

3) gitlabとの接続設定

<https://qiita.com/ksj555/items/688e8219677d53ef8aa1>

① gitlabへアクセスして新規プロジェクトを作成する

i) Dockerコンテナのgitlabサービスを起動して、プロセスを確認する。

```
[root@RHEL75 aml_ai_root]# docker-compose ps
```

Name	Command	State	Ports
aml_ai_gitlab	/assets/wrapper	Up (healthy)	0.0.0.0:3022->22/tcp, 0.0.0.0:443->443/tcp, 0.0.0.0:8081->80/tcp, 0.0.0.0:8001->8001/tcp

ii) 上記赤文字のPortとgitlabが起動している端末のIPアドレスを参考に、ブラウザにURLを入力する。

例)

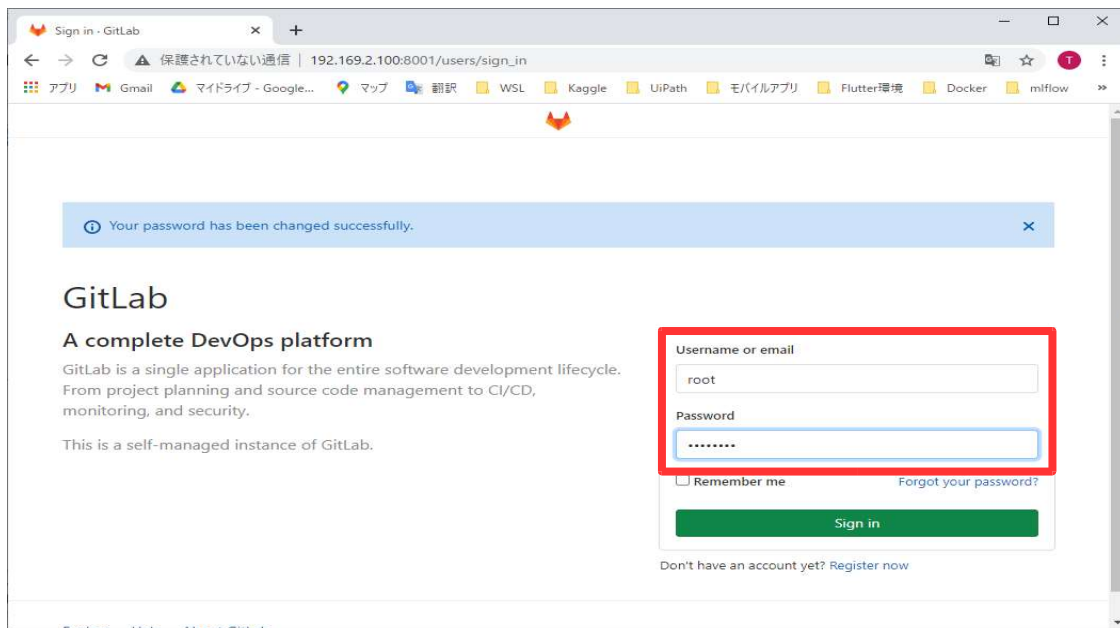
[Http://192.169.2.100:8001/](http://192.169.2.100:8001/)

rootユーザーのパスワードを設定する。

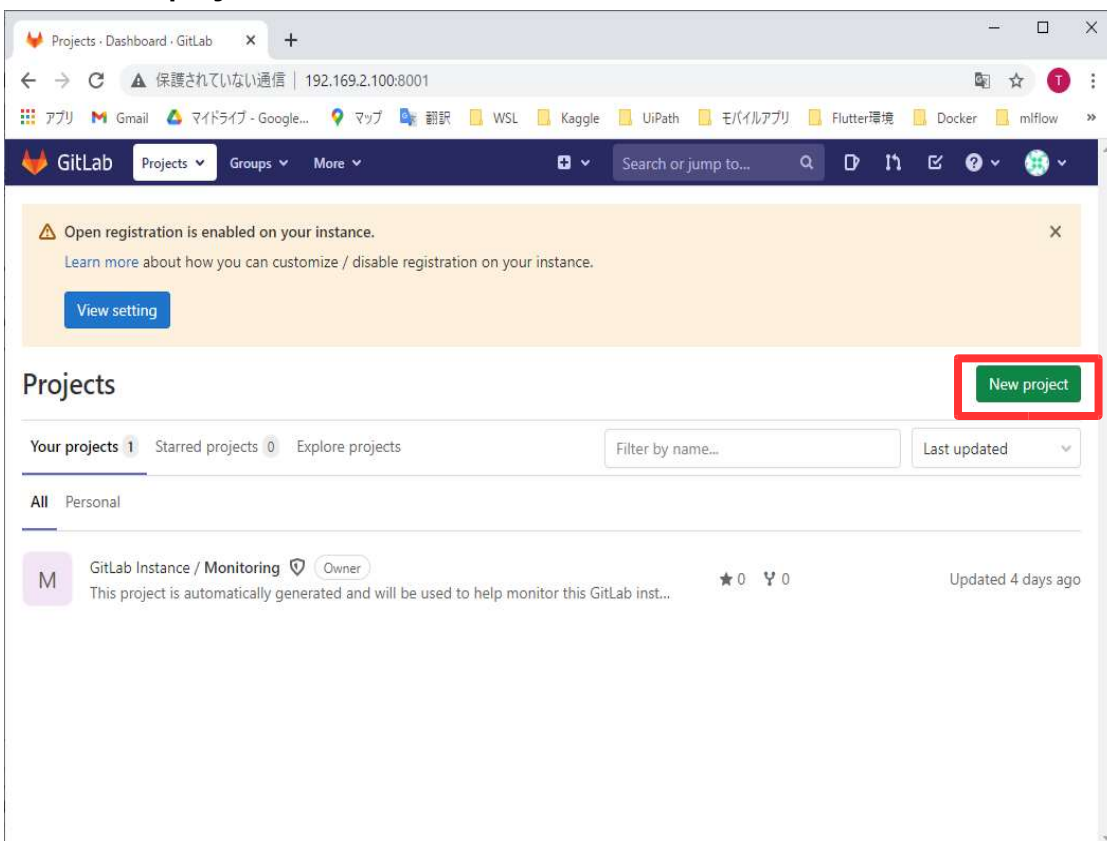
The screenshot shows a web browser window with the GitLab login page. The address bar shows the URL: 192.169.2.100:8001/users/password/edit?reset_password_token=-yWDZkrqGCSLdJFvvpk. A blue notification bar at the top says "Please create a password for your new account." The main heading is "GitLab" with the subtitle "A complete DevOps platform". Below this is a description of GitLab as a single application for the entire software development lifecycle. On the right side, there is a "Change your password" form. This form has two input fields: "New password" and "Confirm new password", both of which are highlighted with a red rectangular box. Below these fields is a "Change your password" button. At the bottom of the page, there are links for "Didn't receive a confirmation email? Request a new one" and "Already have login and password? Sign in".

1.3) gitlab

iii) rootユーザーで、先ほど設定したPWでgitlabにログインする。



iv) 新規プロジェクトの作成する。
New projectをクリック



1.3) gitlab

v) 各項目を入力し、「Create project」ボタンを押す。

Project name : aml_ai を入力
Visibility Level : Private を選択 (デフォルト)

New Project - GitLab

Open registration is enabled on your instance. Learn more about how you can customize / disable registration on your instance. View setting

New project

A project is where you house your files (repository), plan your work (issues), and publish your documentation (wiki), among other things.

All features are enabled for blank projects, from templates, or when importing, but you can disable them afterward in the project settings.

Information about additional Pages templates and how to install them can be found in our Pages getting started guide.

Tip: You can also create a project from the command line. Show command

Blank project Create from template Import project

Project name
aml_ai

Project URL
http://192.169.2.100:8001

Project slug
aml_ai

Want to house several dependent projects under the same namespace? Create a group.

Project description (optional)
Description format

Visibility Level
Private

下にボタンがある

vi) SSH鍵の設定を要求されるが、プロジェクトは作成完了。

Details - Administrator / aml_ai

Open registration is enabled on your instance. Learn more about how you can customize / disable registration on your instance. View setting

You won't be able to pull or push project code via SSH until you add an SSH key to your profile. Add SSH key Don't show again

Administrator > aml_ai > Details

Project 'aml_ai' was successfully created.

aml_ai
Project ID: 2

The repository for this project is empty

You can get started by cloning the repository or start adding files to it with one of the following options.

Clone New file Add README Add LICENSE Add CHANGELOG Add CONTRIBUTING

1.3) gitlab

②Git (クライアント) のインストール

【参考】Visual Studio Code (VS Code) とGitLabを連携させる

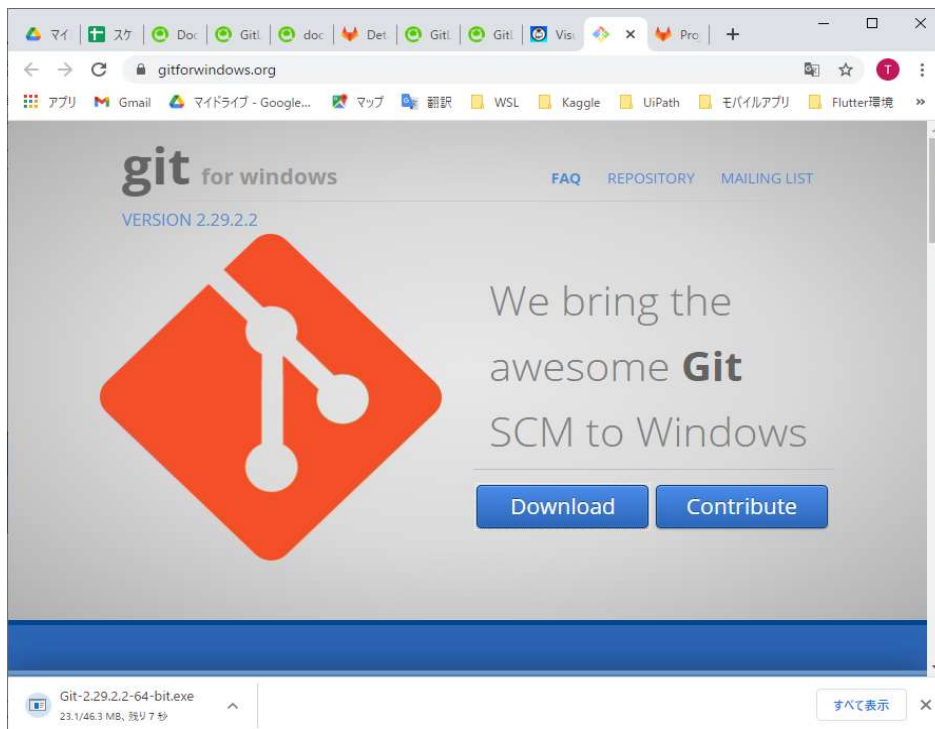
<https://create-fecundity.com/programming/visual-studio-code-gitlab-introduction/#toc1>

【参考】私家版 Git For Windowsのインストール手順

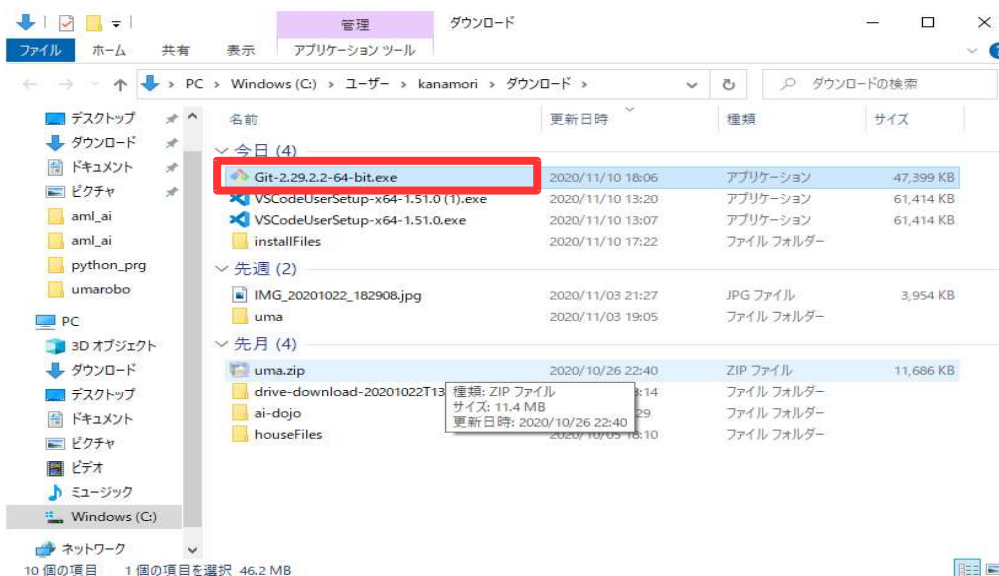
<https://opcdiary.net/technical/programming/%E7%A7%81%E5%AE%B6%E7%89%88->

i) Git for Windowsのページから、Gitのインストーラーをダウンロードする。

git for windows → <https://gitforwindows.org/>



ii) ダウンロードが完了したら、インストーラーをダブルクリックする。

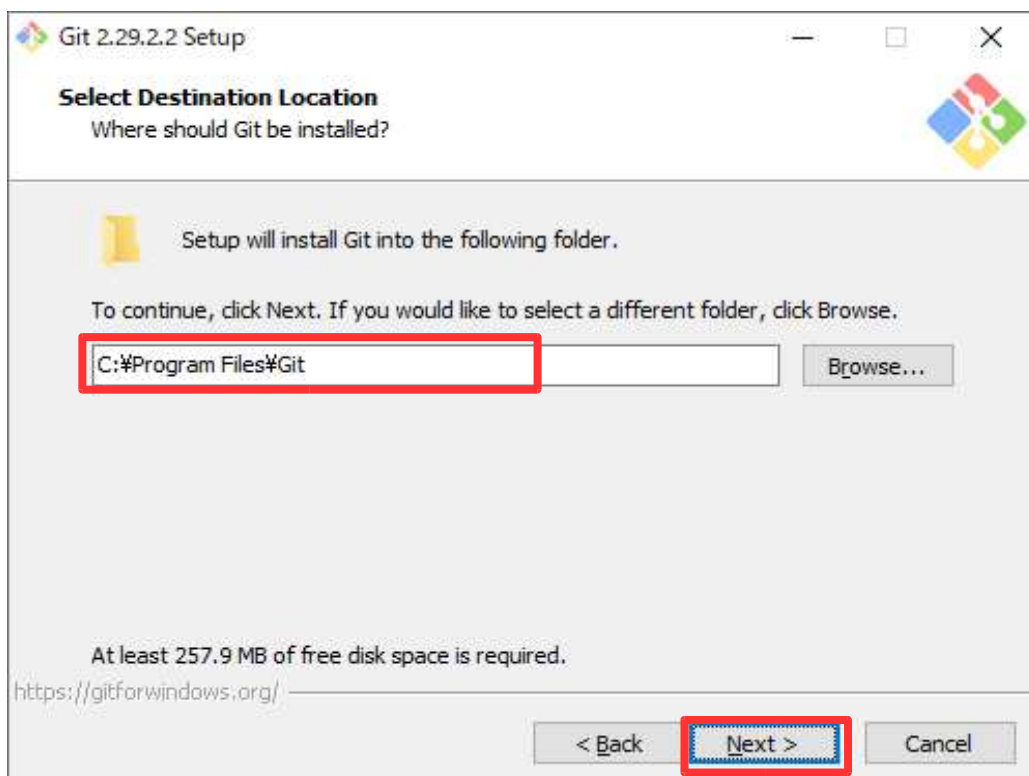


iii) Nextを押す。

1.3) gitlab

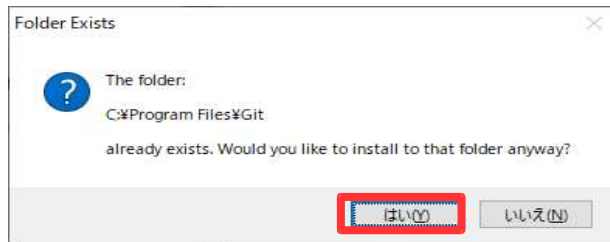


iv) 任意のインストール先を選択して、Nextを押す。

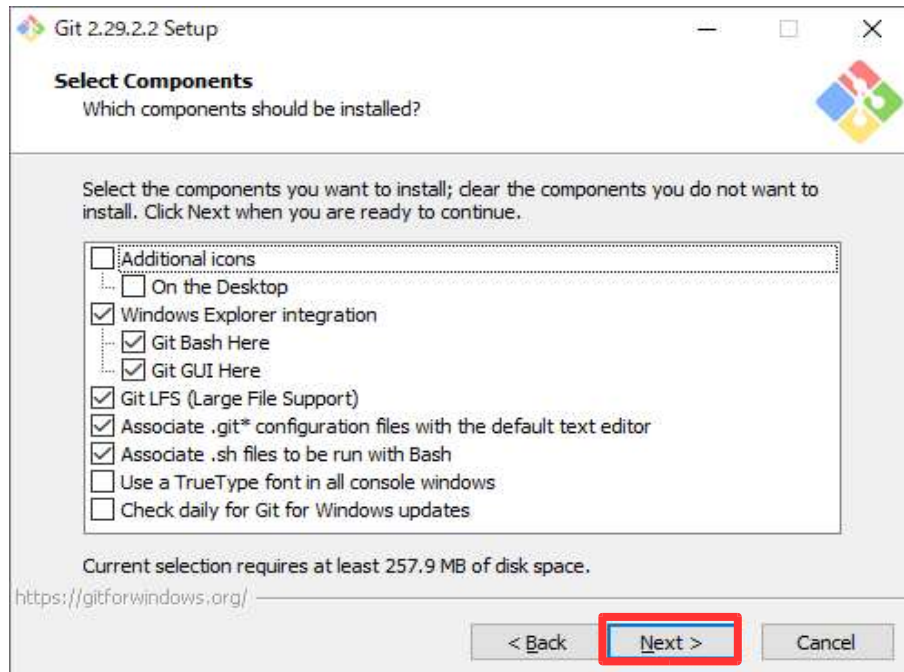


1.3) gitlab

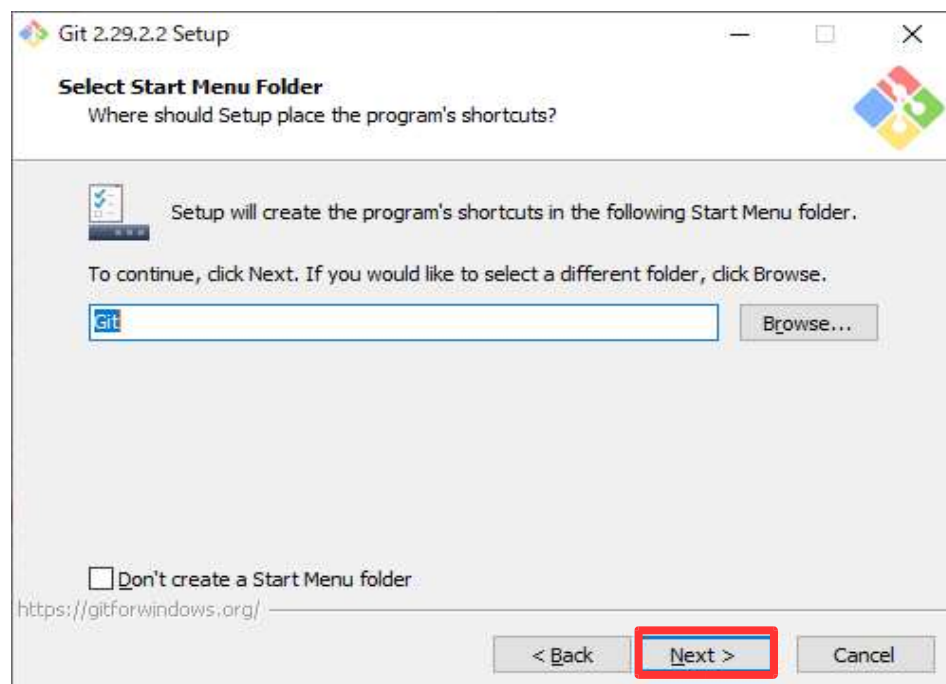
v) 「はい」を押す。



vi) インストールするコンポーネントを選択して、「Next」を押す。(ここではデフォルトのまま)



vii) スタートメニューを追加されるか確認されるので、通常はこのままNextをクリックして先に進む。



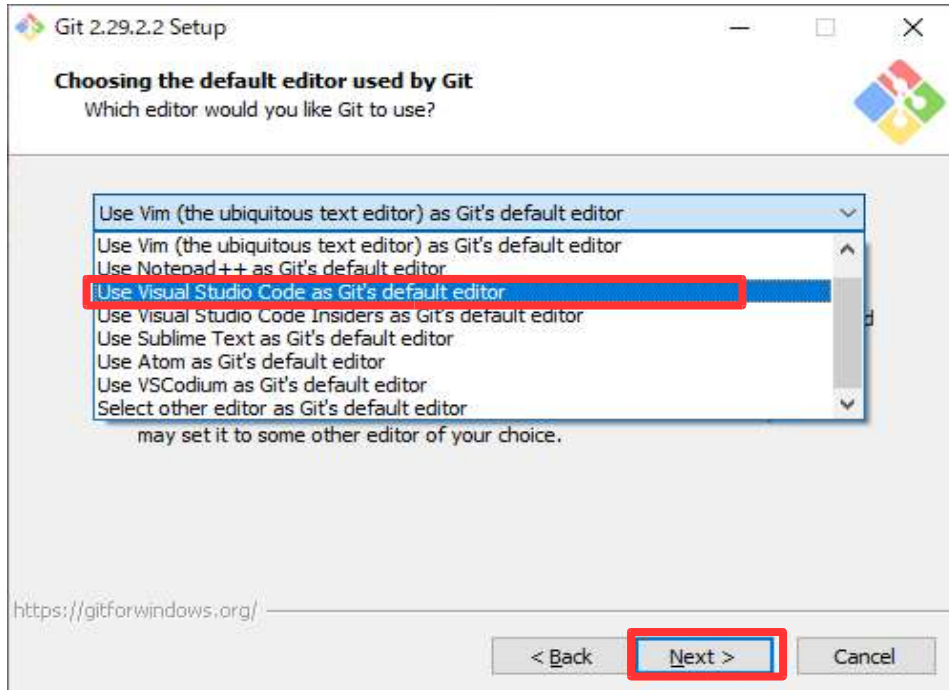
1.3) gitlab

viii) Gitで使用するデフォルトのエディターを選択できる。

『Use Visual Studio Code as Git's Default editor』を選択する。

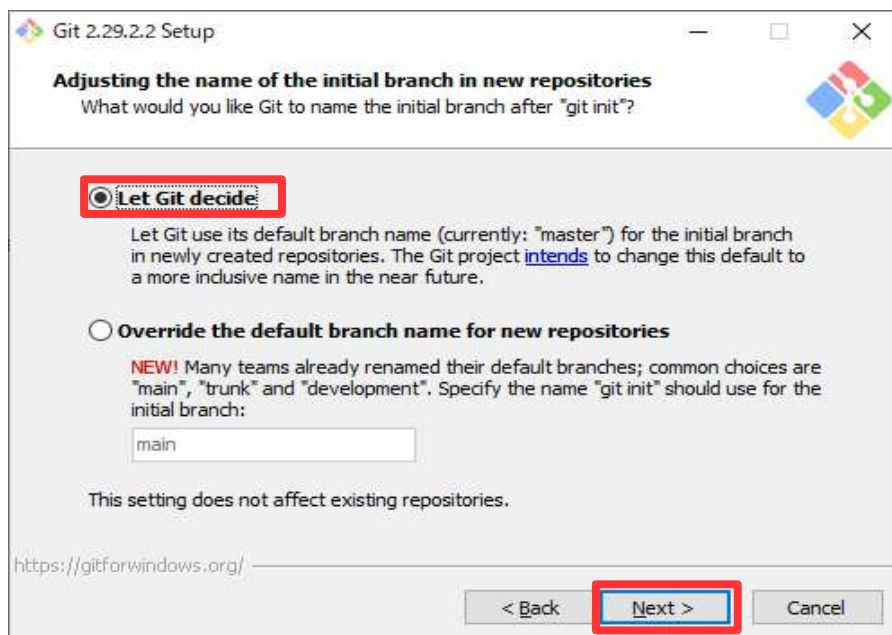
Gitをコマンドプロンプトから使用できるようにする。

こうしておくと、VS Code内で「Ctrl + @」のショートカットキーでコマンドプロンプトとソースコードの切り替えることで、すぐにGitコマンドを打つことができる。



ix) Git initで最初に作成されるブランチ名を選択

- Let Git decide : 従来通り最初に作られるブランチは「master」となります。
- Override the default branch name for new repositories : 下のテキストボックスに指定したブランチ名がgit initコマンドで最初に作られるブランチ名となります。
GitHubの仕様変更にあわせて最初のブランチを「main」にしたい場合にはここで設定しておく。
デフォルト(Let Git Decide)で『Next >』をクリックする。

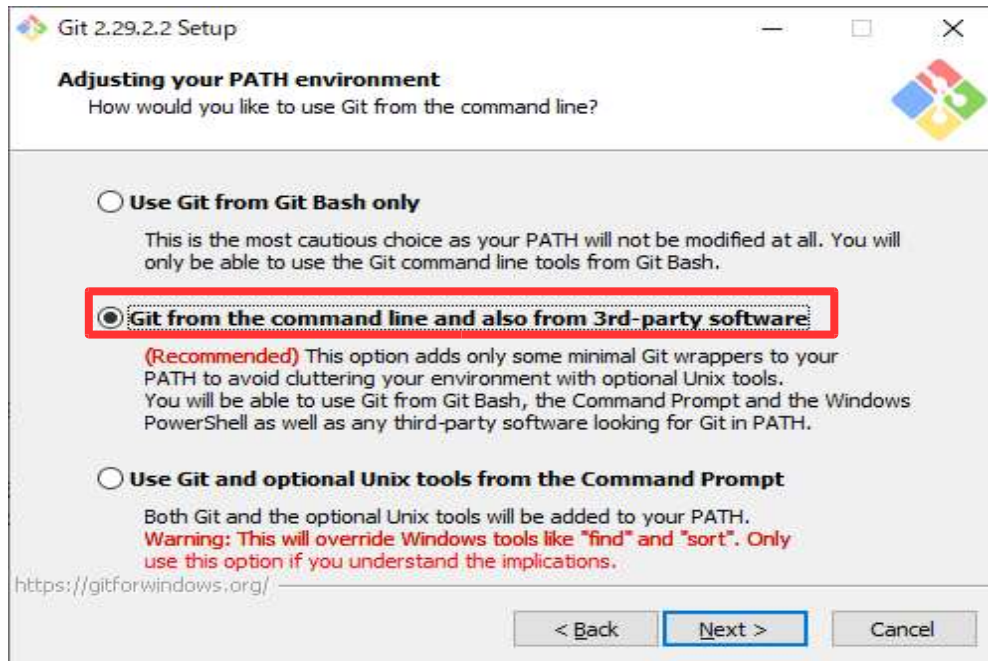


1.3) gitlab

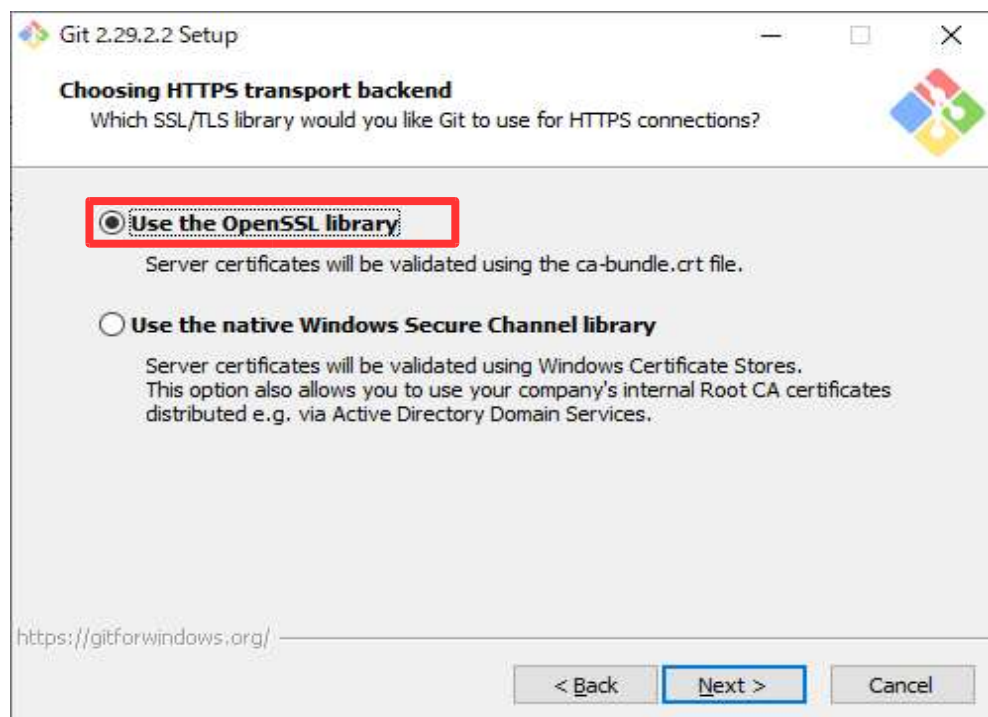
x) ここは環境変数pathに何を追加するかを選択します。

デフォルトの選択項目で次に進む。

コマンドプロンプトやPowerShellからはGitのコマンドだけが使用できます。(これがデフォルトの設定)

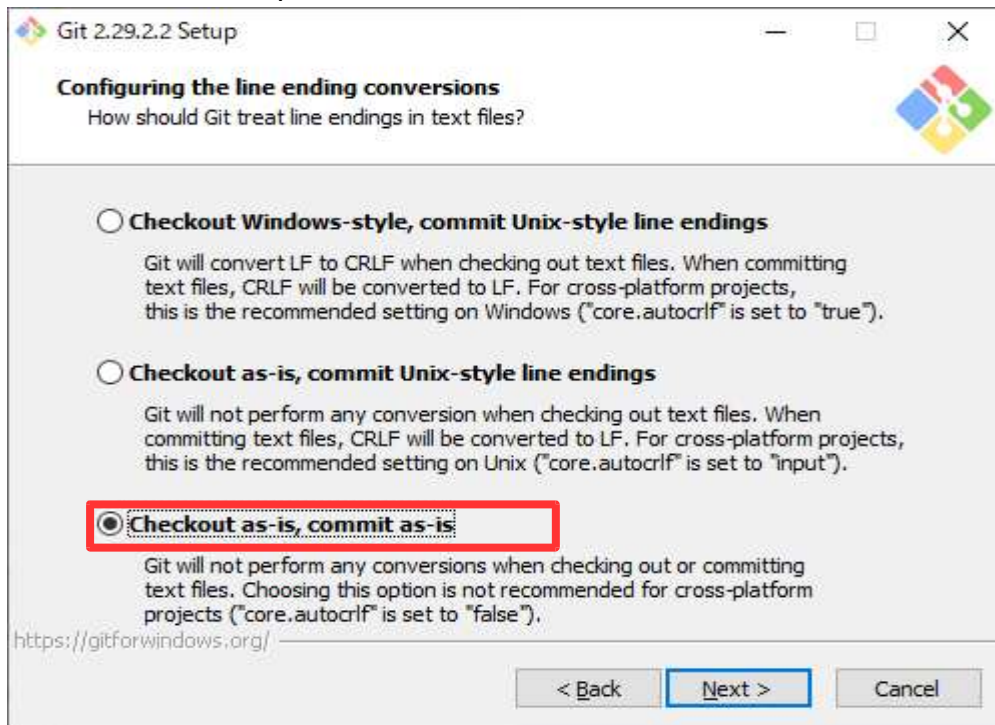


xi) 「Use the OpenSSL library」を選択する。

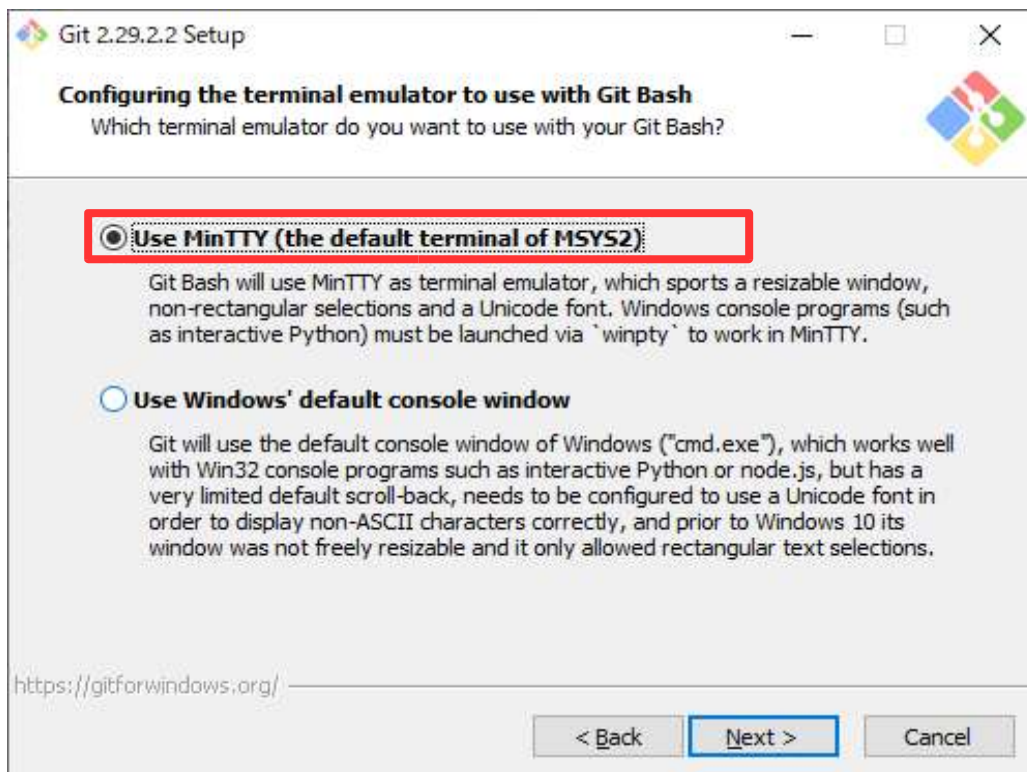


1.3) gitlab

xii) 「Checkout as-is, commit as-is」を選択する。



x iii) 「Use Min TTY(the default terminal of Msys2)」を選択する。



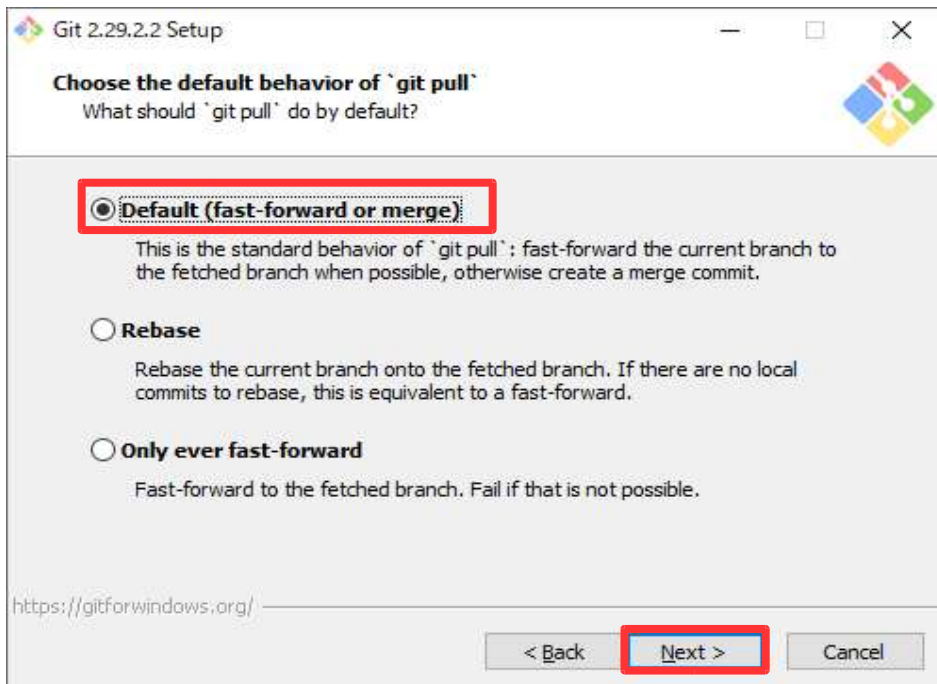
1.3) gitlab

x iv) 「Default(fast-foward or merge)」を選択する。

git pullの方法について設定します。通常はDefaultのままでかまいません。

Rebaseを標準とした場合にはRebaseをfast-forwardのみをデフォルトにしたい場合には

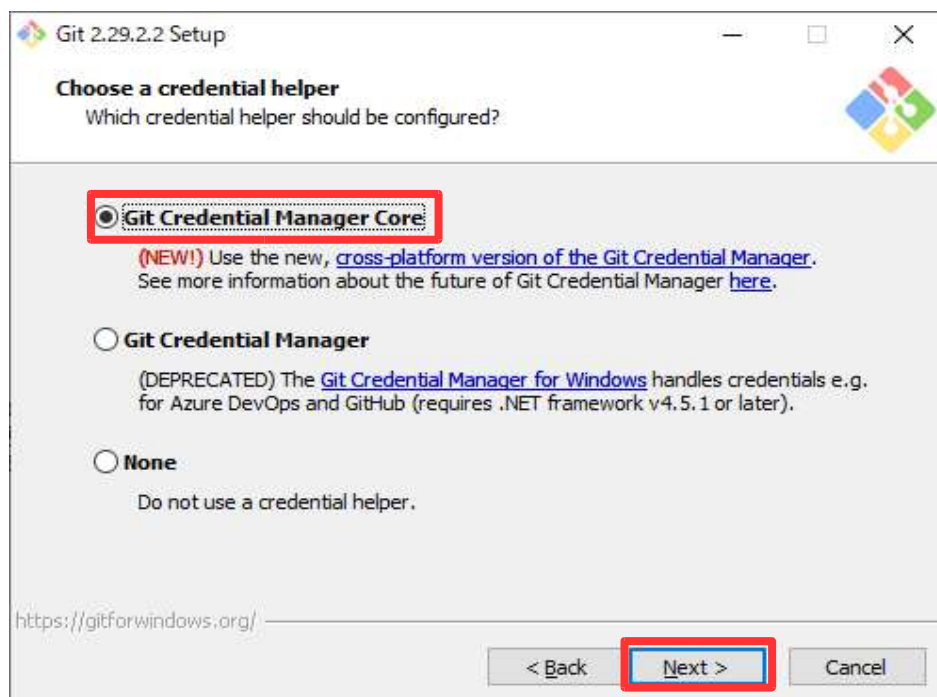
Only ever fast-forwardを選択します。



x v) 「Git Credential Managre Core)」を選択する。

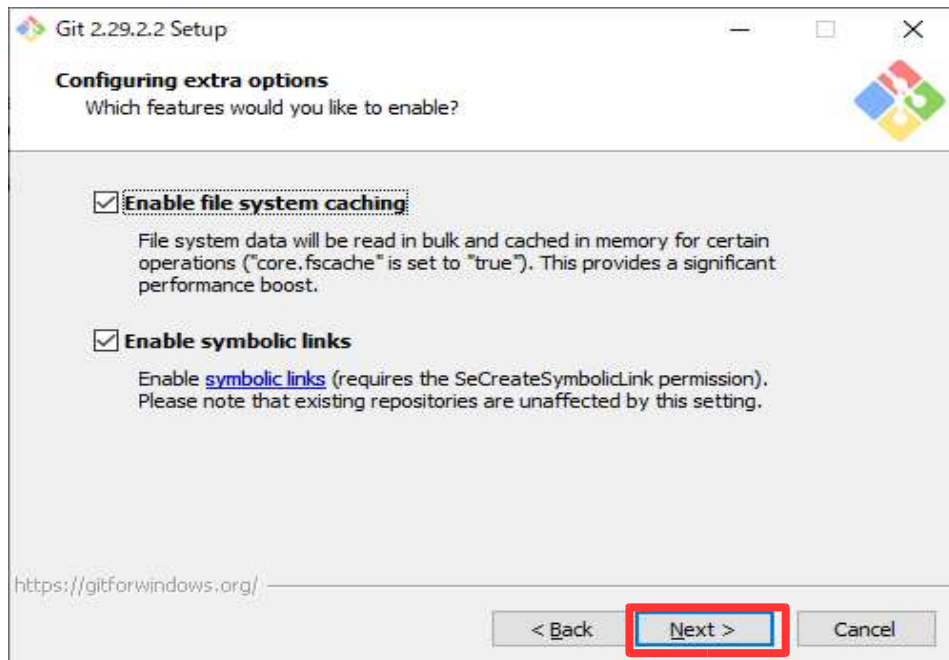
Git credential managerのセットアップを選択します。

通常は更新される予定のGitHubへのサインインに対応しているGit Credential Manager Coreを選択します。何らかの理由で旧バージョン(サポート終了)のGit Credential Managerが必要な人はそちらを選択します。権限管理機能を何も必要としない場合にはNoneを選択します。



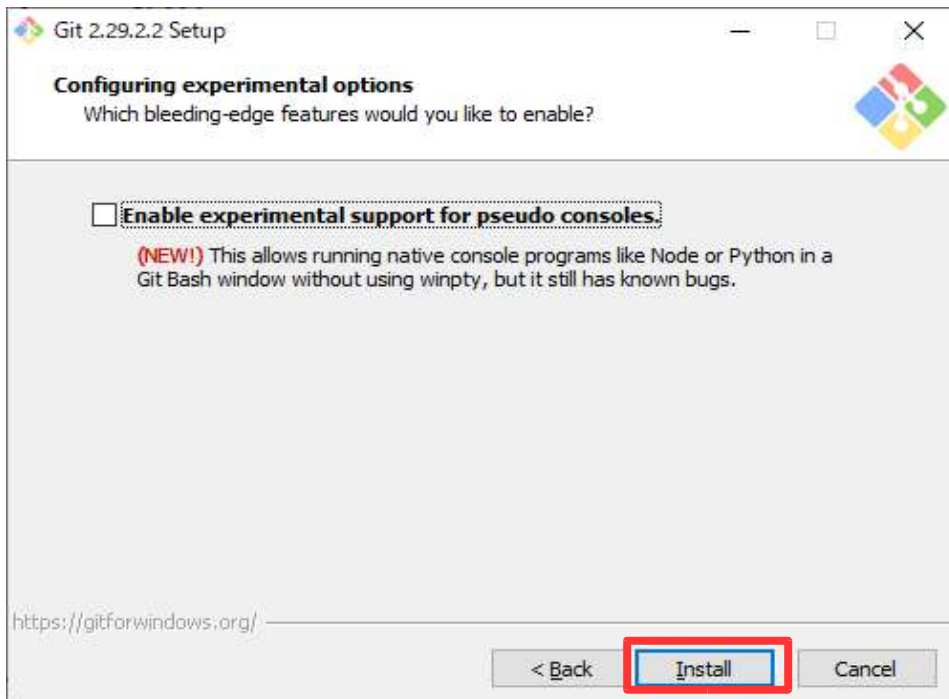
1.3) gitlab

x vi) デフォルトで「Next >」をクリックする。

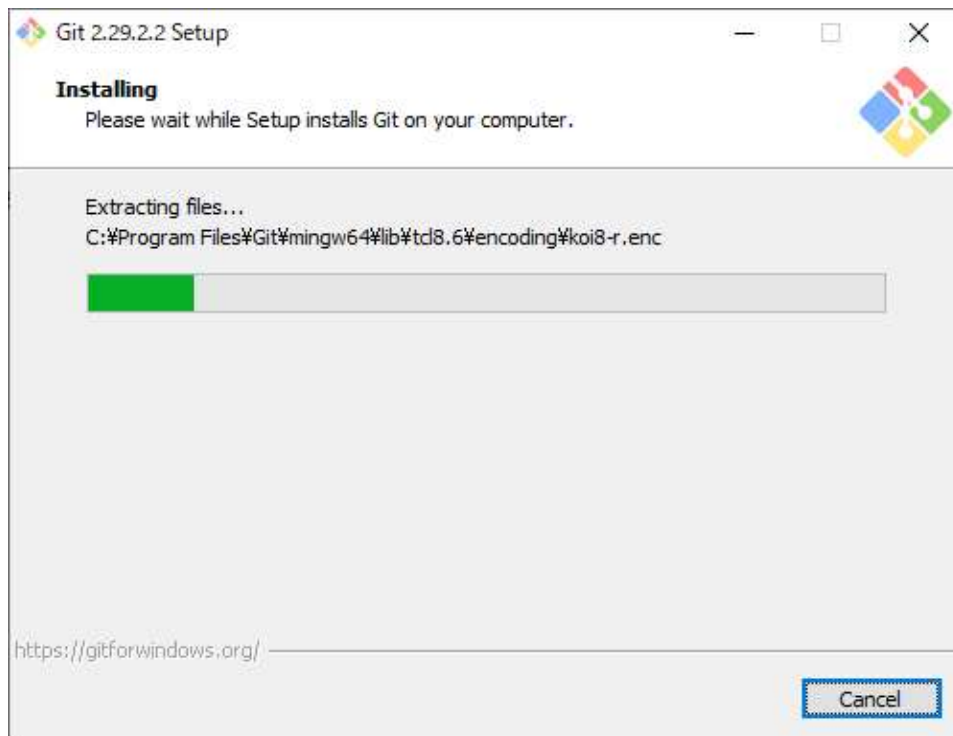


x vii) デフォルトで「Install」をクリックする。

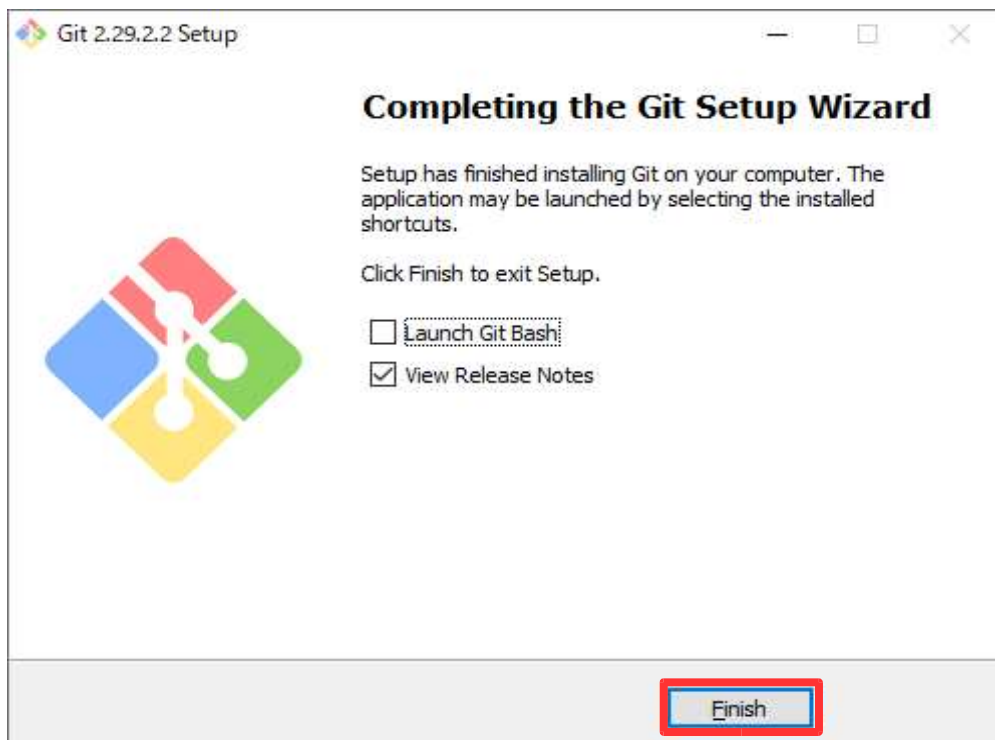
Installをクリックするとここでインストーラーがインストールを開始します。



1.3) gitlab



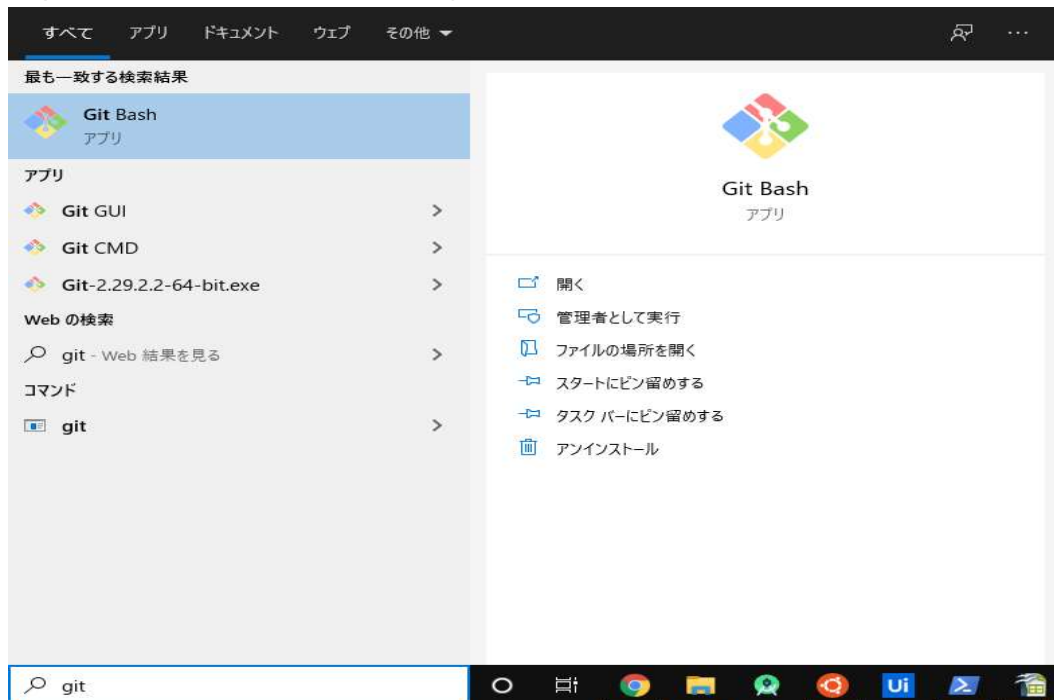
finishi!!



1.3) gitlab

③Git グローバルの設定

i) Windowsマーク隣でGitBashを検索し、Enterを押してGit bashを起動します。



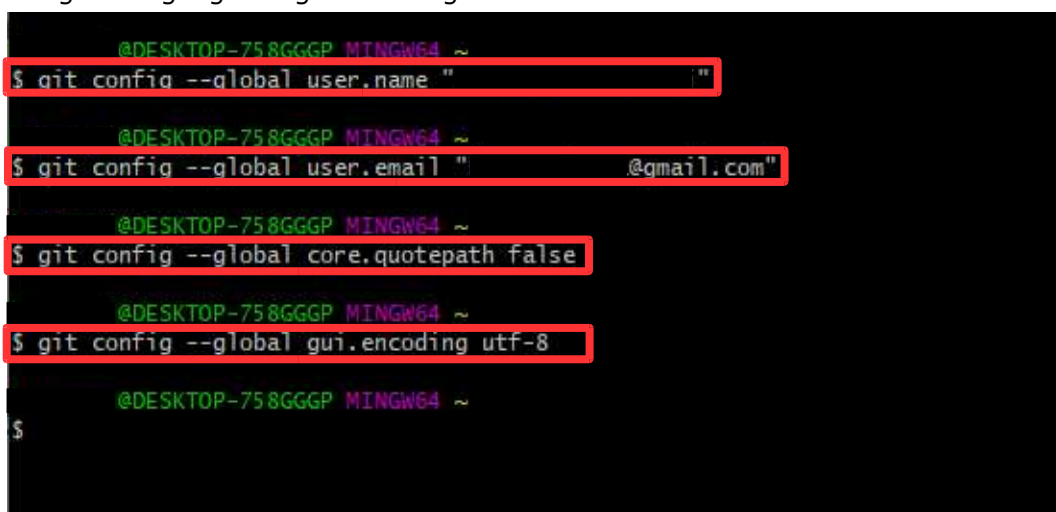
user.nameにはご自身の姓名をローマ字で、
user.emailには普段使用するメールアドレスを使用します。

```
$ git config --global user.name "Mei Sei"  
$ git config --global user.email "hoge@email.com"  
$ git config --global core.quotepath false
```

上の内容が、環境変数homeで設定されたディレクトリの.gitconfigファイルに保存されます。

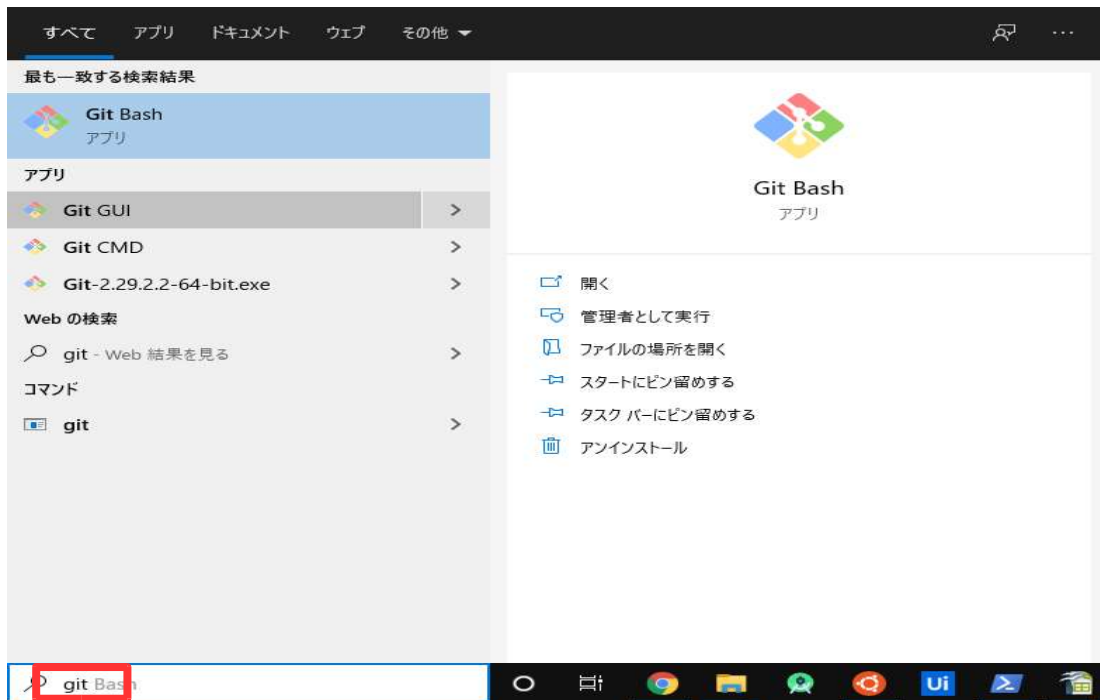
Git GUIで文字化け等あるようでしたら以下の設定も追加してください。

```
git config --global gui.encoding utf-8
```

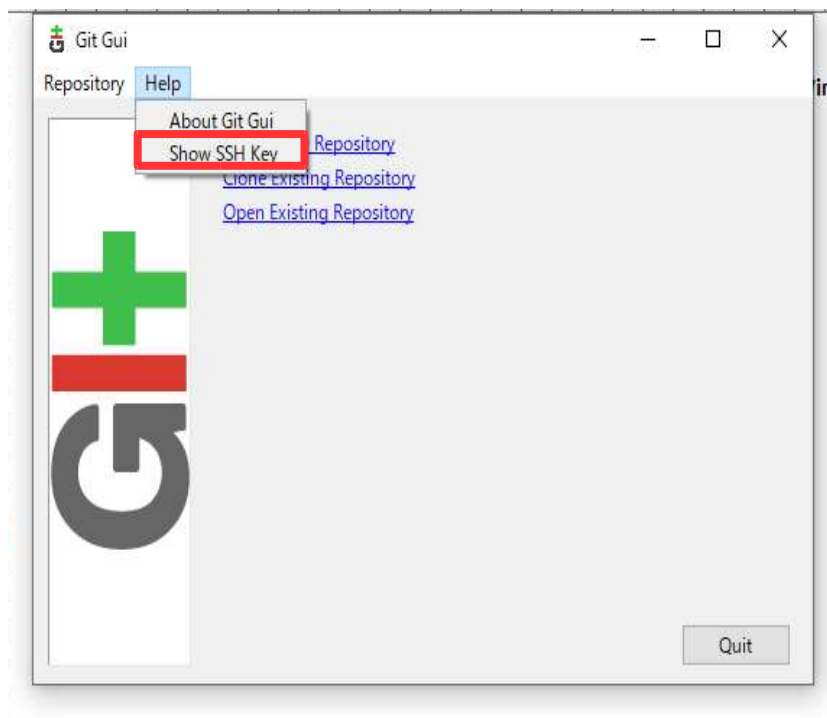


④SSHキーの作成

Windowsマーク隣でGit GUIを検索し、Enterを押してGit GUIを起動します。

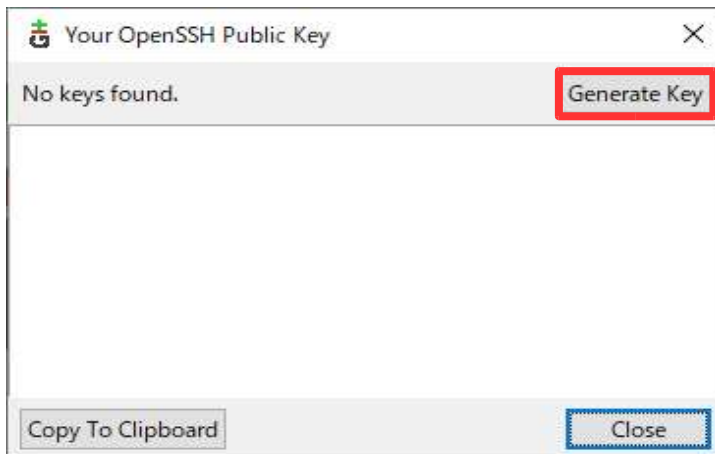


Helpメニューの『Show SSH Key』を選択します。

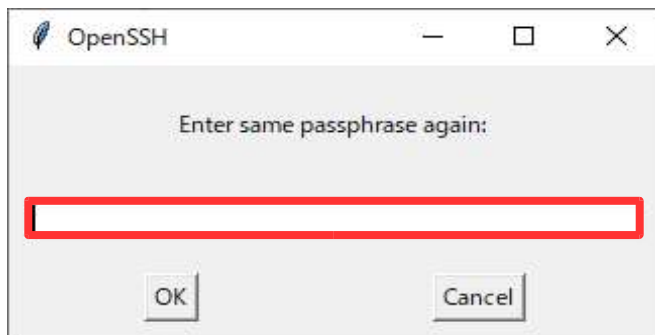


1.3) gitlab

Generate Keyをクリックします。

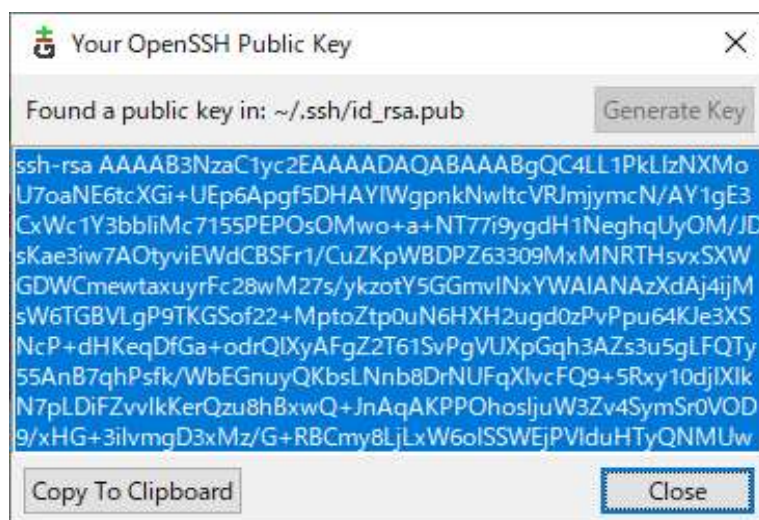


passprase(password)の入力が促されるので、入力します。
再度聞かれるので、同じpassprase(password)を入力します。
→今回はパスフレーズをblankで「Cancel」を押してしまった。。。



なにやら、公開鍵は既に作成されている様子。以下、GitBashで表示された情報。

```
iguchi@DESKTOP-758GGGP MINGW64 ~/.ssh
$ pwd
/c/Users/xxxxx/.ssh
iguchi@DESKTOP-758GGGP MINGW64 ~/.ssh
$ cat id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQGC4LL1PkLzNXMoU7oaNE6tcXGi+UEp6ApGf5I
```



1.3) gitlab

⑤ OpenSSHのインストール

<https://create-fecundity.com/programming/visual-studio-code-gitlab-introduction/#toc1>

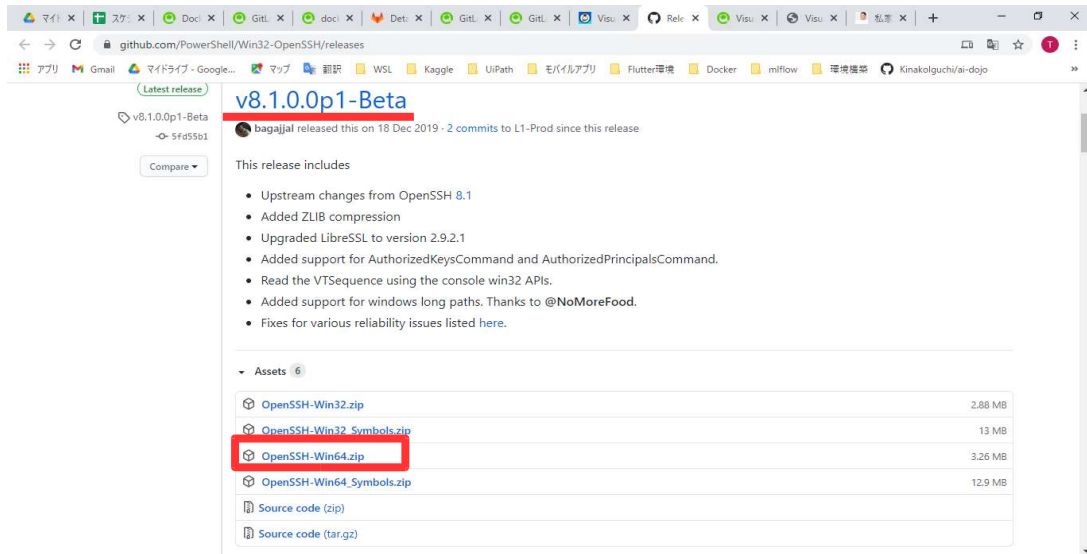
i) OpenSSHのzipファイルをダウンロード

Win32-OpenSSHのダウンロードページからZipファイルをダウンロードする。

<https://github.com/PowerShell/Win32-OpenSSH/releases>

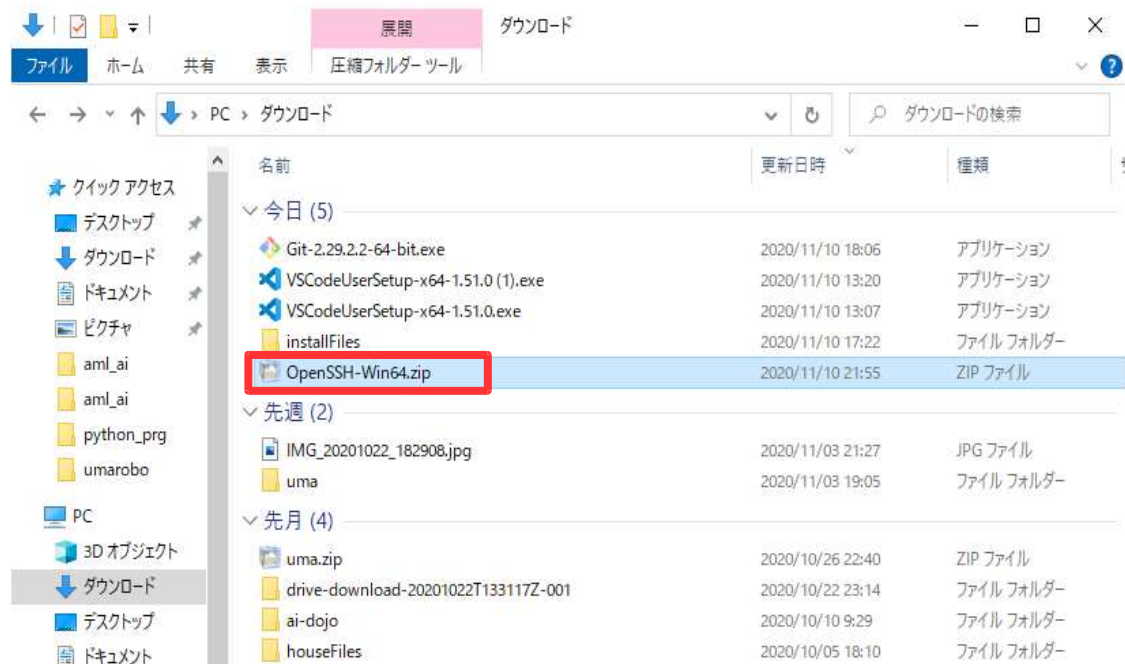
v8.1.0.0p1-Beta

OpenSSH-Win64.zip をダウンロード



ii) OpenSSHのzipファイルをインストール

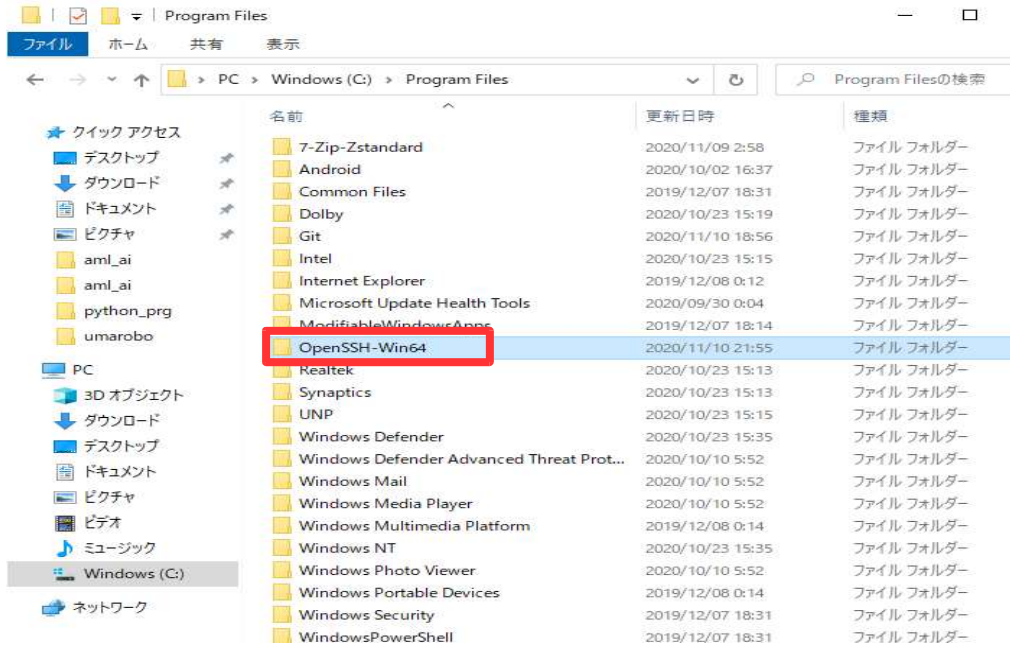
ダウンロードフォルダの「OpenSSH-Win64.zip」をダブルクリック。



1.3) gitlab

OpenSSH-Win64.zip を展開して以下のフォルダー階層で配置します。

C:¥Program Files¥OpenSSH-Win64



iii) 環境変数の設定

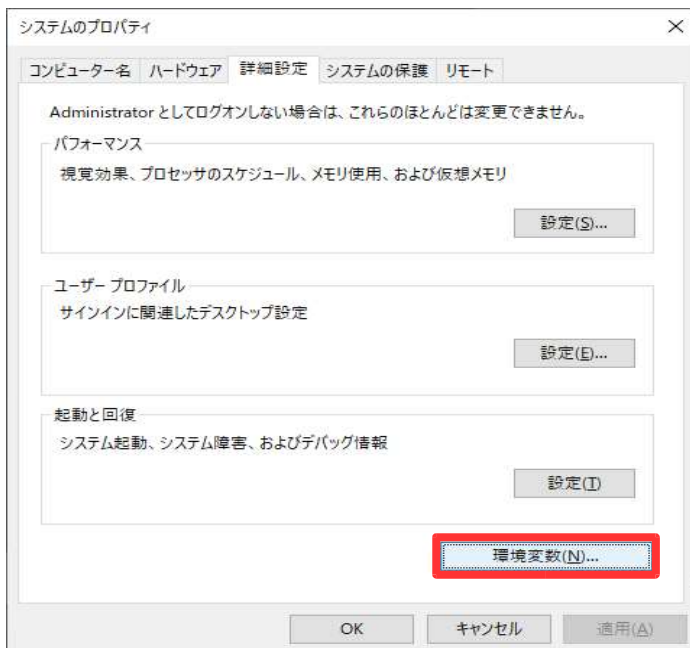
Windowsのシステムの環境変数へ以下の2つの値を設定します。

変数	値
① PATH	C:¥Program Files¥OpenSSH-Win64
② GIT_SSH	C:¥Program Files¥OpenSSH-Win64¥ssh.exe

※PATHは既存の値に追記してください

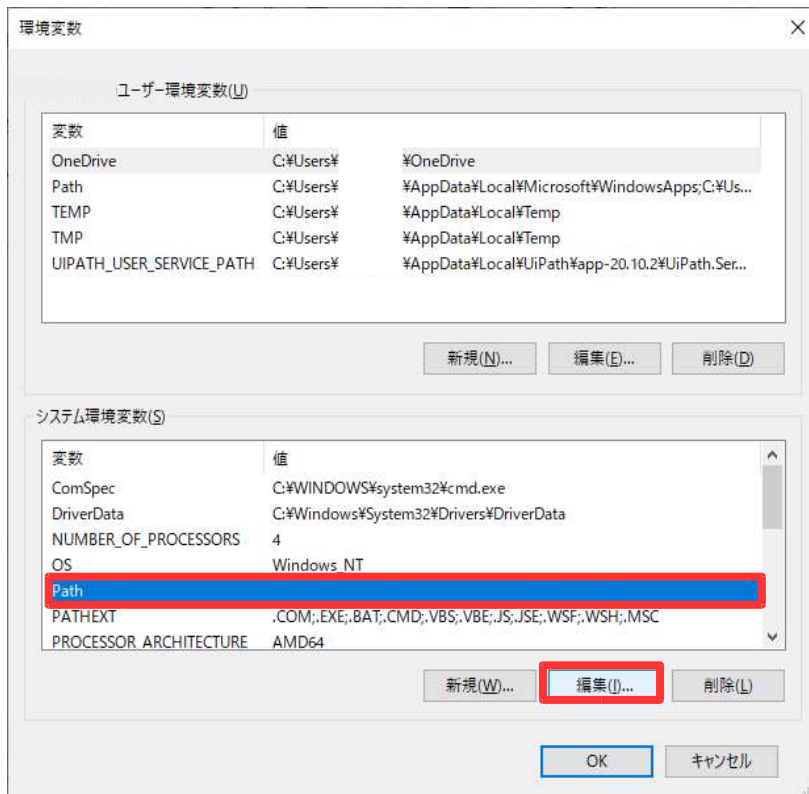
コントロールパネルを開き、

「システムとセキュリティ」→「システム」→「システムの詳細設定」→「環境変数」を選択する。

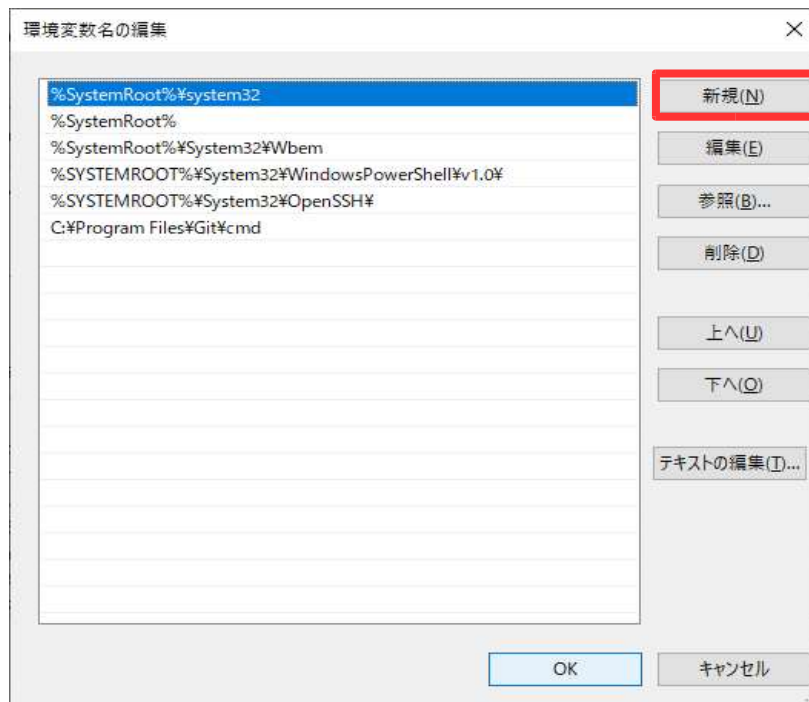


1.3) gitlab

システム環境変数に1つ目の変数「Path」を追加する。
システムの環境変数のPathを選択し、編集をクリック。

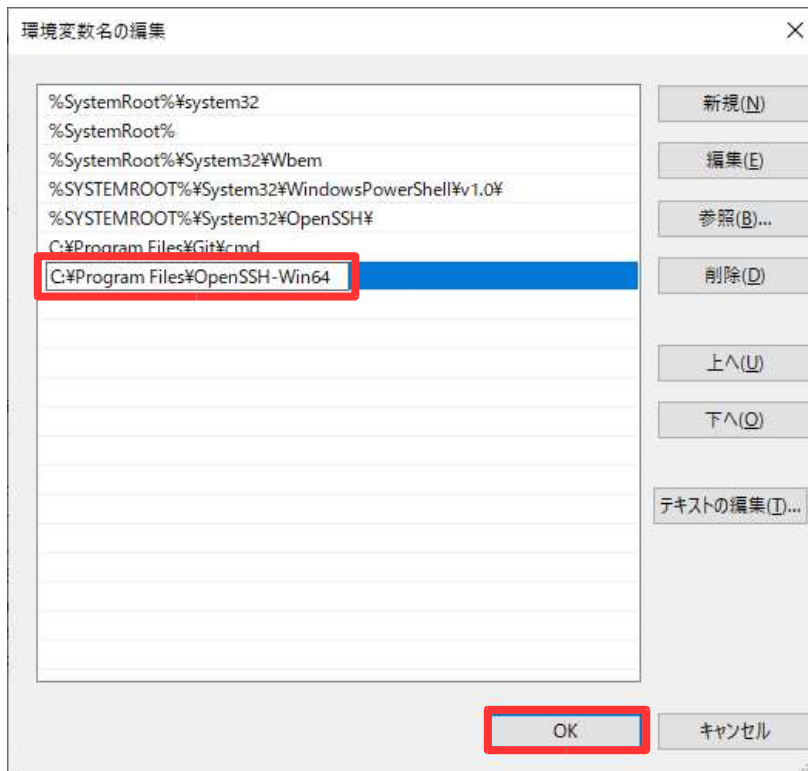


新規をクリック。

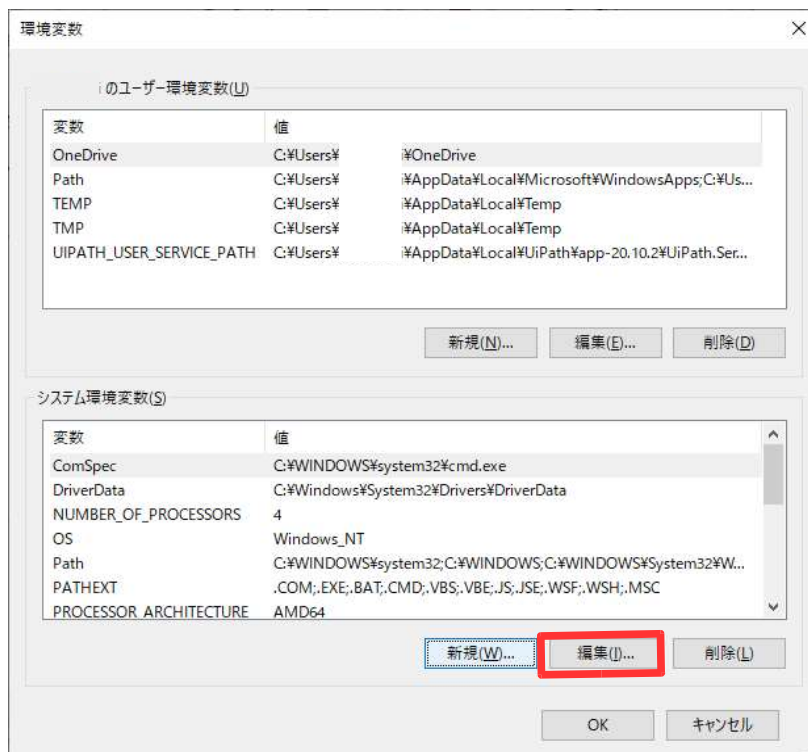


1.3) gitlab

「C:¥Program Files¥OpenSSH-Win64:¥OpenSSH」を追加する。

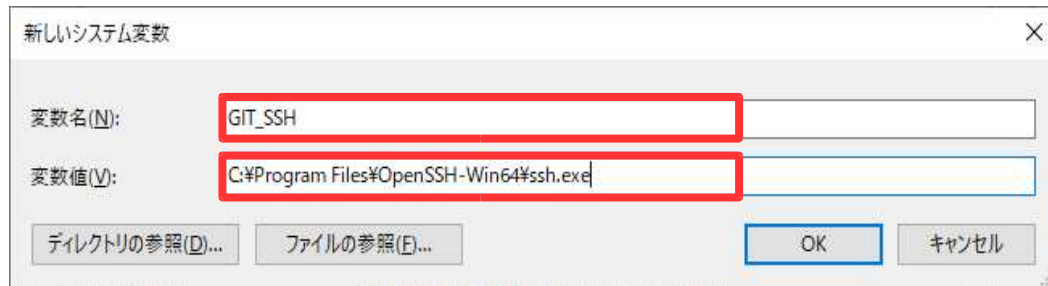


次にシステム環境変数に、2つ目の変数を新しく作成する。
システムの環境変数の「新規」をクリック。

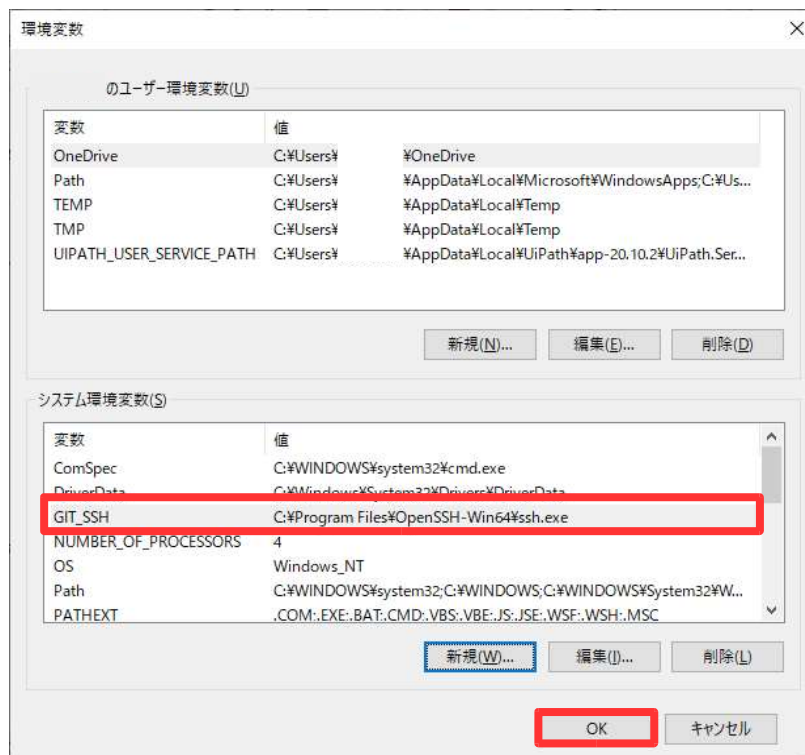


1.3) gitlab

変数名「GIT_SSH」、
変数値「C:¥Program Files¥OpenSSH-Win64¥ssh.exe」を追加する。



追加されたことを確認し、OKをクリック。



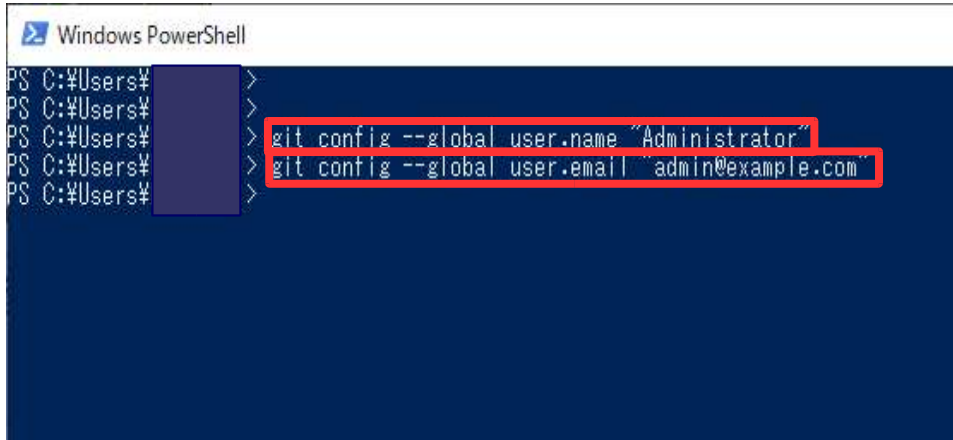
⑥Gitの設定

コマンドプロンプト、もしくはPowerShellを立ち上げる。

GitLabに登録したときのユーザー名とメールアドレスを設定するために、以下のコマンドを入力する。

```
git config --global user.name "Administrator"
```

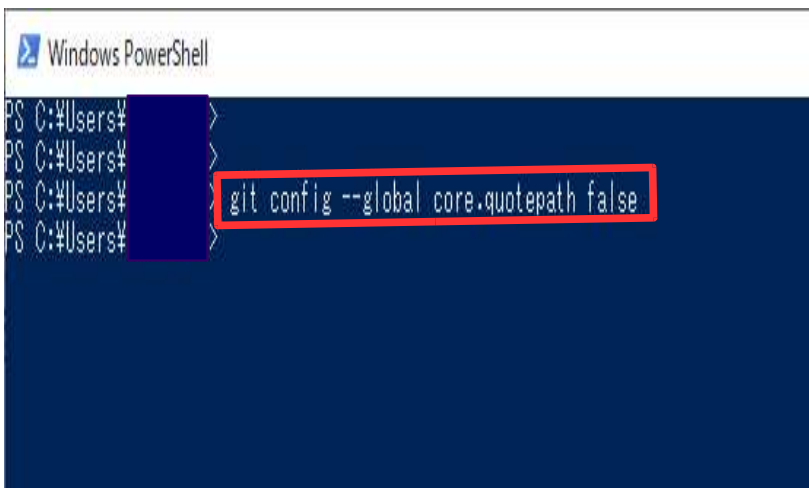
```
git config --global user.email "admin@example.com"
```



A screenshot of a Windows PowerShell terminal window. The title bar reads "Windows PowerShell". The terminal shows a series of prompts "PS C:\Users#\>" followed by two commands: "git config --global user.name \"Administrator\"" and "git config --global user.email \"admin@example.com\"". Both commands are highlighted with a red rectangular box.

Gitで日本語ファイルが文字化けするのを防ぐために、以下のコマンドを入力する。

```
git config --global core.quotePath false
```



A screenshot of a Windows PowerShell terminal window. The title bar reads "Windows PowerShell". The terminal shows a series of prompts "PS C:\Users#\>" followed by the command "git config --global core.quotePath false". The command is highlighted with a red rectangular box.

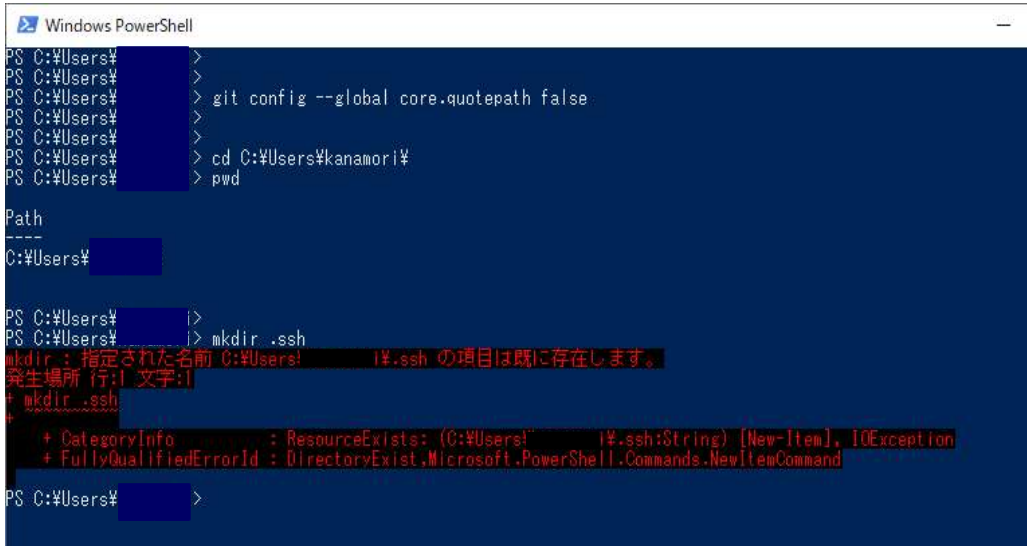
⑦SSH鍵の生成

ユーザアカウントフォルダ内に、「.ssh」フォルダを作成する。

以下のコマンドを実行する。

```
cd C:\Users\%username%\
mkdir .ssh
```

下記エラーは既にsshフォルダがあるというもの。通常は出ないエラーのため問題ない



```
Windows PowerShell
PS C:\Users\%username%\ >
PS C:\Users\%username%\ >
PS C:\Users\%username%\ > git config --global core.quotePath false
PS C:\Users\%username%\ >
PS C:\Users\%username%\ > cd C:\Users\kanamori\
PS C:\Users\%username%\ > pwd
Path
----
C:\Users\%username%\
PS C:\Users\%username%\ >
PS C:\Users\%username%\ > mkdir .ssh
mkdir : 指定された名前 C:\Users\%username%\\.ssh の項目は既に存在します。
発生場所 行:1 文字:1
+ mkdir .ssh
+ ~~~~~
+ CategoryInfo          : ResourceExists: (C:\Users\%username%\\.ssh:String) [New-Item], IOException
+ FullyQualifiedErrorId : DirectoryExist,Microsoft.PowerShell.Commands.NewItemCommand
PS C:\Users\%username%\ >
```

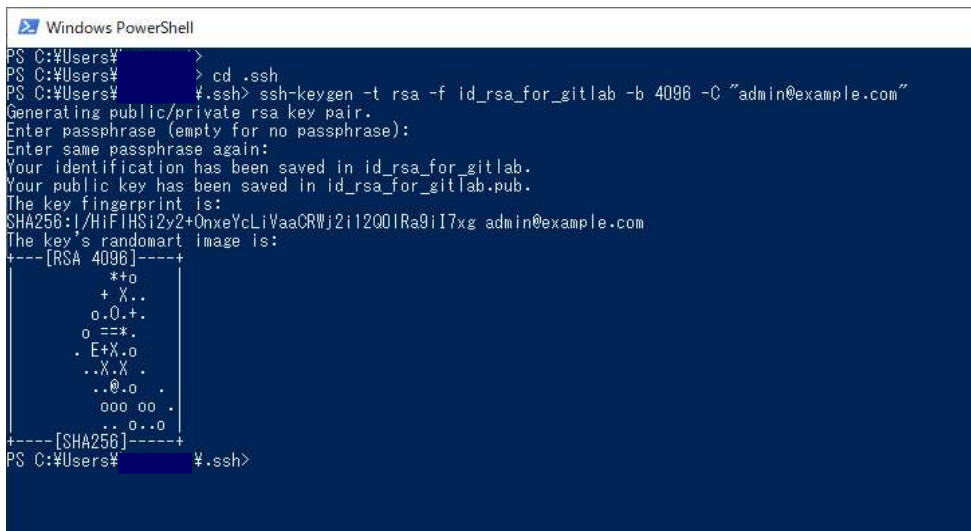
.sshフォルダ内に、公開鍵、秘密鍵を生成する。

以下のコマンドを実行する。

```
cd .ssh
ssh-keygen -t rsa -f id_rsa_for_gitlab -b 4096 -C "admin@example.com"
```

すると、パスワードの入力を促される。

メッセージが表示され、SSH鍵の生成が完了する。

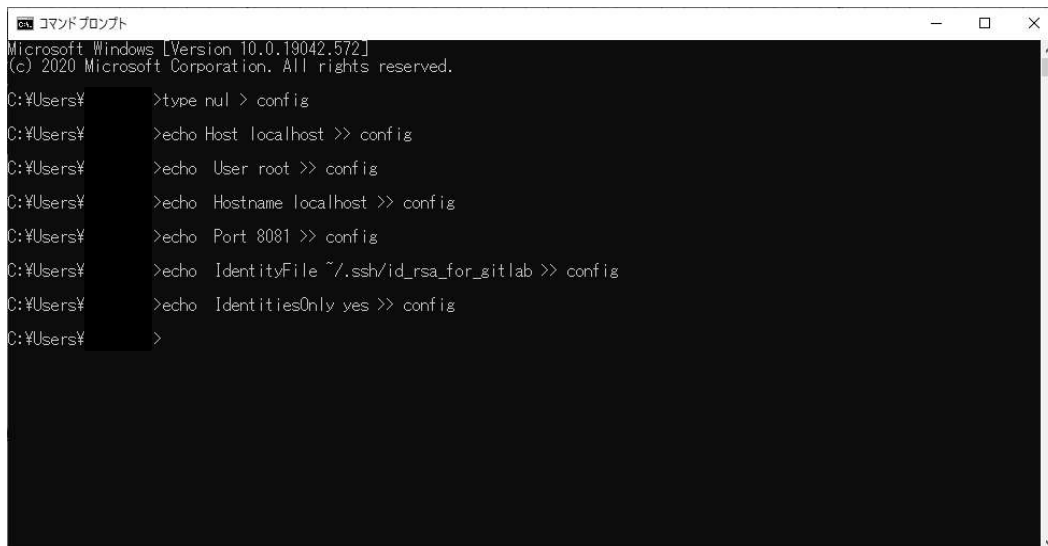


```
Windows PowerShell
PS C:\Users\%username%\ >
PS C:\Users\%username%\ > cd .ssh
PS C:\Users\%username%\ > .ssh> ssh-keygen -t rsa -f id_rsa_for_gitlab -b 4096 -C "admin@example.com"
Generating public/private rsa key pair.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in id_rsa_for_gitlab.
Your public key has been saved in id_rsa_for_gitlab.pub.
The key fingerprint is:
SHA256:1/Hf1HSi2y2+OnxeYcLiVaaCRWj2i12Q01Ra9iI7xg admin@example.com
The key's randomart image is:
+---[RSA 4096]-----+
|
|  .+X..
| o.O.+
| o==*
| .E+X.o
| ..X.X
| ..@.o
| ooo oo
| ..O..O
+---[SHA256]-----+
PS C:\Users\%username%\ > .ssh>
```

1.3) gitlab

コマンドプロンプトにて以下のコマンドを入力して、.ssh内に「config」というファイルを作成する。

```
type nul > config
echo Host localhost >> config
echo User root >> config
echo Hostname localhost >> config
echo Port 8081 >> config
echo IdentityFile ~/.ssh/id_rsa_for_gitlab >> config
echo IdentitiesOnly yes >> config
```



The screenshot shows a Windows Command Prompt window titled "コマンドプロンプト". The window displays the following text:

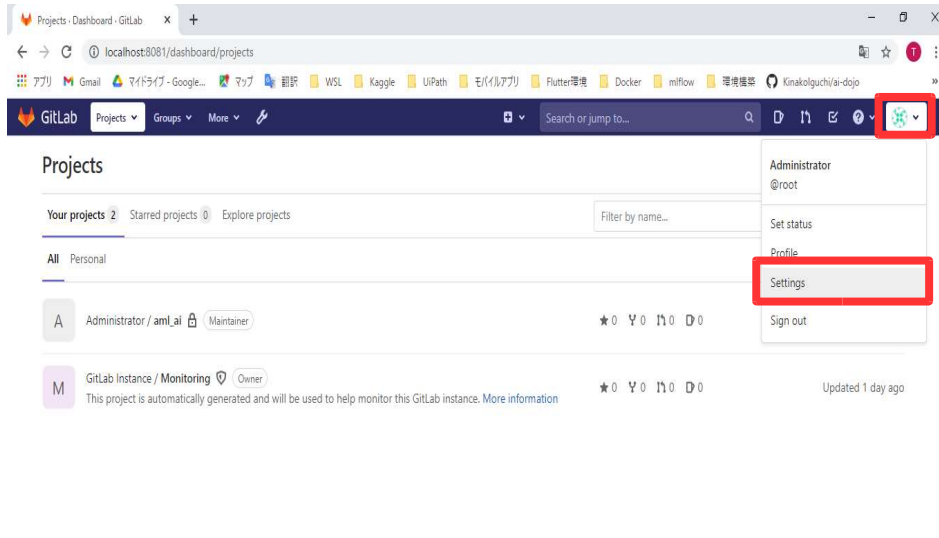
```
Microsoft Windows [Version 10.0.19042.572]
(c) 2020 Microsoft Corporation. All rights reserved.

C:\Users¥>type nul > config
C:\Users¥>echo Host localhost >> config
C:\Users¥>echo User root >> config
C:\Users¥>echo Hostname localhost >> config
C:\Users¥>echo Port 8081 >> config
C:\Users¥>echo IdentityFile ~/.ssh/id_rsa_for_gitlab >> config
C:\Users¥>echo IdentitiesOnly yes >> config
C:\Users¥>
```

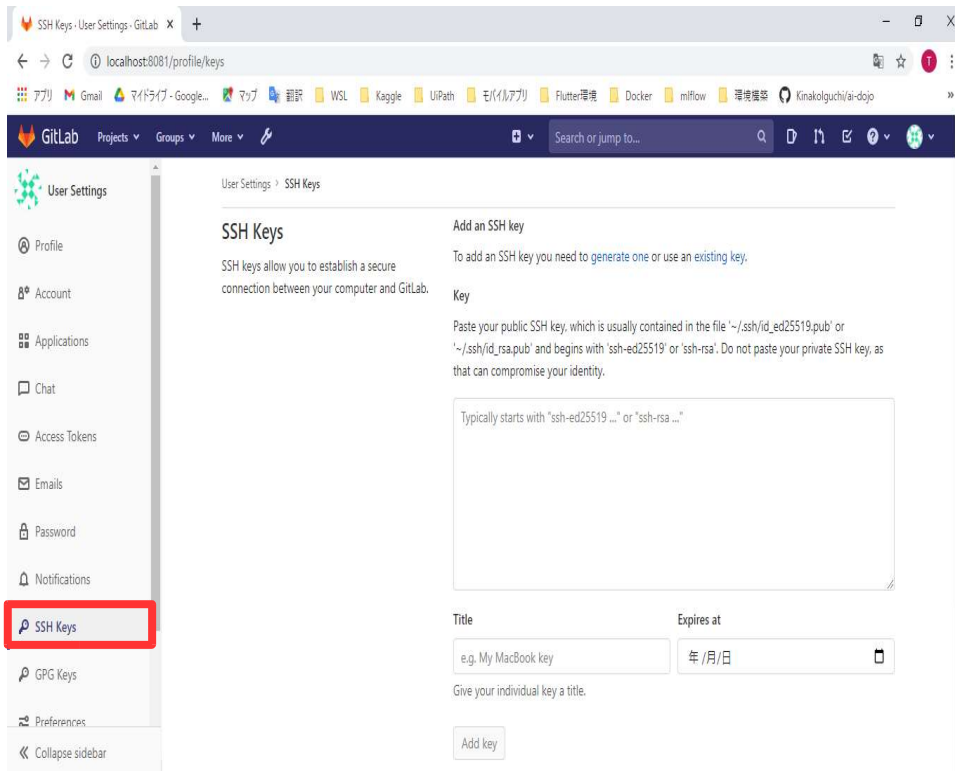

1.3) gitlab

⑧GitLabに公開鍵を登録

右端のユーザーアイコンのプルダウンから「settings」を押下



左のメニューにある「SSH Keys」をクリック



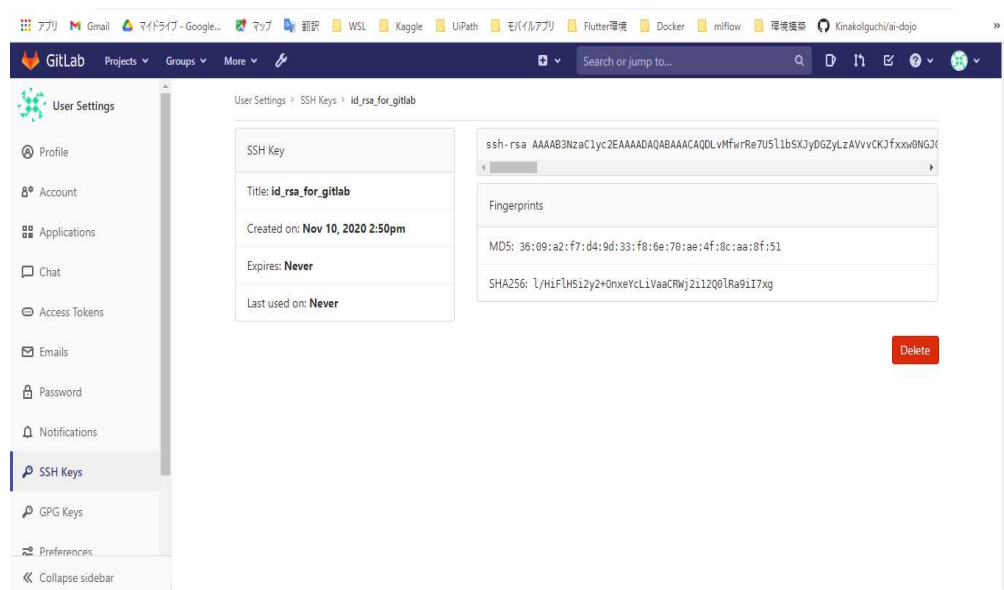
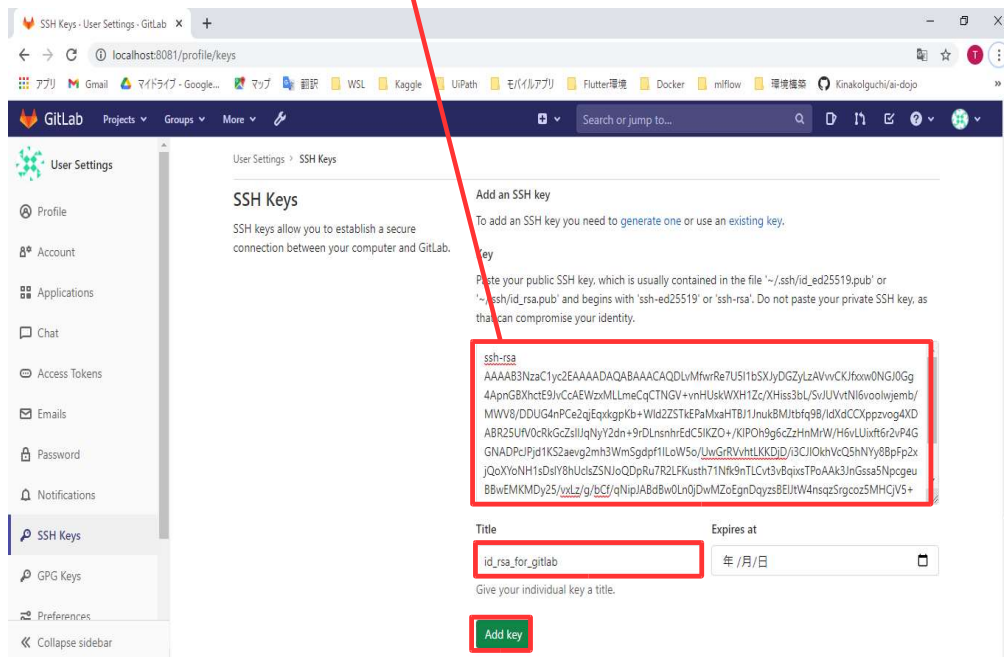
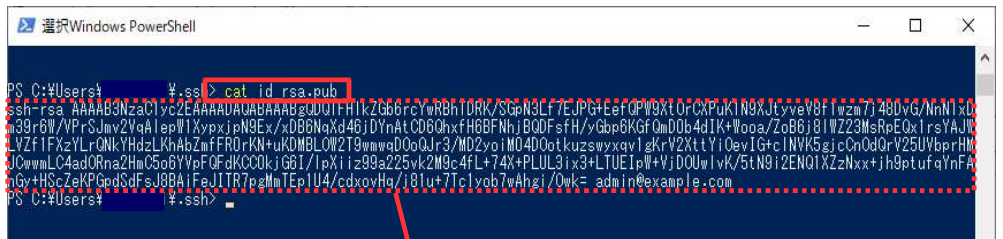
1.3) gitlab

公開鍵「id_rsa_for_gitlab.pub」の番号は、コンソール画面で以下のコマンドで中身を確認。

```
cat id_rsa.pub
```

表示された「ssh-rsa」から始まる番号をコピーする。

「Title」のところは任意のもの。ここでは秘密鍵のファイル名「id_rsa_for_gitlab」とする。



1.3) gitlab

1.3) gitlab

1.3) gitlab