

[ホーム](#) [タイムライン](#) [トレンド](#) [質問](#) [公式イベント](#) [公式コラム](#) [Findyコンテンツ](#) [0](#)

この記事は最終更新日から5年以上が経過しています。

@3no3\_tw

# Raspberry Pi3のLAN外からのSSH接続設定方法

SSH RaspberryPi Raspberrypi3

最終更新日 2019年03月17日 投稿日 2017年08月21日

## はじめに

Raspberry Piを操作するのにキーボードを繋げてディスプレイを繋げて…と準備が面倒で

手元のPC、できれば自宅LAN外からも操作したいと思い、SSH接続の設定方法を記事にしています。

## 環境

- RaspberryPi3 Model B
- OS:Raspbian Jessie with PIXEL
- LANへはRaspberryPi3内蔵の無線接続
- Cygwin Version2.8.2-1にOpenSSHを導入してSSH接続

## 参考サイト



と  
思います。

- [Raspberry Piに外部ネットワークからアクセスできるようにして携帯でペットを遠隔監視する方法](#)
- [Raspberry Pi 3 \(Raspbian Jessie\)の無線LANに固定IPアドレスを設定する](#)
- [インフラエンジニアじゃなくても押さえておきたいSSHの基礎知識](#)
- [Raspberry Pi のセキュリティ設定](#)
- [【初心者でもわかりやすい】Linuxのviの使い方をゼロから！](#)
- [【Cygwin】 Windows8からSSH接続する時のつまづきメモ！](#)

## SSH接続設定

---

SSH接続設定について記述してゆきます。

## Raspberry PiのSSH接続設定

---

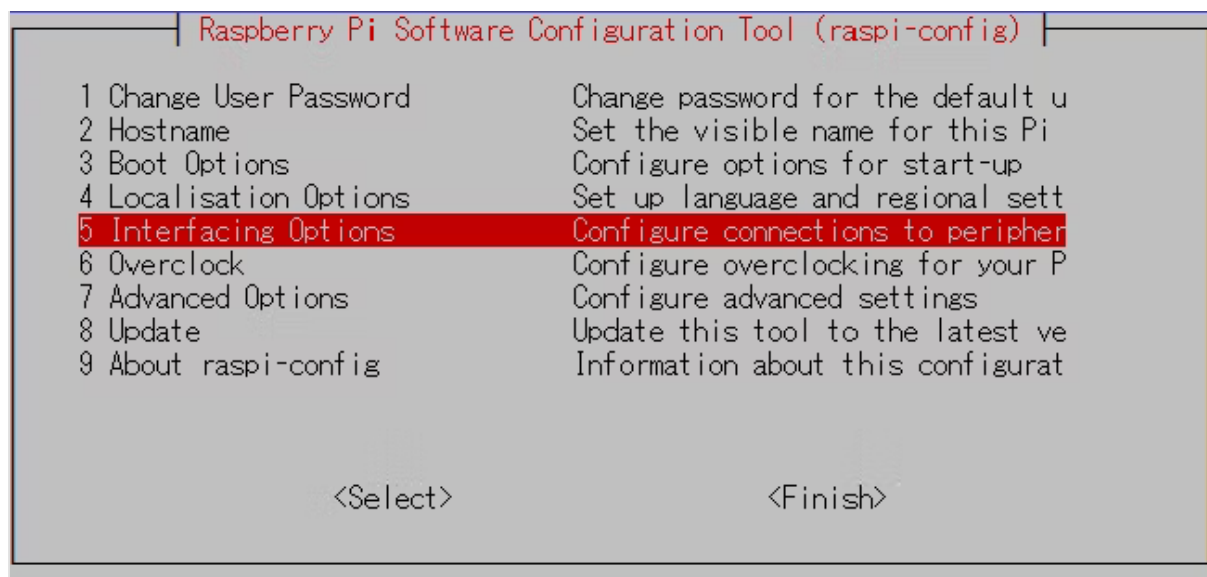
Raspberry PiのSSH接続設定をONにします。

### 1. Raspberry Piのターミナル画面で以下のコマンドを入力して管理画面の表示

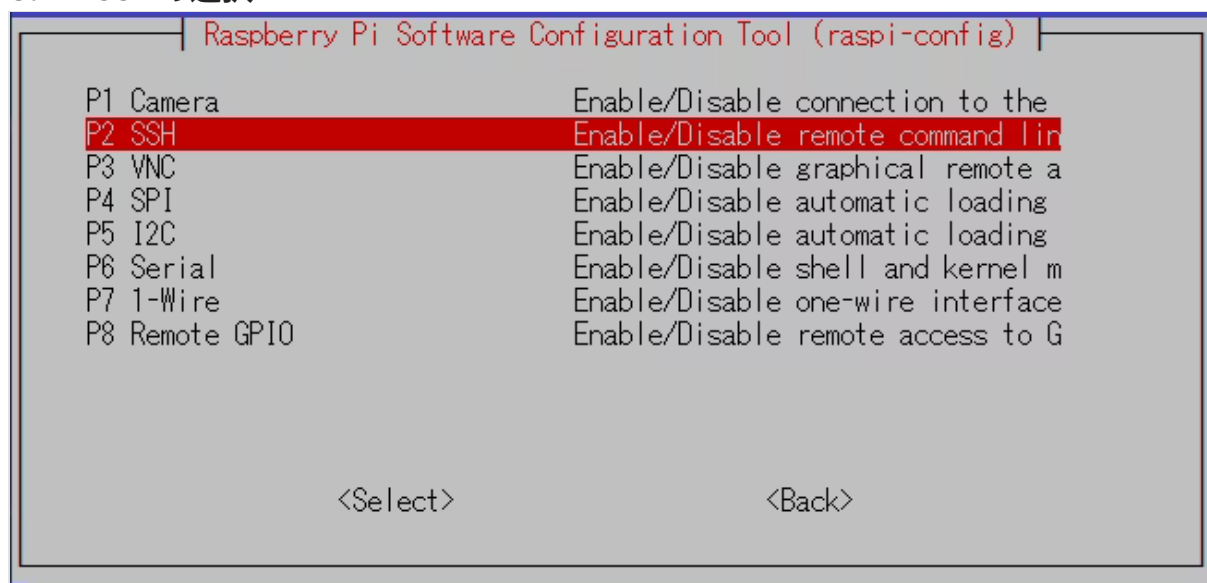
```
$ sudo raspi-config
```

### 2. Interfacing Optionsを選択





### 3. P2 SSHの選択



### 4. 以下の確認画面で「はい(Yes)」を選択

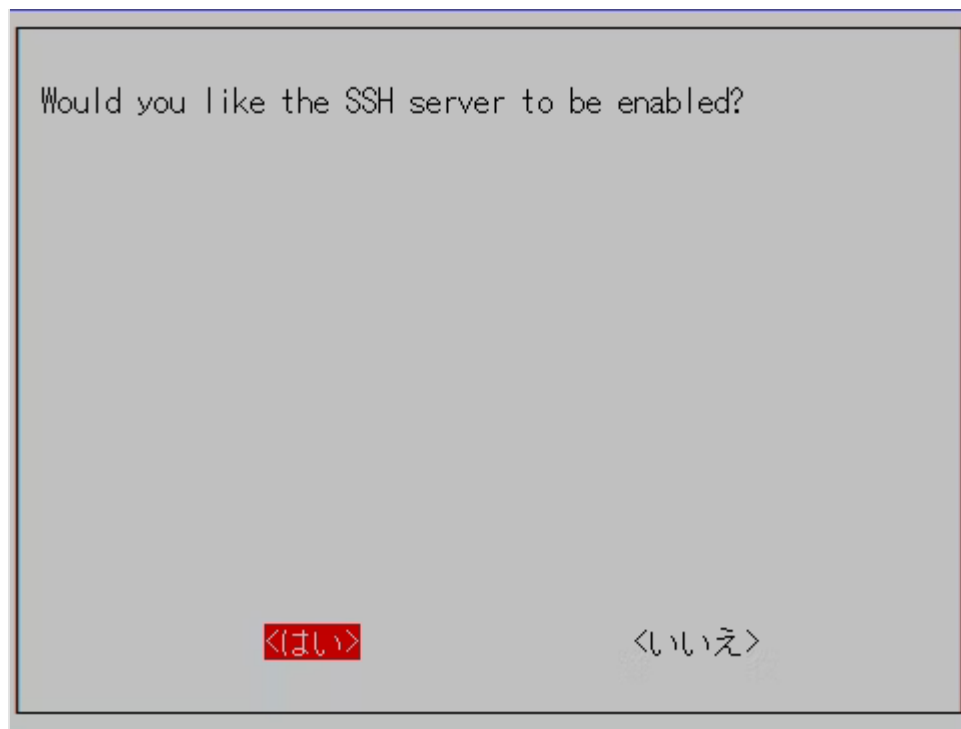


213



259





## 5. 変更を適用するためにRaspberry Piの再起動

Raspberry PiのSSH接続設定は以上です。

## Raspberry PiのローカルIP固定設定

ネットワーク環境は人によってマチマチだと思いますが、インターネット側(WAN)とRouterによって区切られたRaspberry PiやPCが属するネットワーク(LAN)を構築している人が主だと思います。

LANにおいて、Routerの設定によってはLANに属するコンピューターに動的にローカルIPが割り振られています。

SSH接続はこのローカルIPを利用しているので、動的にローカルIPが変更されては接続できなくなってしまいます。

なので、安定してRaspberry PiにSSH接続するためには固定的にローカルIPを設定する必要があります。

### 1. Raspberry Piのターミナル画面から以下のコマンドを実行してdhcpcd.confを開く

```
$ sudo vi /etc/dhcpcd.conf
```



213



259



Vimの操作方法は最初に紹介した以下のサイトがわかりやすいので参照してください。

- [【初心者でもわかりやすい】Linuxのviの使い方をゼロから！](#)

私はこのコマンドをファイルを開いて編集するコマンドなのだろうな程度の認識で実行したら、ファイルの編集方法どころか、ファイルからの離脱方法もわからずに焦った記憶があるので気をつけてください。

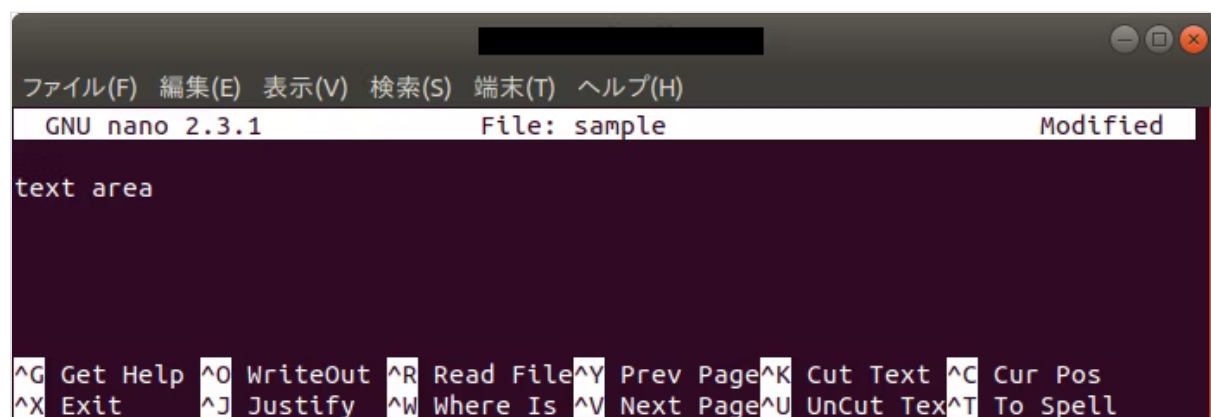
とりあえずEscapeを連打して「:q!」入力でEnterとすれば、変更は保存されずにファイルから離脱できます。

viの操作方法を見て、マジかよ...とウンザリしてしまった方がいたら、もっと親切なエディターのnanoもあるので、こちらを試ってみてはどうでしょうか。

nanoはviと同じように、以下のコマンドで起動できます。

```
$ sudo nano /etc/dhcpd.conf
```

nanoは以下の起動画面を見て頂けたらわかるように、下部にコマンドのガイドを表示してくれています。また、テキストの編集もGUIの環境で使用しているようなエディターとほぼ同じ操作で実行できるので、使いやすいかと思います。



nanoの具体的な操作方法は、以下の投稿を参照して頂くと良いと思います。

- [GNU nanoを使いこなす](#)
- [The GNU nano homepage--Onlile Manual](#)

nanoの使用に関する記述は[@starmineouji](#)さんから頂いたコメントを参考に記述しました。

[@starmineouji](#)さん、コメントをありがとうございました!!



213



259

...

```
/etc/dhcpd.conf
```

```
interface wlan0
static ip_address=192.168.11.x/24      #固定するローカルIP
static routers=192.168.11.1           #デフォルトゲートウェイのIP
static domain_name_servers=192.168.11.1 #DNSサーバーのIP?
```

Routerによって記述は変わるので、Routerの管理画面などから設定を確認して読み替えてください。

"static ip\_address"に今回、固定するローカルIPを設定してください。DHCPで割り当てられるローカルIPの範囲外で設定することをおすすめします。

"static domain\_name\_servers"については調べてみたのですがよくわからず、LAN内でDNSサーバーを運用している場合はそのIPを設定(?)。そうでなければ、デフォルトゲートウェイと同じIPが良いと思います。(私はそれで問題ありませんでした)

### 3. 変更を適用するためにRaspberry Piの再起動

Raspberry PiのローカルIP固定設定は以上です。

## SSH接続確認(同一LAN内から)

ここまでの設定でRaspberry Piと同一のLANからSSHで接続できるようになりました。

実際にSSHで接続できるか確認してみます。

私はCygwinにOpenSSHを導入した環境から接続しています。

CygwinとOpenSSHを導入しようという方は以下のサイトを参照して頂くと良いと思います。

- [【Cygwin】Windows8からSSH接続する時のつまづきメモ！](#)

では、SSH接続をしてみます。以下のコマンドで接続できます。

```
#ssh [User名]@[接続先IP] -p [接続先Port番号]
$ ssh pi@192.168.11.x -p 22
```

特に設定を変更していなければ上記のコマンドで接続できるので、接続先IPだけ設定し、固定ローカルIPに読み替えてください。



213



259

...

設定)と入力すれば接続できます。

## 公開鍵認証方式の設定

ここまででRaspberry Piと同一LAN内からSSH接続できるようになりました。  
あとは、Routerのポート開放を行えばLAN外からもRaspberry PiにSSH接続できるようになります。

ですが、デフォルトのpiユーザー、パスワードもデフォルトのままでパスワード認証方式でLAN外からもRaspberry PiにSSH接続できてしまうというのはセキュリティ的に危険なので、その前に公開鍵認証方式<sup>2</sup>を設定します。  
公開鍵認証方式の説明は以下のサイトがわかりやすいかと思います。

- [共通鍵暗号と公開鍵暗号の解説とSSHでの認証手順](#)

順番に公開鍵認証方式の設定手順について記述してゆきます。

### 1. 秘密鍵と公開鍵の作成

アクセスするPC側のCygwinなどで以下のコマンドを実行

```
$ ssh-keygen -t rsa
```

RSAという暗号方式で秘密鍵と公開鍵が生成されます。  
以下のメッセージで秘密鍵と公開鍵の保存先を聞かれます。

```
Generating public/private rsa key pair.  
Enter file in which to save the key (/home/username/.ssh/id_rsa):
```

問題なければ何も入力しないで大丈夫です。(デフォルトの/home/username/.sshに保存)

Enterで次へ。

以下のメッセージでパスフレーズを聞かれます。

```
Enter passphrase (empty for no passphrase):
```



何も入力しなくても問題ありませんが、セキュリティ的に設定した方が良いです。  
ここで入力したパスフレーズは公開鍵認証方式で接続する時に入力を求められるので忘れずに。  
以下のメッセージでパスフレーズを再度、聞かれます。

```
Enter same passphrase again:
```

上で入力したパスフレーズと同じものを再入力します。  
以下のようなメッセージが出力されれば、秘密鍵と公開鍵の作成は完了です。

```
Your identification has been saved in /home/username/.ssh/id_rsa.  
Your public key has been saved in /home/username/.ssh/id_rsa.pub.  
The key fingerprint is:  
SHA256:xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx username@hostname  
The key's randomart image is:  
+---[RSA 2048]-----+  
|  
|  
|  
|  
|  
|  
|  
|  
|  
+-----[SHA256]-----+
```

## 2. Raspberry Pi側に.sshディレクトリの作成

Raspberry Pi側のpiユーザーでログインして、ターミナルで以下のコマンドを実行。

```
#注)usernameは各自のUser名に読み替えてください  
$ mkdir /home/username/.ssh/
```

ホームディレクトリに.sshディレクトリを作成します。

## 3. Raspberry Pi側に公開鍵を転送する

秘密鍵と公開鍵を作成したPC側のC:\Users\usernameから以下のコマンドを実行



213



259

...



#注)usernameは各自のUser名に読み替えてください

```
#scp -P [ポート番号] [id_rsa.pubのファイルパス] [User名]@[Raspberry Piの固定I  
$ scp -P 22 /home/username/.ssh/id_rsa.pub pi@192.168.11.x:/home/username/
```

公開鍵を作成した.sshに移動してから、ファイル転送を行うコマンドです。

以下のメッセージでパスワードが聞かれるので、piのパスワード(デフォルトでraspberrry)を入力。

```
pi@192.168.11.x's password:
```

Raspberry Piへ公開鍵の転送は完了しました。

次は、公開鍵を利用できるように登録します。

Raspberry Pi側のターミナルで以下のコマンドを実行。

#注)usernameは各自のUser名に読み替えてください

```
$ cat /home/username/Documents/id_rsa.pub >> /home/username/.ssh/author:
```

公開鍵を転送したディレクトリに移動して、id\_rsa.pub(作成した公開鍵)を公開鍵として利用できるようにauthorized\_keysに追記するコマンドです。

authorized\_keysはキーペアを作成するたびに公開鍵を追記する必要があるのですが、すでに記述してあるキー情報を削除してしまわないように注意してください。

次に、authorized\_keysのパーミッションを設定します。

以下のコマンドでパーミッションを設定できます。

#注)usernameは各自のUser名に読み替えてください

```
$ chmod 600 /home/username/.ssh/authorized_keys
```

authorized\_keysの作成ユーザーだけが読み・書きできるように設定されました。

最後に、id\_rsa.pubは不要なので削除します。

#注)usernameは各自のUser名に読み替えてください

```
$ rm /home/username/Documents/id_rsa.pub
```



213



259



以上で公開鍵認証方式の設定は完了です。

SSH接続で以下のように表示されて接続できれば、設定できていることになります。

パスフレーズにはキーを作成した時のものを入力してください。

#注)usernameは各自のUser名に読み替えてください

```
#ssh [User名]@[接続先IP] -i [公開鍵を設置したパス] -p [接続先Port番号]
```

```
$ ssh pi@192.168.11.x -i /home/username/.ssh/id_rsa -p 22
```

Enter passphrase for key '/home/user/.ssh/id\_rsa':

## パスワード認証でのSSH接続の無効化設定

公開鍵認証方式の設定は完了しましたが、piユーザーのパスワード認証でのSSH接続も有効なままなので、パスワード認証でのSSH接続の無効化設定も行います。

### 1. Raspberry Piのターミナルで以下のコマンドを実行

```
$ sudo vi /etc/ssh/sshd_config
```

Vimでsshd\_configを開いたら、PasswordAuthenticationの記述を探してyes→noに修正する。

```
#PasswordAuthentication yes
```

```
PasswordAuthentication no
```

### 2. Raspberry Piの再起動

パスワード認証でのSSH接続の無効化設定は以上です。

## LAN外からRaspberry PiにSSH接続するための設定



213



259



したが、LAN外からは、まだ接続できるようになっていません。

なので、ここではLAN外からもRaspberry PiにSSH接続できるように設定してゆきたいと思います。

といってもLANを構成しているRouter(L3Switchという方もいるかもしれませんが)にポート開放設定をするだけです。

しかも、Routerのポート開放の設定方法はメーカーや機種によって異なるので、各々調べてくださいとなってしまいます。

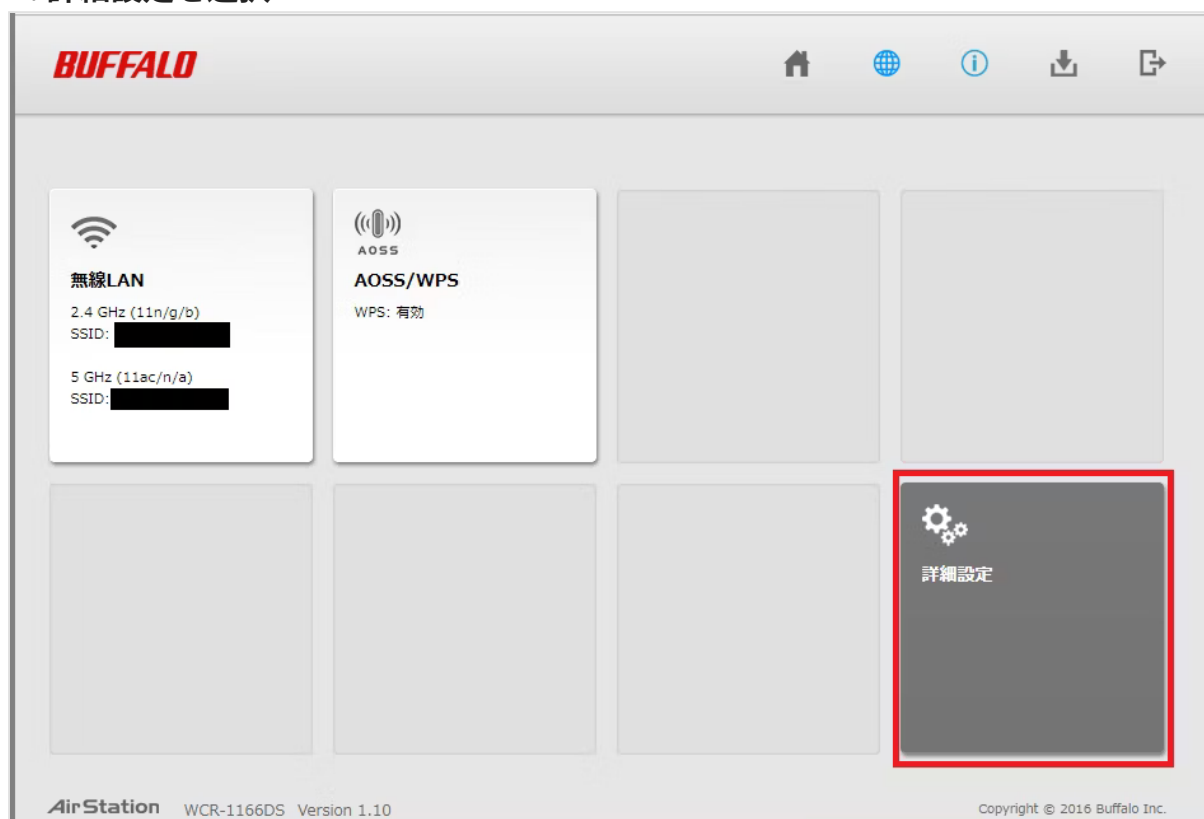
丸投げだと少々無責任なので、私の環境で使用しているRouter WCR-1166DS(BUFFALO製)でのポート開放設定方法を参考程度に記述します。

## 1. Routerの管理画面を開く

BUFFALO製のRouterの場合、以下のサイトから管理画面へのアクセス方法が確認できます。

- [管理画面を開くための手順](#)

## 2. 詳細設定を選択



## 3. 「セキュリティー」→「ポート変換」を選択

Internet  
LAN  
無線設定  
**セキュリティ**  
ファイアウォール  
IPフィルター  
VPN/パススルー  
**ポート変換**  
DMZ  
UPnP  
管理  
ステータス

### ポート変換の新規追加

グループ	新規追加 ▼	新規追加: [任意の名前]
Internet側IPアドレス	エアステーションのInternet側IPアドレス ▼ 手動設定: [ ]	
プロトコル	<input type="radio"/> すべて	
	<input type="radio"/> ICMP	
	<input type="radio"/> 任意	プロトコル番号: [ ]
	<input checked="" type="radio"/> TCP/UDP	SSH(TCPポート:22) ▼ 指定の仕方 任意のTCP/UDPポート: [ ]
LAN側IPアドレス	192.168.11.x	
LAN側ポート	TCP/UDPポート: 22	

新規追加

### ポート変換登録情報

グループ	Internet側IPアドレス LAN側IPアドレス	プロトコル LAN側ポート	操作
	エアステーションのInternet側IPアドレス	TCPポート TCPポート	OFF 修正 削除

#### 4. 上の画面のように入力して「新規追加」を押下

"LAN側IPアドレス"にはRaspberry Piに設定した固定ローカルIPを入力してください。

画面下半分にポート変換登録情報として表示されていることを確認してください。

以上でポート開放設定は完了です。

## SSH接続確認(LAN外から)

LAN外からもSSH接続できるように設定が完了しました。

最後にLAN外からもSSH接続できるか確認します。

SSHの接続コマンドは基本的に同一LAN内からの接続コマンドと同じです。

ですが、気をつけて欲しいのは接続先IPはRaspberry PiがWAN(Internet)に出た時に割り振られているグローバルIPを指定するということです。

グローバルIPの確認方法はRaspberry Piのターミナルで以下のコマンドを実行することで確認できます。

```
$ curl ifconfig.io
```



213



259



また、グローバルIPでの接続はRaspberry Piと同一のLAN内からの環境ではできないので、手軽な方法としてスマートフォンでのテザリング機能を利用して確認するのがおすすめです。

## SSH接続コマンド

```
$ ssh pi@x.x.x.x -p 22    #ssh [User名]@[グローバルIP] -p [接続先Port番号]
```

上記の方法でLAN外からSSH接続できることは確認できたと思いますが、実は一つ懸念点がありまして。

上記の方法はグローバルIPを直接指定してSSH接続しているため、固定でグローバルIPを使用できる環境でない場合には、グローバルIPが変わってしまった時に接続できなくなってしまう可能性があります。

SSH接続する前に都度、グローバルIPを確認できれば良いのですが常にそうできるとは限らないと思うので...

グローバルIPが変わる???という方は、NATとかNAPTあたりを調べてみると良いかもしれません。

- [NAT\(Network Address Translation\)](#)
- [NAPT\(IPマスカレード\)](#)

それで、グローバルIPの問題を解決するためにはDDNSを使用することおすすめします。

## DDNS【Dynamic DNS】

DDNSとは、IPアドレスが頻繁に変わるホストに固定的にドメイン名を割り当て、アドレス変更に即座に追従してDNS情報を更新するシステム。また、その仕組みを利用して提供される動的なDNSサービス。

[IT-用語辞典 e-Words](#)より

動的に変わってしまうグローバルIPに対して、固定のドメイン名を紐付け、そのドメイン名を指定してSSH接続しようということです。

では、DDNSの設定手順を見てゆきましょう。

とは言っても、参考サイトの手順を見てくださいと、ほぼ丸投げですが、やることは2つです。



## 2. グローバルIPが変更された時に、新グローバルIPとドメイン名(こちらは変わらない)の紐付けの自動設定

まず、手順1に関しては[MyDNS.JP](#)のサービスを使いました。

[MyDNS.JP](#)での設定手順は以下のサイトを参考にしてください。

- [DDNS（ダイナミックDNSサービス）とは？無料DDNSサービスの「mydns」で、自宅PCを設定してみた](#)

特に[MyDNS.JP](#)でなければならい、というわけではないので、以下のサイト等を参考に、皆さんの用途に合うものを選択してください。

- [DDNSの無料サービスでオススメな3つの「ieServer」「mydns」「No-IP」の特徴と利用手順をまとめてみた](#)

次に、手順2についてですが、[DiCE](#)というフリーソフトを使います。

この[DiCE](#)が[MyDNS.JP](#)等で設定しなければならない、グローバルIPとドメイン名の紐付け設定をグローバルIPが変わった場合に、自動で行ってくれます。

設定手順については以下のサイトを参考にしてください。

- [DDNSを利用する上で、面倒な「IPアドレスが変更される度にする更新作業」を「DiCE」で自動化してみた](#)

実は、この[DiCE](#)はWindows版とLinux版しか用意されておらず、MacOSXの方はどうすれば良いのかということがあります。

MacOSXの方は以下のような方法があるようです。

(MacOSXの方法は未検証です。すいません...)

- [MyDNSの割当IPを定期的に更新（設定）する方法](#)

最後に、設定したドメイン名でLAN外からSSH接続してみましょう。

接続コマンドは以下です。

```
$ ssh pi@xyz.jp -p 22      #ssh [User名]@[ドメイン名] -p [接続先Port番号]
```

DDNSの設定に関する記述は[@jazzwalker](#)さんから頂いたコメントを参考に記述しました。

[@jazzwalker](#)さん、コメントをありがとうございました!!



213



259

...

## 修正履歴

2018/11/18

### 修正点

- 公開鍵認証方式でSSH接続の場合のsshコマンドに `-i` オプション指定を追加
- ホームディレクトリを表すチルダ(`~`)を一律、`/home/username/` の記述に修正

@katamalixさんから以下の指摘を頂きました。

「鍵認証のところで、鍵のパスを指定するべきではないでしょうか？」

公開鍵認証方式でSSH接続する以下の部分です。

```
$ ssh pi@192.168.11.x -p 22 #修正前
$ ssh pi@192.168.11.x -i ~/username/.ssh/id_rsa -p 22 #@katamalixさん
```

おっしゃる通りです!!

公開鍵を設置したパスを `-i` のオプションで指定すべきですね。

ですが、一部加筆させてください。

(@katamalixさん、指摘された立場なのにゴメンナサイ...)

ホームディレクトリを表すチルダ(`~`)は `/home/username/` を表すので、パスの記述は以下のようにさせてください。

```
#ssh [User名]@[接続先IP] -i [公開鍵を設置したパス] -p [接続先Port番号]
$ ssh pi@192.168.11.x -i /home/username/.ssh/id_rsa -p 22
```

この修正に合わせて、実行するユーザーや `usermod -d` コマンドで動的に変わる可能性のあるホームディレクトリを表すチルダ(`~`)を、一律、`/home/username/` の記述に修正しました。

ところで、公開鍵認証方式のsshコマンドで `-i` のオプションを指定しなくても接続できたような気がしたのだけれどなあ、なんて思っていたのですが、以下のような可能性があるかもしれません。

Raspberry Pi側のconfigファイルで `/etc/ssh/sshd_config` といものがあり、このconfigファイルで `AuthorizedKeysFile` のパラメータに対して、公開鍵ファイルの





```
/etc/ssh/sshd_config
```

#%hはログインUserのホームディレクトリを指す

```
AuthorizedKeysFile %h/.ssh/authorized_keys
```

公開鍵認証方式のsshコマンドで `-i` オプションを省略した場合に、ここで設定されている公開鍵ファイルのパスをデフォルトとして、公開鍵を利用しているのでは、と予想しています。

なので、検証したいところなのですが、ただいまRaspberry Piの機嫌が悪く、Raspberry Piがネットワークに繋がらずSSH接続できない状態で...

本当に予想が正しかったのかどうかの検証結果は後日、更新したいと思います。

->2018/11/24の修正履歴で検証結果を追記しました。

いずれにせよ、`/etc/ssh/sshd_config` の設定は環境によってマチマチだと思うので、様々な環境の方に、この投稿を閲覧して頂いている限りは公開鍵認証方式のsshコマンドで `-i` オプションを明示すべきだと思います。

結論は[@katamalix](#)さんの指摘に感謝ということです。

[@katamalix](#)さん、指摘をありがとうございました!!

## 2018/11/24

### 修正点

- 公開鍵認証方式のsshコマンドで `-i` オプションを省略した場合の検証結果を追記
- 結論としては、公開鍵認証方式のsshコマンドで `-i` オプションを省略すると、`sshd_config`で設定されている `AuthorizedKeysFile` を公開鍵ファイルのパスとして参照する

公開鍵認証方式のsshコマンドで `-i` オプションを省略した場合に、`/etc/ssh/sshd_config` で設定されている公開鍵ファイルのパスをデフォルトとして、公開鍵を利用しているのでは、との予想に対して検証しました。

つまり、公開鍵認証におけるsshコマンドで公開鍵ファイルのパスの指定は省略可能か否かという検証です。

#以下のような-iオプションを指定しないコマンドでも公開鍵認証方式でSSH接続可能か

```
$ ssh pi@192.168.11.x -p 22
```



213



259

...



※前提としてパスワード認証でのSSH接続は無効に設定しているため、公開鍵認証でのみSSH接続可能となっています。

最初に各用語についてですが、sshd\_configはOpenSSHのconfigファイルで、OpenSSHとsshd\_configの詳細については以下の投稿を参照してください。

OpenSSH (Open Secure Shell) は、SSHプロトコルを利用するためのソフトウェアで、SSHサーバおよびSSHクライアントを含む。

[OpenSSH-Wikipedia](#)より引用

- [SSHD\\_CONFIG](#)

では、検証内容について解説してゆきます。

まずはsshd\_configを以下のように設定してSSH接続しました。

```
/etc/ssh/sshd_config
```

#%hはログインUserのホームディレクトリを指す

```
AuthorizedKeysFile %h/.ssh/authorized_keys
```

RaspberryPi側の状態は以下です。

```
$ ls /home/username/.ssh/authorized_keys
/home/username/.ssh/authorized_keys
```

sshd\_configの AuthorizedKeysFile と公開鍵ファイルのパスが一致しているので、 -i オプションを指定しないsshコマンドで接続できました。

次に、RaspberryPi側の状態を以下のように変更して、再度SSH接続。

```
$ mkdir /home/username/.ssh/try
$ mv /home/username/.ssh/authorized_keys /home/username/.ssh/try/
```

sshd\_configの AuthorizedKeysFile と公開鍵ファイルのパスが一致しないので、 -i オプションを指定しないsshコマンドで接続できなくなりました。



213



259



```
/etc/ssh/sshd_config
```

```
#%hはログインUserのホームディレクトリを指す
```

```
AuthorizedKeysFile %h/.ssh/authorized_keys
```

再び `-i` オプションを指定しないsshコマンドで接続できるようになりました。

これらのことから、`-i` オプションを指定しない場合は公開鍵ファイルのパスとして、`sshd_config`の `AuthorizedKeysFile` を参照しているのだなとわかりました。

また、`sshd_config`の `AuthorizedKeysFile` をコメントアウトして削除し、`-i` オプションを指定しないsshコマンドで接続を試みました。

```
/etc/ssh/sshd_config
```

```
#AuthorizedKeysFile %h/.ssh/try/authorized_keys
```

`-i` オプションを指定しないsshコマンドで接続できなくなりました。

まあ、`sshd_config`の設定が無くなってしまったので当然かと思いつつ、公開鍵ファイルは元の場所に戻して、`-i` オプションを指定しないsshコマンドで再度、接続を試みました。

## RaspberryPi側の状態

```
$ mv /home/username/.ssh/try/authorized_keys /home/username/.ssh/
```

```
$ rm -r /home/username/.ssh/try
```

そうすると、`sshd_config`の `AuthorizedKeysFile` が設定が無い状態でも、`-i` オプションを指定しないsshコマンドで接続できてしまうんですね。

どうやら私の環境では、`sshd_config`の `AuthorizedKeysFile` が設定されてない場合には、公開鍵ファイルのパスは `/home/username/.ssh/authorized_keys` がデフォルトになるようです。

ですが、`sshd_config`の `AuthorizedKeysFile` が設定されてない場合の動作は、他のすべての環境でも同じようになるかは保証できません。

以上から、検証結果は以下のようになりました。



213



259



- sshd\_configの AuthorizedKeysFile が設定されていない場合、公開鍵ファイルのパスは /home/username/.ssh/authorized\_keys をデフォルトとして参照する(ただし、この動作がいかなる環境でも当てはまるかは不明)

## 2018/12/16

---

### 修正点

- ファイル編集操作にて、nanoの使用について追記

## 2019/3/17

---

### 修正点

- LAN外からのSSH接続にて、DDNSの設定について追記

1. Unix系で使われているエディター ↗
2. 公開鍵と秘密鍵の組み合わせが正しいペアであると判断されると、接続が許可される認証方式 ↗



目 6



213



259



## 関連記事 Recommended by



### Raspberry PiのSSH設定方法

by tomokin966



### Raspberry Piセットアップメモ(ディスプレイレス)

by u6k



### Raspberry Pi開封から無線LAN経由でSSH接続して初期設定するまでのマ イ・ウェイ（モニタ...

by morizotter



213



259





## [wip] Raspberry Piのセキュリティを高めてみた

by yyano

### コールセンターで働く社員の環境づくりがお客さま対応の品質向上へ～最前線で働く社員

PR ソフトバンク株式会社

### 入社して2年、日本発のAI研究で世界へ。世界最高峰の国際会議「ACL」で論文採択

PR ソフトバンク株式会社

## コメント

この記事にコメントはありません。



コメントする



プレビュー

コミュニティガイドラインに基づき、良識ある内容を心がけましょう。

テキストを入力

0B / 100MB 投稿する



213



259



How developers code is here.

© 2011-2024 Qiita Inc.

ガイドとヘルプ

- About
- 利用規約
- プライバシーポリシー
- ガイドライン
- デザインガイドライン
- ご意見
- ヘルプ
- 広告掲載

コンテンツ

- リリースノート
- 公式イベント
- 公式コラム
- アドベントカレンダー
- Qiita 表彰プログラム
- API
- Findyコンテンツ

SNS

- Qiita（キータ）公式
- Qiita マイルストーン
- Qiita 人気の投稿
- Qiita（キータ）公式

Qiita 関連サービス

- Qiita Team
- Qiita Zine
- Qiita 公式ショップ

運営

- 運営会社
- 採用情報
- Qiita Blog