Windows でのポートフォワードのツール

http://www.fuji-climb.org/pf/JP/

コマンドによるポートフォワード

http://www.sb.soft.iwate-pu.ac.jp/~yushi/memo/ssh_port_forward.html
内容そのまま。

ローカルポートの転送 (-L オプション)

ローカルからリモート方向へのトンネリングを実現する。

例 1 (単純なフォワーディング)

```
% ssh -L 1234:hostA:22 user@hostA
% ssh -L 1234:localhost:22 user@hostA
```

ローカルホストの 1234 ポートにアクセスすると hostA のポート 22 にアクセスできる。

例 2 (ファイアーウォールの外から内部へアクセス可能)

```
% ssh -L 1234:hostB:22 user@hostA
```

ローカルホストの 1234 ポートにアクセスすると、hostA 経由で hostB のポート 22 にアクセスできる。
直接は到達不可能なホストにもアクセス可能

例 3 (ローカルポートのサービスを別ポートに転送)

```
% ssh -L 1234:localhost:22 user@localhost
```

ローカルのポート 22 で行われているサービスを 1234 からもアクセス可能にする

リモートポートの転送 (-R オプション)

リモートホストからローカル方向へのトンネリングを実現する。
例 1 ～ 3 までの逆方向を行えるだけで、基本的な機能は同じ。

例 4

```
% ssh -R 1234:localhost:22 user@hostA
```

hostA のポート 1234 にアクセスするとローカルホストのポート 22 にアクセスできる。
ローカルホストがファイアーウォール内、hostA がグローバルの場合便利。

例 5

```
% ssh -R 1234:hostB:22 user@hostA
```

hostA のポート 1234 にアクセスすると、ローカルホスト経由で hostB のポート 22 にアクセスできる。

例 6

```
% ssh -R 1234:localhost:22 user@localhost
```

結果は例 3 と全く同じ。

ダイナミック転送

ダイナミック転送

ポートフォーワードのセキュリティ

ssh サーバを 192.168.0.1
ssh の接続元を 192.168.0.2
ssh の転送先を 192.168.0.3

とする。

■ローカル転送

```
192.168.0.2 上で
ssh -L 9999:192.168.0.3:9999 192.168.0.1
```

とポートフォワードした場合、

```
ssh_config で GatewayPorts no または、 -g オプションなし
の場合は、
192.168.0.2 上で localhost:9999 のみポートフォワードを受け付ける。
192.168.0.2:9999 の記述は内部、外部問わずポートフォワードを受け付けない。
```

ssh_config で GatewayPorts yes または、 -g オプションをつけると localhost 以外からの接続もポートフォワード可能になる。

■リモート転送

```
192.168.0.2 上で
192.168.0.2# ssh -R 9999:192.168.0.3:9999 192.168.0.1
```

とポートフォワードした場合、

```
192.168.0.1 上で localhost:9999 のみのポートフォワードを受け付ける。
192.168.0.1:9999 の記述は内部、外部問わずポートフォワードを受け付けない。
```

sshd_config で GatewayPorts yes とすると、外部からの接続もポートフォワード可能になる。

ポートフォワーディングを行う際に知っておきたいこと

```
暗号化される経路
暗号化が行われるのはローカルホスト～リモートログイン先のホストまでである。
例 2 の場合、ローカルホスト～ hostA 間は暗号化されるが、hostA ～ hostB 間は暗号化されない。
```

```
ユーザーの権限と利用できるポート
ローカルホストの特権ポートで listen する際には ssh コマンドの実行権限が root と同等でなければならない。
リモートホストの特権ポートで listen する際にはリモートログイン時のユーザー権限が root と同等でなければ
ならない。
```

```
ポートフォワーディングと組み合わせると便利なオプション
```

■よく使うオプションの組合せ例

```
% ssh -C -N -f -L 1234:hostA:22 user@hostA
% ssh -4 -N -f -L 1234:hostA:22 user@hostA
```

このようなオプションで起動すると、ローカルホストの 1234 が hostA のポート 22 につながるようにポートフォワーディングが行われる。
加えて、-C オプションによる圧縮、-f オプションによるバックグラウンドでの動作、-N オプションによるリモートでのコマンド実行無し（通常はシェルが起動してしまう）という指定が可能。

簡単にまとめると、ポートフォワーディングのみを実現するプロセスが作れます。(しかもデータ圧縮をした転送)

■よく使うオプションの簡単な説明

セッションの圧縮 (-C オプション)

SSH を用いた通信経路において、データを圧縮した通信を行う。
設定ファイルに「圧縮レベル」や、「オプションなしでも常時圧縮」などの設定が可能。

リモートでコマンドを実行しない (-N オプション)

リモートでのコマンド実行が行われないため、ポートフォワードのみを行いたい時に有効。
特に指定しない場合、SSH でのコネクションはリモートでシェルが起動する。

バックグラウンドで実行 (-f オプション)

実行するとそれ移行バックグラウンドのプロセスとなる。
停止させるときはプロセスを探して普通に kill すればいい。

プロトコルの指定 (-1 , -2 オプション)

プロトコルのバージョンを指定することができる。
数字がそのままバージョンを表している。

IP バージョンの指定 (-4 , -6 オプション)

IP バージョンを指定することができる。
数字がそのままバージョンを表している。

■多段ポートフォワード

ポートフォワード先からさらにポートフォワードする場合、

ssh コマンド

を使って

```
ssh -t -L 1000:localhost:1000 hostA ssh -L 1000:hostC:1000 hostB
```

のように指定できる。 ssh コマンド はターミナルの割当を行わないため、-t オプションで強制的にターミナルを割り当てる必要がある。

■多段ポートフォワード (ProxyCommand)

https://www.xmisao.com/2013/10/08/ssh-proxy-command.html

```
ssh -oProxyCommand='ssh -W %h:%p X' A
```

または、.ssh/config に

```
Host server
 HostName 1.2.3.4

Host server-anotherserver
 HostName 5.6.7.8
 Port 22
 ProxyCommand ssh -W %h:%p server
```

と書いて、

```
ssh server-anotherserver
```

で多段ポートフォワードできる

config ファイルによるポートフォワード

ローカルフォワード

```
LocalForward PortA Address2:PortB
```

指定方法はコマンドの場合と同じ