP (Elastic IP) IPv4 アドレスの名前が設定されているか確認する。  
> nslookup <申告した名前>.ocs.shikob.net

```
> nslookup foo.ocs.shikob.net
Server:          192.168.100.1
Address: 192.168.100.1#53
```

```
Non-authoritative answer:
Name:foo.ocs.shikob.net
Address: AA.BB.CC.DD
>
```

上記で、Name と Address が申告したものと対応していることを確認する

注意：nslookup コマンドではなく dig が推奨されているが、Windows では標準ではないので、

# Hyper Text Transfer Protocol (HTTP)

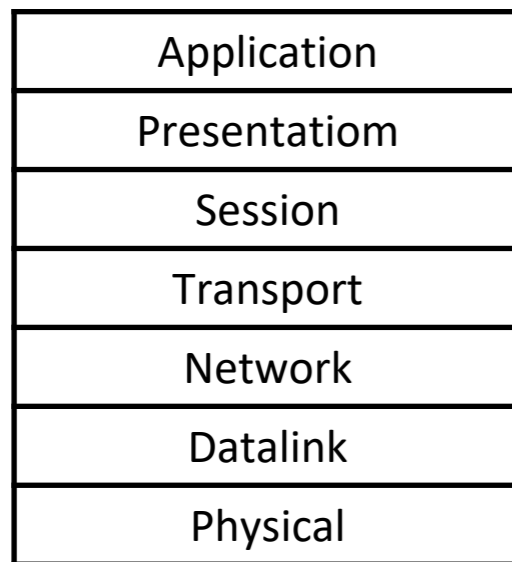
## RFC7230 他

- Web サービスの核となるステートレスなアプリケーションレベルの通信方式
- 8種のメソッドが規定されている、太字がコンテンツ操作：
  - GET** / HEAD / **POST** / **PUT** / **DELETE** / CONNECT / OPTIONS / TRACE

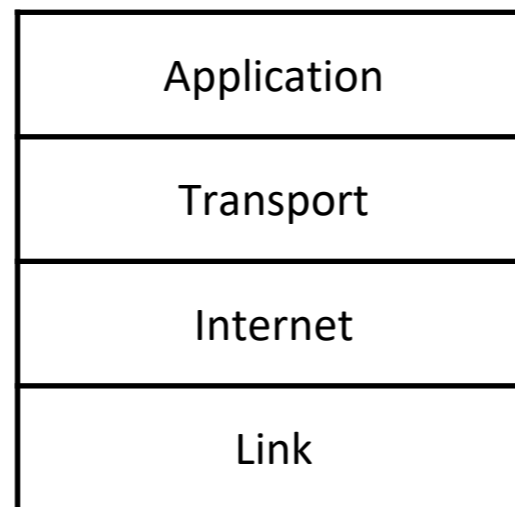
	Safe / Cacheble	Idempotent	機能 (CRUD)
GET	✓	✓	Read
POST	✗	✗	Create
PUT (PATCH)	✗	✓	Create / Update
DELETE	✗	✓	Delete

# ネットワーク階層モデル

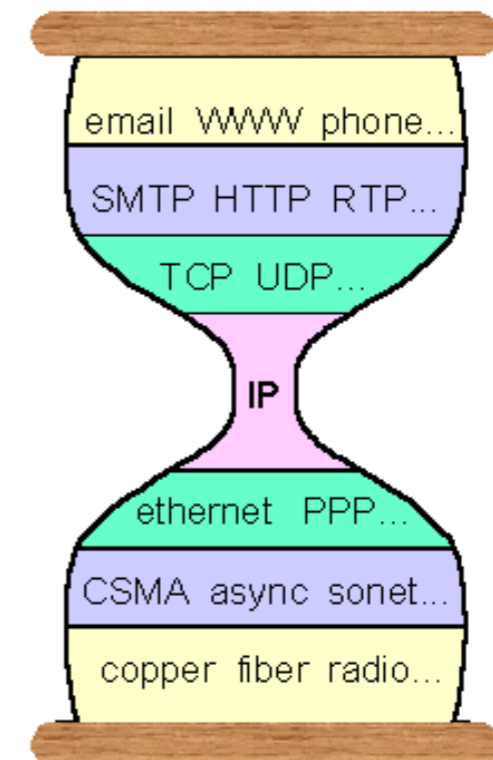
- コンピュータの通信機能の概念モデル
  - OSI 7 層・DARPA 4層が定義されている
- 下位層は上位層に対して抽象化されている。上位層の変更なしに下位層を置き換えることができる。
  - アプリケーションプログラムの改変なしに光ファイバ、無線といった異なる物理層を利用できる。



OSI



DARPA



Steve Deering, Watching the Waist of the Protocol Hopurglass, IETF 2001

# Web サービス セッション管理

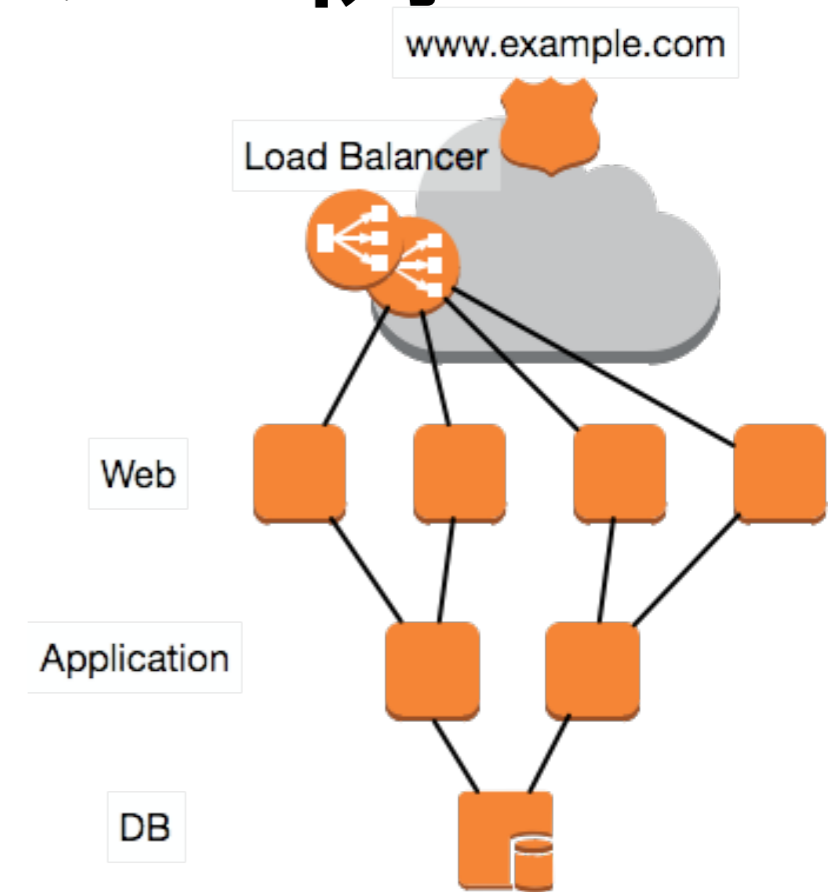
- (HTTP) Cookie : HTTP 上で、サーバ - ユーザエージェント(UA : Web ブラウザ)間の「状態」管理をおこなう目的で仕様化
  - Cookie は HTTP メッセージに重畳して転送される
- 「状態」の例:
  - ユーザのログイン状態、使用言語
  - ユーザトラッキング - e-Commerce など
- 手順:
  - サーバはセッション識別子(Session Identifier) で内部状態を管理、セッション識別子を含む Cookie 情報をUA に転送
  - UA は同じセッションの HTTP メッセージにセッション識別子を含む Cookie 情報を転送



# Web システム

## 多層アーキテクチャの一例

- 3 層Web (Three-Tier Web)アーキテクチャ
  - e-commerce を中心にほぼ標準的に利用されている
  - 開発・運用コストの抑制が目的
    - ユーザインターフェース-データストアまでのソフトウェア結合度を抑え、システムの修正・再利用度を上げる



Tier	Web 開発	機能
Presentation	フロントエンドWeb Server	静的コンテンツ・キャッシュ
Application, or Business Logic	Application Server	動的コンテンツ処理・生成
Data	バックエンドデータベース	関係データベース管理システム(RDBMS)

# 演習：Web サーバに HTTP でアクセスしてみる

- ssh コマンドでユーザ名 ec2-user で Web サーバにログイン
- yum コマンドで telnet を install

```
$ sudo yum -y install telnet
```

- telnet コマンドで TCP 80 番ポートにアクセス、HTTP GET メソッドを実行する

```
$ telnet 127.0.0.0 80  
GET /index.html HTTP/1.0
```

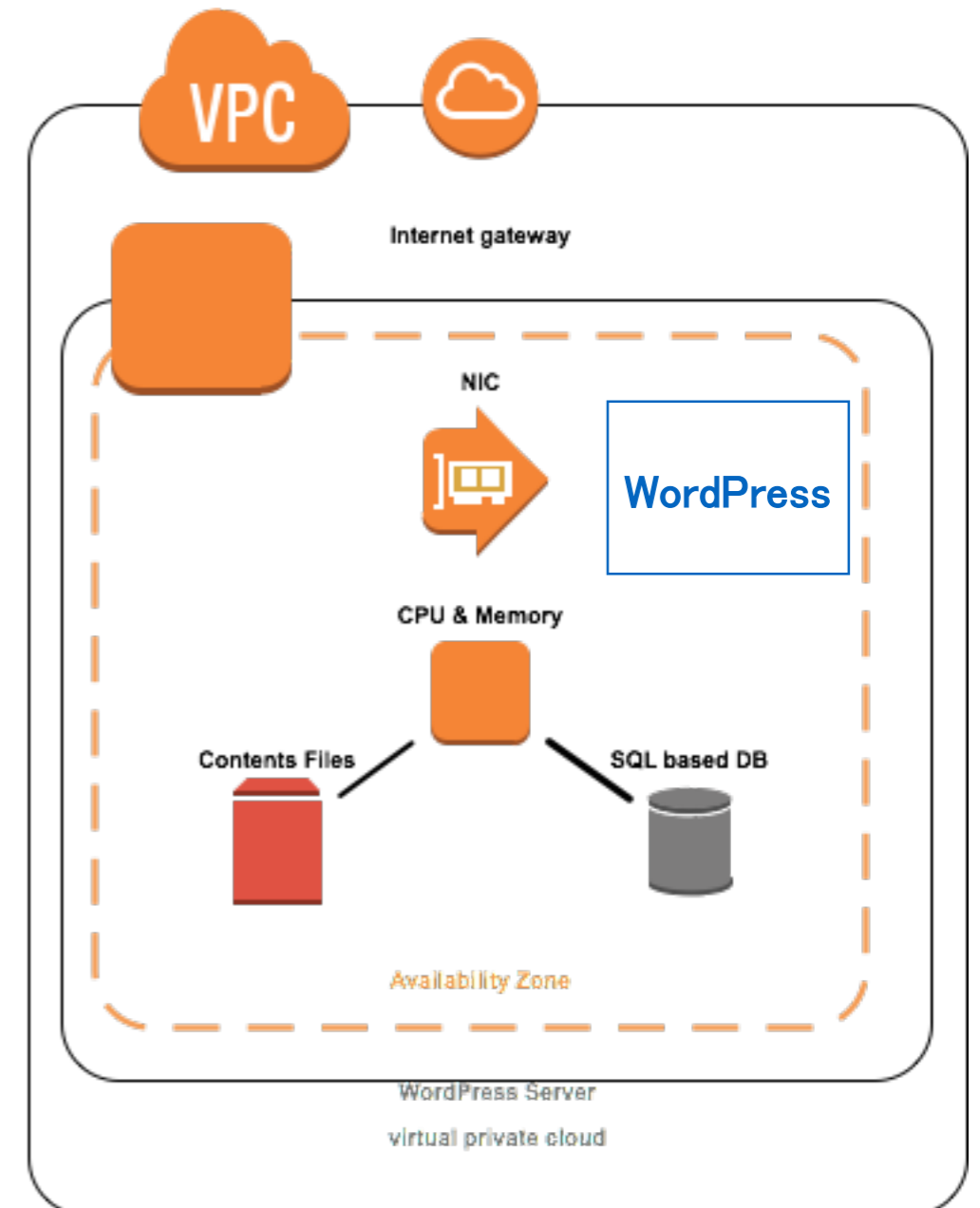
**GET <コンテンツ> HTTP/バージョン**  
**以降を入力、改行は2回**

- 実行後 status code : 200 (OK) が返され、要求したコンテンツを確認

```
HTTP/1.1 200 OK  
Date: Tue, 12 Jun 2018 11:32:22 GMT  
Server: Apache/2.2.34 (Amazon)  
Last-Modified: Wed, 16 May 2018 03:35:03 GMT  
ETag: "6076b-d-56c4a6788cb03"  
Accept-Ranges: bytes  
Content-Length: 13  
Connection: close  
Content-Type: text/html; charset=UTF-8  
  
Hello World!  
Connection closed by foreign host.  
$
```

# 演習 : Wordpress インストール

- 前提 : 前回の演習で、index.php が正常に表示されていること。
- Wordpress : コンテンツ管理システム (Contents Management System: CMS) で、動的な Web ページの提供に広く利用されている。
  - 30% の Web サーバで利用されている。(WP 調べ)
- ページ生成 : PHP (PHP: Hyper Text Processor)
- コンテンツ管理 : RDBMS + ファイル



# 演習: Wordpress インストール1/3

- yum コマンドで必要なソフトウェアパッケージをインストールする、Web サーバを再起動

```
$ sudo yum install -y php70 mysql56-server php70-mysqld  
$ sudo service httpd restart
```

- MySQL(RDBMS: 関係データベース管理システム) を起動する。さらに自動起動を設定

```
$ sudo service mysqld start  
$ chkconfig mysqld on
```

- 初期設定コマンドを実行しパスワードを設定する

```
$ sudo mysql_secure_installation  
Enter current password for root (enter for none): <ここでは改行のみ、パスワードが設定されていないので>  
OK, successfully used password, moving on...  
  
Setting the root password ensures that nobody can log into the MySQL  
root user without the proper authorisation.  
Set root password? [Y/n] Y  
New password: <MySQL のパスワードを入力、忘れないように>  
Re-enter new password:  
<以降の質問は全て Y と答えて良い>
```

- mysql コマンドで Wordpress 向け設定(ユーザ、データベース、権限付与)をおこなう

```
$ mysql -u root -p  
Enter password: <MySQL のパスワードを入力>  
mysql> CREATE USER 'wordpress'@'localhost' IDENTIFIED BY '<Wordpress のパスワード>';  
mysql> CREATE DATABASE wordpress;  
mysql> GRANT ALL PRIVILEGES ON wordpress.* to "wordpress"@"localhost";  
mysql> exit
```

# 演習: Wordpress インストール2/3

- wget コマンドで最新版の Wordpress パッケージ一式を取得、tar コマンドでパッケージを展開する

```
$ wget https://wordpress.org/latest.tar.gz  
$ tar xvzf latest.tar.gz
```

- 設定ファイルをコピーし、修正する。エディタとして nano コマンドが使える、vim でもかまわない。

```
$ cp wordpress/wp-config-sample.php wordpress/wp-config.php  
$ nano wordpress/wp-config.php
```

- データベース関連定義を探し、以下のように修正する

```
define('DB_NAME', 'wordpress');  
define('DB_USER', 'wordpress');  
define('DB_PASSWORD', '<Wordpress のパスワード>');
```

- 認証キー、salt を <https://api.wordpress.org/secret-key/1.1/salt/> から生成する。  
関連定義を探し、置き換える:

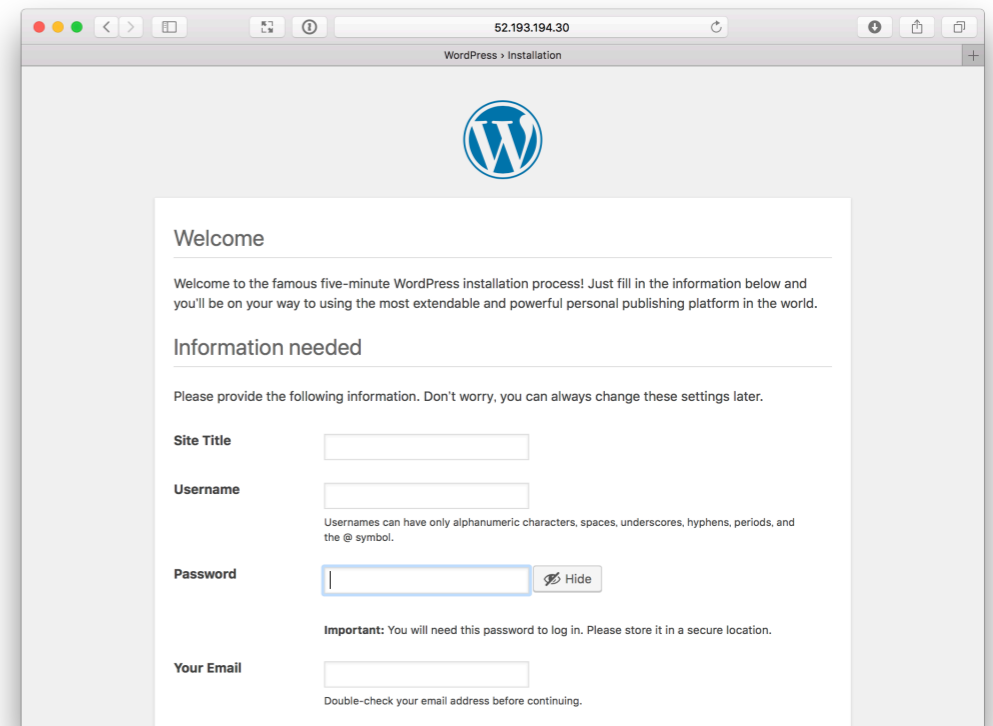
```
define('AUTH_KEY', '置き換える');  
define('SECURE_AUTH_KEY', '置き換える');  
define('LOGGED_IN_KEY', '置き換える');  
define('NONCE_KEY', '置き換える');  
define('AUTH_SALT', '置き換える');  
define('SECURE_AUTH_SALT', '置き換える');  
define('LOGGED_IN_SALT', '置き換える');  
define('NONCE_SALT', '置き換える');
```

- パッケージ全体をコピーする

```
$ cp -r wordpress/* /var/www/html/
```

# 演習: Wordpress インストール3/3

- `http://<DNS 設定で確認した FQDN>/index.php` にアクセスし、Wordpress 設定画面が表示されれば MySQL への接続は成功している。
- Title/Username/Password/Email を設定、install を完了する
- Wordpress で自身を紹介するホームページを作成する。これを出席に代える

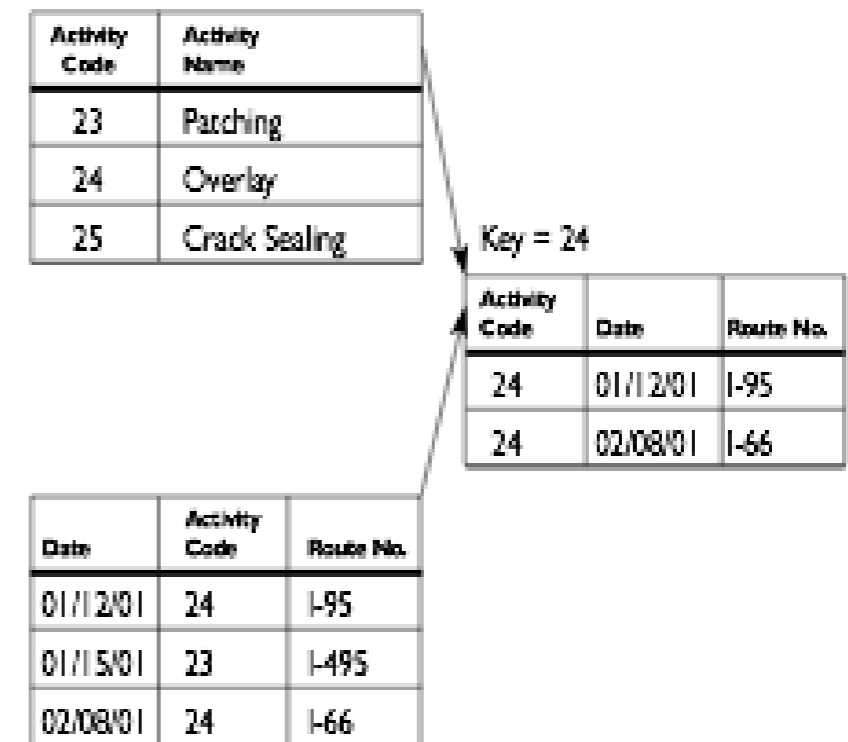


The WordPress software, GPLv2

# 関係データベース管理システム (RDBMS)

- 多くの情報システムのデータストア(\*)バックエンドとして広く利用されている。e.g., MySQL, Oracle Database, Postgress, DB2
  - 永続的なデータ保存・蓄積システム
- CRUDに加え、データベースの整合性(consistency)を保証するトランザクション処理を提供
- アプリケーションは SQL (Structured Query Language) を介してアクセス
- 関係 (Relational) モデル:  
“Data items organized as a set of formally described tables from which data can be accessed or reassembled in many ways without having to reorganize the database tables. Each table (sometimes called a relation) contains one or more data categories in columns. Each row contains a unique instance of data for the categories defined by the columns.”

## Relational Model



Data Integration Glossary,  
US: Department of Transportation, August 2001