

# ホストマシンで起動しているDockerコンテナにクライアントのVSCodeから接続

---

part1ではホストマシンにDockerを用いて仮想環境を用意する手順を説明した。ここでは自分のPCなどのVisual Studio Codeから、リモートでDockerコンテナの起動しているホストマシンにアクセスして操作するための設定について解説する。

## 前提

- クライアントからホストへのssh接続はできるようにしておく
- ホスト・クライアントともにVSCode、Dockerがインストールされており、どちらもローカルで利用できる状態にある

ssh接続については調べたら参考になるサイトがたくさん出てくるはずなのでそれを参考に設定しよう。WindowsにDockerをインストールする際は注意する点がいくつかあるので、以下のサイトあたりを参考に。

- [WindowsでDocker環境を試してみる](#)
- [Docker for Windowsをインストール](#)

---

## ①VSCodeに拡張機能のインストール

以下の拡張機能をホスト・リモートともにインストール(正確には片方だけで良いものもある)

- [Docker](#)
- [Remote SSH](#)
- [Remote Container](#)

---

## ②VSCodeの設定

- クライアント側のVSCodeの設定

VSCodeのsettings.jsonに以下の設定を必要な箇所を訂正して追加

```
{
  "docker.host": "docker.host":"ssh://[リモートでのユーザ名]@[リモートマシンのアドレス]"
  // "docker.host": "docker.host":"ssh://tanaka@192.168.10.10"    example
}
```

VSCodeでsshのconfigを開き、ホストへの接続設定の部分にLocalForwardの項目を追記

```
# 例えばもともと以下のような設定だった場合
Host hostexample1
  HostName 192.168.10.10
  IdentityFile C:/Users/yourname/.ssh/id_rsa
```

```
User Tanaka
Port 22
```

```
# LocalForward 23750 /var/run/docker.sockを追記
Host hostexample1
  HostName 192.168.10.10
  IdentityFile C:/Users/yourname/.ssh/id_rsa
  User Tanaka
  Port 22
  LocalForward 23750 /var/run/docker.sock
```

- ホスト側のVSCodeの設定

VSCodeのsettings.jsonに以下を追記

```
{
  "docker.host": "tcp://localhost:23750"
}
```

[参考](#)

---

### ③VSCodeから接続

1. クライアント上でVSCodeを起動し左下の >< マークからホストマシンにssh接続
2. **ssh接続したVSCodeのウィンドウとは別のウィンドウで**左のリモートエクスプローラーから接続した  
いコンテナを選択 -> 右クリックで **Open Folder in Container**

ssh接続したウィンドウは使わないものの、**開きっぱなしでないとコンテナとの接続が切れる**。これについては現状原因不明。最小化でもしておく。一応多段sshを用いることで解決するらしいが、そもそもコンテナ側にopenssh-serverの設定がされてない場合が多いため、Dockerイメージのビルドの段階からやり直す必要がある上にこの設定がかなり面倒。

---