

Domande simili:



[Come vedere una pagina HTML su Github come una normale pagina HTML renderizzata per vedere l'anteprima nel browser, senza scaricare?](#)

👍 355



[Ospitare un repository Maven su github](#) 👍 312

[Come aggiungere colore al file README.md di Github](#) 👍 309

```
Your branch is based on 'origin/master', but the upstream is gone.
(use "git branch --unset-upstream" to fixup)
```

[Pagine private per un repository Github privato](#)
👍 244

- [Come aggiungere l'ID di monitoraggio di Google Analytics alle pagine GitHub](#) 👍 183

Più dettagli:

LA NUOVA
FIBRA DI TIM
 FINO A 10 VOLTE
 PIÙ VELOCE,
 POTENTE, STABILE
 E SICURA, E UN
 SERVIZIO DI
 ASSISTENZA
 DEDICATA

**SEMPLICE
 MAGNIFICA**

SCOPRI

MAGNIFICA FIBRA FTTH-MG2
 CLIENTI CENTRO E SUD
 29/10/2012
 La tecnologia FTTH-MG2
 è disponibile in fase spe-
 ciale nelle zone del
 l'offerta è attivabile so-
 essere soggetto a limit
 geografiche. Verifica pri
 servizio clienti 387 o pres

— [Jatin Gam](#)

github-pages

Ann. Fibra con Internet Box e attivazione incl
25,95€/mese per 1 anno

Scopri di più

Questo avviso è una novità di Git, che appare per primo in Git 1.8.5. Le note di rilascio contengono solo un breve punto elenco al riguardo:

- "git branch -v -v" (e "git status") non ha fatto distinzione tra un ramo che non si basa su nessun altro ramo, un ramo che è sincronizzato con il suo ramo a monte e un ramo che è configurato con un monte ramo che non esiste più.

Per descrivere cosa significa, devi prima conoscere i "telecomandi", i "rami di tracciamento remoto" e come Git gestisce il "tracciamento a monte". (*I rami di localizzazione remota* sono un termine terribilmente imperfetto: ho iniziato invece a utilizzare *nomi di localizzazione remota* , che ritengo sia un leggero miglioramento. Di seguito, tuttavia, userò "filiale di localizzazione remota" per coerenza con la documentazione di Git.)

Ogni "telecomando" è semplicemente un nome, come `origin` o `octopress` in questo caso. Il loro scopo è quello di registrare cose come l'URL completo dei luoghi da cui tu `git fetch` o gli `git pull` aggiornamenti. Quando usi ¹ Git passa a quel telecomando (usando l'URL salvato) e porta il set appropriato di aggiornamenti. *Registra* anche gli aggiornamenti, utilizzando "filiali di tracciamento remoto". `git fetch remote`,

Un "ramo di tracciamento remoto" (o nome di tracciamento remoto) è semplicemente una registrazione di un nome di ramo come visto per ultimo su un "ramo" remoto. Ogni telecomando è esso stesso un repository Git, quindi ha rami. I rami su "origine" remota sono registrati nel repository locale in `remotes/origin/` . Il testo che ha mostrato dice che c'è un ramo di nome `source` su `origin` e rami con nome `2.1` , `linklog` e così via su `octopress` .

(Un ramo "normale" o "locale", ovviamente, è solo un nome di ramo che hai creato nel tuo repository.)

Infine, puoi impostare un ramo (locale) per "tracciare" un "ramo di tracciamento remoto". Una volta `L` impostato il ramo locale per tenere traccia del ramo di tracciamento remoto `R` , Git chiamerà il `R` suo "upstream" e ti dirà se sei "avanti" e / o "dietro" l'upstream (in termini di commit). È normale (anche raccomandabile) che il ramo locale e i rami di tracciamento remoto utilizzino lo stesso nome (tranne la parte del prefisso remoto), come `source` e `origin/source` , ma non è effettivamente necessario.

E in questo caso, ciò non accade. Hai una filiale locale che `source` traccia un ramo di localizzazione remoto `origin/master` .

Non dovresti conoscere l'esatta meccanica di *come* Git imposta un ramo locale per seguirne uno remoto, ma sono rilevanti di seguito, quindi mostrerò come funziona. Si comincia con il tuo nome filiale locale, `source` . Esistono due voci di configurazione che utilizzano questo nome, `branch.source.remote` e `branch.source.merge` . Dall'output che hai mostrato, è chiaro che entrambi sono impostati, in modo da vedere quanto segue se avessi eseguito i comandi dati:

```
$ git config --get branch.source.remote
origin
$ git config --get branch.source.merge
refs/heads/master
```

Mettendo insieme questi, ² questo dice Git che il ramo `source` tiene traccia "filiali remote-tracking", `origin/master` .

Ma ora guarda l'output di `git branch -a` , che mostra tutti i nomi delle filiali di localizzazione e monitoraggio remoto nel tuo repository. I nomi di tracciamento remoto sono elencati in `remotes/` ... e *non è presente* `remotes/origin/master` . Presumibilmente c'era, una volta, ma ora non c'è più.

Git ti sta dicendo che puoi *rimuovere* le informazioni di tracciamento con `--unset-upstream` . Questo cancellerà sia `branch.source.origin` e `branch.source.merge` , e fermerà l'avviso.

Sembra abbastanza probabile che quello che vuoi, però, sia *passare* dal tracciamento `origin/master` al tracciamento di qualcos'altro: probabilmente `origin/source` , ma forse uno dei `octopress/` nomi.

Puoi farlo con `git branch --set-upstream-to` , ad esempio ³ :

(Vedi anche [Come si fa a tracciare un ramo Git esistente come ramo remoto?](#))

Penso che il modo in cui sei arrivato qui sia che quando hai fatto per la prima volta `git clone` , la cosa da cui hai clonato aveva un ramo `master` . Hai anche avuto un ramo `master` , che è stato impostato per tenere traccia `origin/master` (questa è una configurazione standard normale per git). Questo significava che avevi `branch.master.remote` e `branch.master.merge` impostato, su `origin` e `refs/heads/master` . Ma poi il tuo `origin` telecomando ha cambiato il suo nome da `master` a `source` . Per abbinare, credo che tu abbia anche cambiato il tuo nome locale da `master` a `source` . Questo ha cambiato i *nomi* delle tue impostazioni, da `branch.master.remote` a `branch.source.remote` e da `branch.master.merge` a `branch.source.merge` ... ma ha lasciato i vecchi *valori* , quindi `branch.source.merge` ora era sbagliato.

Fu a questo punto che il collegamento "upstream" si spezzò, ma nelle versioni Git precedenti alla 1.8.5, Git non notò mai l'impostazione spezzata. Ora che hai 1.8.5, lo sta sottolineando.

Questo copre la maggior parte delle domande, ma non quella "devo ripararla". È probabile che tu stia lavorando da anni alla distruzione, facendo (ad esempio). Se continui a farlo, continuerà a aggirare il problema, quindi no, non è *necessario* risolverlo. Se lo desideri, puoi utilizzare per rimuovere l'upstream e interrompere i reclami e non avere alcuna filiale locale contrassegnata come se avesse *alcun* upstream.
`git pull remote branch git pull origin source --unset-upstream source`

Il punto di avere un monte è quello di rendere più convenienti varie operazioni. Ad esempio, `git fetch` seguito da `git merge` generalmente "farà la cosa giusta" se l'upstream è impostato correttamente, e `git status` successivamente `git fetch` ti dirà se il tuo repository corrisponde a quello dell'upstream, per quel ramo.

Se si desidera la comodità, reimpostare l'upstream.

¹ `git pull` utilizza `git fetch` , e a partire da Git 1.8.4, questo (finalmente!) Aggiorna anche le informazioni sul "ramo di monitoraggio remoto". Nelle versioni precedenti di Git, gli aggiornamenti non venivano registrati nelle filiali di tracciamento remoto con `git pull` , solo con `git fetch` . Poiché il tuo Git deve essere almeno la versione 1.8.5, questo non è un problema per te.

² Bene, questo più una linea di configurazione sto deliberatamente ignorando che si trova sotto `remote.origin.fetch` . Git deve mappare il nome "unisci" per capire che è il nome locale completo per il ramo remoto `refs/remotes/origin/master` . La mappatura funziona quasi sempre così, quindi è prevedibile che `master` vada a `origin/master` .

³ Oppure, con `git config` . Se vuoi solo impostare l'upstream `origin/source` sull'unica parte che deve cambiare è `branch.source.merge` , e `git config branch.source.merge refs/heads/source` lo faresti. Ma `--set-upstream-to` dice *quello* che vuoi fare, piuttosto che farti andare, fallo manualmente, quindi è un "modo migliore".

— [Torek](#)
[🔗 fonte](#)

3 +1 per "Se vuoi, puoi usare `--unset-upstream` per avere il ramo locale **contrassegnato** come non avendo alcun upstream affatto."
— [BeatriceThalo](#).

157
👍

La risposta di torek è probabilmente perfetta, ma volevo solo che il disco menzionasse un altro caso diverso da quello descritto nella domanda originale, ma potrebbe apparire lo stesso errore (in quanto potrebbe aiutare gli altri con problemi simili):



Ho creato un repository vuoto (nuovo) utilizzando `git init --bare` uno dei miei server. Poi l'ho `git clone` fatto in un'area di lavoro locale sul mio PC.

Dopo aver eseguito il commit di una versione singola sul repository locale, ho ricevuto l'errore dopo aver chiamato `git status` .


Seguendo la risposta di Torek, capisco che quello che è successo è che il primo commit sul repository della directory di lavoro locale ha creato il ramo "master". Ma sul

Dopo l'esecuzione `git push origin master` dal repository locale, il repository remoto aveva finalmente un ramo master. Ciò ha impedito la visualizzazione dell'errore.

Quindi, per concludere, è possibile che si verifichi un errore del genere per un nuovo repository remoto nuovo con zero commit poiché non ha una diramazione, incluso "master".

— [ElazarR](#)
[🔗 fonte](#)


6



Questo è attualmente nr 1 risultato per questo messaggio di errore `On branch source Your branch is based on 'origin/master', but the upstream is gone.` (use "`git branch --unset-upstream`" to fixup) e, sebbene soggettivo, è più probabile che sia causato dalla clonazione di un repository vuoto così eccezionale da avere una risposta alternativa qui.

— [Bella,](#)


2



Penso che sia per questo che è sempre una buona idea inizializzare il tuo nuovo repository con il file `README.md`, anche se si tratta di un file vuoto


— [Ahmed Hussein](#)

8



Questo potrebbe risolvere il tuo problema.

dopo aver apportato le modifiche è possibile eseguire il commit e quindi




```
git remote add origin https://(address of your repo) it can be https or ssh
then
git push -u origin master
```

spero che funzioni per te.

Grazie

— [Ajay Kotnala](#)
[🔗 fonte](#)


1



Questo aiuterebbe davvero - il motivo è dettagliato nella mia risposta (in breve - questo crea il ramo master mancante nel repository remoto).


— [ElazarR,](#)

7



Per me, si `.git/refs/origin/master` era corrotto.


Ho fatto quanto segue, che ha risolto il problema per me.




```
rm .git/refs/remotes/origin/master
git fetch
git branch --set-upstream-to=origin/master
```

— [smremde](#)
[🔗 fonte](#)

0



In realtà torek ti ha già detto come utilizzare gli strumenti molto meglio di quanto sarei in grado di fare. Tuttavia, in questo caso penso che sia importante sottolineare qualcosa di peculiare se segui le linee guida su <http://octopress.org/docs/deploying/github/> . Vale a dire, avrai **più repository github** nella tua configurazione. Prima di tutto quello con tutto il codice sorgente per il tuo sito web in dire la directory `$WEBSITE` , e poi quello con solo i file generati statici che risiedono in `$WEBSITE/_deploy` . La cosa divertente dell'installazione è che c'è un `.gitignore` file nella `$WEBSITE` directory in modo che questa installazione funzioni davvero.



▼ Basta introduzione. In questo caso l'errore potrebbe anche provenire dal repository in `_deploy` .

```
cd _deploy

git branch -a
* master
remotes/origin/master
remotes/origin/source
```

In .git/config vi normalmente bisogno di trovare qualcosa di simile a questo:

```
[core]
  repositoryformatversion = 0
  filemode = true
  bare = false
  logallrefupdates = true
[remote "origin"]
  url = git@github.com:yourname/yourname.github.io.git
  fetch = +refs/heads/*:refs/remotes/origin/*
[branch "master"]
  remote = origin
  merge = refs/heads/master
```

Ma nel tuo caso il branch master non ha un telecomando.

```
[core]
  repositoryformatversion = 0
  filemode = true
  bare = false
  logallrefupdates = true
[remote "origin"]
  url = git@github.com:yourname/yourname.github.io.git
  fetch = +refs/heads/*:refs/remotes/origin/*
```

Che puoi risolvere:

```
cd _deploy
git branch --set-upstream-to=origin/master
```

Quindi, tutto è come ti ha detto Torek, ma potrebbe essere importante sottolineare che questo potrebbe riguardare la _deploy directory piuttosto che la radice del tuo sito web.

PS: Potrebbe valere la pena usare una shell come zsh con un git plugin per non essere morso da questa cosa in futuro. Mostrerò immediatamente che _deploy riguarda un repository diverso.

— [Anne van Rossum](#)

[🔗 fonte](#)

0

👍

Ho avuto questa domanda due volte, ed è stata sempre causata dalla corruzione del file cache git nel mio ramo locale. L'ho corretto scrivendo l'hash di commit mancante in quel file. Ho ottenuto il giusto hash di commit dal server ed eseguito localmente il seguente comando:

🔄 A

```
cat .git/refs/remotes/origin/feature/mybranch \
echo 1edf9668426de67ab764af138a98342787dc87fe \
>> .git/refs/remotes/origin/feature/mybranch
```

— [user1106400](#)

[🔗 fonte](#)

0

👍

Problema: la tua filiale si basa su 'origin / master', ma l'upstream è sparito.

Soluzione: git branch --unset-upstream

🔄 A

— [Reecha Soni](#)

[🔗 fonte](#)

dovrebbe fare qualcosa al riguardo. Per favore, rivolgiti a questo nella tua risposta.

— [cronoik](#),

0

cancella il tuo ramo locale seguendo il comando

```
git branch -d branch_name
```

potresti anche fare

```
git branch -D branch_name
```

che fondamentalmente forza una cancellazione (anche se locale non unito alla fonte)

— [Pravin](#)
[🔗 fonte](#)

Utilizzando il nostro sito, riconosci di aver letto e compreso le nostre [Informativa sui cookie](#) e [Informativa sulla privacy](#).

Licensed under [cc by-sa 3.0](#) with attribution required.