SSHログイン

vscodeのRemote DeveloperのRemote SSHを使って接続する
User/ユーザ/.ssh/configに追記する

```
Host sakura_vps(root)
      HostName [VPSのIPアドレス]
      User root
```

SSH Targetに追加されるので、クリックしてリモート接続できることを確認できればOK

※Remote SSHに失敗する場合
エラー内容がTarコマンドが無くてエラーになるケース
https://www.hanatare-papa.jp/entry/technology-tool-editor-1

OSアップデート

vscodeのターミナルを新規で開いてコマンド操作を行う
(TerminalでもOK)

```
[root@~]
[root@ /]# yum update
```

セキュリティ対策1

suコマンドで**root**になれるユーザを**wheel**グループのみとする

```
[root@ /]# useradd admin
[root@ /]# passwd admin
[root@ /]# usermod -G wheel admin
```

/etc/pam.d/suを編集する
【編集前】

```
#auth       required    pam_wheel.so use_uid
```

【編集後】

```
auth        required    pam_wheel.so use_uid
```

これでsu コマンドでrootになれるアカウントは、wheelグループに所属しているアカウントのみとなる

セキュリティ対策2

sshのポートを変更する

▼参考
https://labo.kon-ruri.co.jp/sakura-vps-ssh-security-protection/

sshのポート番号は、さくらVPSの場合は1025～49151番を指定する。今回は、2000番を指定

/etc/pam.d/suを編集する
【編集前】

```
#Port 22
```

【編集後】

```
#Port 2000
```

```
[root@ /]# systemctl restart sshd
[root@ /]# systemctl status sshd
[結果]
Active: active (running)
```

/usr/lib/firewalld/services/ssh.xmlを編集する
【編集前】

```
<port protocol="tcp" port="22"/>
```

【編集後】

```
<port protocol="tcp" port="2000"/>
```

ファイヤーウォール起動、自動起動設定、確認

```
[root@ /]# firewall-cmd --state
not running
[root@ /]# systemctl start firewalld.service
[root@ /]# firewall-cmd --state
running
[root@ /]# systemctl enable firewalld.service
[root@ /]# firewall-cmd --reload
[結果]
success
[root@ /]# firewall-cmd --list-all
```

さくらのコンソール画面を開いて、パケットフィルタを無効にする
SSH接続 port 2000で接続できればOK

```
ssh -p 2000 admin@[IPアドレス]
admin@[IPアドレス]'s password:
```

セキュリティ対策3

SSHでのパスワード認証を廃止し、公開鍵認証でのみ認証を可能とする

▼参考
鍵作成
https://labo.kon-ruri.co.jp/sakura-vps-ssh-security-protection/
sshd_configの設定
http://bluearth.cocolog-nifty.com/blog/2018/08/root-7716.html

vscodeの拡張機能「Remote SSH」を使って、SSHでリモートサーバ上に潜って、「root」アカウントで色々と作業をしたい。
(vimを操作に慣れておらず、root以外でサーバ上の作業をした時にPermission defindeエラーが面倒なので、エディタチックにファイルを操作ができる「Remote SSH」を使って、vscode上で作業ができるようにしたい)

下記の設定を行う
・「root」アカウントのSSH認証は、「公開鍵認証」のみでログイン可能
(rootでのパスワード認証は禁止)
※通常の運用では、rootでのSSH認証自体禁止にすることの方が多いみたい?

・「admin」アカウントも同様。

Macのターミナル上で公開鍵・秘密鍵を作成する

```
$ cd ~/.ssh
$ .ssh % ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/Users/xxxxx/.ssh/id_rsa):
sakura_vps_153.XXX.XX.XXX

$ ls
[結果]
sakura_vps_153.XXX.XX.XXX
sakura_vps_153.XXX.XX.XXX.pub
$.ssh % ssh-add -K /Users/ユーザ/.ssh/sakura_vps_153.XXX.XX.XXX

# rootの公開鍵をVPSに登録
$ ssh-copy-id -p 2000 -i /Users/ユーザ/.ssh/sakura_vps_153.XXX.XX.XXX.pub
root@153.XXX.XX.XXX

$ ssh-copy-id -p 2000 -i /Users/ユーザ/.ssh/sakura_vps_153.XXX.XX.XXX.pub
root@153.XXX.XX.XXX
```

VPSにSSHログインすると
・/root/.ssh/authorized_keys
・/home/admin/.ssh/authorized_keys
に公開鍵が登録されていることを確認

/etc/ssh/sshd_configを編集する

【編集前】

```
#PermitRootLogin yes
```

【編集後】

```
#rootのパスワード認証は弾くが、公開鍵認証は有効とする
PermitRootLogin prohibit-password
```

【編集前】

```
PasswordAuthentication yes
```

【編集後】

```
PasswordAuthentication no
```

【編集前】

```
ChallengeResponseAuthentication yes
```

【編集後】

```
ChallengeResponseAuthentication no
```

vscodeのremote sshが参照している設定ファイルを編集する
User/ユーザ/.ssh/config

【編集前】

```
Host sakura_vps(root)
      HostName [VPSのIPアドレス]
      Port [VPSのPort番号]
      User root
```

【編集後】

```
Host sakura_vps(root)
      HostName  [VPSのIPアドレス]
      Port
      User root
      #追記
      IdentityFile /Users/ユーザ/.ssh/sakura_vps_153.XXX.XX.XXX
```

rootで公開鍵方式でSSH潜って、CentOSで適当なアカウント(testuser)を発行し、Macのターミナルで下記のようになっていればOK

```
$ ssh -p 2000 testuser@153.XXX.XX.XXX
testuser@153.XXX.XX.XXX: Permission denied (publickey).
```

Apache

▼参考
https://knowledge.sakura.ad.jp/8541/

Apacheをインストールする

```
[root@ /]# yum install httpd
[root@ /]# systemctl start httpd
[root@ /]# firewall-cmd --add-service=http --zone=public --permanent
success
[root@ /]# firewall-cmd --add-service=https --zone=public --permanent
success
[root@ /]# systemctl restart firewalld
[root@ /]# systemctl enable httpd
```

http://[IPアドレス]/ でApacheのテストページが開けることを確認できればOK

コンテンツのパーミッションを設定する

```
[root@ /]# cd /var/www
[root@ www]# ls -l
drwxr-xr-x 2 root root 4096 Mar 24 23:58 cgi-bin
drwxr-xr-x 2 root root 4096 Mar 24 23:58 html
[root@ www]# chown -R apache:apache /var/www/html/
[root@ www]# ls -l
drwxr-xr-x 2 root   root   4096 Mar 24 23:58 cgi-bin
drwxr-xr-x 2 apache apache 4096 Mar 24 23:58 html
[root@ www]# chmod 775 /var/www/html/
```

※パーミッション755だとvs codeの拡張機能「remote ssh」によるファイル編集追加削除等ができないのでパーミッションは一時的に777にしておいて、後で、755に戻す
▼参考
https://my-web-note.com/wsl-vscode-develop/

```
[root@ /]# chmod 777 /var/www/html/
```

「admin」「root」アカウントのサブグループに「apache」グループを追加する。

これをやることで、「root」「admin」アカウントでも、Remote SSHでvscode上からvar/www/html直下にファイルの編集/追加/削除等ができるようになる。

```
[root@ html]# usermod -aG apache admin
[root@ html]# grep admin /etc/group
[root@ html]# usermod -aG apache root
[root@ html]# grep admin /etc/group
```

「admin」「root」アカウントでvar/www/html直下に対して、ファイル編集/追加/削除等ができることを確認出来ればOK

PHP

▼参考
https://shimaichi.blog/centos7-php8/

PHPをインストールする

```
[root@ html]# yum list installed | grep php
[結果] phpに関するパッケージが無いことを確認
[root@ html]# yum -y install
http://rpms.famillecollet.com/enterprise/remi-release-7.rpm
[root@ html]# yum list | grep php80
[root@ html]# yum -y install --enablerepo=remi-php80 php80 php80-php
php80-php-xml php80-php-xmlrpc php80-php-pecl-mcrypt php80-php-fpm
php80-php-pecl-apcu php80-php-mbstring php80-php-gd php80-php-json
php80-php-pecl-json-post php80-php-pdo php80-php-mysqlnd
php80-php-pecl-mysql php80-php-opcache php80-php-pear php80-php-soap
php80-php-intl php80-php-pear
[結果]Complete!
```

シンボリックリンク貼る、php コマンド確認

```
[root@ html]# php -v
bash: php: command not found
[root@ html]# ln -sf /usr/bin/php80 /usr/bin/php
[root@ html]# php -v
[結果]PHP 8.0.22
```

php.iniの場所を確認する

```
[root@ html]# php -i | grep php.ini
Configuration File (php.ini) Path => /etc/opt/remi/php80

[root@ html]# cp -p /etc/opt/remi/php80/php.ini
/etc/opt/remi/php80/php.ini.BAK
```

/etc/opt/remi/php80/php.iniを編集する

```
#編集前
expose_php = On
#編集後
expose_php = Off

#編集前
post_max_size = 8M
#編集後
post_max_size = 20M

#編集前
upload_max_filesize = 2M
#編集後
```

```
upload_max_filesize = 20M

#編集前
;date.timezone
#編集後
date.timezone = "Asia/Tokyo"

#編集前
;mbstring.language = Japanese
#編集後
mbstring.language = Japanese

#編集前
;mbstring.internal_encoding =
#編集後
mbstring.internal_encoding = UTF-8
#編集前
;mbstring.http_input =
#編集後
mbstring.http_input = UTF-8

#編集前
;mbstring.http_output =
#編集後
mbstring.http_output = pass

#編集前
;mbstring.encoding_translation = Off
#編集後
mbstring.encoding_translation = On

#編集前
;mbstring.detect_order = auto
#編集後
mbstring.detect_order = auto

#編集前
;mbstring.substitute_character = none
#編集後
mbstring.substitute_character = none
```

vi /var/www/html/info.php

```
<?php phpinfo(); ?>
```

Apache再起動

```
[root@ html]# systemctl restart httpd
```

ブラウザでhttp://＜IPアドレス＞/info.phpで動作確認出来ればOK

お名前**com**でドメイン取得

▼参考
https://rainbow-engine.com/onamae-domain-sakuravps-apply/

お名前comで初めてのドメイン取得の場合は、1円でドメインが取得できる

MariaDB

▼参考
https://shimaichi.blog/centos7_9-mariadb10_6/

Centos7にデフォルトで入っているMariaDBを削除

```
[root@ /]# yum list installed | grep mariadb
mariadb-libs.x86_64                 1:5.5.68-1.el7
@base
[root@ /]# yum remove mariadb-libs
[結果]Complete
```

リポジトリファイルを作成する
vi /etc/yum.repos.d/mariadb.repo

```
# MariaDB 10.6 CentOS repository list - created 2021-07-14 12:49 UTC
# http://downloads.mariadb.org/mariadb/repositories/
[mariadb]
name = MariaDB
baseurl = http://yum.mariadb.org/10.6/centos7-amd64
gpgkey=https://yum.mariadb.org/RPM-GPG-KEY-MariaDB
gpgcheck=1
enable=1
```

MariaDBをインストールする

```
[root@ /]# yum list MariaDB*
[root@ /]# yum info MariaDB-server MariaDB-client MariaDB-devel
[root@ /]# yum -y install MariaDB-devel MariaDB-client MariaDB-server
[結果]Complete
[root@ /]# yum list installed | grep mariadb
MariaDB-client.x86_64               10.6.9-1.el7.centos
@mariadb
MariaDB-common.x86_64               10.6.9-1.el7.centos
@mariadb
MariaDB-compat.x86_64               10.6.9-1.el7.centos
@mariadb
MariaDB-devel.x86_64                10.6.9-1.el7.centos
@mariadb
MariaDB-server.x86_64               10.6.9-1.el7.centos
@mariadb
```

```
MariaDB-shared.x86_64                10.6.9-1.el7.centos
@mariadb
galera-4.x86_64                      26.4.12-1.el7.centos
@mariadb
```

MariaDBを起動する

```
[root@ /]# which mysqld
/usr/sbin/mysqld
[root@ /]# systemctl status mariadb
[結果]Active: inactive(dead)
[root@ /]# systemctl start mariadb
[root@ /]# systemctl status mariadb
[結果]Active: inactive(running)
```

MariaDBを自動起動設定する

```
[root@ /]# systemctl enable mariadb
[root@ /]# systemctl is-enabled mariadb
enabled
```

MariaDBの設定コマンド

```
[root@ /]# mariadb-secure-installation
```

設定値は下記を参照
https://shimaichi.blog/centos7_9-mariadb10_6/#:~:text=%E3%81%B0OK%E3%81%A7%E3%81%99%E3%80%82-,%E5%9F%BA%E6%9C%AC%E7%9A%84%E3%81%AA%E3%82%BB%E3%82%AD%E3%83%A5%E3%83%AA%E3%83%86%E3%82%A3%E3%81%AE%E8%A8%AD%E5%AE%9A,-mariadb%2Dsecure%2Dinstallation

MariaDBの設定ファイルバックアップ

```
[root@ /]# cp -p /etc/my.cnf.d/server.cnf /etc/my.cnf.d/server.cnf.BAK
```

/etc/my.cnf.d/server.cnfを編集する
【編集前】

```
[server]
```

【編集後】

```
[server]
character-set-server=utf8
```

```
[root@ /]# systemctl restart mariadb
[root@ /]# systemctl status mariadb
[結果]
[結果]Active: inactive(running)
```

mariadb -vで下記のようになればOK

```
[root@ /]# mariadb -v
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 3
Server version: 10.6.9-MariaDB MariaDB Server
Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.
Reading history-file /root/.mysql_history
Type 'help;' or '\h' for help. Type '\c' to clear the current input
statement.

MariaDB [(none)]>
```

WordPress

▼参考
https://shimaichi.blog/centos7_9-wordpress/

wordpressコンテンツをインストールする

```
[root@ html]# cd /var/www
[root@ html]# wget https://ja.wordpress.org/latest-ja.tar.gz
[root@ html]# tar xvf latest-ja.tar.gz
[root@ html]# chown -R apache:apache .
[root@ html]# ls -l
[root@ html]# systemctl status mariadb
[root@ html]# mysql
```

wordpressのDB作成、ユーザ作成

```
MariaDB [(none)]> create database ユーザ名 default character set utf8;
MariaDB [(none)]> GRANT ALL ON ユーザ名.* TO ユーザー名@localhost
IDENTIFIED BY 'パスワード';
MariaDB [(none)]> select
host,user,select_priv,create_priv,insert_priv,grant_priv from
mysql.user;
MariaDB [(none)]> FLUSH PRIVILEGES;
```

wordpressの初期設定
https://shimaichi.blog/centos7_9-wordpress/#:~:text=Bye-,WordPress%20%E5%88%9D%E6%9C%9F%E8%A8%AD%E5%AE%9A,-Web%E3%83%96%E3%83%A9%E3%82%A6%E3%82%B6%E3%81%A7

無料SSL証明書 Let.s Encryptを導入

▼参考
https://knowledge.sakura.ad.jp/10534/

modsslがインストールされていることを確認する

```
[root@ /]# yum list installed | grep mod_ssl
mod_ssl.x86_64                          1:2.4.6-97.el7.centos.5
@updates
```

https通信がファイアウォールで許可されていることを確認する

```
[root@ /]# firewall-cmd --list-all
[結果]
services: dhcpv6-client http https ssh
```

```
[root@ /]# yum install certbot python2-certbot-apache
Is this ok [y/d/N]: y
Complete!
[root@ /]# certbot --apache -d takacube.com
```

※エラーが表示される場合
Unable to find a virtual host listening on port 80 which is currently needed for Certbot to prove to the CA that you control your domain. Please add a virtual host for port 80.
[root@ /]# vi /etc/httpd/conf/httpd.conf

```
# 証明書取得エラー
NameVirtualHost *:80

<VirtualHost *:80>
ServerAdmin root@takacube.com
DocumentRoot /var/www/wordpress
ServerName takacube.com
</VirtualHost>
```

```
[root@ /]# systemctl restart httpd
```

再度、証明書取得コマンドを実行する

```
[root@ /]# certbot --apache -d takacube.com
IMPORTANT NOTES:
[root@ /]# systemctl restart httpd
```

/etc/httpd/conf/httpd.confを確認すると自動的に追記されている

```
# 証明書取得エラー
NameVirtualHost *:80
```

```
<VirtualHost *:80>
ServerAdmin root@takacube.com
DocumentRoot /var/www/wordpress
ServerName takacube.com
RewriteEngine on
RewriteCond %{SERVER_NAME} =takacube.com
RewriteRule ^ https://%{SERVER_NAME}%{REQUEST_URI} [END,NE,R=permanent]
</VirtualHost>
Include /etc/httpd/conf/httpd-le-ssl.conf
```

```
[root@ /]# systemctl restart httpd
```

chrome上から「この証明書は有効」です。となっていることを確認する
https://takacube.com/

証明書の自動更新

```
[root@ /]# systemctl status crond
Active: active (running)
[root@ /]# crontab -e
```

毎月1日の0時?に証明書の更新
https://dabohaze.site/cron-ssl-auto-update/

```
0 0 1 */3 * /usr/bin/certbot renew && /bin/systemctl restart httpd
```

```
[root@ /]# crontab -l
0 0 1 */3 * /usr/bin/certbot renew && /bin/systemctl restart httpd
```

Apache設定ファイル修正、整理

/etc/httpd/conf/httpd.confのDocumentRootを変更する
【変更前】

```
DocumentRoot "var/www/html"
```

【変更後】

```
DocumentRoot var/www
```

【変更後】

```
<Directory "/var/www/html">
```

【変更後】

```
<Directory "/var/www">
```

VirtualHostをドメイン毎にファイル分けする
現状は、/httpd.confファイルの中に、VirtualHostを記載している。
将来、1つのVPSの中で、複数のドメインを運用するってなった場合は、/httpd.confファイルの中で、ドメイン毎にVirtualHostを記載してしまうと、ごちゃごちゃしてわかりにくいので、「takacube.com」専用にVirtualHostファイルを切って、/httpd.confファイルでIncludeする
/etc/httpd/conf.d/takacube.com.confを新規作成する

```
# 証明書取得エラー
NameVirtualHost *:80

<VirtualHost *:80>
      ServerAdmin root@takacube.com
      DocumentRoot /var/www/wordpress
      ServerName takacube.com
      RewriteEngine on
      RewriteCond %{SERVER_NAME} =takacube.com
      RewriteRule ^ https://%{SERVER_NAME}%{REQUEST_URI}
[END,NE,R=permanent]
</VirtualHost>
Include /etc/httpd/conf/httpd-le-ssl.conf
```

/etc/httpd/conf/httpd.confの中の下記箇所を削除する

```
# 証明書取得エラー
NameVirtualHost *:80

<VirtualHost *:80>
ServerAdmin root@takacube.com
DocumentRoot /var/www/wordpress
ServerName takacube.com
RewriteEngine on
RewriteCond %{SERVER_NAME} =takacube.com
RewriteRule ^ https://%{SERVER_NAME}%{REQUEST_URI} [END,NE,R=permanent]
</VirtualHost>
Include /etc/httpd/conf/httpd-le-ssl.conf
```

※ 必要があれば
/etc/httpd/conf/httpd-le-ssl.confを編集する
【変更後】

```
DocumentRoot /var/www/html
```

【変更後】

```
DocumentRoot /var/www/wordpress
```

phpMyAdmin

▼参考
https://sei-simple.com/it/centos7-phpmyadmin/
https://qiita.com/ekzemplaro/items/f7e0b68568fbbd452b57

```
[root@ html]# yum -y install --enablerepo=remi-php80 phpMyAdmin
[結果]
Installed:
  phpMyAdmin.noarch 0:4.4.15.10-6.el7
```

```
[root@ html]# yum list installed | grep phpMyAdmin
phpMyAdmin.noarch
[root@ html]# cd /etc/httpd/conf.d/
[root@ conf.d]# cp -p phpMyAdmin.conf phpMyAdmin.conf.BAK
```

mod_sslをインストールする

```
[root@ /]# yum search mod_ssl
[root@ /]# yum install mod_ssl
[結果]
Installed:
  mod_ssl.x86_64 1:2.4.6-97.el7.centos.5

Installed:[root@ /]# yum list installed | grep mod_ssl
```

/etc/httpd/conf.d/phpMyAdmin.conf
http://takacube.com/phpMyAdminT2V8jYfM/

phpMyAdmin.confを下記の差分のように編集する
・ModSSLの適用
・自宅のIPアドレスからのみphpMyAdminをアクセスできるようにする
https://takacube.com/wp-content/uploads/2022/09/phpMyAdmin.conf_%E5%B7%AE%E5%88%86.html

```
[root@ html]# systemctl restart httpd
```