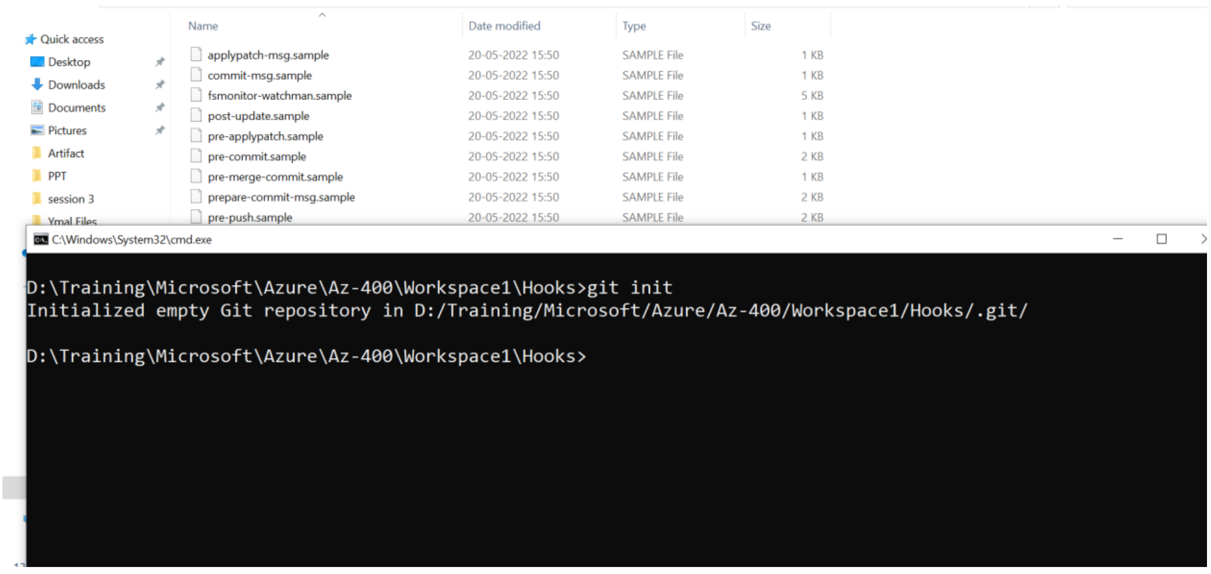
Git hooks are scripts that automatically run every time a particular event occurs in a Git repository. They let you customize Git's internal behaviour and trigger customizable actions at key points in the development life cycle

Hook Name	Event	Description
pre-commit	git commit	This hook is called before obtaining the proposed commit message. Exiting with anything other than zero will abort the commit. It is used to check the commit itself (rather than the message).
prepare-commit-msg	git commit	Called after receiving the default commit message, just prior to firing up the commit message editor. A non-zero exit aborts the commit. This is used to edit the message in a way that cannot be suppressed.
commit-msg	git commit	Can be used to adjust the message after it has been edited in order to ensure conformity to a standard or to reject based on any criteria. It can abort the commit if it exits with a non-zero value.
post-commit	git commit	Called after the actual commit is made. Because of this, it cannot disrupt the commit. It is mainly used to allow notifications.
pre-rebase	git rebase	Called when rebasing a branch. Mainly used to halt the rebase if it is not desirable.
post-checkout	git checkout git clone	Run when a checkout is called after updating the worktree or after git clone. It is mainly used to verify conditions, display differences, and configure the environment if necessary.

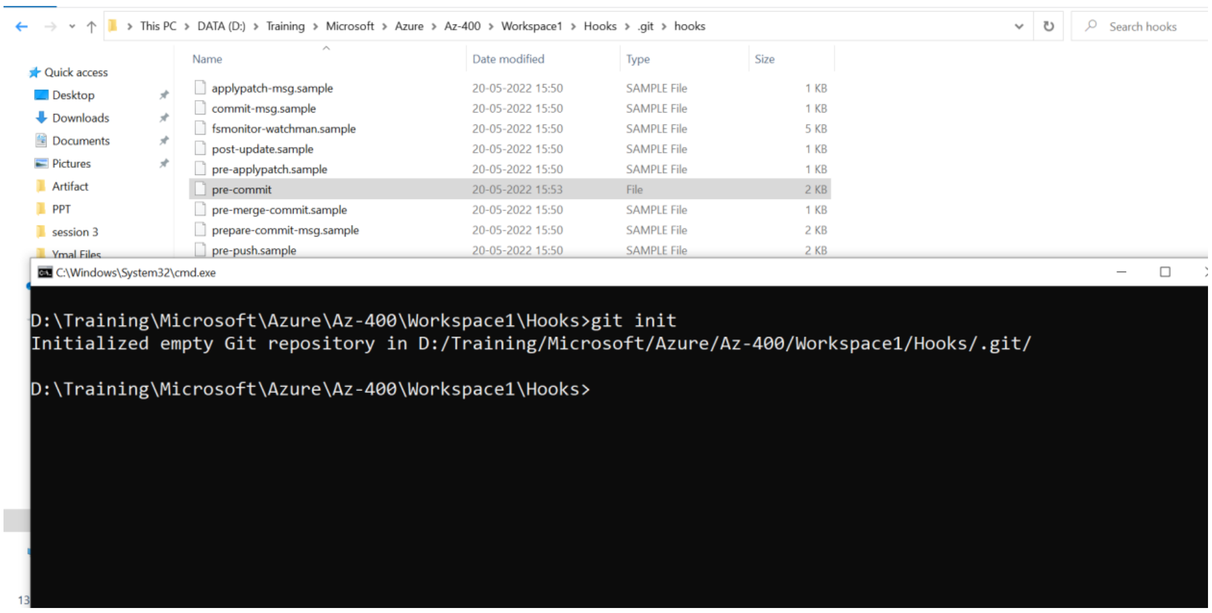
1. Create or Initialize empty git repository (git init)
2. You will find .git folder → hooks



Name	Date modified	Type	Size
applypatch-msg.sample	20-05-2022 15:50	SAMPLE File	1 KB
commit-msg.sample	20-05-2022 15:50	SAMPLE File	1 KB
fsmonitor-watchman.sample	20-05-2022 15:50	SAMPLE File	5 KB
post-update.sample	20-05-2022 15:50	SAMPLE File	1 KB
pre-applypatch.sample	20-05-2022 15:50	SAMPLE File	1 KB
pre-commit.sample	20-05-2022 15:50	SAMPLE File	2 KB
pre-merge-commit.sample	20-05-2022 15:50	SAMPLE File	1 KB
prepare-commit-msg.sample	20-05-2022 15:50	SAMPLE File	2 KB
pre-push.sample	20-05-2022 15:50	SAMPLE File	2 KB

```
D:\Training\Microsoft\Azure\Az-400\Workspace1\Hooks>git init
Initialized empty Git repository in D:/Training/Microsoft/Azure/Az-400/Workspace1/Hooks/.git/

D:\Training\Microsoft\Azure\Az-400\Workspace1\Hooks>
```

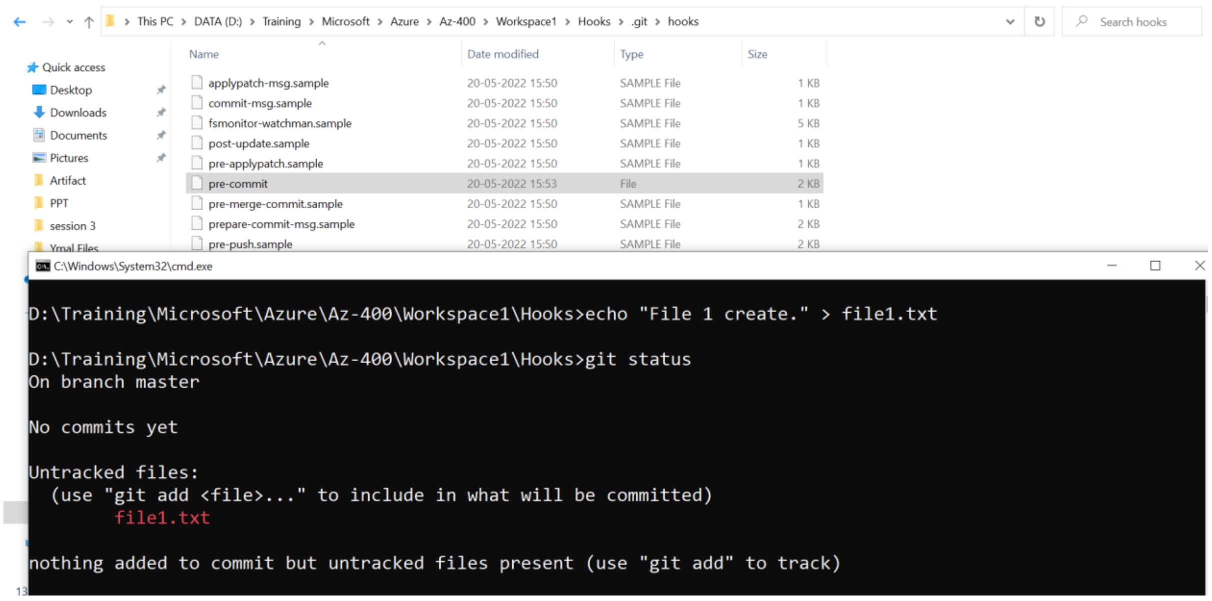
3. To activate git hooks → just remove .sample extension


This PC > DATA (D:) > Training > Microsoft > Azure > Az-400 > Workspace1 > Hooks > .git > hooks
Search hooks
Quick access
Desktop
Downloads
Documents
Pictures
Artifact
PPT
session 3
Ymal Files
C:\Windows\System32\cmd.exe

Name	Date modified	Type	Size
applypatch-msg.sample	20-05-2022 15:50	SAMPLE File	1 KB
commit-msg.sample	20-05-2022 15:50	SAMPLE File	1 KB
fsmonitor-watchman.sample	20-05-2022 15:50	SAMPLE File	5 KB
post-update.sample	20-05-2022 15:50	SAMPLE File	1 KB
pre-applypatch.sample	20-05-2022 15:50	SAMPLE File	1 KB
pre-commit	20-05-2022 15:53	File	2 KB
pre-merge-commit.sample	20-05-2022 15:50	SAMPLE File	1 KB
prepare-commit-msg.sample	20-05-2022 15:50	SAMPLE File	2 KB
pre-push.sample	20-05-2022 15:50	SAMPLE File	2 KB

```
D:\Training\Microsoft\Azure\Az-400\Workspace1\Hooks>git init
Initialized empty Git repository in D:/Training/Microsoft/Azure/Az-400/Workspace1/Hooks/.git/

D:\Training\Microsoft\Azure\Az-400\Workspace1\Hooks>
```

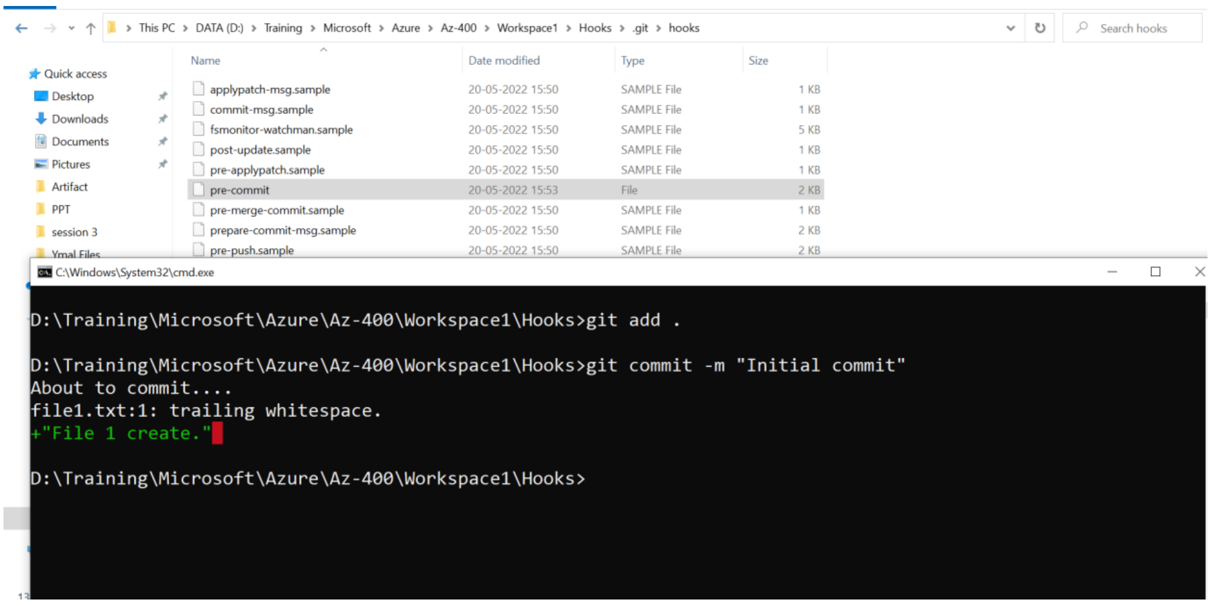
4. Before committing the content of pre-commit will get execute and based on the result of file either commit will be successful or fail


This PC > DATA (D:) > Training > Microsoft > Azure > Az-400 > Workspace1 > Hooks > .git > hooks
Search hooks
Quick access
Desktop
Downloads
Documents
Pictures
Artifact
PPT
session 3
Ymal Files
C:\Windows\System32\cmd.exe

Name	Date modified	Type	Size
applypatch-msg.sample	20-05-2022 15:50	SAMPLE File	1 KB
commit-msg.sample	20-05-2022 15:50	SAMPLE File	1 KB
fsmonitor-watchman.sample	20-05-2022 15:50	SAMPLE File	5 KB
post-update.sample	20-05-2022 15:50	SAMPLE File	1 KB
pre-applypatch.sample	20-05-2022 15:50	SAMPLE File	1 KB
pre-commit	20-05-2022 15:53	File	2 KB
pre-merge-commit.sample	20-05-2022 15:50	SAMPLE File	1 KB
prepare-commit-msg.sample	20-05-2022 15:50	SAMPLE File	2 KB
pre-push.sample	20-05-2022 15:50	SAMPLE File	2 KB

```
D:\Training\Microsoft\Azure\Az-400\Workspace1\Hooks>echo "File 1 create." > file1.txt

D:\Training\Microsoft\Azure\Az-400\Workspace1\Hooks>git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        file1.txt

nothing added to commit but untracked files present (use "git add" to track)
```

5. Stage the contents and commit the changes

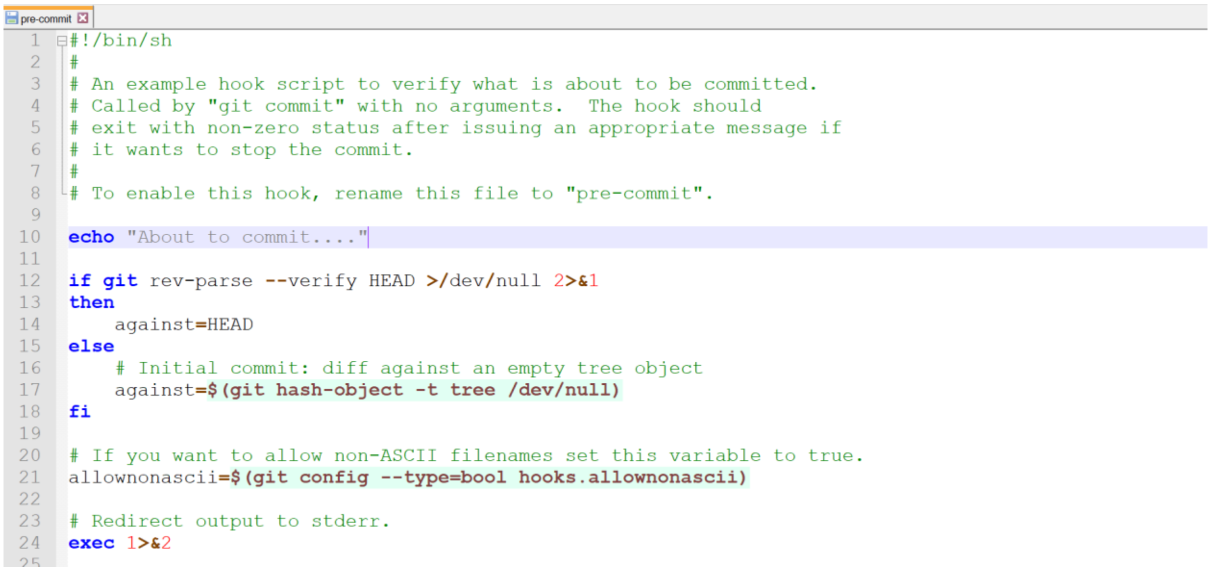


This PC > DATA (D:) > Training > Microsoft > Azure > Az-400 > Workspace1 > Hooks > .git > hooks
Search hooks
Quick access
Desktop
Downloads
Documents
Pictures
Artifact
PPT
session 3
Ymal Files
C:\Windows\System32\cmd.exe

Name	Date modified	Type	Size
applypatch-msg.sample	20-05-2022 15:50	SAMPLE File	1 KB
commit-msg.sample	20-05-2022 15:50	SAMPLE File	1 KB
fsmonitor-watchman.sample	20-05-2022 15:50	SAMPLE File	5 KB
post-update.sample	20-05-2022 15:50	SAMPLE File	1 KB
pre-applypatch.sample	20-05-2022 15:50	SAMPLE File	1 KB
pre-commit	20-05-2022 15:53	File	2 KB
pre-merge-commit.sample	20-05-2022 15:50	SAMPLE File	1 KB
prepare-commit-msg.sample	20-05-2022 15:50	SAMPLE File	2 KB
pre-push.sample	20-05-2022 15:50	SAMPLE File	2 KB

```
D:\Training\Microsoft\Azure\Az-400\Workspace1\Hooks>git add .

D:\Training\Microsoft\Azure\Az-400\Workspace1\Hooks>git commit -m "Initial commit"
About to commit....
file1.txt:1: trailing whitespace.
+"File 1 create."

D:\Training\Microsoft\Azure\Az-400\Workspace1\Hooks>
```

6. Observe “About to commit….” message coming from that file (pre-commit see below)



```
#!/bin/sh
#
# An example hook script to verify what is about to be committed.
# Called by "git commit" with no arguments.  The hook should
# exit with non-zero status after issuing an appropriate message if
# it wants to stop the commit.
#
# To enable this hook, rename this file to "pre-commit".

echo "About to commit...."

if git rev-parse --verify HEAD >/dev/null 2>&1
then
    against=HEAD
else
    # Initial commit: diff against an empty tree object
    against=$(git hash-object -t tree /dev/null)
fi

# If you want to allow non-ASCII filenames set this variable to true.
allownonascii=$(git config --type=bool hooks.allownonascii)

# Redirect output to stderr.
exec 1>&2
```