What is the ‘fatal: remote origin already exists’ error?

fatal: remote origin already exists is a common Git error that occurs when you clone a repository from GitHub, or an external remote repository, into your local machine and then try to update the pointing origin URL to your own repository.

In the context of Kubernetes, the error can occur when you configure orchestrations to include Git repositories. For example, by using: git remote add origin [url].gits

fatal: remote origin already exists is caused by the cloned repository already having a URL configured. Specifically, a URL that leads to the original profile where the repository source is.

What is a remote origin in Git?

remote origin, as the name implies, is the place where code is stored remotely. It is the centralized server or zone where everyone pushes code to and pulls code from.

Remote repositories are versions of your project hosted on Git-compatible platforms such as GitHub, Bitbucket, GitLab, and Assembla. origin is the standard and generic handle that is used to associate the host site’s URL.

For example, you can have an alternative remote URL called dev, which then becomes the handle for a separate repository but for the same code.

When you run git remote -v, you will get a list of handles and associated URLs. So if you have different handlers for the same remote, the console output could look something like this:

D:\GitHub\git remote -v

origin https://github.com/prod\_repo/projectx.git (fetch)

origin https://github.com/prod\_repo/projectx.git (push)

dev https://github.com/dev\_repo/projectx.git (fetch)

dev https://github.com/dev\_repo/projectx.git (push)

This means that you can run the following command: git push dev master

The changes made will get pushed up to the master branch at the URL associated with dev and not origin.

Resolving ‘fatal: remote origin already exists’

For most development environments, origin is the default handler used. Here are 3 ways to resolve fatal: remote origin already exists.

1. Remove the Existing Remote

remote refers to the hosted repository. origin is the pointer to where that remote is. Most of the time, origin is the only pointer there is on a local repository.

If you want to change the pointing URL attached to origin, you can remove the existing origin and then add it back in again with the correct URL.  
To remove your handler, use the remove command on remote, followed by the handler name – which, in our case, is origin.  
Here is an example: git remote remove origin

To check that handler is deleted properly, run the following: git remote -v

You will either get an empty list, or you will get a list of remote handlers that are currently attached to the project with origin removed from the list.  
Now you can run git remote add origin [url].git without encountering the fatal: remote origin already exists error.

2. Update the Existing Remote’s URL

You are not always required to remove the origin handler from remote. An alternative way to solve fatal: remote origin already exists is to update the handler’s pointing URL.

To do this, you can use the set-url command, followed by the handler name (which is origin in our case) and the new URL.

Here is the syntax for updating an existing origin URL: git remote set-url origin [new-url]

Once this is completed, you can now push and pull code from the newly configured Git repository location.

3. Rename the Existing Remote

Alternatively, you can rename origin to something else. This means that instead of deleting the handler’s pointing URL to make room for the new one, you can rename it and keep the original details.

To do this, use the rename command on: remote.

For example, if you want to rename origin to dev, you can use the following command: git remote rename origin dev

Now when you run git remote -v, you will get dev as the handler instead of origin.

D:\GitHub\[some-repo]git remote -v

dev https://github.com/some\_repo/projectx.git (fetch)

dev https://github.com/some\_repo/projectx.git (push)

This will give you room to add a new origin to the list of attached handlers. So when you run git remote add origin [url].git, you will no longer get the fatal: remote origin already exists error prompt.

How to prevent ‘fatal: remote origin already exists’

To prevent fatal: remote origin already exists error from occurring, you can check if the origin handler already exists. If it does not, running the git add remote origin command should not produce this issue.

The most important thing to note here is that origin is only a handler’s short name. It is a reference to the URL, which is where the actual remote repository is hosted.

The handler origin just happens to be the standardized default. This is what makes fatal: remote origin already exists so common. The error itself can occur against any handler, provided that it has the same placeholder name.

To check if origin even exists, run git remote -v to get a list of current remote handlers and the associated URLs.

If origin exists, you can do one of the following:

* remove origin from the remote list via remove command, like so: git remote remove origin
* update origin pointing URL with set-url, like so:git remote set-url origin [new-url]
* rename the existing origin handler to something else via rename command: git remote rename origin [new-name]