Version Control

Part 2

Simple Intro

Why version control?

SupremePlotLow.m	Amos, Matt	2/7/2017	2.47 KB
SupremePlotLow.m	Amos, Matt	2/7/2017	2.45 KB
SupremePlotLow.m	Amos, Matt	2/7/2017	2.49 KB
SupremePlot.m	Amos, Matt	2/7/2017	2.43 KB
SupremePlot.m	Amos, Matt	4/4/2017	3.23 KB

FirstWaveRegister.m	Amos, Matt	1/30/2017	508 bytes
FirstWaveRegister.m	Amos, Matt	3/8/2017	542 bytes
FirstWaveRegister.m	Amos, Matt	3/8/2017	552 bytes
FirstWaveRegister.m	Amos, Matt	3/8/2017	543 bytes

Why version control?

SupremePlotLow.m			2.47 KB
SupremePlotLow.m			2.45 KB
SupremePlotLow.m	Amos, Matt	2/7/2017	2.49 KB
SupremePlot.m	Amos, Matt	2/7/2017	2.43 KB
SupremePlot.m	Amos, Matt	4/4/2017	3.23 KB

FirstWaveRegister.m		1/30/2017	508 bytes
FirstWaveRegister.m	Amos, Matt	3/8/2017	542 bytes
FirstWaveRegister.m	Amos, Matt	3/8/2017	552 bytes
FirstWaveRegister.m	Amos, Matt	3/8/2017	543 bytes

Why version control?

- Collaboration
  - Each member of a team can edit independently
- History of changes
- Create changes in parallel
- Not limited to code - great for notes, .tex …

```
commit e7e1dcc84a2d1d20e2beef7b6e134df7655beaf0
Author: Matt Amos <matty_a1@hotmail.co.uk>
Date:   Mon Mar 19 18:50:21 2018 +0000

    new funcs added to util

commit 53d66732acd323cf653d042d10a7c164a1af03a8
Author: Matt Amos <matty_a1@hotmail.co.uk>
Date:   Mon Mar 19 14:04:30 2018 +0000

    updating for gnasher

commit a573dfad5289f917dbb9ef26dbd74ed11b0cf96e
Author: Matt Amos <matty_a1@hotmail.co.uk>
Date:   Fri Mar 16 22:37:51 2018 +0000

    First Commit of Util
```

```
@@ -371,17 +377,117 @@ def plot_multi_model_mean(cubes_list, mov_ave=None):

     # For mean and standard deviation
     time = std_cube.coord('time_month')
-    fixed_dates = iris.plot._fixup_dates(time_coord, time_coord.points)
+    fixed_dates = iris.plot._fixup_dates(time, time.points)

     std_data = std_cube.data
     mean_data = mean_cube.data
-
+
     plt.plot(fixed_dates, mean_data, label='Model average')
-    plt.fill_between(fixed_dates, mean_data-std_data, mean_data+std_data, facecolor='grey', al
pha=0.4)
+    if std_dev:
+        plt.fill_between(fixed_dates, mean_data-std_data, mean_data+std_data, facecolor='grey'
, alpha=0.4)
```

Example time

- Initialise git repo

```
Matts-MacBook-Pro:maths amosm1$ git init
Initialized empty Git repository in /Users/amosm1/maths/.git/
```

- Initialise git repo

```
Matts-MacBook-Pro:maths amosm1$ git init
Initialized empty Git repository in /Users/amosm1/maths/.git/
```

- Add to the staging area

```
Matts-MacBook-Pro:maths amosm1$ git add square.py
```

- Initialise git repo

```
Matts-MacBook-Pro:maths amosm1$ git init
Initialized empty Git repository in /Users/amosm1/maths/.git/
```

- Add to the staging area

```
Matts-MacBook-Pro:maths amosm1$ git add square.py
```

- Save the changes locally

```
Matts-MacBook-Pro:maths amosm1$ git commit -m 'New Function'
[master (root-commit) e758915] New Function
 1 file changed, 1 insertion(+)
 create mode 100644 square.py
```

- Initialise git repo

```
Matts-MacBook-Pro:maths amosm1$ git init
Initialized empty Git repository in /Users/amosm1/maths/.git/
```

- Add to the staging area

```
Matts-MacBook-Pro:maths amosm1$ git add square.py
```

- Save the changes locally

```
Matts-MacBook-Pro:maths amosm1$ git commit -m 'New Function'
[master (root-commit) e758915] New Function
 1 file changed, 1 insertion(+)
 create mode 100644 square.py
```

- Check we're up to date

```
Matts-MacBook-Pro:maths amosm1$ git status
On branch master
nothing to commit, working tree clean
```

```
1123
1124  def load_and_make_seasonal_1d(in_dir, lat, ndim=4):
1125
1126      # Load all CCMI data
1127      if isinstance(lat, list):
1128          lat_0 = lat[0]
1129          lat_1 = lat[1]
1130      else:
1131          raise ValueError('Latitude must be specified as a list')
1132
1133      # Load files and group separate time files into lists
1134      files = [os.path.join(in_dir, file)
1135               for file in os.listdir(in_dir) if file[:2] != '._']
1136      con = iris.Constraint(latitude=lambda cell: lat_0 < cell < lat_1)
1137      in_cubes = iris.cube.CubeList(
1138          [cube for cube in iris.load(files, con) if cube.ndim >= ndim])
1139      cubes_list = group_models(in_cubes)
1140
1141      # Concatenate and collapse (area-weighting)
1142      fin_cubes = []
1143      for cubes in cubes_list:
1144          rem_and_eq(cubes)
1145          if len(cubes) != 1:
1146
1147              try:
1148                  out_cube = iris.cube.CubeList(cubes).concatenate_cube()
1149              except iris.exceptions.ConcatenateError:
```

```
amosm1$ git log -p -- util.py
```

```
-def load_and_make_seasonal_1d(in_dir, lat):
+def load_and_make_seasonal_1d(in_dir, lat, ndim=4):

     # Load all CCMI data
     if isinstance(lat, list):
@@ -1034,7 +1121,7 @@ def load_and_make_seasonal_1d(in_dir, lat):
     # Load files and group separate time files into lists
     files = [os.path.join(in_dir, file) for file in os.listdir(in_dir) if file[:2]!='._']
     con = iris.Constraint(latitude=lambda cell: lat_0 < cell < lat_1 )
-    in_cubes = iris.cube.CubeList([cube for cube in iris.load(files, con) if cube.ndim >= 3])
+    in_cubes = iris.cube.CubeList([cube for cube in iris.load(files, con) if cube.ndim >= ndim
])
     cubes_list = group_models(in_cubes)

     # Concatenate and collapse (area-weighting)
@@ -1078,8 +1165,14 @@ def load_and_make_seasonal_1d(in_dir, lat):
     return s_model_cubes
```

```
Matts-MacBook-Pro:analysis amosm1$ git log -p -- util.py
```

```
1125
1126        # Load all CCMI data
1127        if isinstance(lat, list):
```

```
-def load_and_make_seasonal_1d(in_dir, lat):
+def load_and_make_seasonal_1d(in_dir, lat, ndim=4):
```

```
    # Load all CCMI data
    if isinstance(lat, list):
@@ -1034,7 +1121,7 @@ def load_and_make_seasonal_1d(in_dir, lat):
    # Load files and group separate time files into lists
    files = [os.path.join(in_dir, file) for file in os.listdir(in_dir) if file[:2]!='._']
    con = iris.Co
-    in_cubes = iris.cube.CubeList([cube for cube in iris.load(files, con) if cube.ndim >= 3])
+    in_cubes = iris.cube.CubeList([cube for cube in iris.load(files, con) if cube.ndim >= ndim
])
    cubes_list = group_models(in_cubes)

    # Concatenate and collapse (area-weighting)
@@ -1078,8 +1165,14 @@ def load_and_make_seasonal_1d(in_dir, lat):
    return s_model_cubes
```

```
1146
1147                try:
1148                    out_cube = iris.cube.CubeList(cubes).concatenate_cube()
1149                except iris.exceptions.ConcatenateError:
```

amosm1$ git checkout square.py


DONT
PANIC.

GitHub

Search or jump to... / Pull requests Issues Marketplace Explore

mattramos / shared-atmos-scripts

Watch 0 Star 0 Fork 0

Code Issues 0 Pull requests 0 Projects 0 Wiki Insights Settings

Branch: master shared-atmos-scripts / CCMI-Download / Create new file Upload files Find file History

mattramos Update README.md Latest commit 0598e30 an hour ago

..

README.md	Update README.md	an hour ago
ccmi_data_retrieve.py	First Commit	7 days ago
ccmi_download.sh	First Commit	7 days ago
ccmi_filelist_retrieve.sh	First Commit	7 days ago

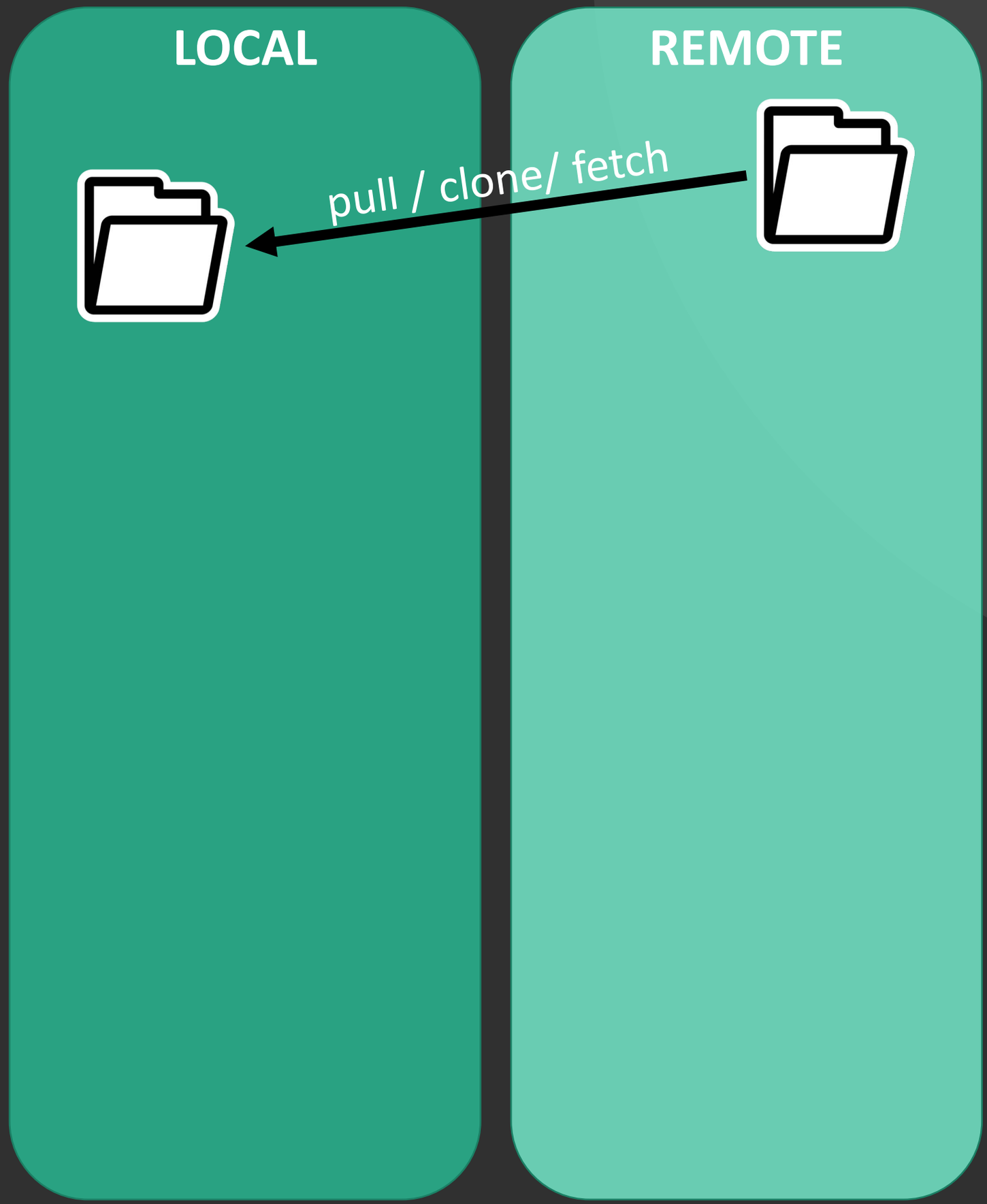
@@ -80,4 +78,4 @@ Points to note

```
80  - The first time you run, you will need to run with 'refresh' as the last argument this
81        may take a few minutes whilst the file structure is collected. It is stored so
82        this process need only be repeated infrequently
83  -- CESM models are not on BADC
```

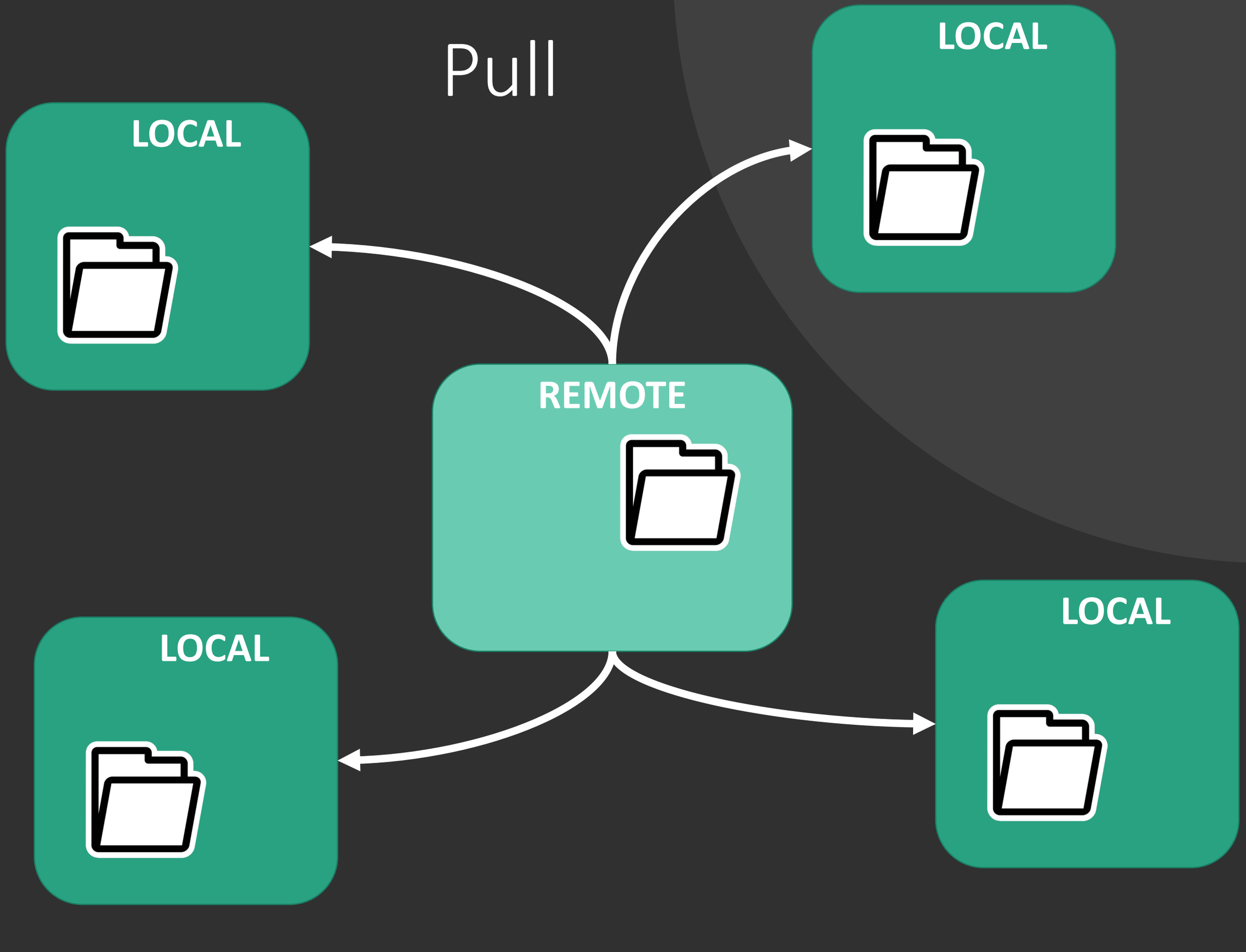
```
78  - The first time you run, you will need to run with 'refresh' as the last argument this
79        may take a few minutes whilst the file structure is collected. It is stored so
80        this process need only be repeated infrequently
81  +- CESM models are not on BADC
```

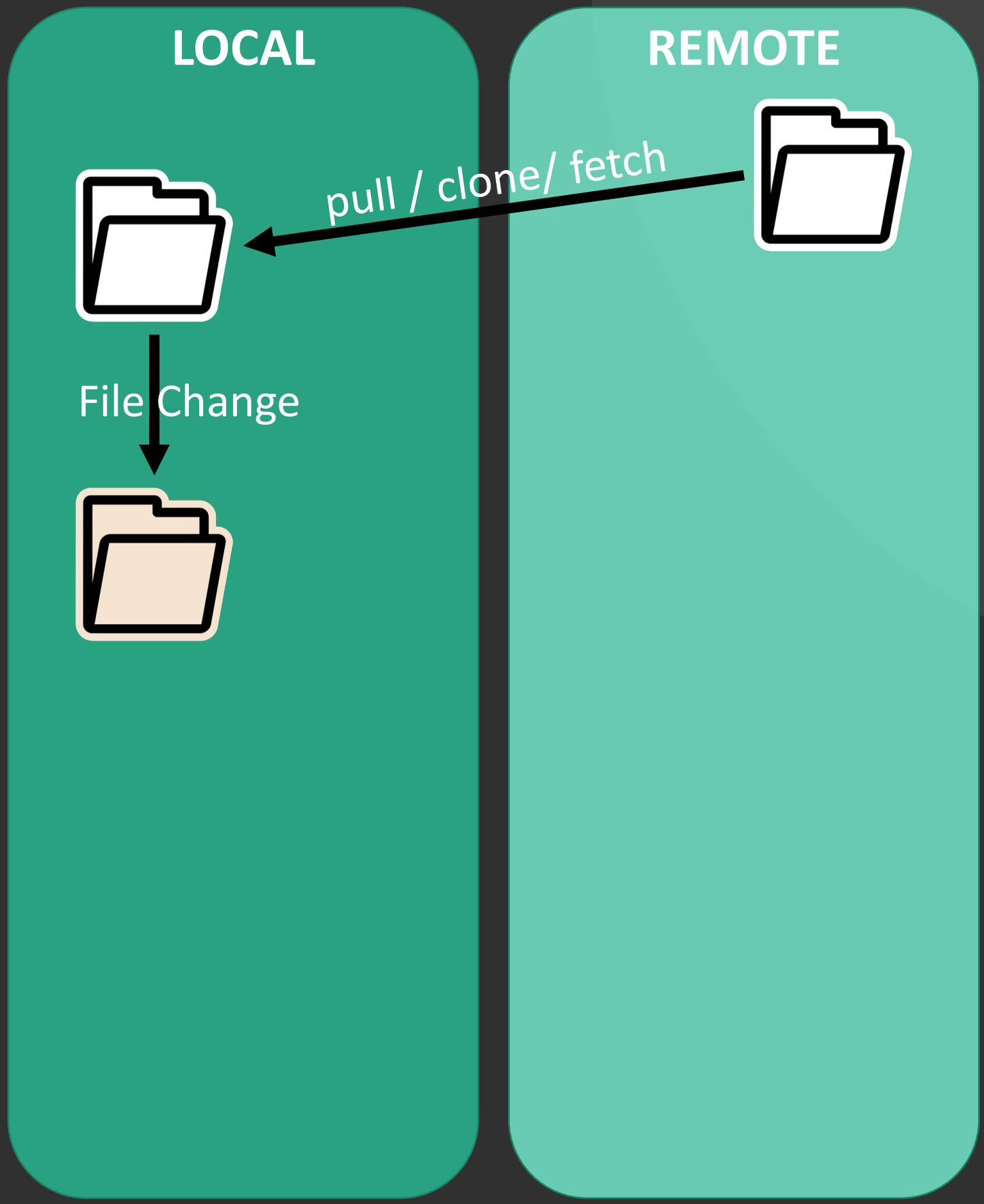
git

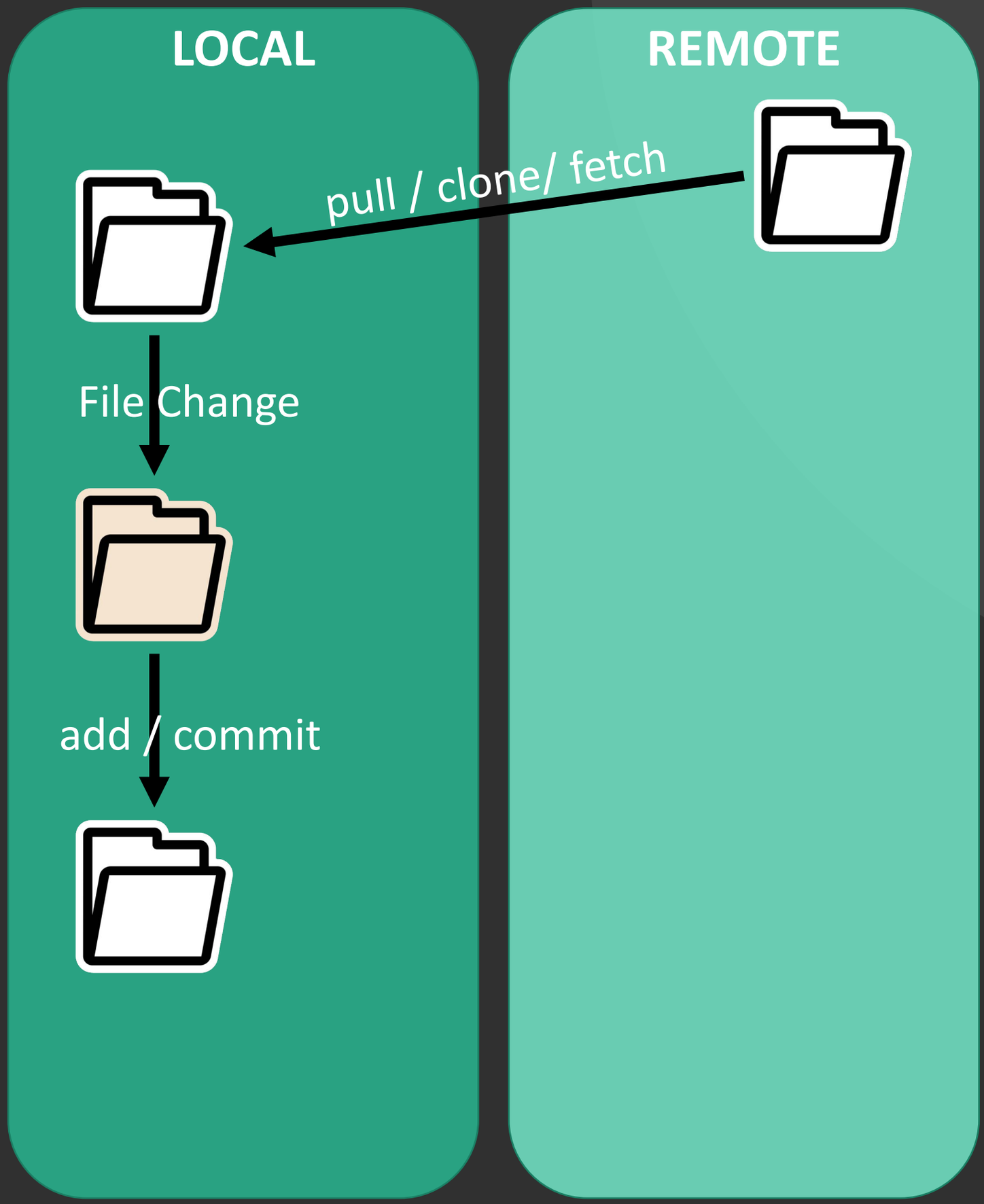
≠

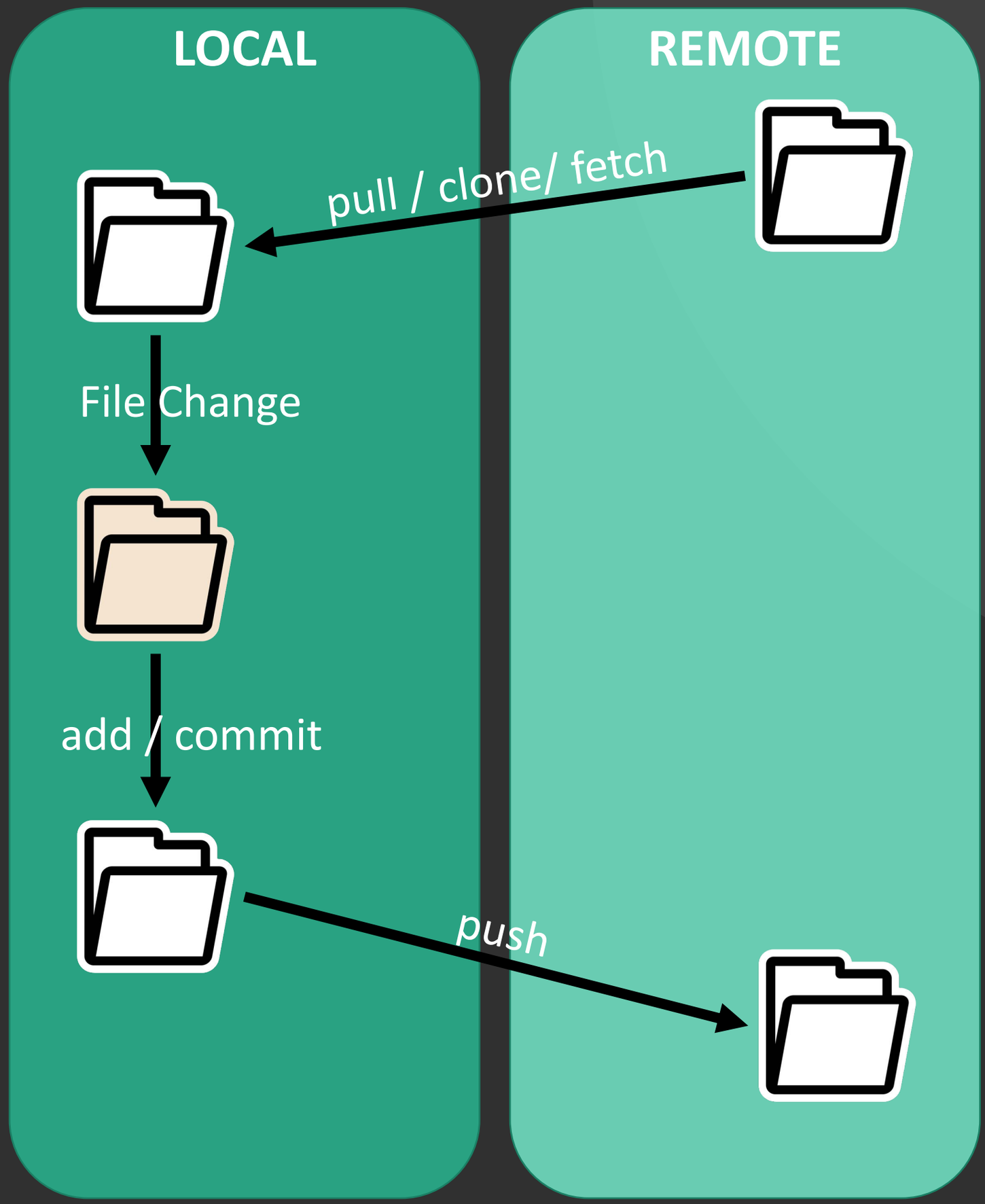

LOCAL
REMOTE
pull / clone/ fetch

Pull




LOCAL
REMOTE
pull / clone/ fetch
File Change


LOCAL
REMOTE
pull / clone/ fetch
File Change
add / commit


LOCAL
REMOTE
pull / clone/ fetch
File Change
add / commit
push

That’s pretty much it

1. Get github account download git
2. Create repo on github
3. git config --global user.name "First Last"
4. git config --global user.email "user@domain"
5. git init
6. git add (staging area)
7. git commit
8. git remote add origin https://github.com/user/repo.git
9. git push -u origin master

More?

- Find the difference between versions
  - git diff
- Restore to specific version
  - git checkout 660213b file.py
- Add branches for collaboration

Personally

- Saved me a lot of time
- Avoided that ‘OH NO!’ moment
- Sharing files
- Learnt by making it habit to use git every day

Questions??

- LEC repository for sharing code
- https://github.com/mattramos/shared-atmos-scripts/
    - Where I put scripts and functions to share
    - CCMI downloads
    - IRIS wrappers and functions

Quick Tutorial

I thought I'd share these slides on github of course and it seems like the perfect time to make a mini tutorial of the process.

Once the repo is set up these processes become a lot quicker

0. Get git and git hub

- Download git (conda install -c anaconda git, or https://git-scm.com/downloads)
- Get a github academic account https://help.github.com/articles/applying-for-an-academic-research-discount/
- Set up git
  - git config --global user.name "First Last”
  - git config --global user.email "user@domain”
  - Can also set up ssh keys https://help.github.com/articles/generating-a-new-ssh-key-and-adding-it-to-the-ssh-agent/

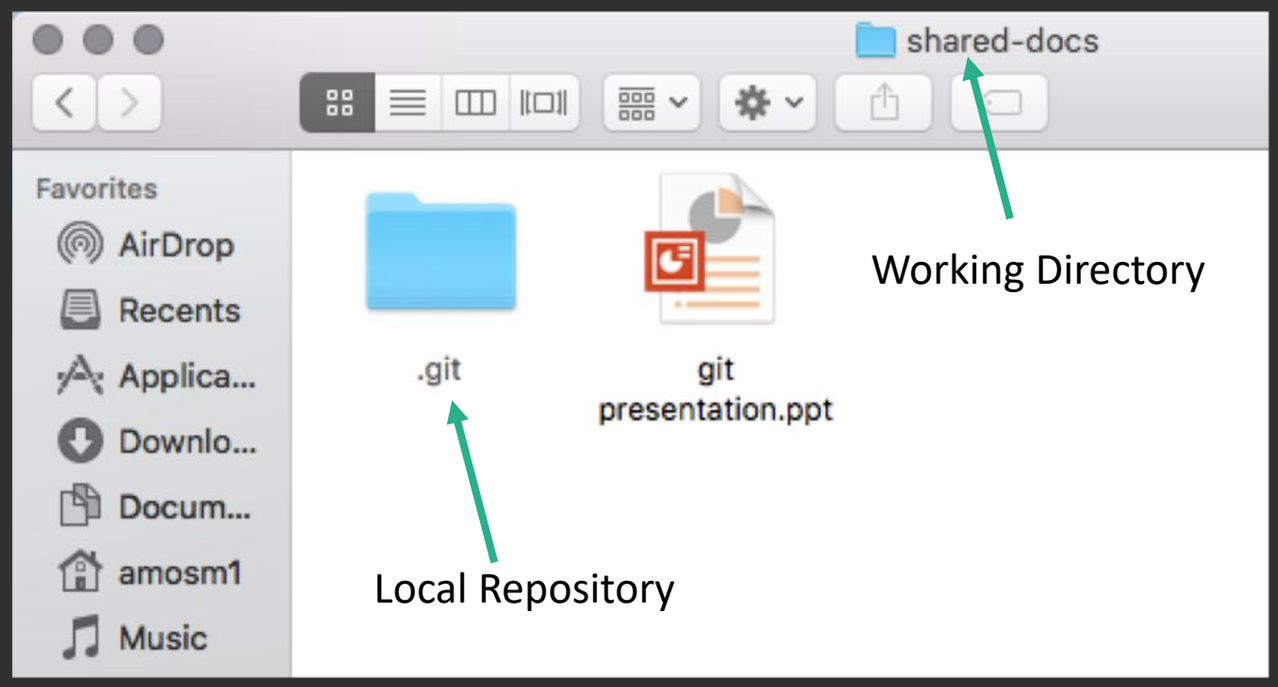
1. Initialise git

- Find a directory you want to add to git/github

```
[dyn-10-32-100-59:~ amosm1$ cd Documents/
[dyn-10-32-100-59:Documents amosm1$ mkdir shared-docs
[dyn-10-32-100-59:Documents amosm1$ cd shared-docs
[dyn-10-32-100-59:shared-docs amosm1$ git init
Initialized empty Git repository in /Users/amosm1/Documents/shared-docs/.git/
```

Initialise a local repository

- Made a new working directory and initialise a git repository (.git folder)



2. Adding to local repo

- Add the file to the staging area (git add filename)

```
[dyn-10-32-100-59:shared-docs amosm1$ git add git\ presentation.ppt
```

- git status shows us about the files git is watching

```
[dyn-10-32-100-59:shared-docs amosm1$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)

        new file:   git presentation.ppt

dyn-10-32-100-59:shared-docs amosm1$
```

We have the master version (not a branch)

git is watching this file but it isn't yet saved to the local repo

3. Saving to the local repo

- We commit (save) the file to the local repo.

```
[dyn-10-32-100-59:shared-docs amosm1$ git commit -m 'added the presentation about git'
[master (root-commit) 07ba3ff] added the presentation about git
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 git presentation.ppt
```

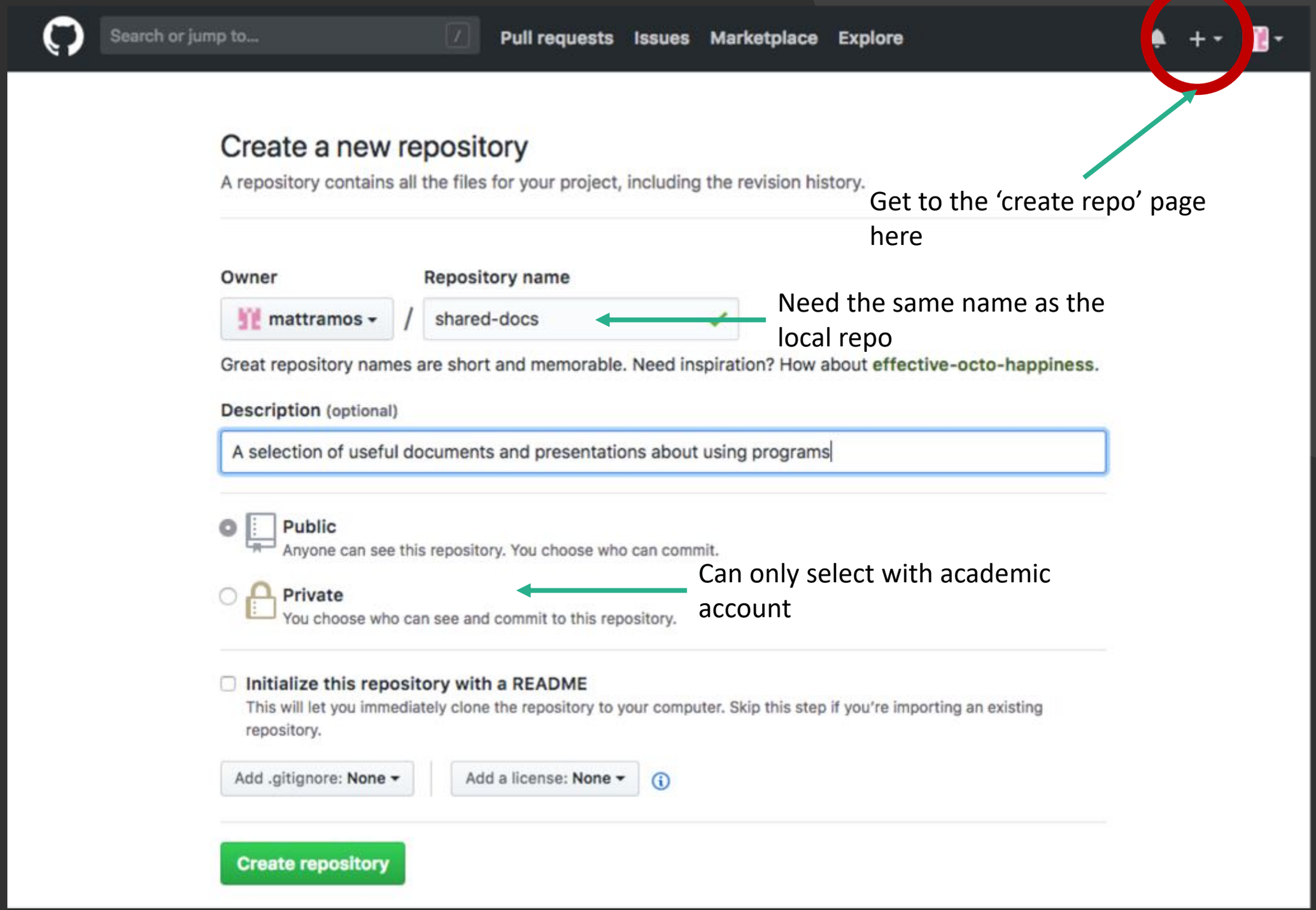
History id

Useful message about changes

- git now shows that we are up to date locally

```
[dyn-10-32-100-59:shared-docs amosm1$ git status
On branch master
nothing to commit, working tree clean
```

4. Create remote repo on github



5. Push to github remote repo

- Create a new remote, called origin. Essentially shows git where the remote repo is

```
amosm1$ git remote add origin https://github.com/mattramos/shared-docs.git
```