

`git push` 命令用于将本地分支的更新，推送到远程主机。它的格式与 `git pull` 命令相似。

```
$ git push <远程主机名> <本地分支名>:<远程分支名>
```

使用语法

```
git push [--all | --mirror | --tags] [--follow-tags] [--atomic] [-n | --dry-run] [--receive-pack=<git-receive-pack>]
      [--repo=<repository>] [-f | --force] [-d | --delete] [--prune] [-v | --verbose]
      [-u | --set-upstream] [--push-option=<string>]
      [--[no-]signed|--sign=(true|false|if-asked)]
      [--force-with-lease[=<refname>[:<expect>]]]
      [--no-verify] [<repository> [<refspec>...]]
```

描述

使用本地引用更新远程引用，同时发送完成给定引用所需的对象。可以在每次推入存储库时，通过在那里设置挂钩触发一些事件。

当命令行不指定使用 `<repository>` 参数推送的位置时，将查询当前分支的 `branch.*.remote` 配置以确定要在哪里推送。如果配置丢失，则默认为 `origin`。

示例

以下是一些示例 -

```
$ git push origin master
```

上面命令表示，将本地的 `master` 分支推送到 `origin` 主机的 `master` 分支。如果 `master` 不存在，则会被新建。

如果省略本地分支名，则表示删除指定的远程分支，因为这等同于推送一个空的本地分支到远程分支。

```
$ git push origin :master
```

等同于

```
$ git push origin --delete master
```

上面命令表示删除 `origin` 主机的 `master` 分支。如果当前分支与远程分支之间存在追踪关系，则本地分支和远程分支都可以省略。

```
$ git push origin
```

上面命令表示，将当前分支推送到 `origin` 主机的对应分支。如果当前分支只有一个追踪分支，那么主机名都可以省略。

```
$ git push
```

如果当前分支与多个主机存在追踪关系，则可以使用 `-u` 选项指定一个默认主机，这样后面就可以不加任何参数使用 `git push`。

```
$ git push -u origin master
```

上面命令将本地的 `master` 分支推送到 `origin` 主机，同时指定 `origin` 为默认主机，后面就可以不加任何参数使用 `git push` 了。

不带任何参数的 `git push`，默认只推送当前分支，这叫做 `simple` 方式。此外，还有一种 `matching` 方式，会推送所有有对应的远程分支的本地分支。Git 2.0 版本之前，默认采用 `matching` 方法，现在改为默认采用 `simple` 方式。如果要修改这个设置，可以采用 `git config` 命令。

```
$ git config --global push.default matching
```

或者

```
$ git config --global push.default simple
```

还有一种情况，就是不管是否存在对应的远程分支，将本地的所有分支都推送到远程主机，这时需要使用 `--all` 选项。

```
$ git push --all origin
```

上面命令表示，将所有本地分支都推送到 `origin` 主机。

如果远程主机的版本比本地版本更新，推送时 Git 会报错，要求先在本地做 `git pull` 合并差异，然后再推送到远程主机。这时，如果你一定要推送，可以使用 `--force` 选项。

```
$ git push --force origin
```

上面命令使用 `--force` 选项，结果导致在远程主机产生一个“非直进式”的合并(non-fast-forward merge)。除非你很确定要这样做，否则应该尽量避免使用 `--force` 选项。

最后，`git push` 不会推送标签(tag)，除非使用 `-tags` 选项。

```
$ git push origin --tags
```

将当前分支推送到远程的同名的简单方法，如下 -

```
$ git push origin HEAD
```

将当前分支推送到源存储库中的远程引用匹配主机。这种形式方便推送当前分支，而不考虑其本地名称。如下 -

```
$ git push origin HEAD:master
```

其它示例

1.推送本地分支 `lbranch-1` 到新大远程分支 `rbranch-1`：

```
$ git push origin lbranch-1:refs/rbranch-1
```

2.推送 `lbranch-2` 到已有的 `rbranch-1`，用于补充 `rbranch-1`：

```
$ git checkout lbranch-2
```

```
$ git rebase rbranch-1
```

```
$ git push origin lbranch-2:refs/rbranch-1
```

3.用本地分支 **lbranch-3** 覆盖远程分支 **rbranch-1** :

```
$ git push -f origin lbranch-2:refs/rbranch-1
```

或者 -

```
$ git push origin :refs/rbranch-1 //删除远程的 rbranch-1 分支
```

```
$ git push origin lbranch-1:refs/rbranch-1
```

4.查看 **push** 的结果

```
$ gitk rbranch-1
```

5.推送 tag

```
$ git push origin tag_name
```

6.删除远程标签

```
$ git push origin :tag_name
```