2020年5月28日

大学オンライン授業向けサーバー

ホームページの運用マニュアル

ホスティングのサーバ設定と

OS と Drupal などのインストール

改訂版 2020年5月28日

デジタルマーケティング・トランスフォーメーション

オープンソースソフトウェアで事業の情報化

デジタルマーケティングで事業の活性化

担当

	目次	
第1章	さくらのクラウドの設定	1
	•	1
	● さくらクラウドのユーザー管理	1
	✓ ユーザーとアカントの関係	1
	✓ 「ユーザー」 xxxxxx の設定です。 他のアカウントへの	アク
	セスができない状態です。	5
	● ログイン	5
	● サーバ環境の設定(作業日:2020/05/11)	8
	● プランの変更	10
第2章	OS(Ubuntu)のインストール	11
	● 基本設定	11
	✓ vim インストール	12
	✓ Curlのインストール	14
	● GUIのインストール	14
	✓ デスクトップ環境 Mate をインストール	14
	✓ 日本語環境をインストール	14
	✓ システムの再起動 reboot	15
	✓ GUI でユーザーログインができない現象	18
	✓ GUIの再インストール	19
第3章	ファイアウォールの設定(2020年5月)	21
	● ファイアウォール(ufw)設定	21
	✓ サーバー内部から Localhost のポート公開情報を確認	21
	✓ 外のサーバーから接続した場合のポート公開状況の確認	21
	✓ ufwのインストールと設定	22
第4章	データベースのインストール	25
	● MySQLデータベースのインストール	25
	✓ Ubuntu 18.04.4 で標準パッケージとしてバージョンを確認	忍25
	✓ Drupal 9 のデータベースサポート条件	26
	✓ MySQLの初期設定	27
<i>tete</i> _ 	✓ MySQL の動作確認	29
第5章	Webサーハ (Nginx) のインストール	32
	✓ インストールの確認	33
	✓ Nginxの起動	33
	 ✓ Nginx の自動起動の設定 	34

	● 仮想ドメインの設定	34
	✓ シンボリックリンクの設定	35
	● インストール環境の確認	35
第6章	Let's Encrypt で SSL 化の設定	41
	• Certbot インストール	41
	● d.jca.jp 用の Certbot インストール	42
	● m.jca.jp 用の Certbot インストール	44
	● Certbotの自動更新設定	45
第7章	PHP インストール	47
	● PHP7.4のインストール	47
	● php7.4のバージョン確認	47
	● インストールされた PHP7.4 関連モジュール	47
	● PHP と Nginx 関連モジュールのインストール	48
	● その他の PHP 関連モジュールのインストール	50
	● インストールされた PHP 関連モジュールの確認	50
	● Nginx での PHP 設定	51
	✓ www.confの設定	51
	✓ ソケットのパーミッションを変更	52
	✓ ドメインごとの conf ファイルの編集	52
	✓ php.ini の編集	53
	✓ Nginxの再起動	53
	✓ 自動起動の設定	54
	✓ PHP の環境確認	54
第8章	Composer インストール	55
第9章	Drupal のインストールと設定	55
	● d.jca.jp サイト	56
	 m.jca.jp サイト 	56
	✓ Nginx の設定	56
	✓ データベースの設定	59
第10章	を Backup と update の設定	59
	 バックアップのシェルスクリプト 	59
	✓ backup.sh ファイルの内容	60
	 アップデイトのシェルスクリプト 	61

61
63
63
64
64
66
68
70
71
72
73
74
74
74
75
75
75

第1章 さくらのクラウドの設定

● さくらクラウドのユーザー管理

さくらクラウドでは、一つのクラウドアカウント(利用料の支払い・決済のカード情報など が共有されるアカウント)で、複数のユーザーを追加設定して、そのユーザーごとに、クラ ウド環境を独自に構築・管理できます。さくらクラウドのマニュアルでは、

https://manual.sakura.ad.jp/cloud/controlpanel/user-account.html

✓ ユーザーとアカントの関係

- コントロールパネルへのログイン認証の単位となる「**ユーザ」**
- リソースの作業空間(実際のクラウド構築環境)となる「**アカウント」**



「アカウント」ごとにクラウド環境の支払いが計算され、最後のさくらインターネットへの 支払は、「管理者:会員 ID」の支払い口座になります。



設定は、ログインするときに、さくらインターネット会員の登録ユーザーアカウント (xxxxxxxx)でログインします。 <u>https://secure.sakura.ad.jp/cloud/</u>

さくらインターネット会員としてログイン			
And a set of the set o			
•••••			
☑会員IDを保存			
ログイン			

左メニューの「**アカウント」**と「**ユーザー」**で確認できます。



「アカウント」 xxxxxxx の設定内容

他のアカウントにはアクセスできない状態です。

さくらのクラウド ホーム:	C		🔔 お知らせ 🛛 🧬 🛀 😂 🖘 - ログアウト
₩ サービス	🔄 アカウント一覧へ戻る 🛛 📝	保存 🛛 🕅 隙	アカウント: 「「「」」」」
🙆 ダッシュボード	リソース ID	113200780232	
🔗 アカウント		University Online W	orkshop
🚢 그-ザ	名前*	1~32文字	
👂 API≠—	アカウントコード		
● 2段階認証		-	
倉 イベントログ			
💵 請求情報			 アクセスホロ ワシース間見 単振爆作 設定編集 作成・削除
		g 3)	□ 請求問覧 □ オブジェクトストレージ
			🔘 アクセス不可 🔿 リソース閲覧 🔿 電源操作
	コーザのマクセフ境限		○ 設定編集 ○ 作成・削除
			□ 請求閲覧 □ オブジェクトストレージ
			□ ウェブアクセラレータ □ イベントログ
			○ アクセス不可 ○ リソース閲覧 ○ 電源操作
			🔘 設定編集 🔘 作成・削除
			□ 請求閲覧 ✓ オブジェクトストレージ
			✓ ツェノアクセラレータ ✓ イベントロク

✓ 「ユーザー」 xxxxxxx の設定です。 他のアカウントへのアクセスができない状態です。

さくらのクラウド ホーム:	C		🔒 お知らせ 🔹 ログアウト			
₩ サービス	🔄 ユーザー覧へ戻る 🛛 保存	🖸 削除	ユーザ、			
🐠 ダッシュボード	リソース ID	113200780160				
💋 アカウント						
🚢 ユーザ	名前*	1~32文字				
API≠-	フーザコード					
① 2段階認証		The state of the s				
自 イベントログ	2段階認証	無効				
10 請求情報	パスワード	•••••	•			
		8~64文字, 英数字を混ぜてください				
	マカウントスのマクセフ接限		 アクセス不可 〇 リソース閲覧 〇 電源操作 〇 設定編集 作成・削除 請求閲覧 □ オブジェクトストレージ □ ウェブアクセラレータ 「イベントログ 			
	テカランドへのアクセス権限		 ○ アクセス不可 ○ リソース閲覧 ○ 電源操作 ○ 設定編集 ● 作成・削除 □ 請求閲覧 ✓ オブジェクトストレージ ✓ ウェブアクセラレータ ✓ イベントログ 			

• ログイン

設定は次の URL から行いました。 https://secure.sakura.ad.jp/cloud/

さくらのクラウドユーザとしてログイン					
An and the second					
パスワード					
ユーザコードと会員IDを保存					
ログイン					
ユーザコード・パスワードが分からない場合はさくらインターネット会員 としてログインするか、ユーザの管理者にお問い合わせください。					

上記のパスワードは xxxxxxx でログインします。



ログイン後、「**アカウント」xxxxxxxx** を選択クリック

C		÷	💄 お知らせ	<u>& </u>
@ アカウント未選択 ▼				サービスリダイレクト
	アカウントを選択し	, τ < ;	ださい	
	 C Pカウント未選択 ▼ C C	 C C P かかうント未選択▼ C C	アカウント未選択▼ アカウント未選択▼ アカウントを選択してく アカウントを選択してく	C アカウント未選択 ▼ アカウントを選択してください

ログインするとクラウド環境になり、「石狩第 1 ゾーン」を選択します。 サーバーの口を クリックし、「詳細」でコンソールなどを管理できます。サーバーの停止は、Ubuntu コマ ンドで shutdown を実行するか、「電源操作」で起動、シャットダウンができます。

● 石狩第1ゾーン ▼	c		*	0		
パーゼ 🔳		サーバー覧 (1/上限 100)	十追加	▮ 詳細	🕛 電源操作 ▼	🛃 削除
📾 ストレージ	~	Q, V				
な ネットワーク	~	名 CPU メモリ NIC 収容ホスト名	作成日時			
📰 アプライアンス	~	🗌 🏠 🍕 1 1GB 🌐 133.242.21.74 🗘 sac-is1a-sv6	37 2020/05/11	18:01		•
⊕ グローバル	~	ステータス: 2 起動 2 停止 2 準備中/起動処理中				
セキュアモバイル	~					
🛍 マップ						
🗼 リソースマネージャ	,					
📕 LAB	~					

● サーバ環境の設定(作業日:2020/05/11)

石狩第1ゾーン

1 CPU メインメモリ 1GB SSD40GB OS: Ubuntu 18.04.4 64bit

サーバの設定							
ホスト名 xxxxxxxx 管理ユーザーパスワード(ubuntu) xxxxxxxx							
リソース ID 113200780391 IP アドレス 133.242.21.74							
ネットマスク 255.255.255.0 ゲートウェイ 133.242.21.1							
ネットワーク 133.242.21.0/24 帯域幅 100Mbps							

管理画面での設定

🥝 情報 🛛 📠 NIC (1) 🔄 ディスク (1) 💿 ISOイメージ なし 🔞 シンプル監視 (0) 🚇 アクティビティ 💂 コンソール

リソース ID	113200780391
名前	n an
説明	
アイコン	Obuntu
プラン	プラン/1Core-1GB(新プラン)
NICドライバ	virtio
ステータス	UP
有効状態	利用可能
作成日時	2020/05/11 18:01:40 (2時間前)
修正日時	2020/05/11 18:03:59 (2時間前)
状態変更日時	2020/05/11 18:03:59 (2時間前) down → up
ハイパーバイザ	SAKURA Internet [CLOUD SERVICE 3.2]
収容ホスト名	sac-is1a-sv441

ネットワークの設定

🤕 情報 🔚 NIC (1) 🔤 ディスク (1) 💿 ISOイメージ なし 🛽 🕲 シンプル監視 (0) 💻 アクティビティ 🖳 コンソール

このリージョンの推奨ネームサーバ: 133.242.0.3, 133.242.0.4									
#	接続	IPv4アドレス	ホスト名	ゲートウェイ	ネットワーク	帯域幅 パケットフィル			
0	🌐 共有セグメント インターネット	133.242.21.74 [未割当	133.242.21.1 🕒	133.242.21.0/24 [100Mbps	未割当	▼	

ディスクの設定

🧿 情報) m NIC	(1)	🦲 ディスク (1)	💿 ISOイメージ なし	🚳 シンプル監視 (0)	學 アクティビティ	狊 コンソール	
接続	#	נע	V-ZID	名前			ソース	サイズ
virtio	1	11	3200780393	univ1jcafe			💾 Ubuntu Server 18.04.4 LTS 64bit	40GB 🔻

🗊 情報	📱 接続先サーバ	遲 アクティビティ
------	----------	-----------

リソース ID	113200780393
名前	univ1jcafe
プラン	SSDプラン
接続	virtio
ソース	アーカイブ:113200264075 (Ubuntu Server 18.04.4 LTS 64bit)
サイズ	40GB
有効状態	利用可能
作成日時	2020/05/11 18:01:44 (3時間前)
収容ストレージ名	sac-is1a-iscsi5-st106

● プランの変更

さくらクラウドの仮想コア(CPU)、メモリーを変更することができます。アクセスの増減 に対応してプラン変更可能です。変更する場合は、OS をシャットダウンして、「プラン 変更」をクリックし、プランを選択して、起動します。



第2章 OS(Ubuntu)のインストール

OSの設定は、コンソールを使って行いました。さくらクラウドの標準で Ubuntu 18.04.4 をインストール場合のプログラムの設定。

● 基本設定

別の Linux コンピュータ(例えば、事務所の Linux サーバなど)で、外部から Nmap ポ ートスキャンで開いているポートを確認しました。 \$ nmap 133.242.21.74

Starting Nmap 7.70 (https://nmap.org) at 2020-05-11 20:51 JST Nmap scan report for 133.242.21.74 Host is up (0.045s latency). Not shown: 997 closed ports PORT STATE SERVICE 22/tcp open ssh 25/tcp filtered smtp 12345/tcp filtered netbus

Nmap done: 1 IP address (1 host up) scanned in 3.66 seconds

SSH ポートが開いており、root(ubuntu)で接続できる状態です。新規ユーザアカウント (office, jcafe)を作成し、sudo コマンドを実行できるようにします。

ユーザ追加: ユーザ名 office パスワード **xxxxxxxx** ユーザ名 jcafe パスワード **xxxxxxxx** # adduser office

Visudo コマンドでユーザ(office, jcafe)を設定

User privilege specification root ALL=(ALL:ALL) ALL office ALL=(ALL:ALL) ALL jcafe ALL=(ALL:ALL) ALL

✓ vim インストール

sudo apt install vim

自分用に修正

vi ~/.vimrc

" vim の独自拡張機能を使用(vi との互換性無し)

```
set nocompatible
" 文字コードを指定
set encoding=utf-8
"ファイルエンコードを指定(先頭から順に成功するまで読み込む)
set fileencodings=utf-8,iso-2022-jp,sjis,euc-jp
" 自動認識させる改行コードを指定
set fileformats=unix,dos
"バックアップを取得
"逆は[set nobackup]
set backup
"バックアップを取得するディレクトリを指定
set backupdir=~/backup
"検索履歴を残す世代数
set history=50
"検索時に大文字小文字を区別しない
set ignorecase
"検索語に大文字を混ぜると検索時に大文字を区別する
set smartcase
"検索語にマッチした単語をハイライト
"逆は [ set nohlsearch ]
set hlsearch
"インクリメンタルサーチを使用(検索語の入力最中から随時マッチする文字列の検索を
開始)
"逆は[set noincsearch]
set incsearch
" 行番号を表示
"逆は [set nonumber]
set number
" 改行($)やタブ(^I)を可視化
set list
"括弧入力時に対応する括弧を強調
set showmatch
"自動インデントを有効にする
"逆は[noautoindent]
set autoindent
"構文ごとに色分け表示
"逆は[syntax off]
syntax on
" [syntax on ] の場合のコメント文の色を変更
highlight Comment ctermfg=LightCyan
"ウィンドウ幅で行を折り返す
" 逆は [ set nowrap ]
set wrap
```

✓ Curl のインストール

sudo apt install curl
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
libcurl4
The following NEW packages will be installed:
curl libcurl4
0 upgraded, 2 newly installed, 0 to remove and 0 not upgraded.
Need to get 373 kB of archives.
After this operation, 1,038 kB of additional disk space will be used.
Do you want to continue? [Y/n]

• GUIのインストール

✓ デスクトップ環境 Mate をインストール

sudo apt -y install ubuntu-mate-desktop

✓ 日本語環境をインストール

sudo apt -y install language-pack-ja-base language-pack-ja ibus-mozc

sudo localectl set-locale LANG=ja_JP.UTF-8 LANGUAGE="ja_JP:ja"

source /etc/default/locale

echo \$LANG

ja_JP.UTF-8 # 変更された

✓ システムの再起動 reboot

日本語環境が起動します。

(2) 情報	NIC (1)	🧾 ディスク (1)	💿 ISOイメージ なし	🕲 シンプル監視 (0)	奥 アクティビティ	皇 コンソール	
univ1	jcafe	×	切断 🔋 ペースト	🔻 🔲 ESC 🛄	Alt 🔲 Ctrl 📧	Ctrl+Alt+Del	न ポップアップ
uni	v1					🔂 📼	jp 21:30 🖒
						4	
		cafe					
2	ol	ffice		E ji			Δ.
	Ľ	パスワード		SED			
	U	buntu User					

ディスプレイ解像度を変更

インストール後は 800x600 なので、1024x768 に変更したいと考え、GUI のディスプレ イ設定や xrandr コマンドなどでは、できないので、GRUB ファイルで直接指定する方法 を知り、修正しました。

https://qiita.com/nsd24/items/c06294d1de40f2e9870b

https://qastack.jp/ubuntu/54067/how-do-i-safely-change-grub2-screen-resolution

GRUB ファイルを編集して、解像度を指定します。

sudo vim /etc/default/grub

GRUB_GFXMODE=1024x768 を追加

- - -

The resolution used on graphical terminal

note that you can use only modes which your graphic card supports via VBE

you can see them in real GRUB with the command `vbeinfo'

#GRUB_GFXMODE=640x480

GRUB_GFXMODE=1024x768

- - -

sudo update-grub

その後、OS 再起動。再起動後、日本語入力の設定を確認し、「IBus の設定」で「入力メ ソッド」に「日本語—Mozc」を追加設定します。



「日本語-Mozc」が起動したら、デスクトップ画面の右上の「A」の言語入力アイコンをク リックして、インプットを「Hiragana」を選択すると日本語入力になります。「Direct Input」は英語モード。



✓ GUI でユーザーログインができない現象

その後、GUI でログインしたところ、office ではログインできない、jcafe や ubuntu では ログインできる、現象が起こりました。ssh では office でログインできます。

そこで、ssh で office でログインして Linux カーネルが起動時に出力したメッセージを 表示するコマンド dmesg で、 sudo dmesg | less すると、

[3.678260] audit: type=1400 audit(1589251102.352:2): apparmor="STATUS" operation="profile_load" profile="unconfined" name="/usr/lib/lightdm/lightdm-guest-session" pid=370 comm="apparmor parser"

[3.678264] audit: type=1400 audit(1589251102.352:3): apparmor="STATUS" operation="profile_load" profile="unconfined" name="/usr/lib/lightdm/lightdm-guest-session//chromium" pid=370 comm="apparmor_parser"

[3.684354] audit: type=1400 audit(1589251102.360:4): apparmor="STATUS" operation="profile_load" profile="unconfined" name="/sbin/dhclient" pid=372 comm="apparmor_parser"

[3.684356] audit: type=1400 audit(1589251102.360:5): apparmor="STATUS" operation="profile_load" profile="unconfined" name="/usr/lib/NetworkManager/nm-dhcp-client.action" pid=372 comm="apparmor_parser"

[3.684358] audit: type=1400 audit(1589251102.360:6): apparmor="STATUS" operation="profile_load" profile="unconfined" name="/usr/lib/NetworkManager/nm-dhcp-helper" pid=372 comm="apparmor_parser"

[3.684359] audit: type=1400 audit(1589251102.360:7): apparmor="STATUS" operation="profile_load" profile="unconfined" name="/usr/lib/connman/scripts/dhclient-script" pid=372 comm="apparmor_parser"

```
[ 3.700122] audit: type=1400 audit(1589251102.376:8):
apparmor="STATUS" operation="profile_load" profile="unconfined"
name="/usr/bin/man" pid=374 comm="apparmor_parser"
```

[3.700124] audit: type=1400 audit(1589251102.376:9): apparmor="STATUS" operation="profile_load" profile="unconfined" name="man_filter" pid=374 comm="apparmor_parser"

[3.700125] audit: type=1400 audit(1589251102.376:10): apparmor="STATUS" operation="profile_load" profile="unconfined" name="man_groff" pid=374 comm="apparmor_parser" という記録がありました。Ubuntu 18.04 でログインできない現象をネット検索すると、 ゲストセッションを無効にするとログインできそうな説明がありました。

https://qiita.com/JeJeNeNo/items/6f061a0ea8a7cfe198c1

でもダメでしたので、master というユーザーを追加して GUI でログインしました。 master / **xxxxxxxx**

✓ GUIの再インストール

MATE デスクトップ環境では、ユーザー名:office でログインできない、GUI で日本語入 カができない(システム管理だけではそれほど日本語入力が必須ではないが)ので、勉強 のために、tasksel コマンド、MATE デスクトップのほかに、GNOME デスクトップ環境、 Ubuntu サーバー環境などを追加インストールし、再起動。



第3章 ファイアウォールの設定(2020年5月)

● ファイアウォール(ufw)設定

ファイアウォールの設定を ufw に変更しました。

現在のポート公開状況を再確認します。nmap をインストール。

sudo apt install nmap

✓ サーバー内部から Localhost のポート公開情報を確認

sudo nmap localhost

Starting Nmap 7.60 (https://nmap.org) at 2020-05-12 13:54 JST Nmap scan report for localhost (127.0.0.1) Host is up (0.0000020s latency). Other addresses for localhost (not scanned): ::1 Not shown: 998 closed ports PORT STATE SERVICE 22/tcp open ssh 631/tcp open ipp

Nmap done: 1 IP address (1 host up) scanned in 1.65 seconds

✓ 外のサーバーから接続した場合のポート公開状況の確認

sudo nmap 133.242.21.74

Starting Nmap 7.40 (https://nmap.org) at 2020-05-12 13:57 JST Nmap scan report for 133.242.21.74 Host is up (0.026s latency). Not shown: 997 closed ports PORT STATE SERVICE 22/tcp open ssh 25/tcp filtered smtp 12345/tcp filtered netbus

Nmap done: 1 IP address (1 host up) scanned in 154.34 seconds

✓ ufw のインストールと設定

このページを参考(Debian 10 buster ですが)

https://www.mk-mode.com/blog/2019/10/26/debian-10-firewall-setting/

※ufw コマンドの一例

# uf	fw status	# <= 状況確認
# uf	w enable	# <= ファイアウォールの有効化
# uf	fw disable	# <= ファイアウォールの無効化
# uf	fw default deny	# <= 全アクセスの拒否
# uf	fw default allow	# <= 全アクセスの許可
# uf	fw deny smtp	# <= SMTP サービスのアクセスを拒否
# uf	fw allow ftp	# <= FTP サービスのアクセスを許可
# uf	w deny 25/tcp	# <= TCP ポート 25 でのアクセスを拒否
# uf	w allow 80/udp	# <= UDP ポート 80 でのアクセスを許可
# uf	w allow 53	# <= TCP ポート 53, UDP ポート 53 でのアクセスを拒否
# uf	fw allow in http	# <= HTTP サービスのアクセスを IN のみ許可
# uf	fw reject out sm	tp # <= SMTP サービスのアクセスを OUT のみ却下
# uf	fw allow proto to	p from any to any port 8080:8090
	# <	<= TCP ポート 8080 ~ 8090 での全アクセスを許可
# uf	fw allow proto to	p from 192.168.11.0/24 to any port 9999
	# <	<= TCP ローカルネットワーク 192.168.11.0/24 からの
	#	TCP ポート 9999 でのアクセスを許可
# uf	fw delete 3	# <= 3 番目のルールを削除

インストール apt-get install ufw

パッケージリストを読み込んでいます... 完了

依存関係ツリーを作成しています

状態情報を読み取っています... 完了

ufw はすでに最新バージョン (0.36-0ubuntu0.18.04.1) です。

IPv6 は使わない設定にします。

/etc/default/ufw を編集して IPV6=no に設定します。

sudo ufw status Status: inactive

sudo ufw default deny

Default incoming policy changed to 'deny'

(be sure to update your rules accordingly)

sudo ufw status

Status: inactive

sudo ufw allow 80/tcp

Rules updated

sudo ufw allow 443/tcp

Rules updated

sudo ufw allow 22/tcp <= 開発・サポート環境の時のみ

Rules updated

sudo ufw enable

Command may disrupt existing ssh connections. Proceed with operation (y|n)? y Firewall is active and enabled on system startup

sudo ufw status

Status: active

То	Action I	From
80/tcp	ALLOW	Anywhere
443/tcp	ALLOW	Anywhere
22/tcp	ALLOW	Anywhere

外部の Linux サーバーから nmap でポート公開の確認

Starting Nmap 7.40 (https://nmap.org) at 2020-05-12 17:34 JST Nmap scan report for 133.242.21.74 Host is up (0.027s latency). Not shown: 997 filtered ports PORT STATE SERVICE 22/tcp open ssh 80/tcp closed http 443/tcp closed https

Nmap done: 1 IP address (1 host up) scanned in 17.56 seconds

第4章 データベースのインストール

• MySQLデータベースのインストール

✓ Ubuntu 18.04.4 で標準パッケージとしてバージョンを確認

MySQL の場合:

sudo apt-cache policy mysql-server*

mysql-server-5.7:

インストールされているバージョン: (なし)

候補: 5.7.30-0ubuntu0.18.04.1

mysql-server:

インストールされているバージョン: (なし) 候補: 5.7.30-0ubuntu0.18.04.1

mysql-server-core-5.7: インストールされているバージョン: (なし) 候補: 5.7.30-0ubuntu0.18.04.1

MariaDB の場合

sudo apt-cache policy mariadb-server* mariadb-server-core-10.1: インストールされているバージョン: (なし) 候補: 1:10.1.44-0ubuntu0.18.04.1 バージョンテーブル: mariadb-server-10.1:

インストールされているバージョン: (なし) 候補: 1:10.1.44-0ubuntu0.18.04.1 mariadb-server:

インストールされているバージョン: (なし) 候補: 1:10.1.44-0ubuntu0.18.04.1

✓ Drupal 9 のデータベースサポート条件

2020年5月12日現在の仕様として

https://www.drupal.org/docs/9/how-drupal-9-is-made-and-what-isincluded/environment-requirements-of-drupal-9

Database backend and other hosting requirements

If using Drupal 9 with MySQL or Percona, version 5.7.8 is required.

If using Drupal 9 with MariaDB, version 10.3.7 is required.

If using Drupal 9 with SQLite, version 3.26 is required. (PHP 7.4 does not use the system provided SQLite, so take extra care to make sure your PHP is compiled with at least this version).

If using Drupal 9 with PostgreSQL, version 10 is required with the pg_trgm extension.

とあるので、Drupal 9 を考えて、Ubuntu 標準の MySQL Server 5.7 をインストールします。

sudo apt install mysql-server

パッケージリストを読み込んでいます... 完了

依存関係ツリーを作成しています

状態情報を読み取っています... 完了

以下の追加パッケージがインストールされます:

libaio1 libevent-core-2.1-6 libhtml-template-perl mysql-client-5.7 mysql-clientcore-5.7 mysql-common

mysql-server-5.7 mysql-server-core-5.7

提案パッケージ:

libipc-sharedcache-perl mailx tinyca

以下のパッケージが新たにインストールされます:

libaio1 libevent-core-2.1-6 libhtml-template-perl mysql-client-5.7 mysql-client-

core-5.7 mysql-common

mysql-server mysql-server-5.7 mysql-server-core-5.7

アップグレード:0個、新規インストール:9個、削除:0個、保留:0個。

19.1 MB のアーカイブを取得する必要があります。

この操作後に追加で 155 MB のディスク容量が消費されます。

続行しますか? [Y/n]

✓ MySQLの初期設定

sudo mysql_secure_installation

Securing the MySQL server deployment.

Connecting to MySQL using a blank password. The 'validate_password' plugin is installed on the server. The subsequent steps will run with the existing configuration of the plugin. Please set the password for root here.

New password: <--- xxxxxxxx

Re-enter new password:

Do you wish to continue with the password provided?(Press y|Y for Yes, any other key for No) : Y By default, a MySQL installation has an anonymous user, allowing anyone to log into MySQL without having to have a user account created for them. This is intended only for testing, and to make the installation go a bit smoother. You should remove them before moving into a production environment.

Remove anonymous users? (Press y|Y for Yes, any other key for No) : Y Success.

Normally, root should only be allowed to connect from 'localhost'. This ensures that someone cannot guess at the root password from the network.

Disallow root login remotely? (Press y|Y for Yes, any other key for No) : y Success.

By default, MySQL comes with a database named 'test' that anyone can access. This is also intended only for testing, and should be removed before moving into a production environment.

Remove test database and access to it? (Press y|Y for Yes, any other key for No) : y

- Dropping test database... Success.

- Removing privileges on test database... Success.

Reloading the privilege tables will ensure that all changes made so far will take effect immediately.

Reload privilege tables now? (Press y|Y for Yes, any other key for No) : y Success.

All done!

MySQLの root のパスワード xxxxxxxx

✓ MySQLの動作確認

sudo mysql -uroot -p

Enter password: Welcome to the MySQL monitor. Commands end with ; or ¥g. Your MySQL connection id is 8 Server version: 5.7.30-0ubuntu0.18.04.1 (Ubuntu)

Copyright (c) 2000, 2020, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

Type 'help;' or '¥h' for help. Type '¥c' to clear the current input statement.

mysql> show databases; +----+ | Database | +----+ | information_schema | | mysql | | performance_schema | | sys | +----+ 4 rows in set (0.01 sec)

mysql>

Drupal 8 などをインストールする際に必要となる root 権限と同じ MySQL のユーザーア カウント(たとえば、phpmyadmin)を作成し、すべてのデータベースに権限を付与しま す。

ユーザーパスワードポリシーの設定確認

mysql> show variables like 'validate_password%';

+-----+
| Variable_name | Value |
+-----+
validate_password_check_user_name	OFF
validate_password_dictionary_file	
validate_password_length	8
validate_password_length	8
validate_password_mixed_case_count	1
validate_password_number_count	1
validate_password_policy	MEDIUM
validate_password_special_char_count	1
+-----+

7 rows in set (0.01 sec)

mysql> SET GLOBAL validate_password_length = 6; Query OK, 0 rows affected (0.00 sec)

mysql> SET GLOBAL validate_password_number_count = 0; Query OK, 0 rows affected (0.00 sec)

mysql> show variables like 'validate_password%';
+-----+
| Variable_name | Value |
+-----+
| validate_password_check_user_name | OFF
validate_password_dictionary_file	
validate_password_length	6
validate_password_number_count	1
validate_password_number_count	0
validate_password_policy	MEDIUM
validate_password_special_char_count	1
+-----+

7 rows in set (0.00 sec)

パスワードの文字数を6、number_count =0 に設定変更

ユーザーの作成

mysql> create user 'phpmyadmin'@'localhost' IDENTIFIED BY ' **xxxxxxxx** '; Query OK, 0 rows affected (0.00 sec)

ユーザー名 phpmydmin / xxxxxxxx

ユーザーに権限を設定

mysql> GRANT USAGE ON *.* TO 'phpmyadmin'@'localhost' IDENTIFIED BY '
xxxxxxx ' with max_queries_per_hour 0 max_connections_per_hour 0
max_updates_per_hour 0 max_user_connections 0;
Query OK, 0 rows affected, 1 warning (0.00 sec)

mysql> flush privileges; Query OK, 0 rows affected (0.01 sec)

すべてのデータベースに対してユーザーの権限を設定します

mysql> GRANT SELECT,INSERT,UPDATE,DELETE,CREATE,ALTER,INDEX,DROP,CREATE TEMPORARY TABLES,LOCK TABLES ON *.* TO 'phpmyadmin'@localhost IDENTIFIED BY ' **xxxxxxxx** '; Query OK, 0 rows affected, 1 warning (0.01 sec)

mysql> flush privileges; Query OK, 0 rows affected (0.00 sec)
Ubuntuの Nginx インストールは http://nginx.org/en/linux_packages.html#Ubuntu

Debian10 BusterのNginx インストールも参考にしました。 https://www.mk-mode.com/blog/2019/12/20/debian-10-nginx-installationby-official-apt/

Install the prerequisites:

sudo apt install curl gnupg2 ca-certificates lsb-release

To set up the apt repository for stable nginx packages:

```
sudo echo "deb http://nginx.org/packages/ubuntu `lsb_release -cs` nginx"
¥
```

> | sudo tee /etc/apt/sources.list.d/nginx.list

deb http://nginx.org/packages/ubuntu bionic nginx

import an official nginx signing key so apt could verify the packages authenticity:

curl -fsSL https://nginx.org/keys/nginx_signing.key | sudo apt-key add -OK

Verify that you now have the proper key:

sudo apt-key fingerprint ABF5BD827BD9BF62 pub rsa2048 2011-08-19 [SC] [有効期限: 2024-06-14] 573B FD6B 3D8F BC64 1079 A6AB ABF5 BD82 7BD9 BF62 uid [不明] nginx signing key <signing-key@nginx.com>

To install nginx:

sudo apt update sudo apt install nginx

✓ インストールの確認

nginx -v

nginx version: nginx/1.18.0

✓ Nginx の起動

sudo systemctl start nginx

sudo systemctl status nginx

 nginx.service - nginx - high performance web server Loaded: loaded (/lib/systemd/system/nginx.service; enabled; vendor preset: enabled)
 Active: active (running) since Tue 2020-05-12 19:40:59 JST; 7s ago Docs: http://nginx.org/en/docs/
 Process: 7351 ExecStart=/usr/sbin/nginx -c /etc/nginx/nginx.conf
 (code=exited, status=0/SUCCESS)
 Main PID: 7364 (nginx)
 Tasks: 2 (limit: 1103) CGroup: /system.slice/nginx.service tq7364 nginx: master process /usr/sbin/nginx -c /etc/nginx/nginx.conf mq7365 nginx: worker process

May 12 19:40:58 univ1 systemd[1]: Starting nginx - high performance web server... May 12 19:40:59 univ1 systemd[1]: Started nginx - high performance web server.

✓ Nginx の自動起動の設定

sudo systemctl enable nginx

Synchronizing state of nginx.service with SysV service script with

/lib/systemd/systemd-sysv-install.

Executing: /lib/systemd/systemd-sysv-install enable nginx

sudo systemctl is-enabled nginx enabled

もし、自動起動しないようにするには

sudo systemctl disable nginx sudo systemctl is-enabled nginx

● 仮想ドメインの設定

✓ シンボリックリンクの設定

cd ../sites-enabled/

office@univ1:/etc/nginx/sites-enabled\$ ls -la

合計 8

drwxr-xr-x 2 root root 4096 May 12 20:54 .

drwxr-xr-x 5 root root 4096 May 12 21:00 ...

lrwxrwxrwx 1 root root 26 May 12 20:54 default -> ../sites-available/default

lrwxrwxrwx 1 root root 40 May 12 20:24 d.jca.jp.conf -> /etc/nginx/sites-

available/d.jca.jp.conf

office@univ1:/etc/nginx/sites-enabled\$ sudo In -s /etc/nginx/sites-

available/m.jca.jp.conf ./

office@univ1:/etc/nginx/sites-enabled\$ ls -la

合計 8

drwxr-xr-x 2 root root 4096 May 12 21:16 .

drwxr-xr-x 5 root root 4096 May 12 21:00 ...

Irwxrwxrwx 1 root root 26 May 12 20:54 default -> ../sites-available/default
Irwxrwxrwx 1 root root 40 May 12 20:24 d.jca.jp.conf -> /etc/nginx/sitesavailable/d.jca.jp.conf

lrwxrwxrwx 1 root root 40 May 12 21:16 m.jca.jp.conf -> /etc/nginx/sitesavailable/m.jca.jp.conf

● インストール環境の確認

/etc/nginx

```
drwxr-xr-x 2 root root 4096 May 12 20:50 conf.d

-rw-r--r-- 1 root root 1007 Apr 21 21:43 fastcgi_params

-rw-r--r-- 1 root root 2837 Apr 21 21:43 koi-utf

-rw-r--r-- 1 root root 2223 Apr 21 21:43 koi-win

-rw-r--r-- 1 root root 5231 Apr 21 21:43 mime.types

lrwxrwxrwx 1 root root 22 Apr 21 21:43 modules -> /usr/lib/nginx/modules

-rw-r--r-- 1 root root 684 May 12 21:00 nginx.conf

-rw-r--r-- 1 root root 643 Apr 21 21:43 nginx.conf_org

-rw-r--r-- 1 root root 636 Apr 21 21:43 scgi_params

drwxr-xr-x 2 root root 4096 May 12 21:25 sites-available

drwxr-xr-x 2 root root 4096 May 12 21:16 sites-enabled

-rw-r--r-- 1 root root 664 Apr 21 21:43 uwsgi_params

-rw-r--r-- 1 root root 3610 Apr 21 21:43 win-utf
```

./conf.d/
-rw-r--r-- 1 root root 1093 Apr 21 21:43 default.conf

Is -la ./sites-available/ -rw-r--r-- 1 root root 1189 May 12 20:43 default -rw-r--r-- 1 root root 209 May 12 21:06 d.jca.jp.conf -rw-r--r-- 1 root root 169 May 12 20:31 d.jca.jp.conf_org -rw-r--r-- 1 root root 209 May 12 21:13 m.jca.jp.conf

```
Is -la ./sites-enabled/
Irwxrwxrwx 1 root root 26 May 12 20:54 default -> ../sites-available/default
Irwxrwxrwx 1 root root 40 May 12 20:24 d.jca.jp.conf -> /etc/nginx/sites-
available/d.jca.jp.conf
Irwxrwxrwx 1 root root 40 May 12 21:16 m.jca.jp.conf -> /etc/nginx/sites-
available/m.jca.jp.conf
```

設定ファイルの内容:

```
/etc/nginx/nginx.conf
```

user nginx; worker_processes 1;

```
error_log /var/log/nginx/error.log warn;
pid /var/run/nginx.pid;
```

```
events {
    worker_connections 1024;
}
```

```
http {
    include /etc/nginx/mime.types;
    default_type application/octet-stream;
```

access_log /var/log/nginx/access.log main;

sendfile on;
#tcp_nopush on;

keepalive_timeout 65;

#gzip on;

```
#include /etc/nginx/conf.d/*.conf;
include /etc/nginx/sites-enabled/*;
}
```

←----!

```
/etc/nginx/conf.d/default.conf
```

```
server {
  listen
            80;
  server_name localhost;
  #charset koi8-r;
  #access_log /var/log/nginx/host.access.log main;
  location / {
     root /usr/share/nginx/html;
     index index.html index.htm;
  }
  #error page 404
                             /404.html;
  # redirect server error pages to the static page /50x.html
  #
  error_page 500 502 503 504 /50x.html;
  location = /50x.html 
     root /usr/share/nginx/html;
  }
  # proxy the PHP scripts to Apache listening on 127.0.0.1:80
  #
  #location \sim ¥.php$ {
      proxy_pass http://127.0.0.1;
  #
  #}
  # pass the PHP scripts to FastCGI server listening on 127.0.0.1:9000
  #
  #location \sim ¥.php$ {
      root
  #
                  html;
  #
      fastcqi pass 127.0.0.1:9000;
      fastcgi_index index.php;
  #
  #
      fastcgi param SCRIPT FILENAME /scripts$fastcgi script name;
  #
      include
                   fastcgi_params;
  #}
  # deny access to .htaccess files, if Apache's document root
  # concurs with nginx's one
  #
  #location ~ /¥.ht {
      deny all;
  #
  #}
```

}

/etc/nginx/sites-avaiable/default

```
server {
            80;
  listen
  #server name localhost;
  server_name 133.242.21.74;
                                             ←---- !
  #charset koi8-r;
  #access_log /var/log/nginx/host.access.log main;
  location / {
     # root /usr/share/nginx/html;
     root /var/www/html;
                                        ←---- I
     index index.html index.htm;
  }
                            /404.html;
  #error_page 404
  # redirect server error pages to the static page /50x.html
  #
  error page 500 502 503 504 /50x.html;
  location = /50x.html 
     #root /usr/share/nginx/html;
     root /var/www/html;
                                        ←---- !
  }
  # proxy the PHP scripts to Apache listening on 127.0.0.1:80
  #
  \#location \sim ¥.php$ {
  #
      proxy_pass http://127.0.0.1;
  #}
  # pass the PHP scripts to FastCGI server listening on 127.0.0.1:9000
  #
  #location \sim ¥.php$ {
  #
                 html;
      root
      fastcqi pass 127.0.0.1:9000;
  #
  #
      fastcqi index index.php;
      fastcgi_param SCRIPT_FILENAME /scripts$fastcgi_script_name;
  #
  #
      include
                  fastcqi params;
```

```
#}
# deny access to .htaccess files, if Apache's document root
# concurs with nginx's one
#
#location ~ /¥.ht {
# deny all;
#}
}
```

/etc/nginx/sites-avaiable/d.jca.jp.conf

```
server {
    listen 80;
    server_name d.jca.jp;
```

```
error_log /var/log/nginx/d_jca_jp-error.log warn;
access_log /var/log/nginx/d_jca_jp-access.log main;
```

←----!

←---- !

←---- !

/etc/nginx/sites-avaiable/m.jca.p.conf

```
server {
listen 80;
server_name m.jca.jp;
```

error_log /var/log/nginx/m_jca_jp-error.log warn; <----- ! access_log /var/log/nginx/m_jca_jp-access.log main; <----- !

第6章 Let's Encrypt で SSL 化の設定

• Certbot インストール

参考にしたサイト: <u>https://www.codeflow.site/ja/article/how-to-secure-nginx-with-</u> let-s-encrypt-on-ubuntu-18-04

Let's Encrypt のレポジトリを追加

sudo add-apt-repository ppa:certbot/certbot

更新と確認 sudo apt update

certbot インストール sudo apt install python-certbot-nginx

念のために、Nginx の設定ファイルの確認 sudo nginx -t 念のために Nginx 再起動 sudo systemctl reload nginx

Certbot のバージョン確認 sudo certbot --version certbot 0.31.0

● d.jca.jp 用の Certbot インストール

念のために、最初に certbot だけ起動テスト

sudo certbot

Please read the Terms of Service at

https://letsencrypt.org/documents/LE-SA-v1.2-November-15-2017.pdf. You must agree in order to register with the ACME server at

https://acme-v02.api.letsencrypt.org/directory

(A)gree/(C)ancel: A

Would you be willing to share your email address with the Electronic Frontier Foundation, a founding partner of the Let's Encrypt project and the non-profit organization that develops Certbot? We'd like to send you email about our work encrypting the web, EFF news, campaigns, and ways to support digital freedom.

(Y)es/(N)o: Y

Which names would you like to activate HTTPS for?

1: d.jca.jp

2: m.jca.jp

Select the appropriate numbers separated by commas and/or spaces, or leave input

blank to select all options shown (Enter 'c' to cancel): c Please specify --domains, or --installer that will help in domain names

autodiscovery, or --cert-name for an existing certificate name.

IMPORTANT NOTES:

 Your account credentials have been saved in your Certbot configuration directory at /etc/letsencrypt. You should make a secure backup of this folder now. This configuration directory will also contain certificates and private keys obtained by Certbot so making regular backups of this folder is ideal.
 office@univ1:/etc/nginx/sites-available\$

途中で止める

次にドメイン名 d.jca.jp を指定してインストール

sudo certbot --nginx -d d.jca.jp

Saving debug log to /var/log/letsencrypt/letsencrypt.log Plugins selected: Authenticator nginx, Installer nginx Obtaining a new certificate Performing the following challenges: http-01 challenge for d.jca.jp Waiting for verification... Cleaning up challenges Deploying Certificate to VirtualHost /etc/nginx/sites-enabled/d.jca.jp.conf

Please choose whether or not to redirect HTTP traffic to HTTPS, removing HTTP access.

 No redirect - Make no further changes to the webserver configuration.
 Redirect - Make all requests redirect to secure HTTPS access. Choose this for new sites, or if you're confident your site works on HTTPS. You can undo this change by editing your web server's configuration.

Select the appropriate number [1-2] then [enter] (press 'c' to cancel): 2 ←-! Redirecting all traffic on port 80 to ssl in /etc/nginx/sites-enabled/d.jca.jp.conf

Congratulations! You have successfully enabled https://d.jca.jp

You should test your configuration at: https://www.ssllabs.com/ssltest/analyze.html?d=d.jca.jp

IMPORTANT NOTES:

Congratulations! Your certificate and chain have been saved at: /etc/letsencrypt/live/d.jca.jp/fullchain.pem
Your key file has been saved at: /etc/letsencrypt/live/d.jca.jp/privkey.pem
Your cert will expire on 2020-08-11. To obtain a new or tweaked version of this certificate in the future, simply run certbot again with the "certonly" option. To non-interactively renew *all* of your certificates, run "certbot renew"

- If you like Certbot, please consider supporting our work by:

Donating to ISRG / Let's Encrypt: https://letsencrypt.org/donate Donating to EFF: https://eff.org/donate-le

2の Redirect を指定したので、d.jca.jp は常時 SSL 対応になります。

● m.jca.jp 用の Certbot インストール

sudo certbot --nginx -d m.jca.jp

Saving debug log to /var/log/letsencrypt/letsencrypt.log Plugins selected: Authenticator nginx, Installer nginx Obtaining a new certificate Performing the following challenges: http-01 challenge for m.jca.jp Waiting for verification... Cleaning up challenges Deploying Certificate to VirtualHost /etc/nginx/sites-enabled/m.jca.jp.conf

Please choose whether or not to redirect HTTP traffic to HTTPS, removing HTTP access.

 No redirect - Make no further changes to the webserver configuration.
 Redirect - Make all requests redirect to secure HTTPS access. Choose this for new sites, or if you're confident your site works on HTTPS. You can undo this change by editing your web server's configuration.

Select the appropriate number [1-2] then [enter] (press 'c' to cancel): 2 ←---! Redirecting all traffic on port 80 to ssl in /etc/nginx/sites-enabled/m.jca.jp.conf

Congratulations! You have successfully enabled https://m.jca.jp

You should test your configuration at: https://www.ssllabs.com/ssltest/analyze.html?d=m.jca.jp

IMPORTANT NOTES:

Congratulations! Your certificate and chain have been saved at: /etc/letsencrypt/live/m.jca.jp/fullchain.pem Your key file has been saved at: /etc/letsencrypt/live/m.jca.jp/privkey.pem Your cert will expire on 2020-08-11. To obtain a new or tweaked version of this certificate in the future, simply run certbot again with the "certonly" option. To non-interactively renew *all* of your certificates, run "certbot renew"

- If you like Certbot, please consider supporting our work by:

Donating to ISRG / Let's Encrypt: https://letsencrypt.org/donate Donating to EFF: https://eff.org/donate-le

● Certbot の自動更新設定

このスクリプトは1日に2回実行され、有効期限の30日以内にある証明書を自動的に更新します。

sudo certbot renew --dry-run

Saving debug log to /var/log/letsencrypt/letsencrypt.log

Processing /etc/letsencrypt/renewal/d.jca.jp.conf

Cert not due for renewal, but simulating renewal for dry run Plugins selected: Authenticator nginx, Installer nginx Renewing an existing certificate Performing the following challenges: http-01 challenge for d.jca.jp Waiting for verification... Cleaning up challenges

new certificate deployed with reload of nginx server; fullchain is /etc/letsencrypt/live/d.jca.jp/fullchain.pem

Processing /etc/letsencrypt/renewal/m.jca.jp.conf

Cert not due for renewal, but simulating renewal for dry run Plugins selected: Authenticator nginx, Installer nginx Renewing an existing certificate Performing the following challenges:

http-01 challenge for m.jca.jp

Waiting for verification...

Cleaning up challenges

new certificate deployed with reload of nginx server; fullchain is /etc/letsencrypt/live/m.jca.jp/fullchain.pem

** DRY RUN: simulating 'certbot renew' close to cert expiry ** (The test certificates below have not been saved)

(The test certificates below have not been saved.)

Congratulations, all renewals succeeded. The following certs have been renewed:

/etc/letsencrypt/live/d.jca.jp/fullchain.pem (success) /etc/letsencrypt/live/m.jca.jp/fullchain.pem (success) ** DRY RUN: simulating 'certbot renew' close to cert expiry ** (The test certificates above have not been saved.)

IMPORTANT NOTES:

- Your account credentials have been saved in your Certbot configuration directory at /etc/letsencrypt. You should make a secure backup of this folder now. This configuration directory will also contain certificates and private keys obtained by Certbot so making regular backups of this folder is ideal.

● PHP7.4 のインストール

参考にしたサイト: https://computingforgeeks.com/how-to-install-php-on-ubuntu/

sudo apt -y install software-properties-common sudo add-apt-repository ppa:ondrej/php sudo apt-get update

sudo apt -y install php7.4

● php7.4 のバージョン確認

php -v

PHP 7.4.5 (cli) (built: Apr 19 2020 07:36:30) (NTS)
Copyright (c) The PHP Group
Zend Engine v3.4.0, Copyright (c) Zend Technologies
with Zend OPcache v7.4.5, Copyright (c), by Zend Technologies

● インストールされた PHP7.4 関連モジュール

\$ sudo apt list --installed | grep 'php*'

libapache2-mod-php7.4/bionic,now 7.4.5-1+ubuntu18.04.1+deb.sury.org+1 amd64 libgphoto2-6/bionic,now 2.5.16-2 amd64 libgphoto2-l10n/bionic,bionic,now 2.5.16-2 all libgphoto2-port12/bionic,now 2.5.16-2 amd64 libgraphene-1.0-0/bionic,now 1.8.0-1 amd64 libgraphite2-3/bionic,now 1.3.11-2 amd64 libhyphen0/bionic,now 2.8.8-5 amd64 libphonenumber7/bionic,now 7.1.0-5ubuntu5 amd64 php-common/bionic,bionic,now 2:76+ubuntu18.04.1+deb.sury.org+6 all php7.4/bionic,bionic,now 7.4.5-1+ubuntu18.04.1+deb.sury.org+1 all php7.4-cli/bionic,now 7.4.5-1+ubuntu18.04.1+deb.sury.org+1 amd64 php7.4-common/bionic,now 7.4.5-1+ubuntu18.04.1+deb.sury.org+1 amd64 php7.4-json/bionic,now 7.4.5-1+ubuntu18.04.1+deb.sury.org+1 amd64 php7.4-opcache/bionic,now 7.4.5-1+ubuntu18.04.1+deb.sury.org+1 amd64 php7.4-readline/bionic,now 7.4.5-1+ubuntu18.04.1+deb.sury.org+1 amd64 php7.4-readline/bionic,now 7.4.5-1+ubuntu18.04.1+deb.sury.org+1 amd64 php7.4-readline/bionic,now 7.4.5-1+ubuntu18.04.1+deb.sury.org+1 amd64 php7.4-readline/bionic,now 7.4.5-1+ubuntu18.04.1+deb.sury.org+1 amd64 python3-cryptography/bionic-updates,bionic-security,now 2.1.4-1ubuntu1.3 amd64 ubuntu-mate-wallpapers-photos/bionic,bionic,now 18.04.11 all xserver-xephyr/bionic-updates,now 2:1.19.6-1ubuntu4.4 amd64

● PHP と Nginx 関連モジュールのインストール

Nginx と PHP の設定

sudo systemctl disable --now apache2

Synchronizing state of apache2.service with SysV service script with /lib/systemd/systemd-sysv-install. Executing: /lib/systemd/systemd-sysv-install disable apache2

sudo apt-get install nginx php7.4-fpm

パッケージリストを読み込んでいます... 完了 依存関係ツリーを作成しています 状態情報を読み取っています... 完了 nginx はすでに最新バージョン (1.18.0-1~bionic) です。 提案パッケージ: php-pear 以下のパッケージが新たにインストールされます: php7.4-fpm アップグレード: 0 個、新規インストール: 1 個、削除: 0 個、保留: 11 個。 1,402 kB のアーカイブを取得する必要があります。 この操作後に追加で 4,729 kB のディスク容量が消費されます。 続行しますか? [Y/n] Y

動作確認

sudo systemctl status php7.4-fpm nginx

• php7.4-fpm.service - The PHP 7.4 FastCGI Process Manager

Loaded: loaded (/lib/systemd/system/php7.4-fpm.service; enabled; vendor preset: enabled)

Active: active (running) since Wed 2020-05-13 17:39:08 JST; 1min 26s ago Docs: man:php-fpm7.4(8)

Process: 12632 ExecStopPost=/usr/lib/php/php-fpm-socket-helper remove /run/php/php-fpm.sock /etc/php/7.4/fpm/pool.d/www.conf 74 (code=exite

Process: 12647 ExecStartPost=/usr/lib/php/php-fpm-socket-helper install /run/php/php-fpm.sock /etc/php/7.4/fpm/pool.d/www.conf 74 (code=exi Main PID: 12634 (php-fpm7.4)

Status: "Processes active: 0, idle: 2, Requests: 0, slow: 0, Traffic: 0req/sec" Tasks: 3 (limit: 1103)

CGroup: /system.slice/php7.4-fpm.service

tq12634 php-fpm: master process (/etc/php/7.4/fpm/php-fpm.conf) tq12645 php-fpm: pool www mq12646 php-fpm: pool www

May 13 17:39:08 univ1 systemd[1]: Stopped The PHP 7.4 FastCGI Process Manager.

May 13 17:39:08 univ1 systemd[1]: Starting The PHP 7.4 FastCGI Process Manager...

May 13 17:39:08 univ1 systemd[1]: Started The PHP 7.4 FastCGI Process Manager.

• nginx.service - nginx - high performance web server

Loaded: loaded (/lib/system/nginx.service; enabled; vendor preset: enabled)

Active: active (running) since Wed 2020-05-13 17:15:30 JST; 25min ago Docs: http://nginx.org/en/docs/

Main PID: 1128 (nginx)

Tasks: 2 (limit: 1103)

CGroup: /system.slice/nginx.service

tq1128 nginx: master process /usr/sbin/nginx -c /etc/nginx/nginx.conf mq1130 nginx: worker process

May 13 17:15:29 univ1 systemd[1]: Starting nginx - high performance web server...

May 13 17:15:30 univ1 systemd[1]: nginx.service: Can't open PID file /var/run/nginx.pid (yet?) after start: No such file or directory May 13 17:15:30 univ1 systemd[1]: Started nginx - high performance web server.

● その他の PHP 関連モジュールのインストール

参考にしたサイト: https://www.colinodell.com/blog/201911/how-to-install-php-74

bcmath,bz2,intl,gd,mbstring,mysql,zip

sudo apt-get install php7.4-cli php7.4-fpm php7.4-bcmath php7.4-curl php7.4-gd php7.4-intl php7.4-json php7.4-mbstring php7.4-mysql php7.4-opcache php7.4-sqlite3 php7.4-xml php7.4-zip

もしくは

sudo apt-get install -y php7.4-

{cli,fpm,bcmath,curl,bz2,intl,gd,json,mbstring,mysql,opcache,xml,sqlite3,zip}

● インストールされた PHP 関連モジュールの確認

sudo apt list --installed | grep 'php*'

```
libapache2-mod-php7.4/bionic,now 7.4.5-1+ubuntu18.04.1+deb.sury.org+1
amd64
libgphoto2-6/bionic,now 2.5.16-2 amd64
libgphoto2-port12/bionic,now 2.5.16-2 all
libgphoto2-port12/bionic,now 2.5.16-2 amd64
libgraphene-1.0-0/bionic,now 1.8.0-1 amd64
libgraphite2-3/bionic,now 1.3.11-2 amd64
libphonenumber7/bionic,now 2.8.8-5 amd64
libphonenumber7/bionic,now 7.1.0-5ubuntu5 amd64
php-common/bionic,bionic,now 2:76+ubuntu18.04.1+deb.sury.org+6 all
php7.4/bionic,bionic,now 7.4.5-1+ubuntu18.04.1+deb.sury.org+1 all
php7.4-bcmath/bionic,now 7.4.5-1+ubuntu18.04.1+deb.sury.org+1 amd64
php7.4-cli/bionic,now 7.4.5-1+ubuntu18.04.1+deb.sury.org+1 amd64
php7.4-common/bionic,now 7.4.5-1+ubuntu18.04.1+deb.sury.org+1 amd64
```

php7.4-curl/bionic,now 7.4.5-1+ubuntu18.04.1+deb.sury.org+1 amd64 php7.4-fpm/bionic,now 7.4.5-1+ubuntu18.04.1+deb.sury.org+1 amd64 php7.4-intl/bionic,now 7.4.5-1+ubuntu18.04.1+deb.sury.org+1 amd64 php7.4-intl/bionic,now 7.4.5-1+ubuntu18.04.1+deb.sury.org+1 amd64 php7.4-mbstring/bionic,now 7.4.5-1+ubuntu18.04.1+deb.sury.org+1 amd64 php7.4-mbstring/bionic,now 7.4.5-1+ubuntu18.04.1+deb.sury.org+1 amd64 php7.4-opcache/bionic,now 7.4.5-1+ubuntu18.04.1+deb.sury.org+1 amd64 php7.4-readline/bionic,now 7.4.5-1+ubuntu18.04.1+deb.sury.org+1 amd64 php7.4-sqlite3/bionic,now 7.4.5-1+ubuntu18.04.1+deb.sury.org+1 amd64

xserver-xephyr/bionic-updates,now 2:1.19.6-1ubuntu4.4 amd64

● Nginx での PHP 設定

✓ www.confの設定

/etc/php/7.4/fpm/pool.d/www.conf 修正

```
. . . . . . . . .
;user = www-data
user = nginx
                         ←-!
;group = www-data
group = nginx
                        ←-!
. . . . . . . . .
;listen.owner = www-data
listen.owner = nginx
                                     ←-!
;listen.group = www-data
listen.group = nginx
                              ←-!
. . . . . . . . . . . .
;security.limit_extensions = .php .php3 .php4 .php5 .php7
```

✓ ソケットのパーミッションを変更

sudo chown -R nginx:nginx /var/run/php/php7.4-fpm.sock

✓ ドメインごとの conf ファイルの編集

d.jca.jp ドメインの場合: /etc/nginx/sites-available/d.jca.jp.conf

server {

server_name d.jca.jp;

```
error_log /var/log/nginx/d_jca_jp-error.log warn;
access_log /var/log/nginx/d_jca_jp-access.log main;
```

```
root /var/www/html/d.jca.jp/web;
index index.php index.html index.htm; ←-- !
```

```
location / {
```

```
#root /usr/share/nginx/html/d.jca.jp;
#root /var/www/html/d.jca.jp;
#index index.html index.htm;
```

```
}
```

```
location ~ ¥.php$ {
   root /var/www/html/d.jca.jp/web;
   fastcgi_pass unix:/run/php/php7.4-fpm.sock;
   fastcgi_index index.php;
   fastcgi_param SCRIPT_FILENAME
$document_root$fastcgi_script_name;
   include fastcgi_params;
  }
```

```
listen 443 ssl; # managed by Certbot
```

```
ssl_certificate /etc/letsencrypt/live/d.jca.jp/fullchain.pem; # managed by
Certbot
  ssl_certificate_key /etc/letsencrypt/live/d.jca.jp/privkey.pem; # managed by
Certbot
  include /etc/letsencrypt/options-ssl-nginx.conf; # managed by Certbot
  ssl_dhparam /etc/letsencrypt/ssl-dhparams.pem; # managed by Certbot
}
server {
  if ($host = d.jca.jp) {
     return 301 https://$host$request_uri;
  } # managed by Certbot
  listen
            80;
  server_name d.jca.jp;
  return 404; # managed by Certbot
}
```

```
✓ php.iniの編集
```

```
/etc/php/7.4/fpm/php.ini
```

memory_limit = -1

.....

```
upload_max_filesize = 20M <a>←----後でこの値を修正します(73 ページで確認)
```

.....

✓ Nginx の再起動

sudo systemctl restart php7.4-fpm sudo systemctl restart nginx

✓ 自動起動の設定

sudo systemctl enable nginx

Synchronizing state of nginx.service with SysV service script with /lib/systemd/systemd-sysv-install. Executing: /lib/systemd/systemd-sysv-install enable nginx

sudo systemctl enable php7.4-fpm

Synchronizing state of php7.4-fpm.service with SysV service script with /lib/systemd/systemd-sysv-install. Executing: /lib/systemd/systemd-sysv-install enable php7.4-fpm

✓ PHP の環境確認

PHP Version 7.4.5



System	Linux univ1 4.15.0-99-generic #100-Ubuntu SMP Wed Apr 22 20:32:56 UTC 2020 x86_64			
Build Date	Apr 19 2020 07:36:30			
Server API	FPM/FastCGI			
Virtual Directory Support	disabled			
Configuration File (php.ini) Path	/etc/php/7.4/fpm			
Loaded Configuration File	/etc/php/7.4/fpm/php.ini			
Scan this dir for additional .ini files	/etc/php/7.4/fpm/conf.d			
Additional .ini files parsed	<pre>/etc/php/7.4/fpm/conf.d/10-mysqlnd.ini, /etc/php/7.4/fpm/conf.d/10-opcache.ini, /etc/php/7.4/fpm/conf.d/10-pdo.ini, /etc/php/7.4/fpm/conf.d/15-xml.ini, /etc/php/7.4/fpm/conf.d/20-bcmath.ini, /etc/php/7.4/fpm/conf.d/20-b22.ini, /etc/php/7.4/fpm/conf.d/20-calendar.ini, /etc/php/7.4/fpm/conf.d/20-ctype.ini, /etc/php/7.4/fpm/conf.d/20-curl.ini, /etc/php/7.4/fpm/conf.d/20-dom.ini, /etc/php/7.4/fpm/conf.d/20-exif.ini, /etc/php/7.4/fpm/conf.d/20-dom.ini, /etc/php/7.4/fpm/conf.d/20-exif.ini, /etc/php/7.4/fpm/conf.d/20-gd.ini, /etc/php/7.4/fpm/conf.d/20-exif.ini, /etc/php/7.4/fpm/conf.d/20-gd.ini, /etc/php/7.4/fpm/conf.d/20-gettext.ini, /etc/php/7.4/fpm/conf.d/20-iconv.ini, /etc/php/7.4/fpm/conf.d/20-gettext.ini, /etc/php/7.4/fpm/conf.d/20-iconv.ini, /etc/php/7.4/fpm/conf.d/20-mbstring.ini, /etc/php/7.4/fpm/conf.d/20-ison.ini, /etc/php/7.4/fpm/conf.d/20-mbstring.ini, /etc/php/7.4/fpm/conf.d/20-pdo_sqlite.ini, /etc/php/7.4/fpm/conf.d/20-pdo_mysql.ini, /etc/php/7.4/fpm/conf.d/20-pdo_sqlite.ini, /etc/php/7.4/fpm/conf.d/20-phar.ini, /etc/php/7.4/fpm/conf.d/20-posix.ini, /etc/php/7.4/fpm/conf.d/20-simplexml.ini, /etc/php/7.4/fpm/conf.d/20-sokets.ini, /etc/php/7.4/fpm/conf.d/20-simplexml.ini, /etc/php/7.4/fpm/conf.d/20-sysvmsg.ini, /etc/php/7.4/fpm/conf.d/20-sysvsem.ini, /etc/php/7.4/fpm/conf.d/20-sysvsmsg.ini, /etc/php/7.4/fpm/conf.d/20-sysvsem.ini, /etc/php/7.4/fpm/conf.d/20-sysvsmsg.ini, /etc/php/7.4/fpm/conf.d/20-sysvsem.ini, /etc/php/7.4/fpm/conf.d/20-sysvsmsg.ini, /etc/php/7.4/fpm/conf.d/20-sysvsem.ini, /etc/php/7.4/fpm/conf.d/20-sysvsmsg.ini, /etc/php/7.4/fpm/conf.d/20-sysvsem.ini, /etc/php/7.4/fpm/conf.d/20-sysvsm.ini, /etc/php/7.4/fpm/conf.d/20-sysvsem.ini, /etc/php/7.4/fpm/conf.d/20-xsl.ini, /etc/php/7.4/fpm/conf.d/20-sysvsem.ini, /etc/php/7.4/fpm/conf.d/20-xsl.ini, /etc/php/7.4/fpm/conf.d/20-sysvsem.ini, /etc/php/7.4/fpm/conf.d/20-xsl.ini, /etc/php/7.4/fpm/conf.d/20-zip.ini</pre>			

参考にしたサイト: https://utsu-programmer.com/environment/composer-install/

php -r "copy ('https://getcomposer.org/installer', 'composer-setup.php') ;";

sudo php composer-setup.php --install-dir=/usr/local/bin --filename=composer;

All settings correct for using Composer Downloading...

Composer (version 1.10.6) successfully installed to: /usr/local/bin/composer Use it: php /usr/local/bin/composer

composer -v

Composer version 1.10.6 2020-05-06 10:28:10

Usage: command [options] [arguments]

第9章 Drupalのインストールと設定

● d.jca.jp サイト

Nginx の設定ファイル: /etc/nginx/sites-avaiable/d.jca.jp.conf Nginx のドメインルート: /var/www/html/d.jca.jp/web

データベース名: d8_d_jcajp ユーザー名: phpmyadmin パスワード: **xxxxxxxxxx**

● m.jca.jp サイト

Drupal 8 の DrupalRoot は /var/www/html/m.jca.jp/web

✓ Nginxの設定

設定フィル: /etc/nginx/sites-available/m.jca.jp.conf

m.jca.jp.conf のソース

server {
 server_name m.jca.jp; ←-- !

#root /usr/share/nginx/html/d.jca.jp; root /var/www/html/m.jca.jp/web; index index.html index.htm index.php;

←--!

```
# --> for Drupal
location = /favicon.ico {
     log not found off;
     access_log off;
  }
  location = /robots.txt {
     allow all;
     log_not_found off;
     access_log off;
  }
  # Very rarely should these ever be accessed outside of your lan
  location \sim^* ¥.(txt|log)$ {
     allow 192.168.0.0/16;
     deny all;
  }
  location ~ ¥..*/.*¥.php$ {
     return 403;
  }
  location ~ ^/sites/.*/private/ {
     return 403;
  }
location ~ (^|/)¥. {
     return 403;
  }
  location / {
     # try files $uri @rewrite; # For Drupal <= 6
     try_files $uri /index.php?$query_string; # For Drupal >= 7
  }
  location @rewrite {
     rewrite ^{(.*)} /index.php?q=$1;
  }
# location ~ '¥.php$|^/update.php' {
       fastcgi_split_path_info ^(.+?¥.php)(|/.*)$;
#
#
       #NOTE: You should have "cgi.fix pathinfo = 0;" in php.ini
```

```
#
      include fastcgi_params;
#
      include snippets/fastcqi-php.conf;
#
      fastcgi_param SCRIPT_FILENAME $request_filename;
#
      fastcgi_intercept_errors on;
#
      fastcqi pass unix:/run/php/php7.0-fpm.sock;
#
    }
location ~ ^/sites/.*/files/styles/ { # For Drpal >= 7
     try files $uri @rewrite;
  }
  location ~* ¥.(js|css|png|jpg|jpeg|gif|ico)$ {
     expires max;
     log not found off;
  }
# <--- end Drupal 8
location ~ ¥.php$ {
  root /var/www/html/m.jca.jp/web;
                                                         ←-- !
  fastcgi_pass unix:/run/php/php7.4-fpm.sock;
  fastcqi index index.php;
  fastcgi param SCRIPT FILENAME $document root$fastcgi script name;
  include fastcgi params;
 }
  listen 443 ssl; # managed by Certbot
  ssl certificate /etc/letsencrypt/live/m.jca.jp/fullchain.pem; # managed by
Certbot
  ssl_certificate_key /etc/letsencrypt/live/m.jca.jp/privkey.pem; # managed by
Certbot
  include /etc/letsencrypt/options-ssl-nginx.conf; # managed by Certbot
  ssl dhparam /etc/letsencrypt/ssl-dhparams.pem; # managed by Certbot
}
server {
  if ($host = m.jca.jp) {
     return 301 https://$host$request uri;
  } # managed by Certbot
```

```
58
```

listen 80; server_name m.jca.jp; return 404; # managed by Certbot

}

✓ データベースの設定

d.jca.jp サイトのバックアップからデータベースを作成してデータベースをインポートしました。その後、Drupal のディレクトリを解凍して、settings.php ファイルで以下のデ ータベース設定を調整し、復元しました。

- データベース名: d8_m_jcajp
- ユーザー名: phpmyadmin
- パスワード: **xxxxxxxxxx**

第10章 Backupとupdateの設定

バックアップのシェルスクリプト

ユーザー xxxxxxxxx で Ubuntu にログイン xxxxxxxxxx / xxxxxxxxx

cd /home/jcafe/backup/ sudo ./backup.sh

✓ backup.sh ファイルの内容

```
cd /home/jcafe/backup/backup.sh
```

#!/bin/sh

chown -R nginx:nginx /var/www/html/d.jca.jp chown -R nginx:nginx /var/www/html/m.jca.jp chown -R nginx:nginx /var/www/html/opigno.jca.jp

cd /home/jcafe/backup rm -rf ./data mkdir -p ./data cp -ar ./backup.sh ./data/backup.sh

Drupal 8 d.jca.jp
cd ../
mysqldump -u root -pxxxxxxx d8_d_jcajp > /home/jcafe/work/d8_d_jcajp.sql
mv /home/jcafe/work/d8_d_jcajp.sql ./backup/data/d8_d_jcajp.sql
cd /var/www/html
tar cvzf /home/jcafe/backup/data/d_jca_jp.tar.gz ./d.jca.jp
cd /home/jcafe/backup

Drupal 8 m.jca.jp
cd ../
mysqldump -u root -pxxxxxxx d8_m_jcajp >
/home/jcafe/work/d8_m_jcajp.sql
mv /home/jcafe/work/d8_m_jcajp.sql ./backup/data/d8_m_jcajp.sql
cd /var/www/html
tar cvzf /home/jcafe/backup/data/m_jca_jp.tar.gz ./m.jca.jp
cd /home/jcafe/backup

Drupal 8 opigno.jca.jp
cd ../
mysqldump -u root -pxxxxxxx d8_opigno_jcajp >
/home/jcafe/work/d8_opigno_jcajp.sql
mv /home/jcafe/work/d8_opigno_jcajp.sql ./backup/data/d8_opigno_jcajp.sql
cd /var/www/html
tar cvzf /home/jcafe/backup/data/opigno_jca_jp.tar.gz ./opigno.jca.jp
cd /home/jcafe/backup

backup Private Directory
#tar cvzf ./backup/data/home_mydomain.tar.gz /home/drupal/mydomain

backup Nginx directory
cd /etc
tar cvzf /home/jcafe/backup/data/nginx.tar.gz ./nginx

cd /home/jcafe/backup

ls -lah ./data

tar cvzf ./d8_d_jcajp_`date +%Y%m%d_%H%M`.tar.gz ./data

ls -alh ../backup

アップデイトのシェルスクリプト

/home/jcafe/backup/ sudo ./drush_update.sh

✓ drush_update.shの内容

/home/jcafe/backup/drush_update.sh

#!/bin/sh

composer self-update
cd /home/jcafe/work
composer self-update
pwd

d.jca.jp

echo "start d.jca.jp ¥n" cd /var/www/html/d.jca.jp ./vendor/drush/drush/drush cr composer update ./vendor/drush/drush/drush cr ./vendor/drush/drush/drush updatedb ./vendor/drush/drush/drush cron sudo chown -R nginx:nginx ../d.jca.jp echo "Finished d.jca.jp ¥n" pwd

m.jca.jp echo "start m.jca.jp ¥n" cd /var/www/html/m.jca.jp ./vendor/drush/drush/drush cr composer update ./vendor/drush/drush/drush cr ./vendor/drush/drush/drush updatedb ./vendor/drush/drush/drush cron sudo chown -R nginx:nginx ../m.jca.jp echo "Finished m.jca.jp ¥n" pwd

opigno.jca.jp echo "start opigno.jca.jp ¥n" cd /var/www/html/opigno.jca.jp ./vendor/drush/drush/drush cr #Check if updates available echo "Check if Opigno updates available ¥n" composer outdated opigno/opigno lms #Check if updates will run without any issues echo "Check if Opigno updates will run without any issues ¥n" composer update --dry-run composer update composer prohibits opigno/opigno lms ./vendor/drush/drush/drush cr ./vendor/drush/drush/drush updatedb ./vendor/drush/drush/drush cron sudo chown -R nginx:nginx ../opigno.jca.jp echo "Finished opigno.jca.jp ¥n" pwd

● 追加したモジュールの内容

/var/www/html/d.jca.jp/web/modules/contrib

acknowledgements	dropdown_lang	juage	linkit	shs	
address	ds	login_se	ecurity	simplenews	
admin_content_notificati	on entity	n	nailchimp		
simplenews_scheduler					
adminimal_admin_toolba	ar entity_brows	er	mailsyst	tem	
simple_sitemap					
admin_toolbar	entity_print	me	nu_admin_	per_menu	
sitemap					
asset_injector	entity_reference_re	evisions	menu_bloc	ck smtp	
auto_entitylabel	entity_usage	m	enu_breadc	rumb	
social_media_links					
backup_migrate	exclude_node_ti	tle	menu_per	_role	
superfish					
block_class	field_formatter_clas	ss m	ietatag		
swiftmailer					
captcha	field_group	modu	le_filter		
switch_page_theme					
charts	field_group_table	mult	tiple_registr	ration	
taxonomy_manager					
chosen	field_permissions	noc	leaccess		
taxonomy_menu					
cleaner	flippy	node_vi	ew_permiss	sions text_resize	Э
coder	fontawesome	opti	mizedb	token	
config_filter	form_mode_manage	er	paragraph	S	
translation_views					
config_split	fullcalendar_view	pat	hauto	userprotect	-
content_lock	gdpr_compliance	C	_uick_node	_clone	
user_registrationpasswor	rd				
content_moderation_notifications google_analytics rabbit_hole					
video					

content_planner	google_tag	recaptcha	
video_embed_field			
crop	group	redirect	
views_bulk_operations			
csv_serialization	image_widget_c	rop reg_confi	rm_email
views_data_export			
ctools	imce	role_delegation	views_fieldsets
date_range_formatter	inline_entity_	_form rrssb	
views_infinite_scroll			
domain	lang_dropdown	scheduler	
views_slideshow			
domain_theme_switch	legal	search_api	webform
draggableviews	libraries	shield	
xls serialization			

/var/www/html/d.jca.jp/web/modules/custom

mt_cookieconsent mt_countdown

第12章 Opignoのインストールと設定

✓ 最初の HTTP 状態での Nginx 設定

Opigno のインストール先: /var/www/html/opigno.jca.jp/ のディレクトリ

/etc/nginx/sites-available/opigno.jca.jp.conf

←--!

server {
 server_name opigno.jca.jp;

←--!

```
error_log /var/log/nginx/opigno_jca_jp-error.log warn;
                                                        ←--!
  root /var/www/html/opigno.jca.jp/web;
                                                ←--!
  index index.php index.html index.htm;
# --> for Drupal
location = /favicon.ico {
     log_not_found off;
     access log off;
  }
  location = /robots.txt {
     allow all;
     log not found off;
     access_log off;
  }
  # Very rarely should these ever be accessed outside of your lan
  location \sim^* ¥.(txt|log)$ {
     allow 192.168.0.0/16;
     deny all;
  }
  location ~ 4..*/.*
     return 403;
  }
  location ~ ^/sites/.*/private/ {
     return 403;
  }
location ~ (^|)¥. {
     return 403;
  }
  location / {
     # try_files $uri @rewrite; # For Drupal <= 6</pre>
     try files $uri /index.php?$query string; # For Drupal >= 7
  }
  location @rewrite {
```

```
rewrite ^/(.*)$ /index.php?q=$1;
}
```

```
# location ~ '¥.php$|^/update.php' {
```

- # fastcgi_split_path_info ^(.+?¥.php)(|/.*)\$;
- # #NOTE: You should have "cgi.fix_pathinfo = 0;" in php.ini
- # include fastcgi_params;
- # include snippets/fastcgi-php.conf;
- # fastcgi_param SCRIPT_FILENAME \$request_filename;
- # fastcgi_intercept_errors on;
- # fastcgi_pass unix:/run/php/php7.0-fpm.sock;
- # }

```
location ~ ^/sites/.*/files/styles/ { # For Drpal >= 7
    try_files $uri @rewrite;
    }
    location ~* ¥.(js|css|png|jpg|jpeg|gif|ico)$ {
        expires max;
        log_not_found off;
    }
```

```
# <--- end Drupal 8
```

```
✓ SSLの設定
```

opigno.jca.jp 用の Certbot インストール

sudo certbot --nginx -d opigno.jca.jp

Saving debug log to /var/log/letsencrypt/letsencrypt.log Plugins selected: Authenticator nginx, Installer nginx Obtaining a new certificate Performing the following challenges: http-01 challenge for opigno.jca.jp Waiting for verification... Cleaning up challenges Deploying Certificate to VirtualHost /etc/nginx/sites-enabled/opigno.jca.jp.conf

Please choose whether or not to redirect HTTP traffic to HTTPS, removing HTTP access.

 No redirect - Make no further changes to the webserver configuration.
 Redirect - Make all requests redirect to secure HTTPS access. Choose this for new sites, or if you're confident your site works on HTTPS. You can undo this change by editing your web server's configuration.

Select the appropriate number [1-2] then [enter] (press 'c' to cancel): 2 Redirecting all traffic on port 80 to ssl in /etc/nginx/sitesenabled/opigno.jca.jp.conf

Congratulations! You have successfully enabled https://opigno.jca.jp

You should test your configuration at:

https://www.ssllabs.com/ssltest/analyze.html?d=opigno.jca.jp

IMPORTANT NOTES:

 Congratulations! Your certificate and chain have been saved at: /etc/letsencrypt/live/opigno.jca.jp/fullchain.pem
 Your key file has been saved at: /etc/letsencrypt/live/opigno.jca.jp/privkey.pem
 Your cert will expire on 2020-08-25. To obtain a new or tweaked version of this certificate in the future, simply run certbot again with the "certonly" option. To non-interactively renew *all* of your certificates, run "certbot renew"

- If you like Certbot, please consider supporting our work by:

Donating to ISRG / Let's Encrypt: https://letsencrypt.org/donate
Donating to EFF:

https://eff.org/donate-le

✓ Nginx の設定

/etc/nginx/sites-available/opigno.jca.jp.conf

opigno.jca.jp.conf ファイルの設定確認

```
server {
                                                                  ←---!
server_name opigno.jca.jp;
  error_log /var/log/nginx/opigno_jca_jp-error.log warn;
                                                                  ←---Ⅰ
  access_log /var/log/nginx/opigno_jca_jp-access.log main;
                                                                  ←---!
  root /var/www/html/opigno.jca.jp/web;
                                                                  ←---!
  index index.php index.html index.htm;
# --> for Drupal
location = /favicon.ico {
     log_not_found off;
     access_log off;
  }
  location = /robots.txt {
     allow all;
     log_not_found off;
     access_log off;
  }
  # Very rarely should these ever be accessed outside of your lan
  location \sim^* ¥.(txt|log)$ {
     allow 192.168.0.0/16;
     deny all;
  }
  location ~ 4..*/.*4.php {
     return 403;
  }
```

```
location ~ ^/sites/.*/private/ {
    return 403;
}
location ~ (^|/)¥. {
    return 403;
    }
location / {
        # try_files $uri @rewrite; # For Drupal <= 6
        try_files $uri /index.php?$query_string; # For Drupal >= 7
    }
location @rewrite {
        rewrite ^/(.*)$ /index.php?q=$1;
}
```

```
# location ~ '¥.php$|^/update.php' {
```

- # fastcgi_split_path_info ^(.+?¥.php)(|/.*)\$;
- # #NOTE: You should have "cgi.fix_pathinfo = 0;" in php.ini
- # include fastcgi_params;
- # include snippets/fastcgi-php.conf;
- # fastcgi_param SCRIPT_FILENAME \$request_filename;
- # fastcgi_intercept_errors on;
- # fastcgi_pass unix:/run/php/php7.0-fpm.sock;

```
# }
```

```
location ~ ^/sites/.*/files/styles/ { # For Drpal >= 7
    try_files $uri @rewrite;
    }
    location ~* ¥.(js|css|png|jpg|jpeg|gif|ico)$ {
        expires max;
        log_not_found off;
    }
```

```
# <--- end Drupal 8
```

```
listen 443 ssl; # managed by Certbot
```

ssl_certificate /etc/letsencrypt/live/opigno.jca.jp/fullchain.pem; # managed
by Certbot

ssl_certificate_key /etc/letsencrypt/live/opigno.jca.jp/privkey.pem; #
managed by Certbot

include /etc/letsencrypt/options-ssl-nginx.conf; # managed by Certbot ssl_dhparam /etc/letsencrypt/ssl-dhparams.pem; # managed by Certbot

}

```
server {
    if ($host = opigno.jca.jp) {
        return 301 https://$host$request_uri;
    } # managed by Certbot
```

```
server_name opigno.jca.jp;
listen 80;
return 404; # managed by Certbot
```

}

✓ Nginx のシンタックスチェック

sudo nginx -t

nginx: the configuration file /etc/nginx/nginx.conf syntax is ok

nginx: configuration file /etc/nginx/nginx.conf test is successful

Nginx の再起動

sudo systemctl reload nginx sudo systemctl restart nginx

✓ Certbot の自動更新設定

このスクリプトは1日に2回実行され、有効期限の30日以内にある証明書を自動的に更新します。

sudo certbot renew --dry-run

Saving debug log to /var/log/letsencrypt/letsencrypt.log

Processing /etc/letsencrypt/renewal/d.jca.jp.conf Cert not due for renewal, but simulating renewal for dry run Plugins selected: Authenticator nginx, Installer nginx Renewing an existing certificate new certificate deployed with reload of nginx server; fullchain is /etc/letsencrypt/live/d.jca.jp/fullchain.pem Processing /etc/letsencrypt/renewal/m.jca.jp.conf Cert not due for renewal, but simulating renewal for dry run Plugins selected: Authenticator nginx, Installer nginx Renewing an existing certificate Processing /etc/letsencrvpt/renewal/opigno.ica.ip.conf Cert not due for renewal, but simulating renewal for dry run Plugins selected: Authenticator nginx, Installer nginx Renewing an existing certificate Performing the following challenges: http-01 challenge for opigno.jca.jp Waiting for verification... Cleaning up challenges new certificate deployed with reload of nginx server; fullchain is /etc/letsencrypt/live/opigno.jca.jp/fullchain.pem ** DRY RUN: simulating 'certbot renew' close to cert expiry ** (The test certificates below have not been saved.) Congratulations, all renewals succeeded. The following certs have been renewed: /etc/letsencrypt/live/d.jca.jp/fullchain.pem (success) /etc/letsencrypt/live/m.jca.jp/fullchain.pem (success) /etc/letsencrypt/live/opigno.jca.jp/fullchain.pem (success) ** DRY RUN: simulating 'certbot renew' close to cert expiry (The test certificates above have not been saved.)

✓ Opigno をインストール中にワーニング

POST AND UPLOAD SETTINGS

PHP max post size should be bigger than max upload size. Neglecting this is known to cause issues in some installations.

/etc/php/7.4/fpm/php.ini と /etc/php/7.4/cli/php.ini ファイルの修正

·····.

Nginx 再起動 sudo systemctl reload nginx sudo systemctl restart nginx

Opigno のインストール

Drupal およびそのディストリビューションやパッケージ化されたシステムは、なるべく、 独自ドメイン、サブドメインなどに Composer でインストールする方法が好ましいです。 Composer や Drush コマンドでシステム管理する際に、Drupal Root などの環境変数を 間違えないように、別の環境が好ましいです。

今回は、/var/www/html/opigno.jca.jp/ 以下にインストールしました。 opigno.jca.jp でドメインとしてアクセスします。

Opigno の Drupal Root : /var/www/html/opigno.jca.jp/web

sudo composer create-project opigno/opigno-composer /var/www/html/opigno.jca.jp

ファイルダウンロード後に、https://opigno.jca.jp/ でインストールを実行

✓ データベースの設定

```
$databases['default']['default'] = array (
    'database' => 'd8_opigno_jcajp',
    'username' => 'phpmyadmin',
    'password' => ' xxxxxxxx ',
    'prefix' => '',
    'host' => 'localhost',
    'port' => '3306',
    'namespace' => 'Drupal¥¥Core¥¥Database¥¥Driver¥¥mysql',
    'driver' => 'mysql',
);
```

✓ settings.phpの設定

```
$settings['trusted_host_patterns'] = [
    '^opigno¥.jca¥.jp$',
];
```

✓ drush status

sudo ······vendor/drush/drush/drush status

Drupal version : 8.8.5 Site URI : http://default DB driver : mysql DB hostname : localhost DB port : 3306 DB username : phpmyadmin DB name : d8_opigno_jcajp Database : Connected Drupal bootstrap : Successful Default theme : platon Admin theme : platon PHP binary : /usr/bin/php7.4 PHP config : /etc/php/7.4/cli/php.ini PHP OS : Linux Drush script : /var/www/html/opigno.jca.jp/vendor/drush/drush/drush Drush version : 9.7.2 Drush temp : /tmp Drush configs : /var/www/html/opigno.jca.jp/vendor/drush/drush/drush.yml Install profile : opigno lms : /var/www/html/opigno.jca.jp/web Drupal root Site path : sites/default Files, Public : sites/default/files Files, Temp : /tmp

✓ Swiftmailer モジュールを追加インストール

メールなどの送信のために、Swiftmailer モジュールを追加インストールしました。

sudo composer require 'drupal/swiftmailer: ^2.0'

• Opigno のバックアップ

他の Drupal 8 と同じファイルです。

/home/jcafe/backup/backup.sh

● Opigno のアップデイト

他の Drupal 8 と同じファイルです。

/home/jcafe/backup/drush_update.sh