
title : NIIC(DA) 構成管理操作手順書
ver : 1.0.0
date : 2022/03/23
author : 伊藤忠テクノソリューションズ

改版履歴

版名	改定者	改定日	改定ページ	改定内容	備考
初版	CTC	2022/03/23			初版作成

目次

はじめに

1. クライアント環境構築
 - 1.1 Git Bashインストール
 - 1.2 SourceTreeインストール
2. NIIC(DA) Gitリポジトリ作成
3. NIIC(DA) Gitローカル操作
 - 3.1 リモートリポジトリからローカルGitへソース取得
 - 3.2 ソース編集
 - 3.2.1 SourceTreeへローカルリポジトリ登録
 - 3.2.2 ソース修正
 - 3.2.3 ソース追加
 - 3.2.4 ソース削除
 - 3.2.5 ソース最新化
4. その他

はじめに

このドキュメントは、NIIC (DA) 構成管理のソース管理の基本操作手順について説明します。
ソース管理のツールを専用ツールから、Gitへと移行します。

Git (ギット) は、プログラムのソースコードなどの変更履歴を記録・追跡するための**分散型バージョン管理システム**である。Linuxカーネルのソースコード管理に用いるためにリーナス・トーバルズによって開発され、それ以降ほかの多くのプロジェクトで採用されている。Linuxカーネルのような巨大プロジェクトにも対応できるように、動作速度に重点が置かれている。現在のメンテナは濱野純 (英語: Junio C Hamano/Google) で、2005年7月から担当している。
出典: フリー百科事典『ウィキペディア (Wikipedia) 』

1. クライアント環境構築

1.1 Git Bashインストール

別紙：08.NIIC(DA) 構成管理ツール移行構築手順（クライアントツールインストール）

_20220209-01C.xlsx

※CTCの開発環境はインストール済み

1.2 SouceTreeインストール

別紙：08.NIIC(DA) 構成管理ツール移行構築手順（クライアントツールインストール）

_20220209-01C.xlsx

※CTCの開発環境はインストール済み

2. NIIC (DA) Gitリポジトリ作成

ssifリポジトリを例に作成の流れを説明します。

実際には運用開始時点で準備が完了していますので、作成する必要はありません。

ローカルの自PCに作成する場合もこの流れを踏襲できます。

サーバー:nidsvr01b (検証運用管理サーバー)

ディレクトリ:/export/home/smc/source/git

ユーザ:source

```
$ cd /export/home/smc/source/git
$ pwd
$ mkdir ssif # ローカルリポジトリ
$ mkdir ssif.git # リモートリポジトリ
$ cd ssif.git
$ git init --bare --shared=true # ベアリポジトリ作成
$ cd ..
$ cd ssif
$ pwd
$ git init # gitの初期化
$ cp -ip /mnt/win/shared/ssif.git.gz . # ssifのリソースをコピー
$ gunzip ssif.git.gz
$ tar xvf ssif.tar # リソースの展開
$ ls -lR # 展開の確認
$ git add . # リソースをインデックスに登録します。「.」はファイル名でも可能、
# 「.」はファイルすべて
$ git status # 現在のコミットなどの状態を表示
$ git commit -m "first commit" # コミットは必ずコメント(-m "xxxxxx")を付けます。
$ git log # コミットの履歴を確認
$ git remote add origin /export/home/smc/source/git/ssif.git # リモートリポジトリを関連付け
# originはデフォルトのマスタ名
$ git remote -v # リモートリポジトリを確認
$ git push origin master # リモートのoriginへmasterをpushします。
```

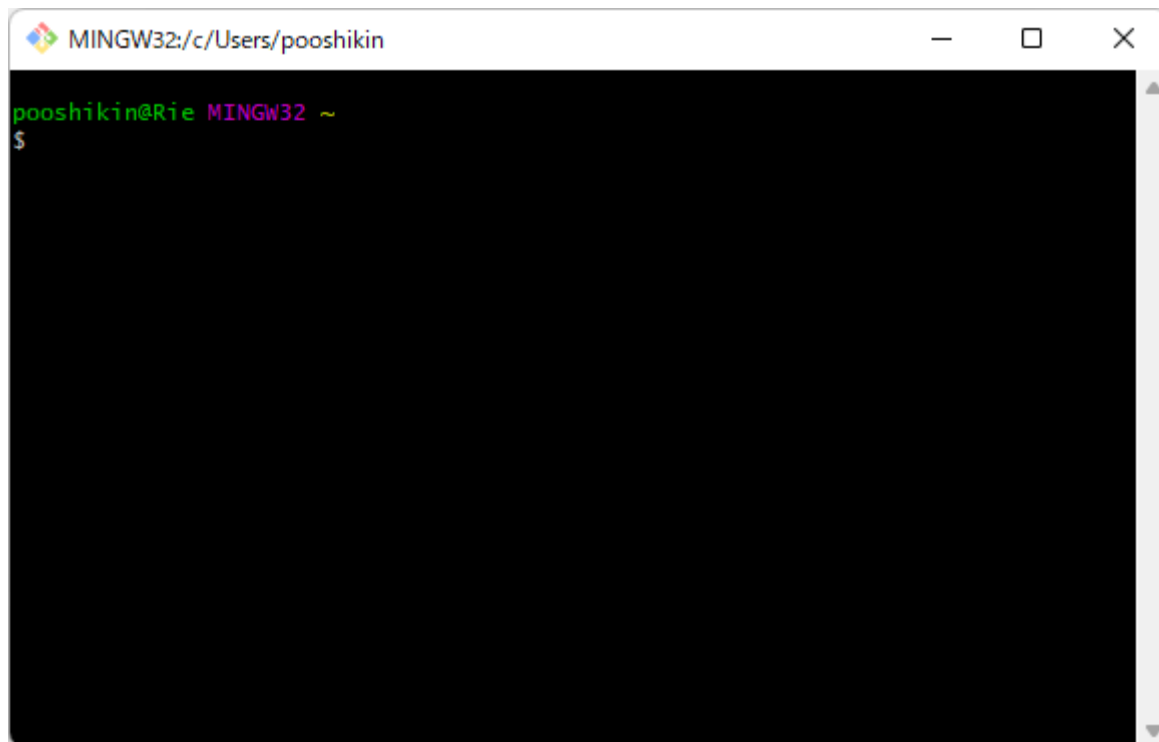
上記の作業を経て、すべてのマスターリポジトリとリモートリポジトリが作成されます。

ユーザは、リモートリポジトリをローカルにcloneして実作業を行うことになります。

3. NIIC (DA) Gitローカル操作

3.1 リモトリポジトリからローカルGitへソース取得

Windows > スタート > Git BashにてGit Bashのターミナルを起動します。

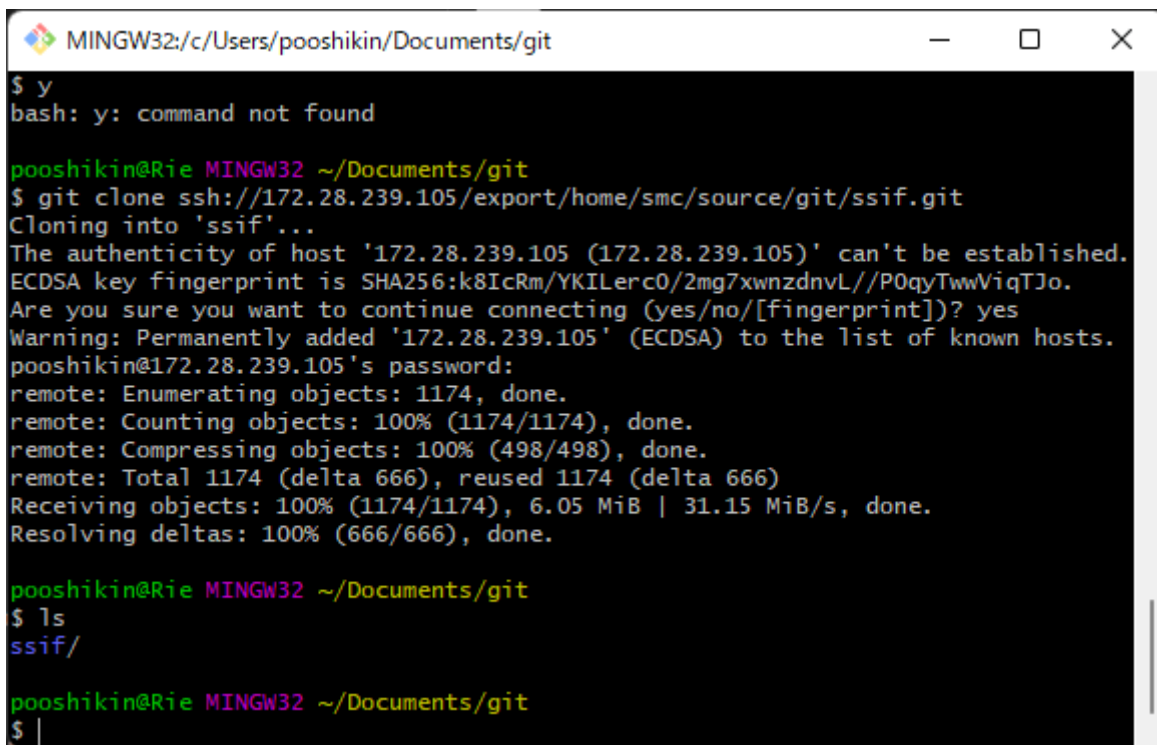


- ssifのソースを取得する場合の例

```
$ cd Documents          # Documentsに移動
$ pwd
$ mkdir git             # gitディレクトリを作成
$ cd git
$ git clone ssh://172.28.239.105/export/home/smc/source/git/ssif.git
Cloning into 'ssif'...
The authenticity of host '172.28.239.105 (172.28.239.105)' can't be established.
ECDSA key fingerprint is SHA256:k8IcRm/YKILerc0/2mg7xwnzdnvL//P0qyTwwViqTJo.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '172.28.239.105' (ECDSA) to the list of known hosts.
pooshikin@172.28.239.105's password: # パスワードを入力 P番またはS番のパスワード
remote: Enumerating objects: 1174, done.
remote: Counting objects: 100% (1174/1174), done.
remote: Compressing objects: 100% (498/498), done.
remote: Total 1174 (delta 666), reused 1174 (delta 666)
Receiving objects: 100% (1174/1174), 6.05 MiB | 31.15 MiB/s, done.
Resolving deltas: 100% (666/666), done.
```

上記のIPはデモ用です。実際の運用管理サーバー(10.47.70.10)

ssifディレクトリが作成されています。

A screenshot of a terminal window titled 'MINGW32:/c/Users/pooshikin/Documents/git'. The terminal shows the following commands and output:

```
$ y
bash: y: command not found

pooshikin@Rie MINGW32 ~/Documents/git
$ git clone ssh://172.28.239.105/export/home/smc/source/git/ssif.git
Cloning into 'ssif'...
The authenticity of host '172.28.239.105 (172.28.239.105)' can't be established.
ECDSA key fingerprint is SHA256:k8IcRm/YKILerc0/2mg7xwnzdnhvL//P0qyTwwViqTJo.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '172.28.239.105' (ECDSA) to the list of known hosts.
pooshikin@172.28.239.105's password:
remote: Enumerating objects: 1174, done.
remote: Counting objects: 100% (1174/1174), done.
remote: Compressing objects: 100% (498/498), done.
remote: Total 1174 (delta 666), reused 1174 (delta 666)
Receiving objects: 100% (1174/1174), 6.05 MiB | 31.15 MiB/s, done.
Resolving deltas: 100% (666/666), done.

pooshikin@Rie MINGW32 ~/Documents/git
$ ls
ssif/

pooshikin@Rie MINGW32 ~/Documents/git
$ |
```

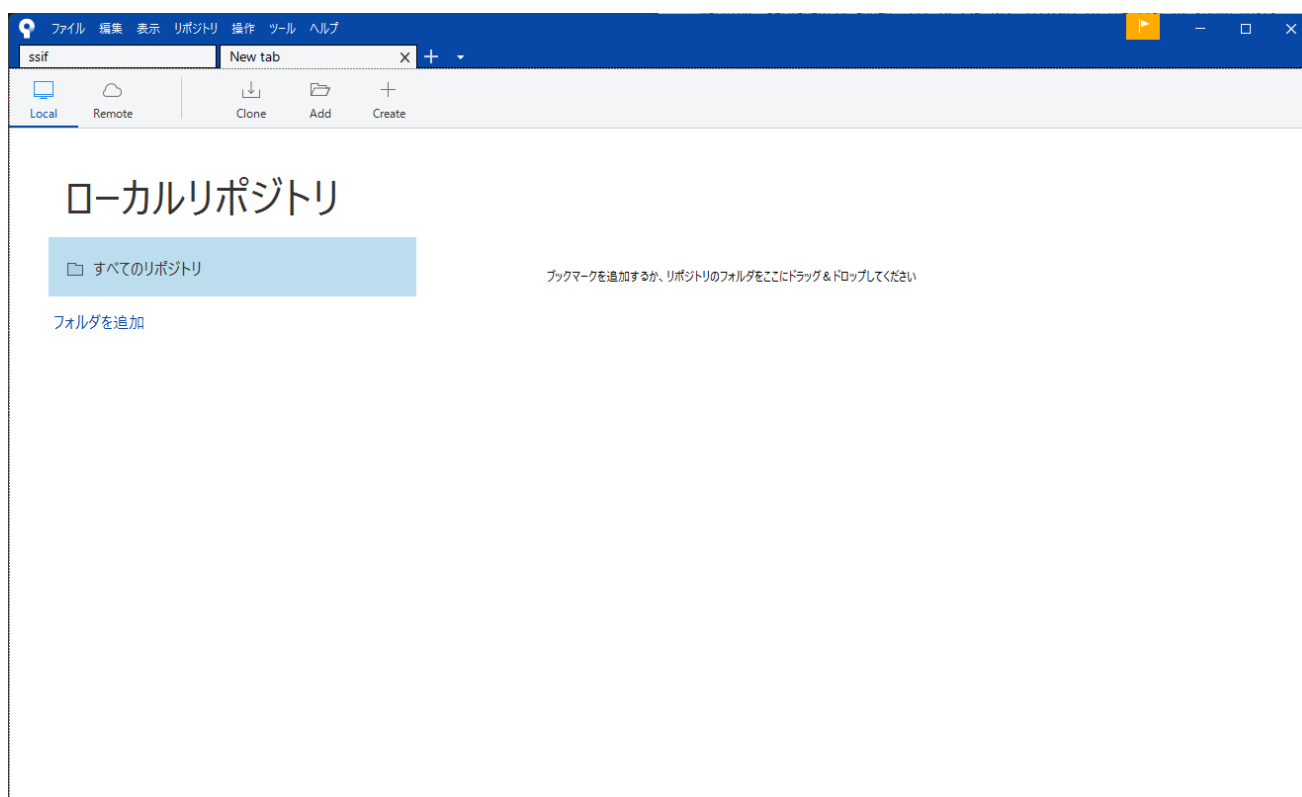
これでローカルのGitへssifソースの取得は完了です。

※同様に必要なシステムのリポジトリをローカルに取得します。

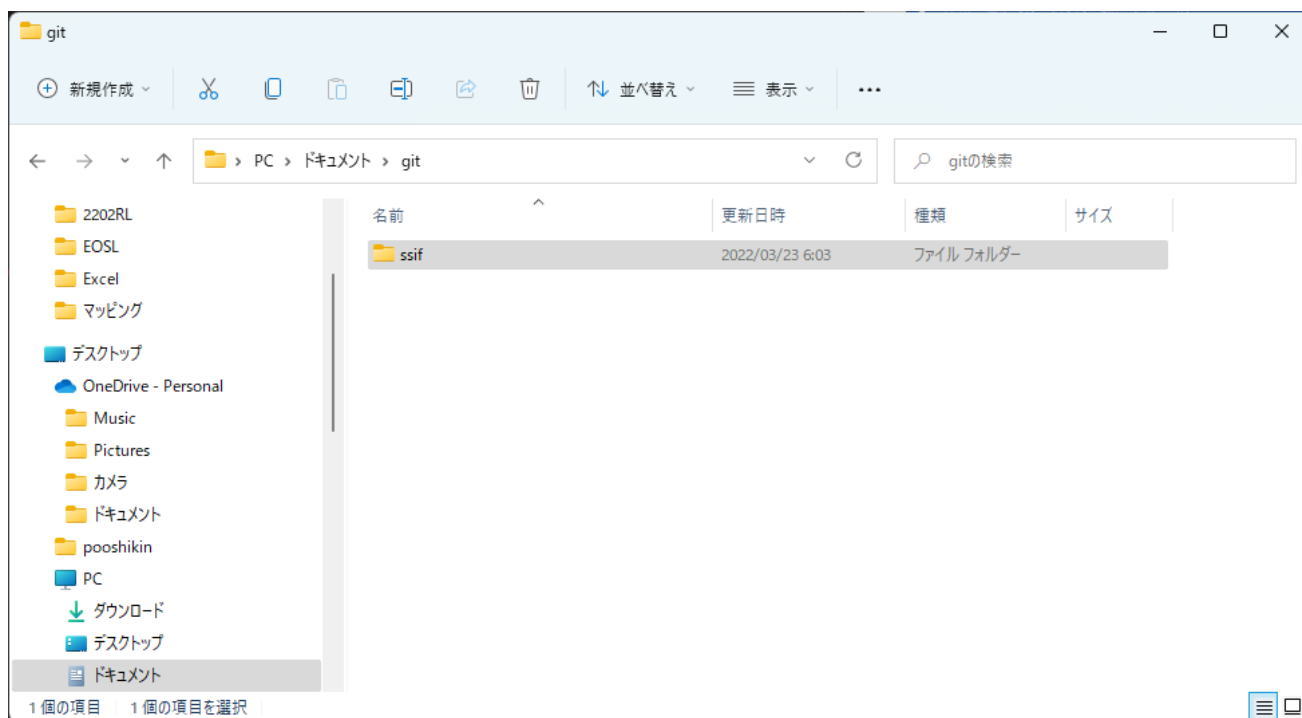
3.2 ソース編集

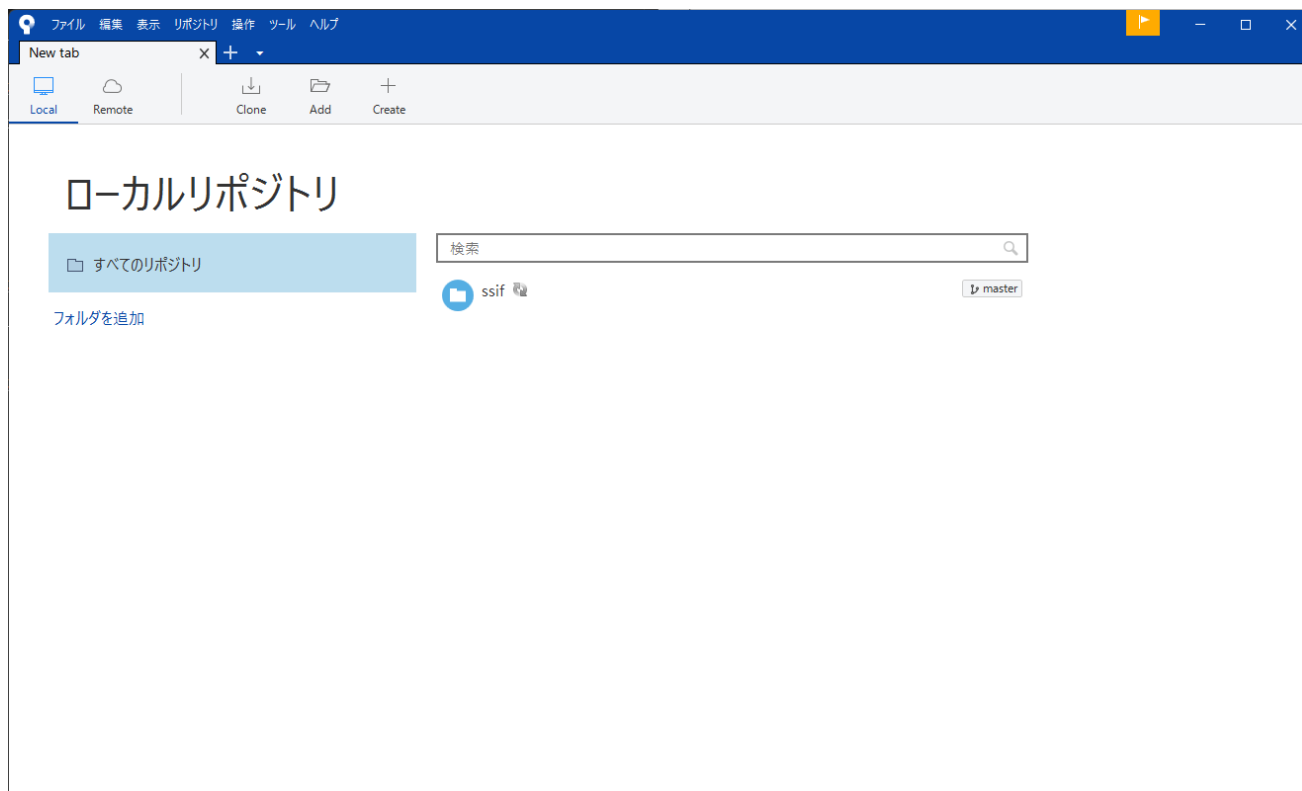
3.2.1 SourceTreeへローカルリポジトリ登録

デスクトップ > SourceTree

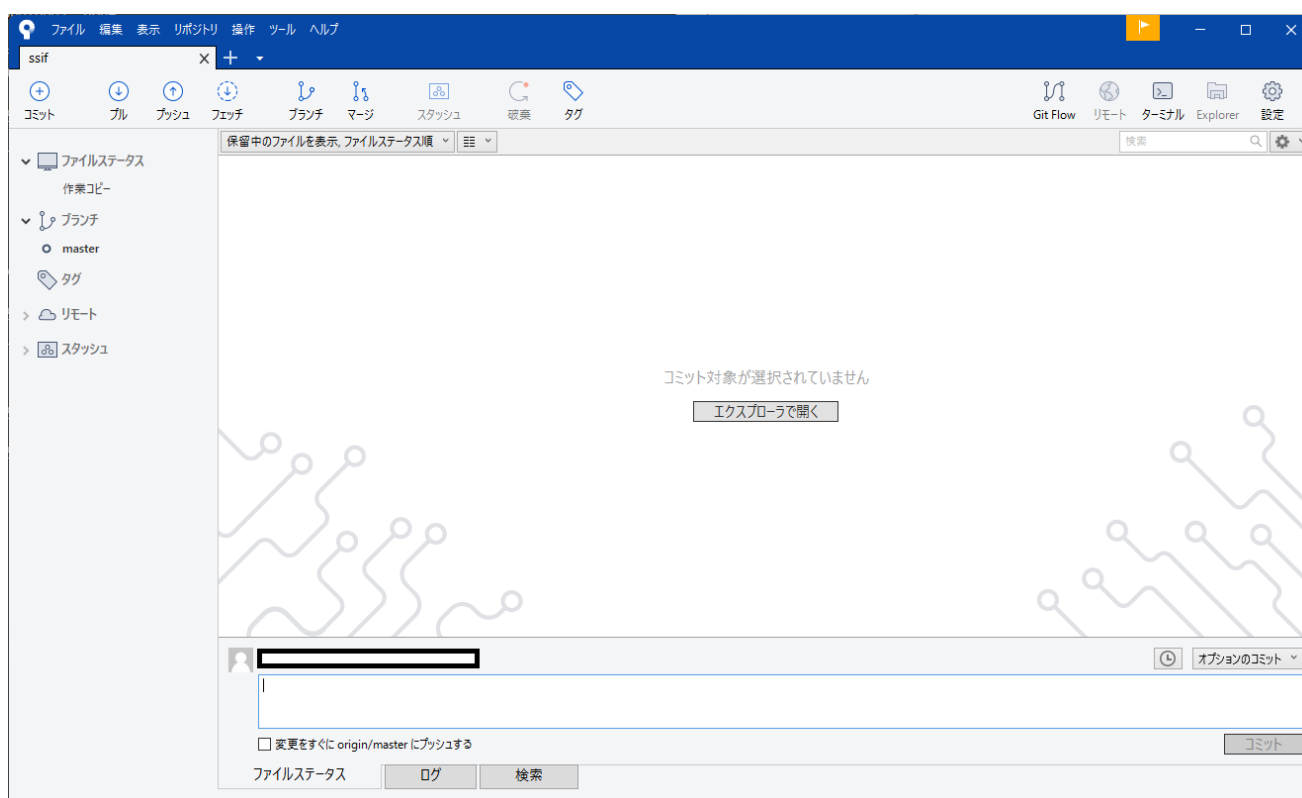


ローカルリポジトリへ前節でCloneしたssifのディレクトリをエクスプローラーからドラッグアンドドロップで登録します。





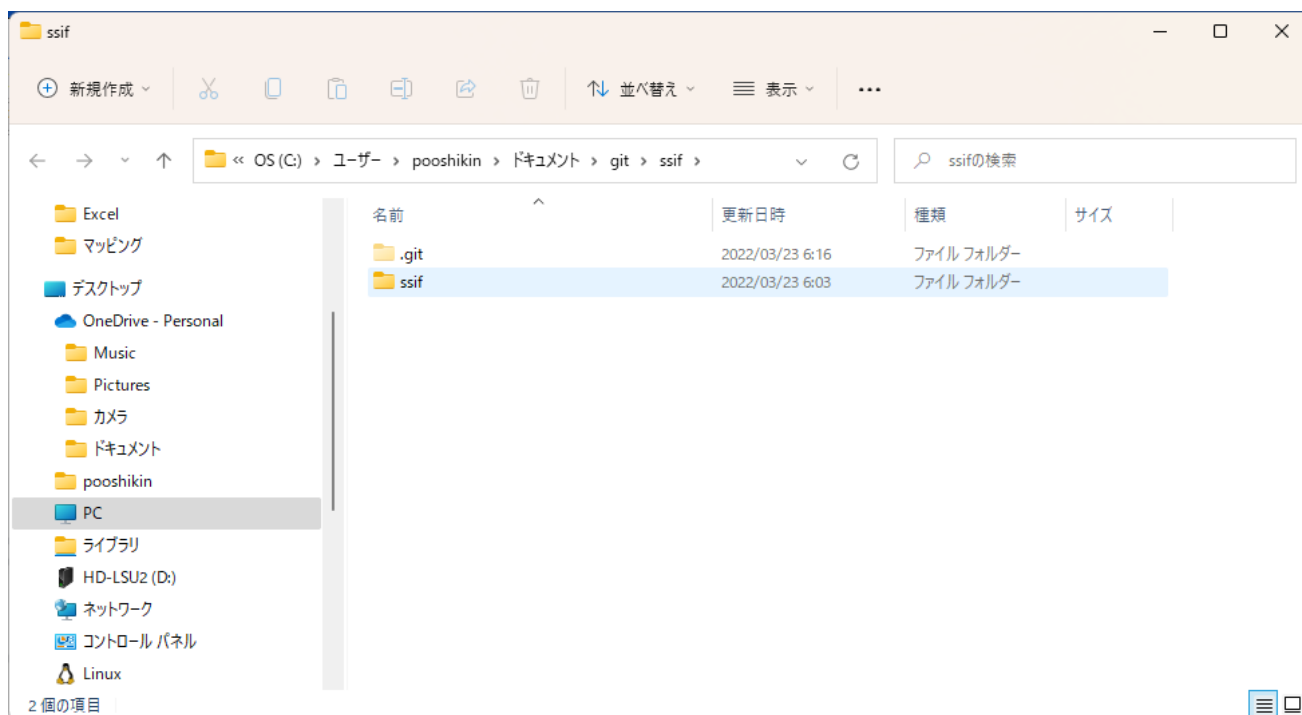
登録されたssifをダブルクリックします。



これでSourceTreeでソースを管理できるようになります。

3.2.2 ソース修正

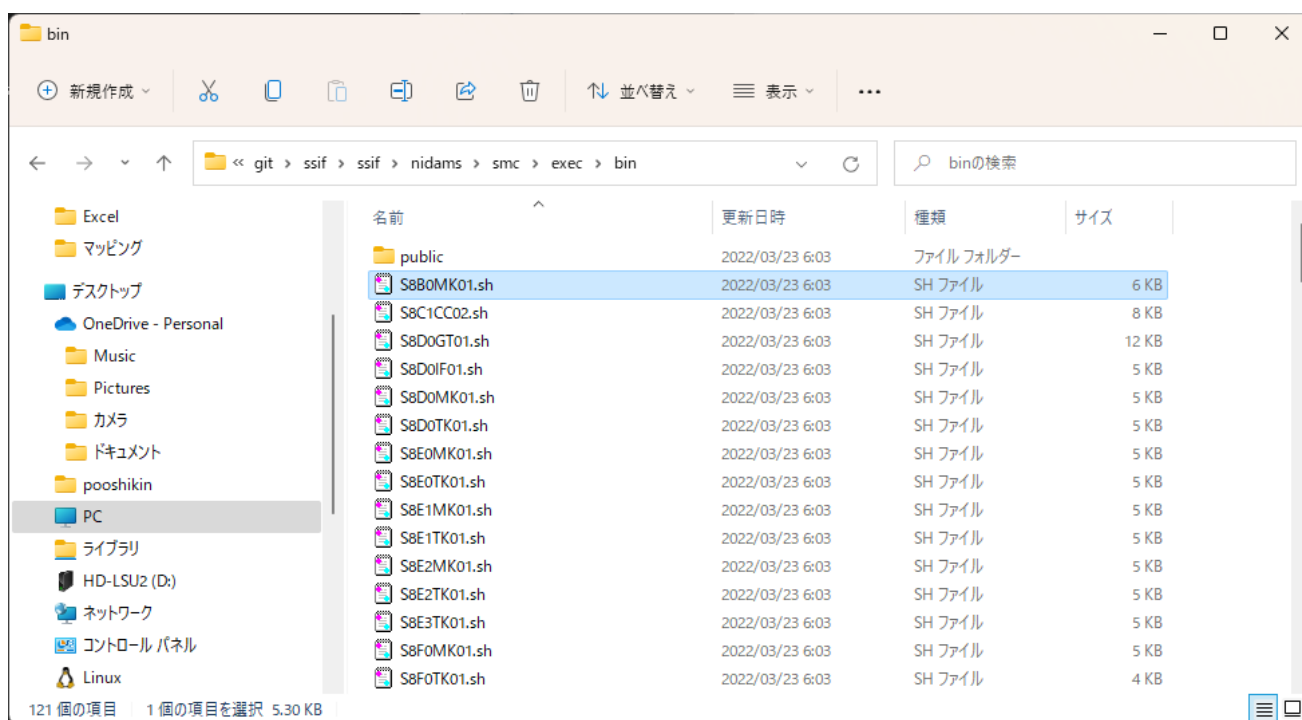
SourceTreeの「エクスプローラーで開く」をクリックします。



.gitはGitの管理ディレクトリです。

ssifディレクトリの階層に入り修正するソースを選択します。

「S8B0MK01.sh」を編集してみます。（あらかじめ、桜エディターなどでshを関連付けします）

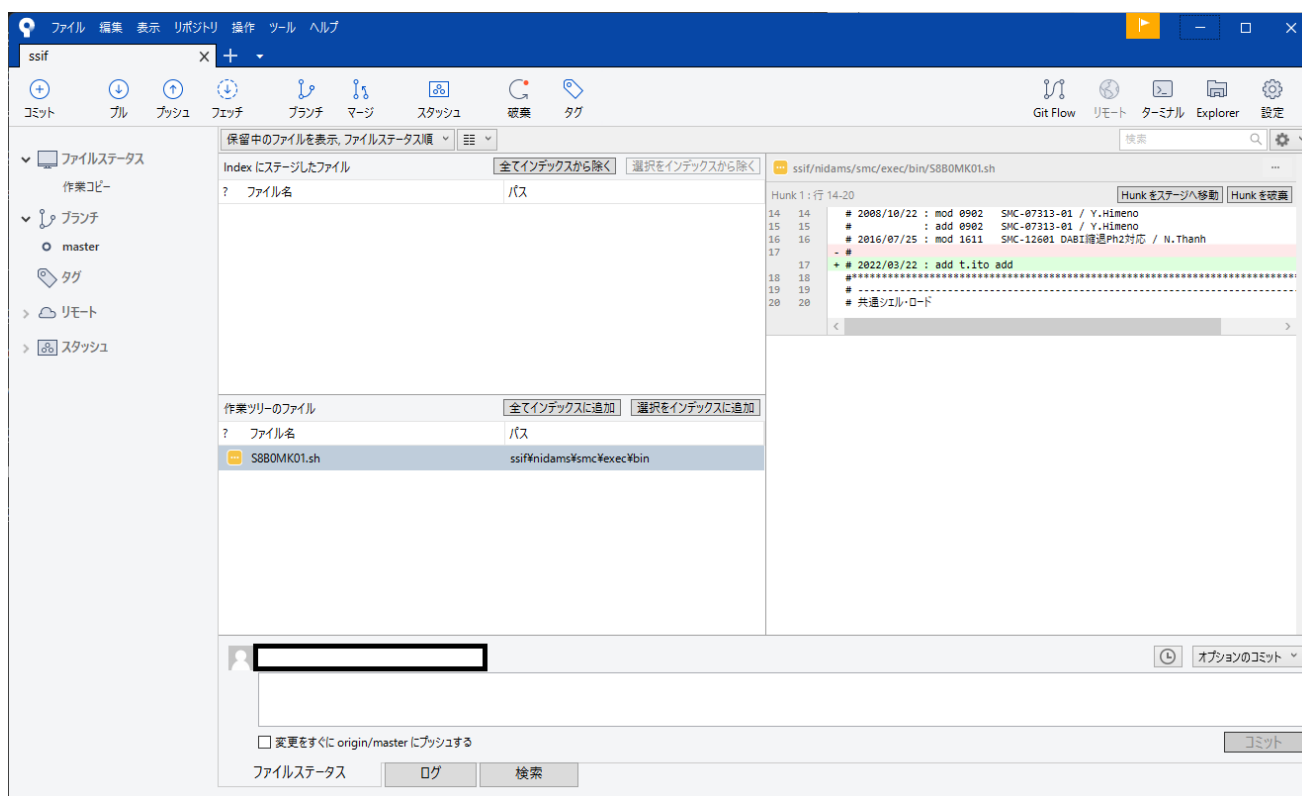


17行目に追加して、上書き保存します。

```

1 #!/usr/sunos/bin/sh
2 #bin/sh
3 SMC-12601: 2016/07/25: Mod. End
4 *****
5 著者: 伊藤忠テクノサイエンス株式会社
6 システム名: SMC
7 ファイル名: S8B0MK01.sh
8 バージョン: $Revision$
9 概要: create BBO: S/O: files: function: call
10 作成日: 2003/08/30
11 作成者: M.Mochizuki
12 *****
13 # 2008/10/06 : mod. 0902 SMC-07313-01: EOSL案件対応 / T.Yamada
14 # 2008/10/22 : mod. 0902 SMC-07313-01: Y.Himeno
15 # 2016/07/25 : add. 0902 SMC-07313-01: Y.Himeno
16 # 2016/07/25 : mod. 1611 SMC-12601: DABI繰越Ph2対応 / N.Thanh
17 # 2022/03/22 : add t.ito add
18 *****
19
20 # 共通シェル・ロード
21 -----
22 # SMC-07313-01: mod. start: 2008/10/22: Y.Himeno
23 # ./opt/smc/exec/bin/S8B0CM01.sh
24 # ${SMC_HOME}/exec/bin/S8B0CM01.sh
25 # SMC-07313-01: mod. end: 2008/10/22: Y.Himeno
26
27 # SMC-07313-01: EOSL案件対応: mod. start: 2008/10/06: T.Yamada
28 # ./opt/smc/exec/etc/S8B0CM01.rc
29 # ${SMC_HOME}/exec/etc/S8B0CM01.rc
30 # SMC-07313-01: EOSL案件対応: mod. end: 2008/10/06: T.Yamada
31
32
33 # -----
34 # エラー処理
35 -----
  
```

SourceTreeに戻ると編集したファイルが表示され、右上に変更点が表示されています。
作業ツリーの変更ファイルの「選択ファイルをインデックスに追加」をクリックします。



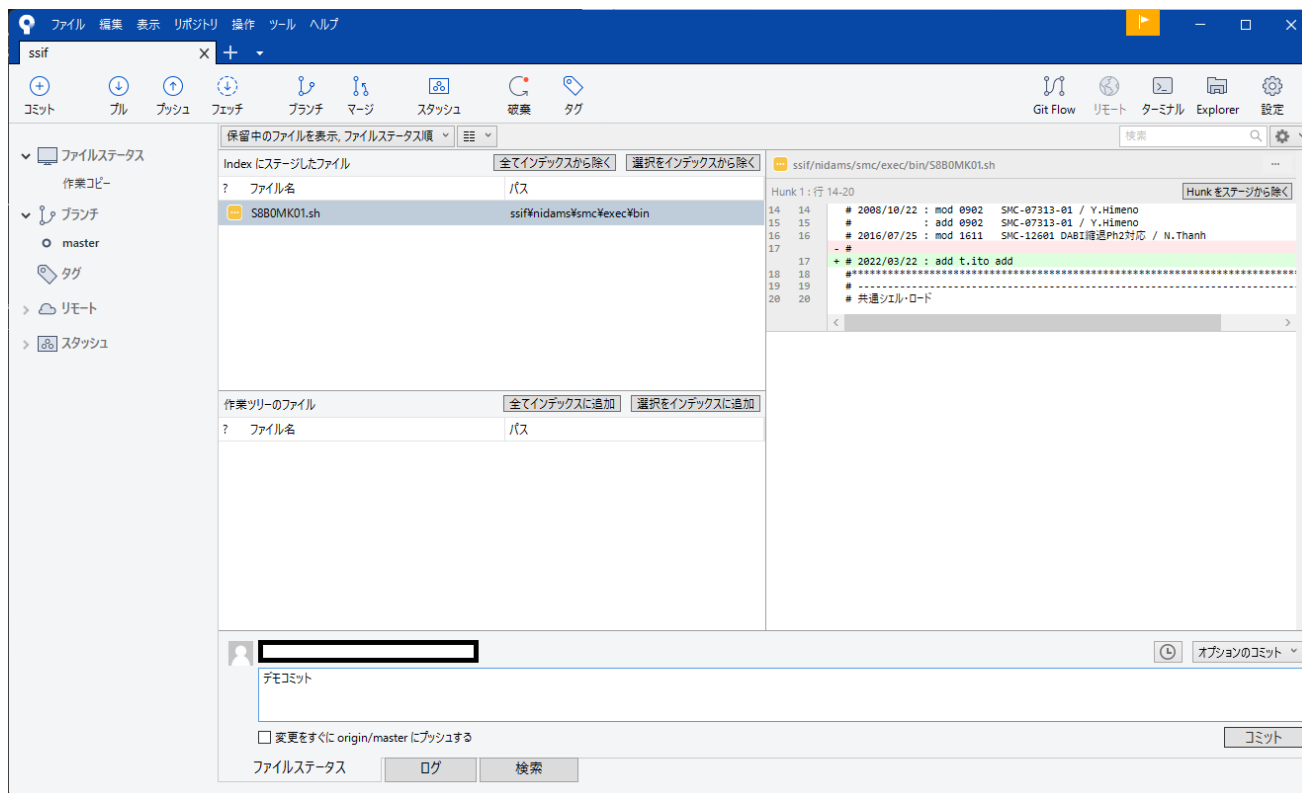
Indexにステージしたファイルが追加されます。

ここでIndexからはずす場合は、選択をインデックスから除くで行います。

この時点でファイルはコミットされていません。

コミットするには、下部のテキストボックスにコメントを入力して、コミットボタンを押下します。

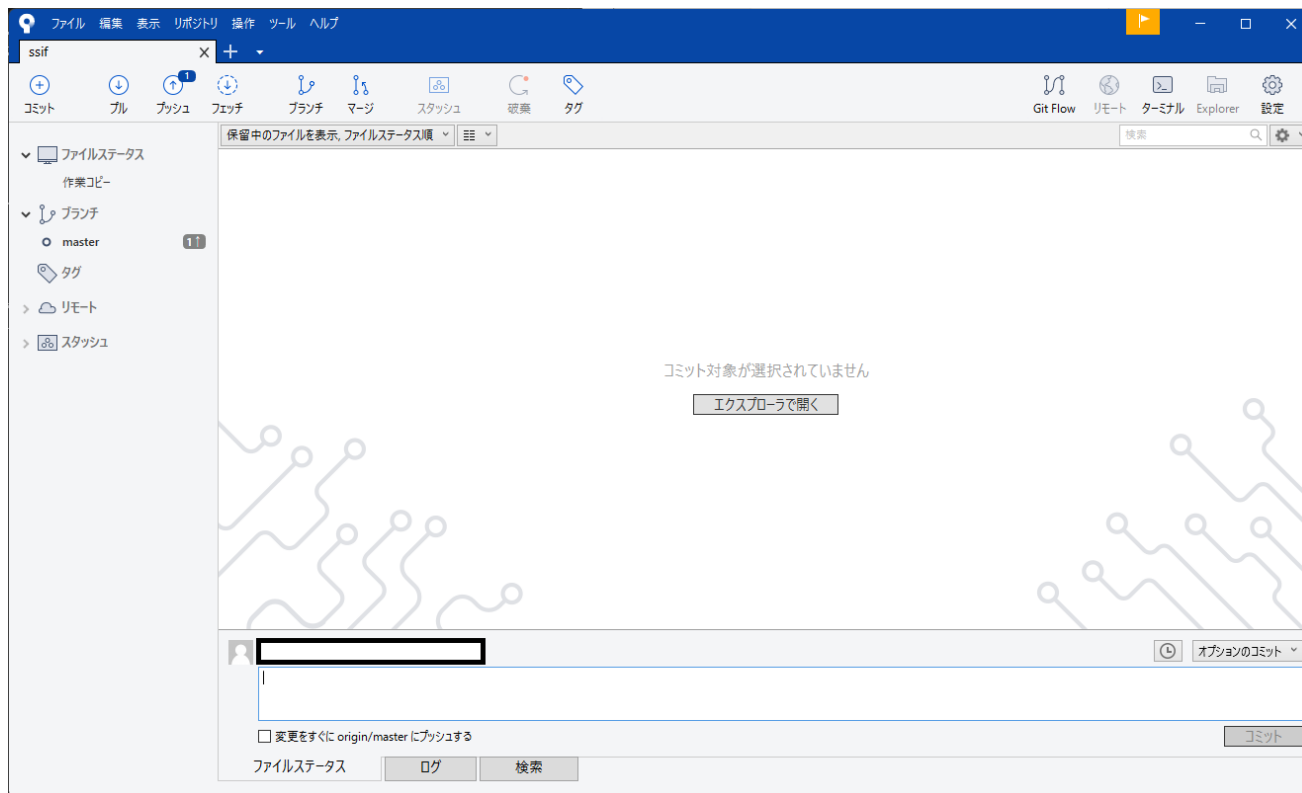
コメントを「デモコミット」を入力しています。



プッシュボタンに「1」が表示されています。

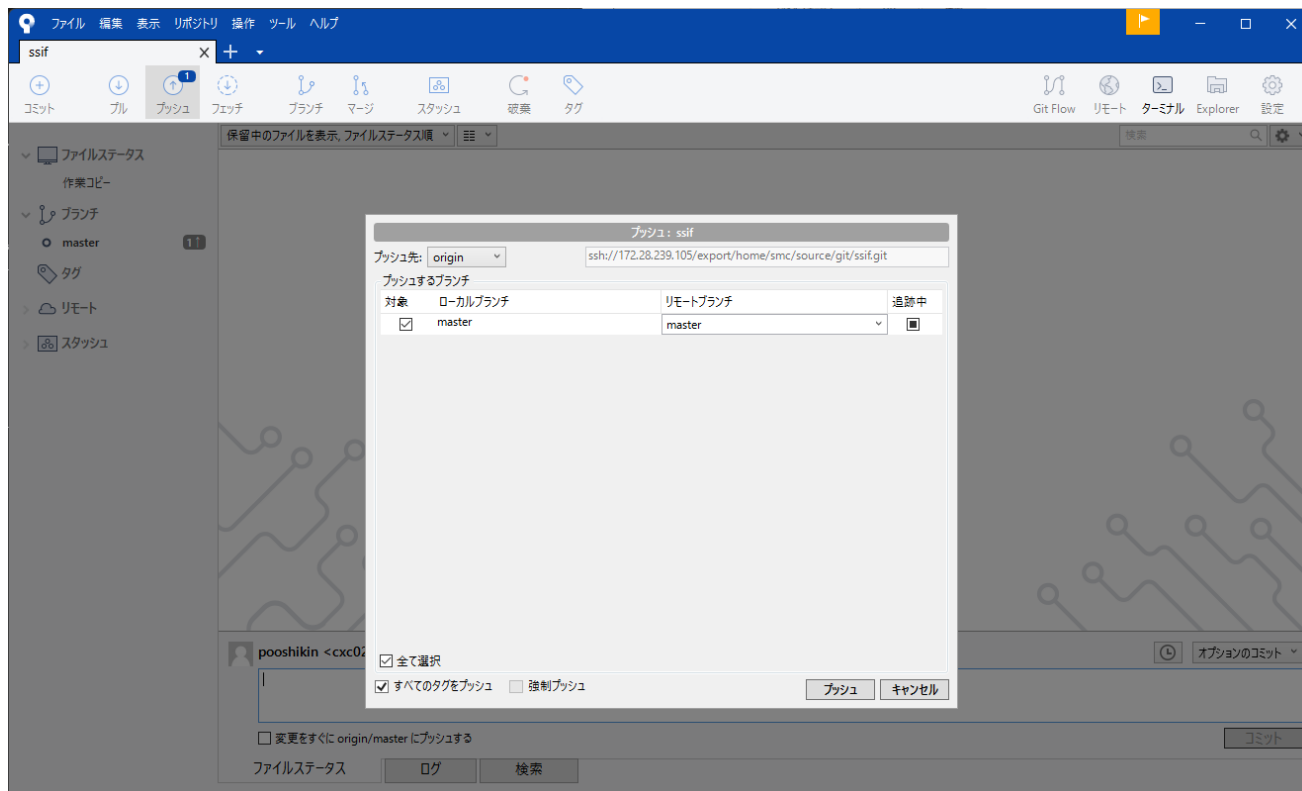
これは変更したファイルをssifのリモートリポジトリへpushする操作になります。

ボタンを押下します。



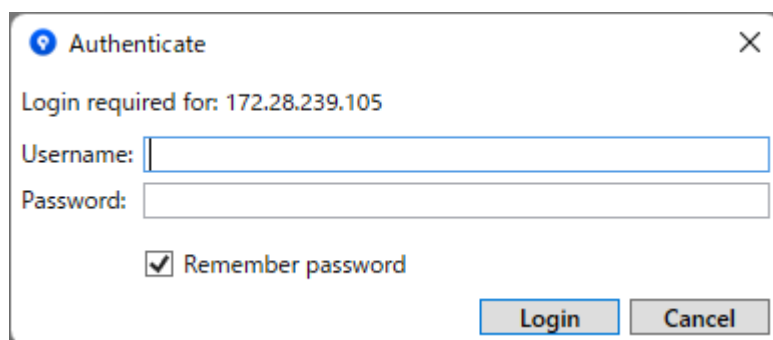
プッシュダイアログが表示されます。

ローカルブランチ (master) からリモートブランチ (master) へpushします。



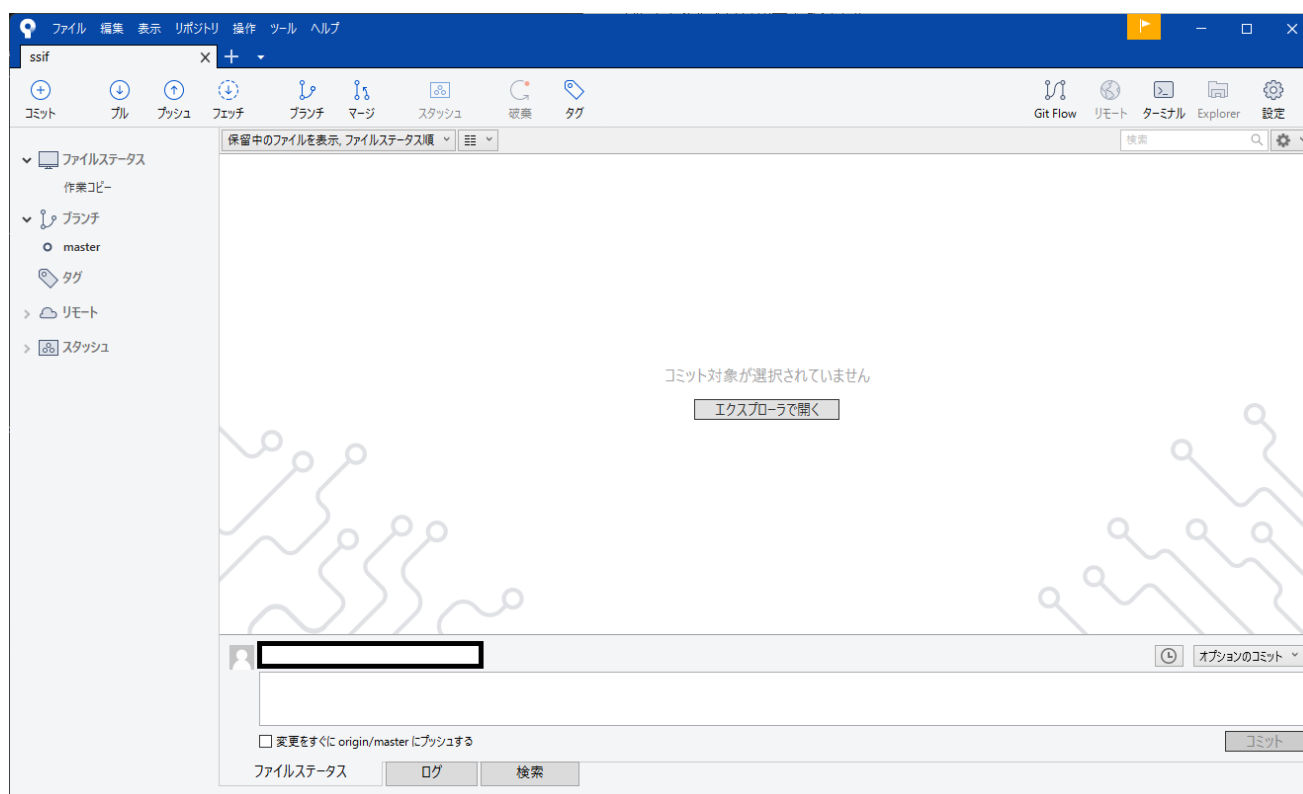
ユーザとパスワードが求められます。(P番/S番とパスワード)

※一度記憶すると次回から表示されなくなります。



An 'Authenticate' dialog box with a close button (X) in the top right corner. It contains the text 'Login required for: 172.28.239.105'. Below this are two input fields: 'Username:' and 'Password:'. A checkbox labeled 'Remember password' is checked. At the bottom right are two buttons: 'Login' and 'Cancel'.

編集中のファイルがなくなった状態です。



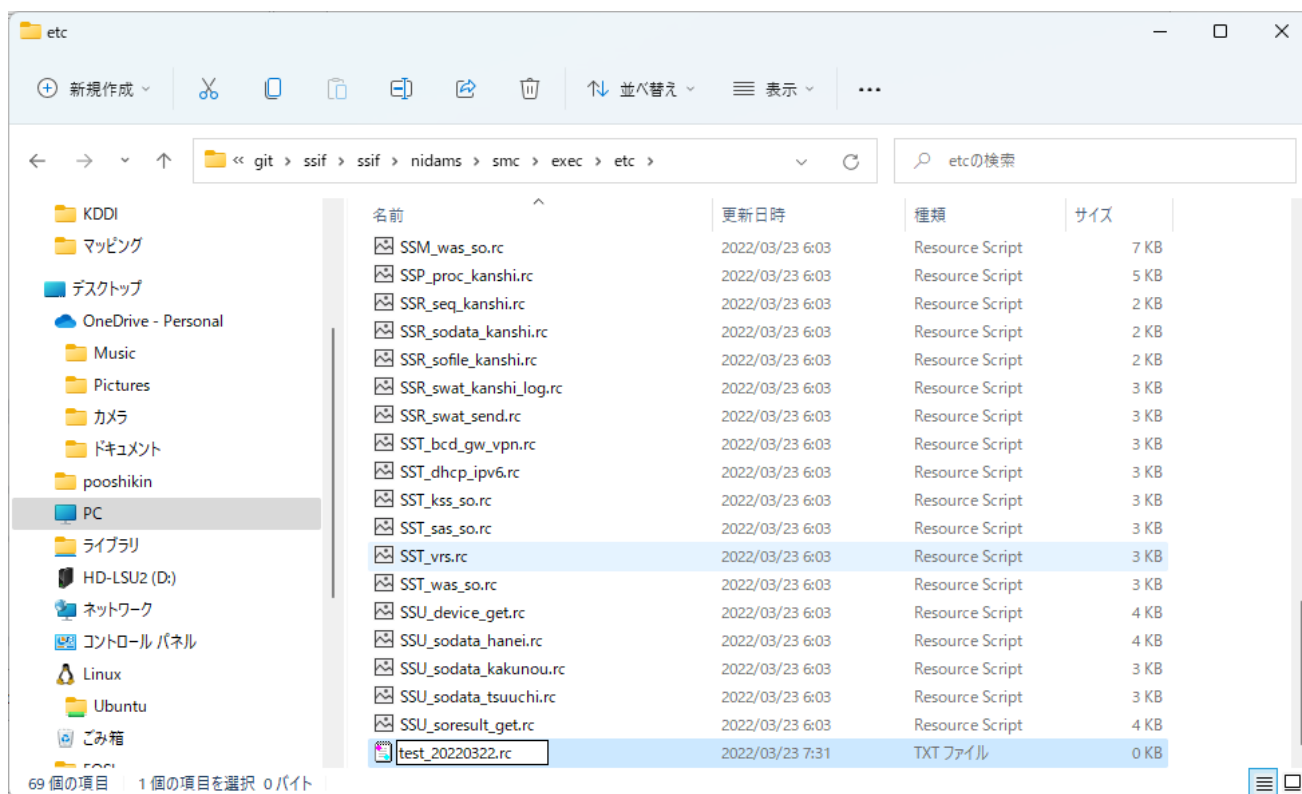
3.2.3 ソース追加

SourceTreeの「エクスプローラーで開く」をクリックします。

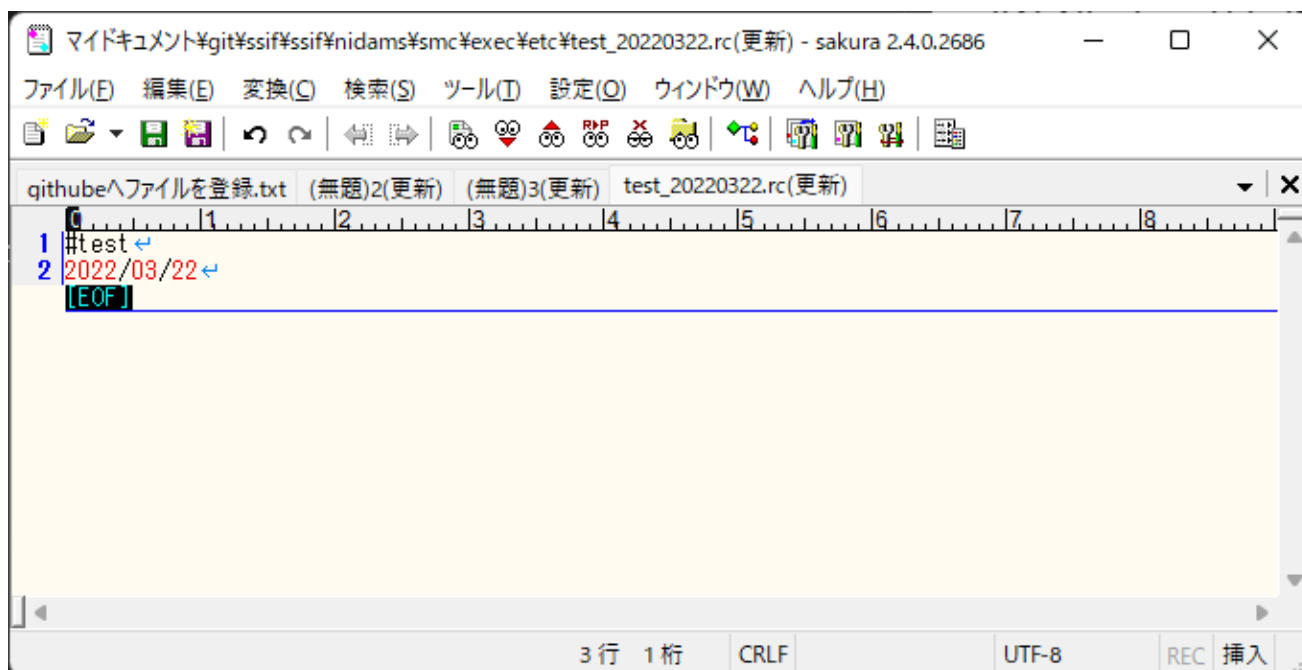
ディレクトリ

```
ssif>nidams>nidams>smc>exec>etc>
```

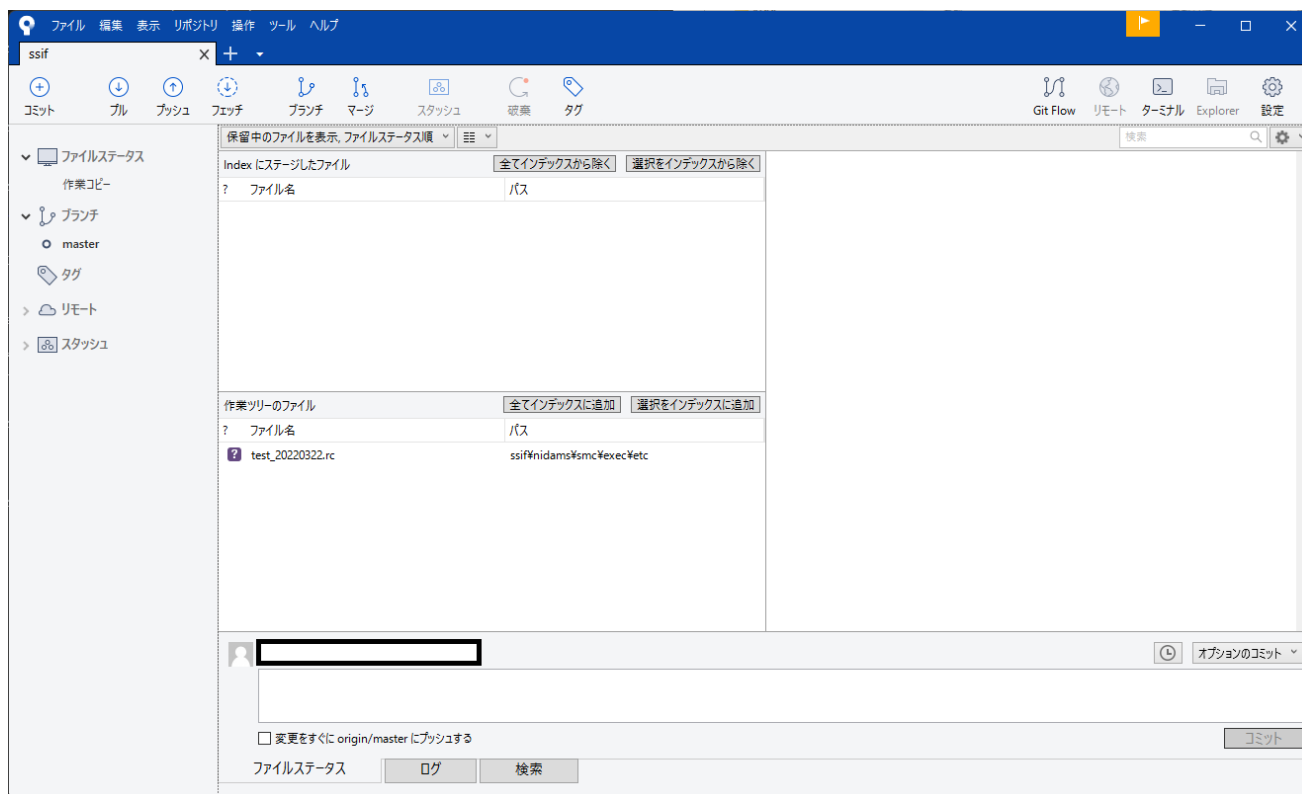
「test_20220322.rc」を新規で作成します。
(外部ファイルを配置も可能です。)



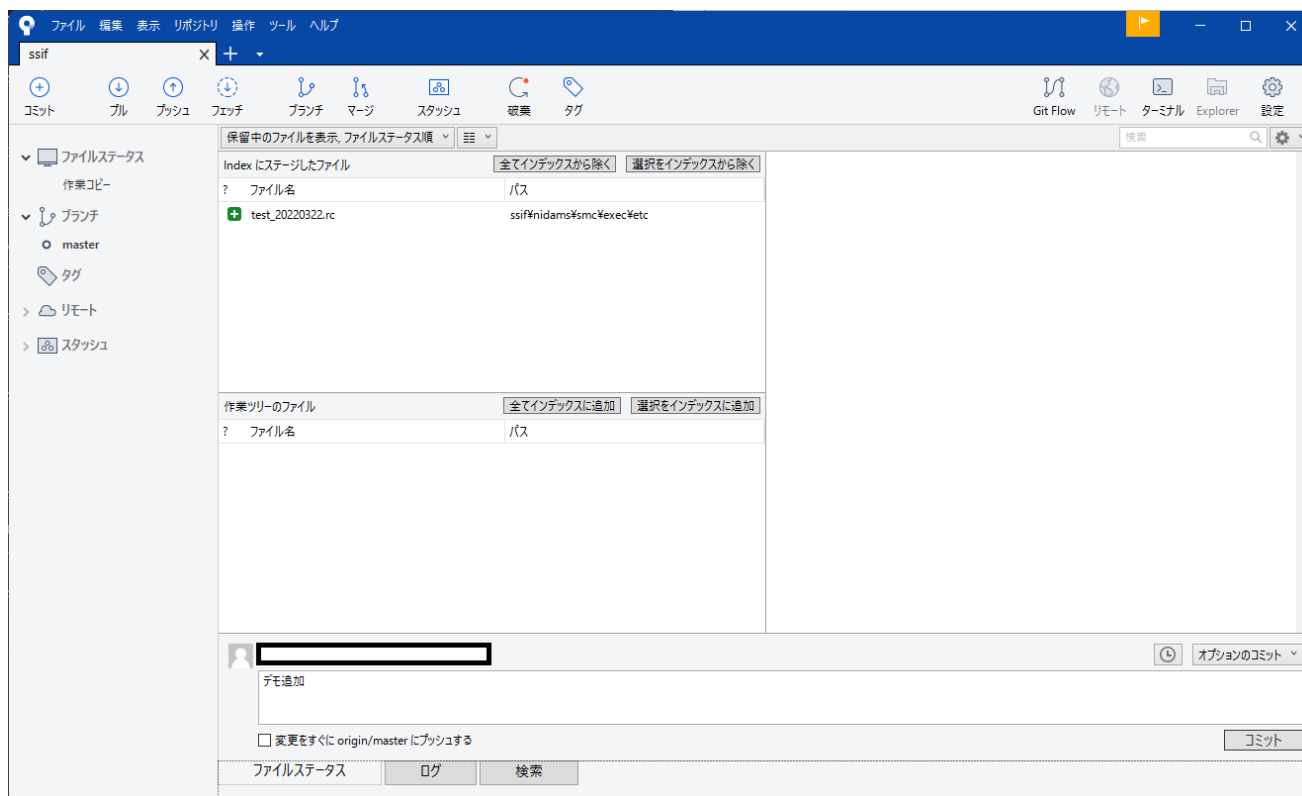
下記の内容で保存します。



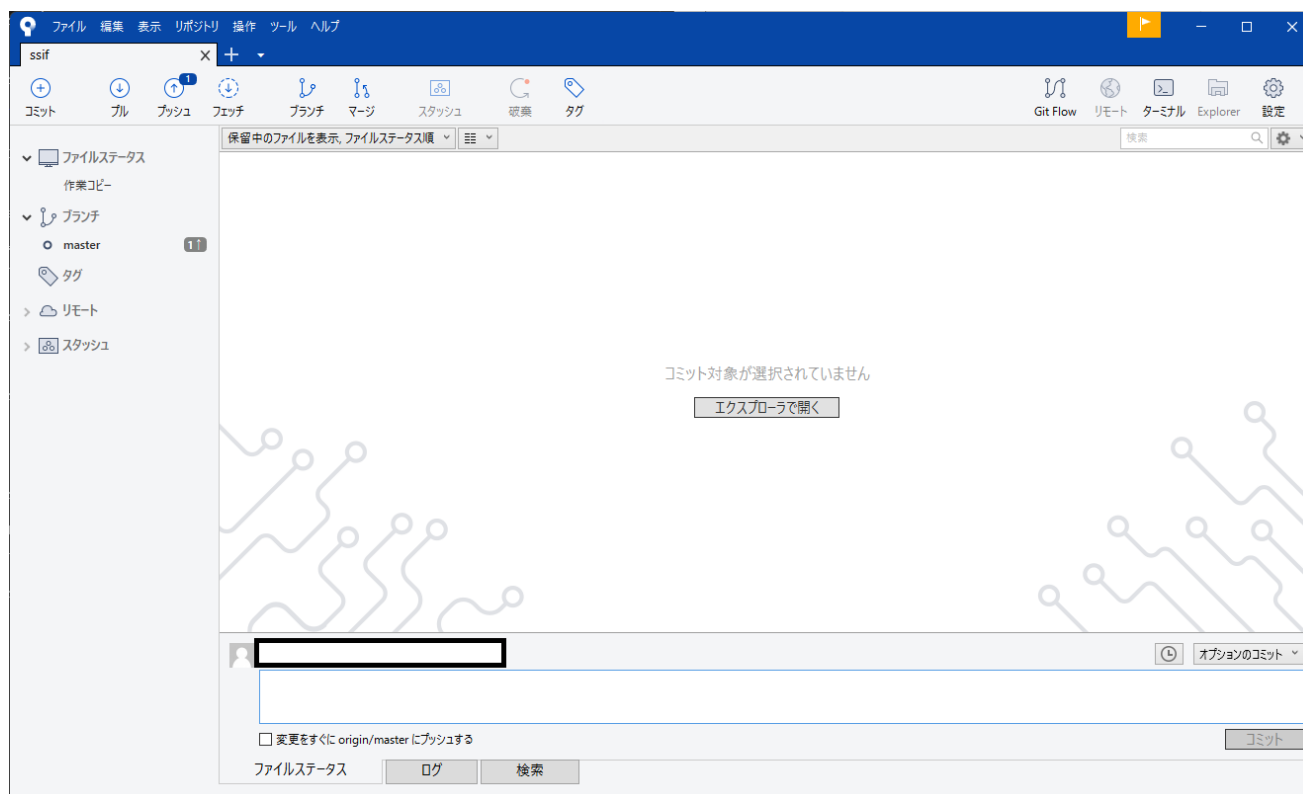
SourceTreeの作業ツリーに追加したファイルが表示されます。
追加したファイルを選択して、「選択をインデックスへ追加」をクリックします。



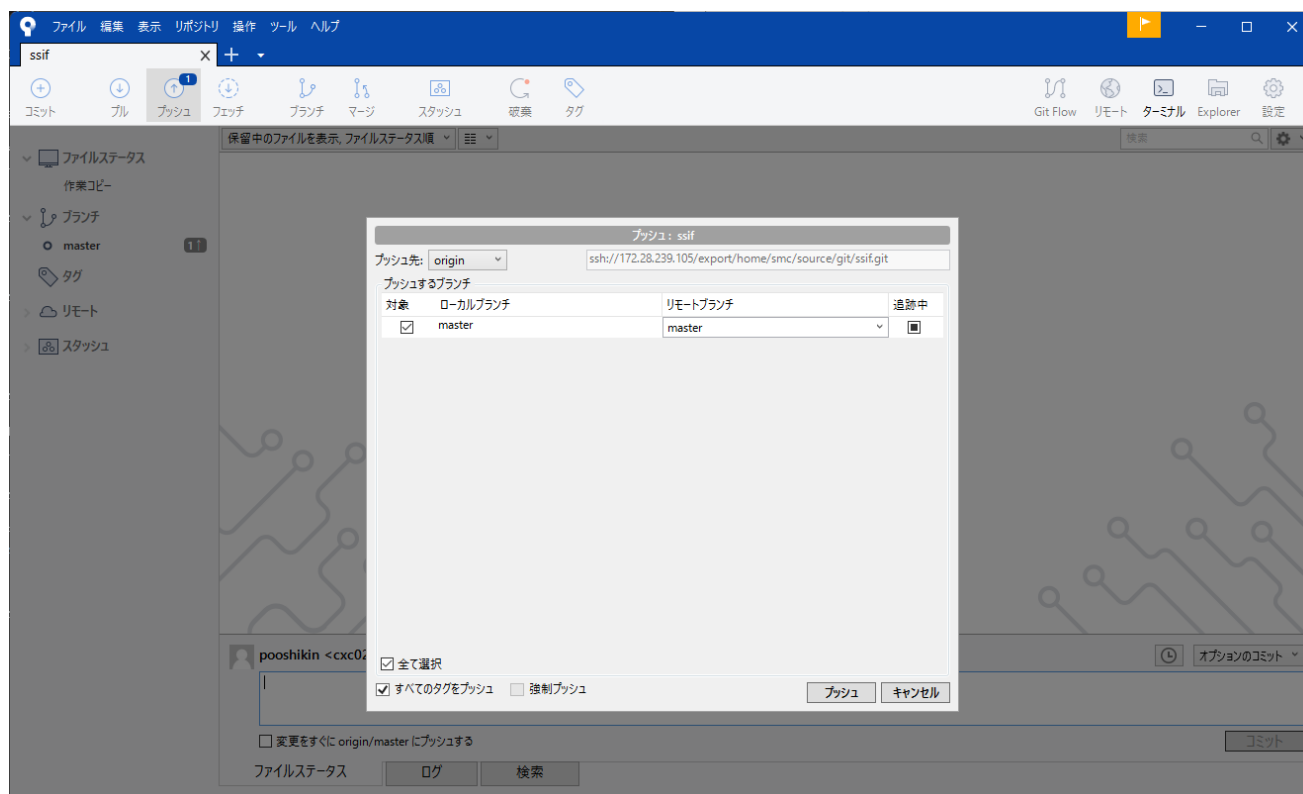
Indexにステージしたファイルが追加されます。
コメントに「デモ追加」と入力してコミットします。



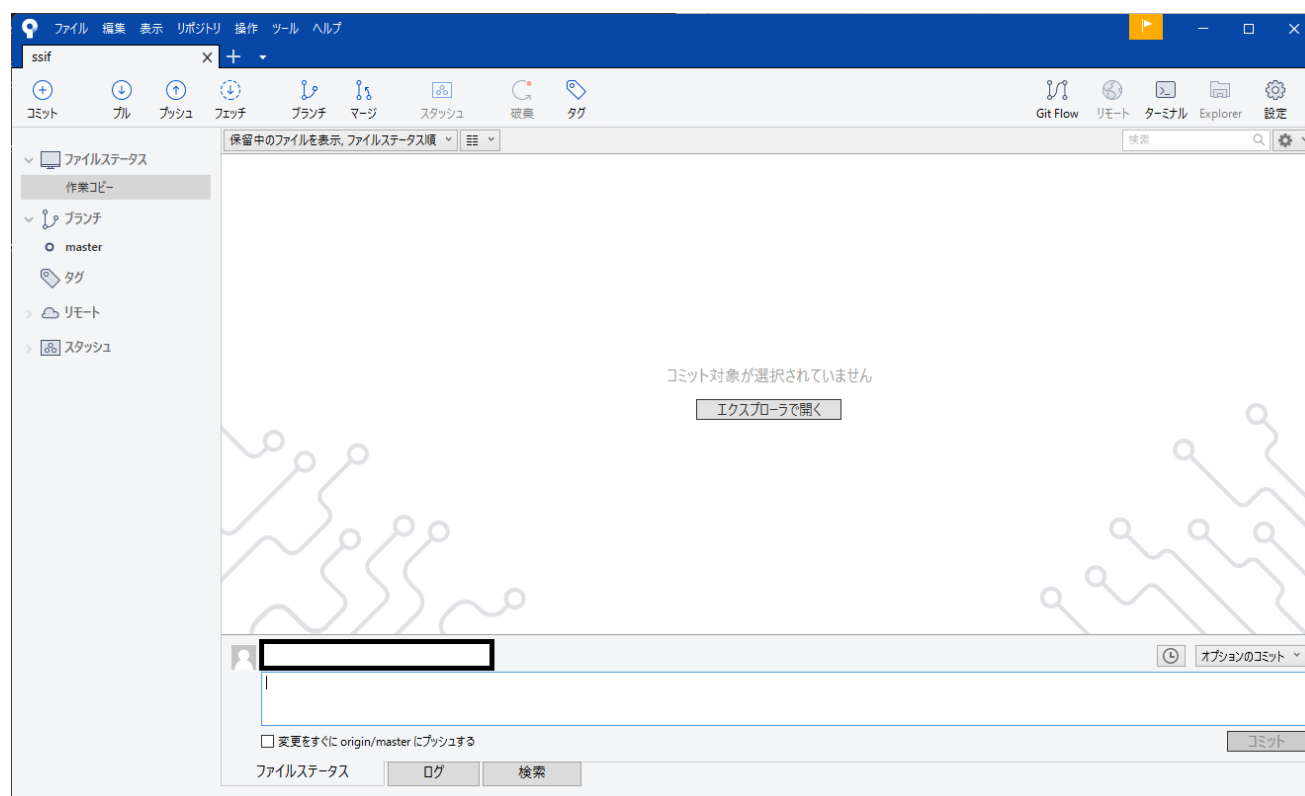
プッシュボタンに「1」が表示されています。
これは追加したファイルをssifのリモートリポジトリへpushする操作になります。
ボタンを押下します。



プッシュダイアログが表示されます。
ローカルブランチ (master) からリモートブランチ (master) へpushします。



編集集中のファイルがなくなった状態です。



3.2.4 ソース削除

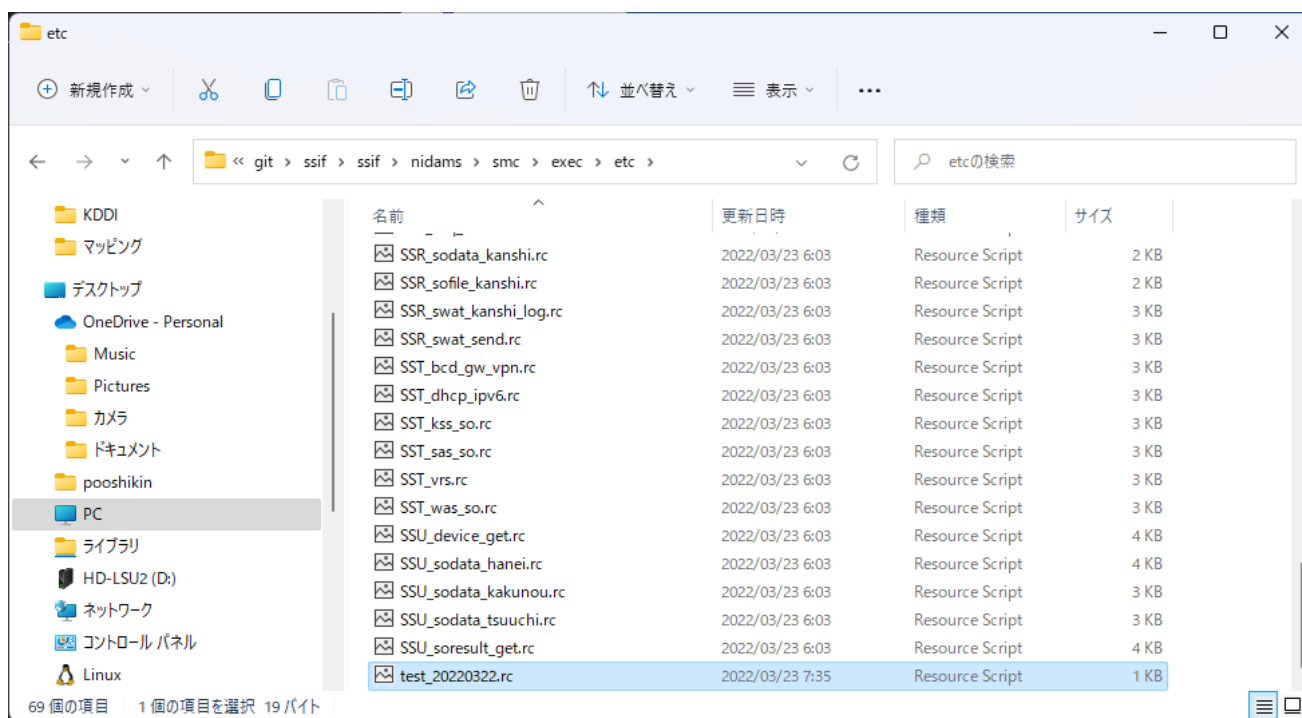
前節で追加した「test_20220322.rc」を削除します。

SourceTreeの「エクスプローラーで開く」をクリックします。

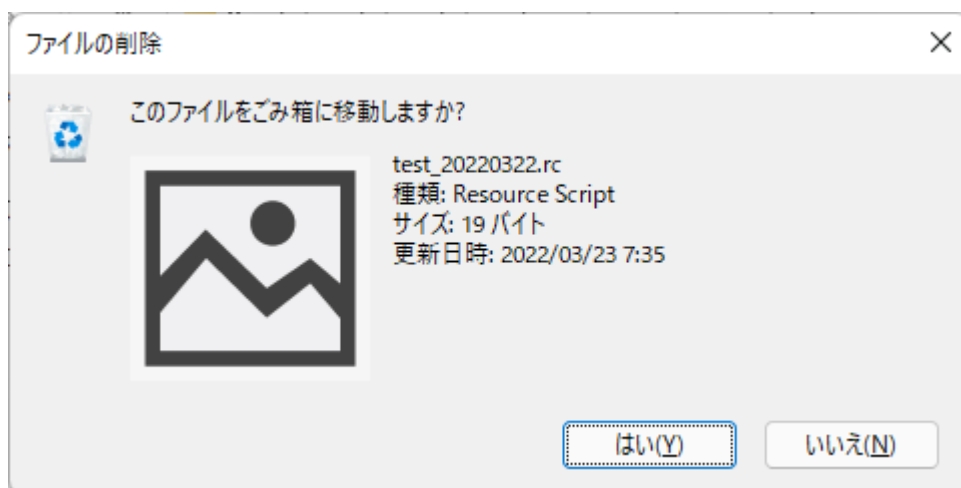
ディレクトリ

```
ssif>nidams>nidams>smc>exec>etc>
```

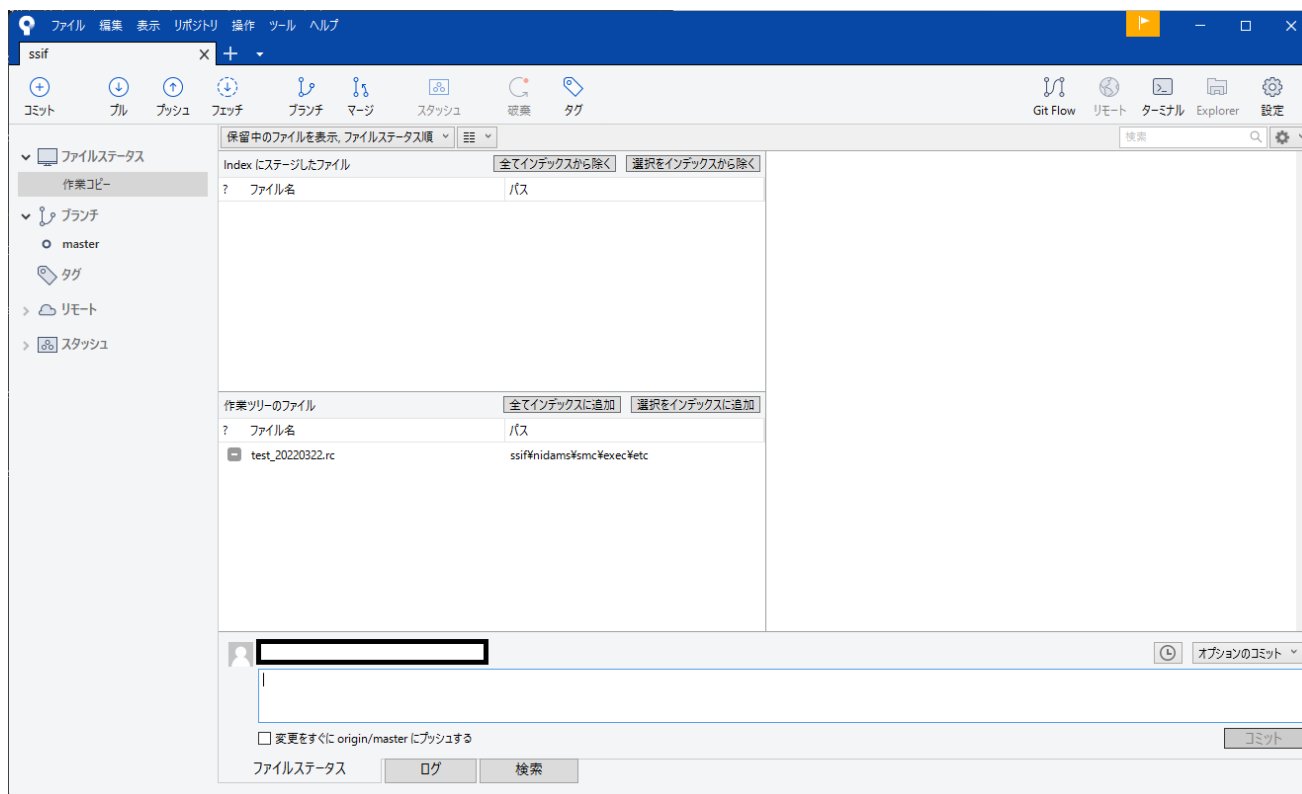
deleteキーで削除します。



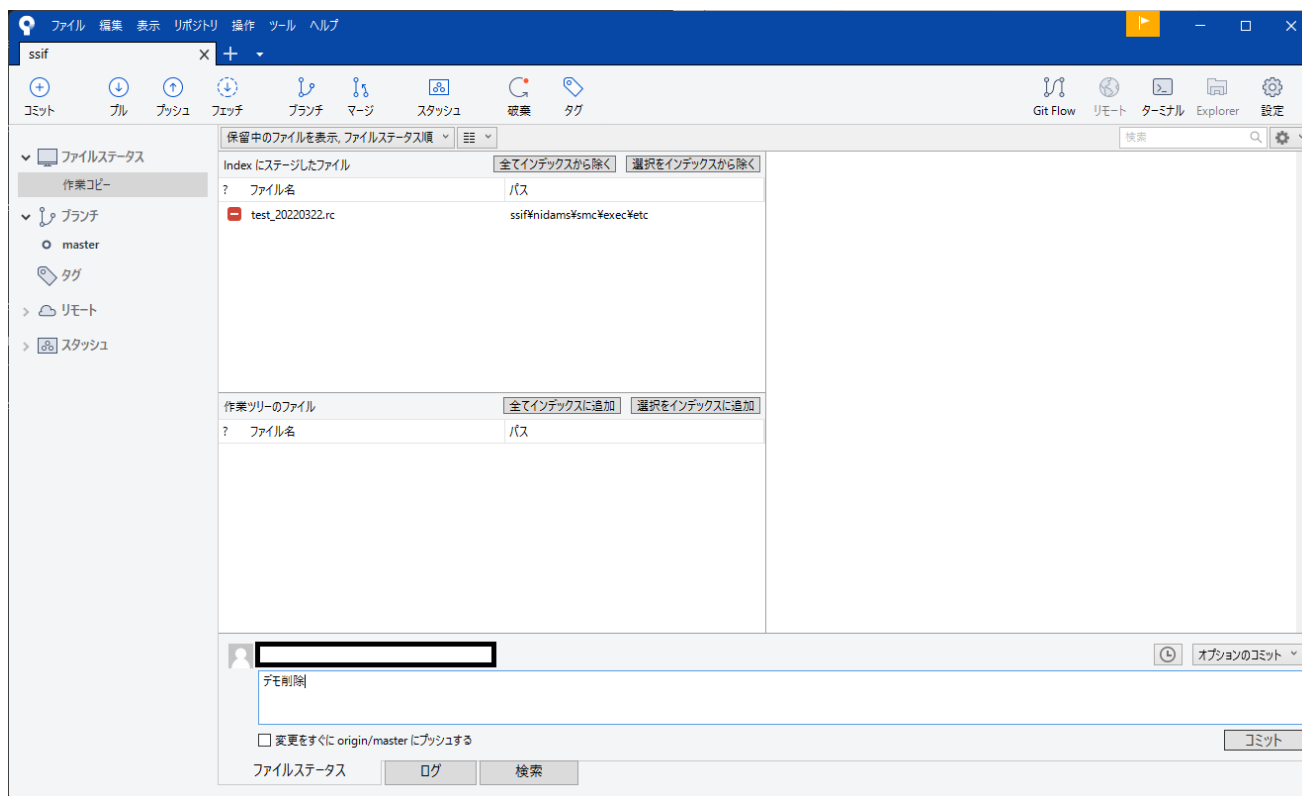
はいで削除します。



SourceTreeの作業ツリーに削除したファイルが表示されます。
削除したファイルを選択して、「選択をインデックスへ追加」をクリックします。



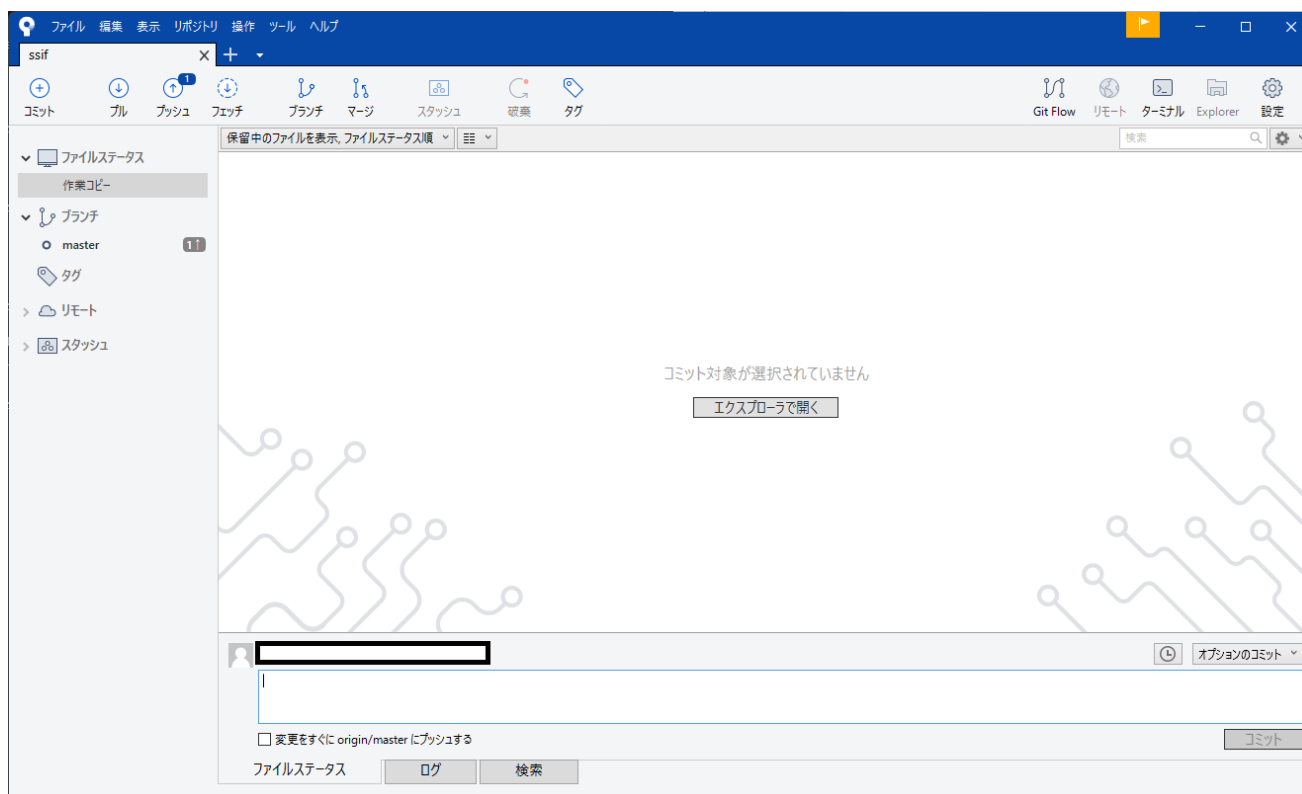
Indexにステージしたファイルが追加されます。
コメントに「デモ削除」と入力してコミットします。



プッシュボタンに「1」が表示されています。

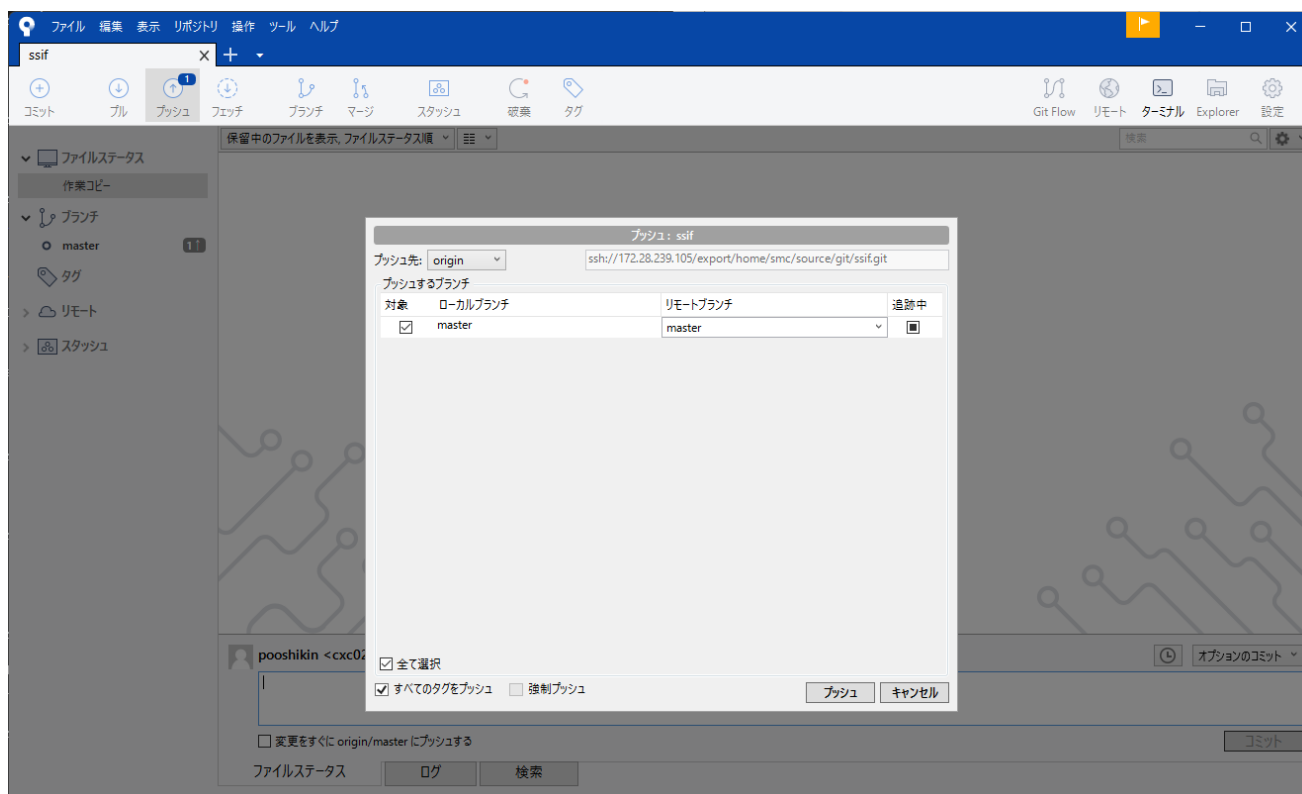
これは削除したファイルをssifのリモートリポジトリへpushする操作になります。

ボタンを押下します。



プッシュダイアログが表示されます。

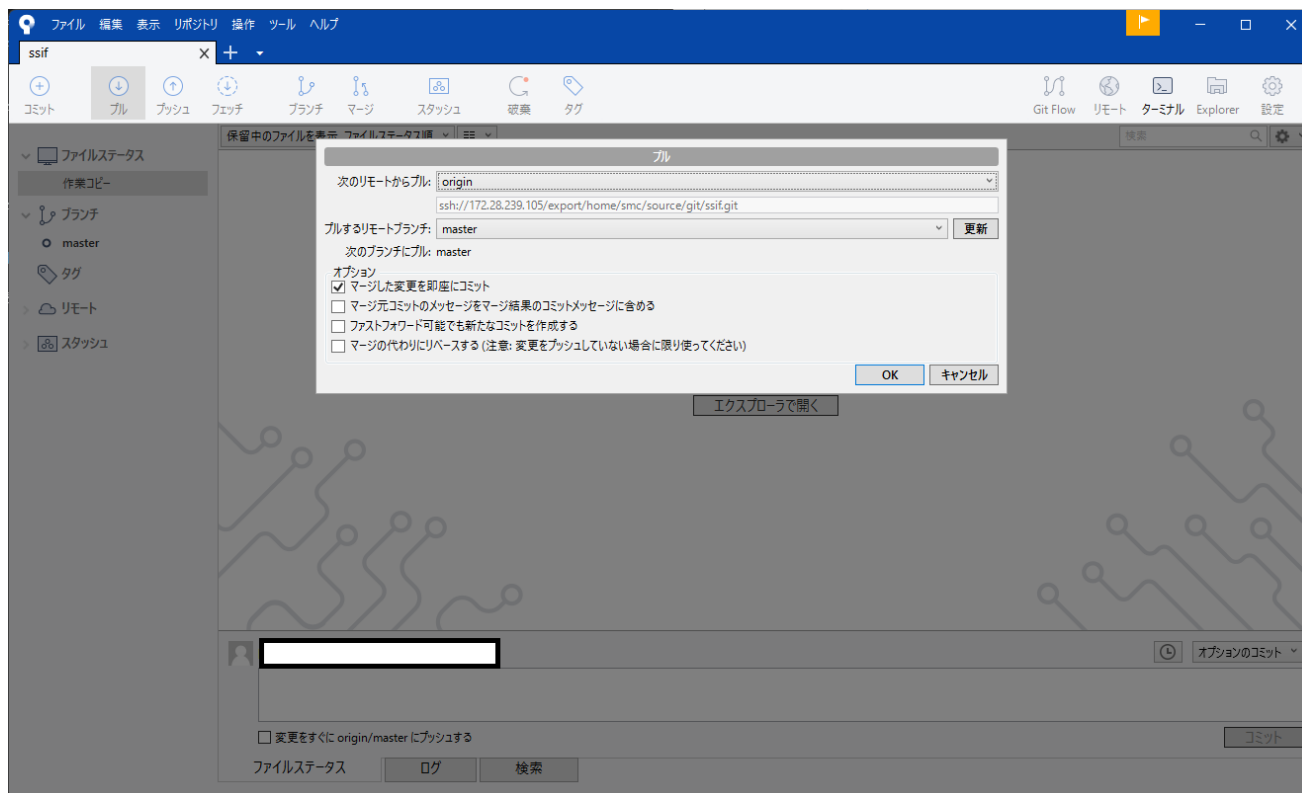
ローカルブランチ (master) からリモートブランチ (master) へpushします。



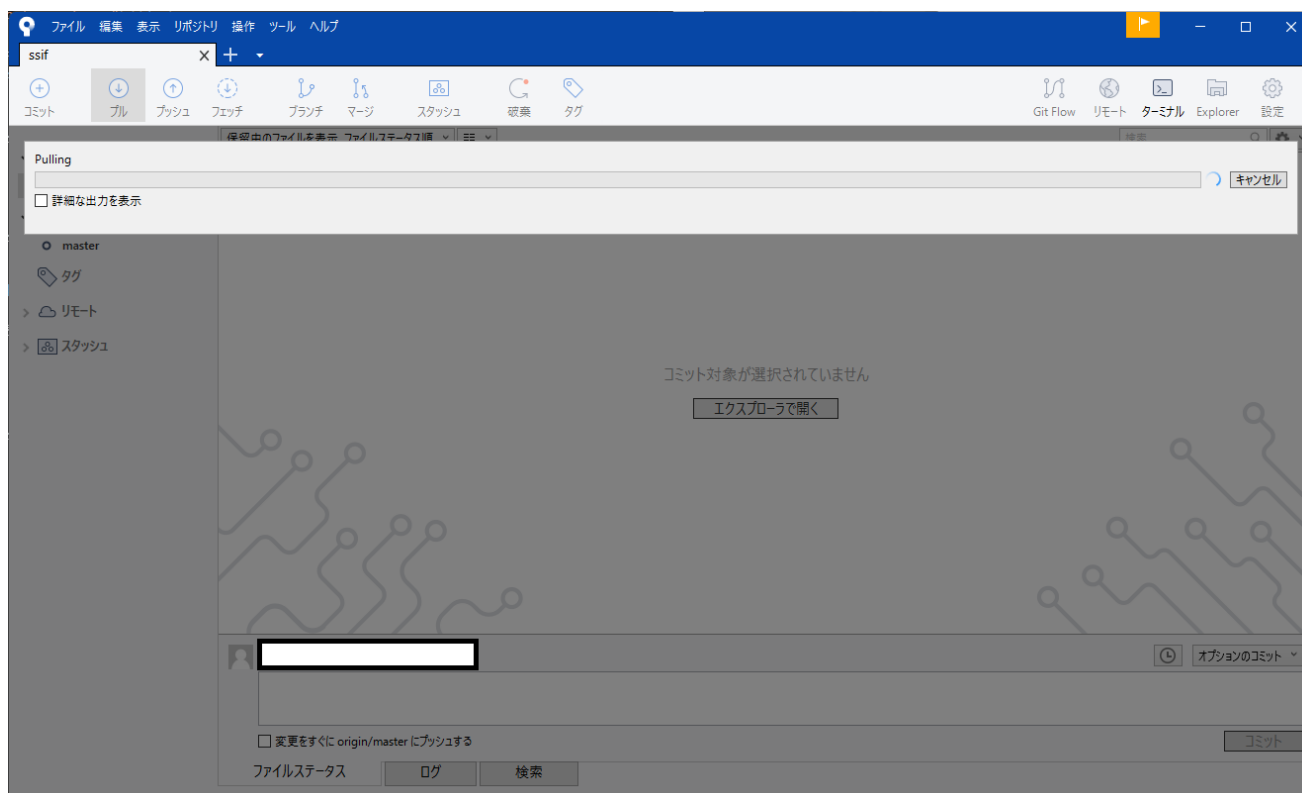
3.2.5 ソース最新化

リモートリポジトリは複数のユーザーで更新されるため、定期的に最新化する必要があります。
SorceTreeでssifのタブが表示されている状態で、「pullボタン」を押下することで、サーバーのリモートリポジトリの最新の状態を取得できます。

OKを押下します。



pullの状態です。



4. その他

以上でNIIC(DA)のGitの基本操作手引きになります。
サーバに登録されているマスターが別紙にて提供しています。

[NIIC(DA)_Gitマスター_20220418-01C.xlsx]

参照の上、ローカルにGitのソース環境を構築してください。

設計チーム並びに開発担当会社は、ソースの編集と実行モジュールなどをGitへ登録後、リリース連絡票を起票し、構成管理宛にリリース連絡票を送付ください。

構成管理はリリース連絡票を元にGitリポジトリからリリース物件を取得し物件を作成します。

※物件作成ツールについては別途説明とツールを提供いたします。