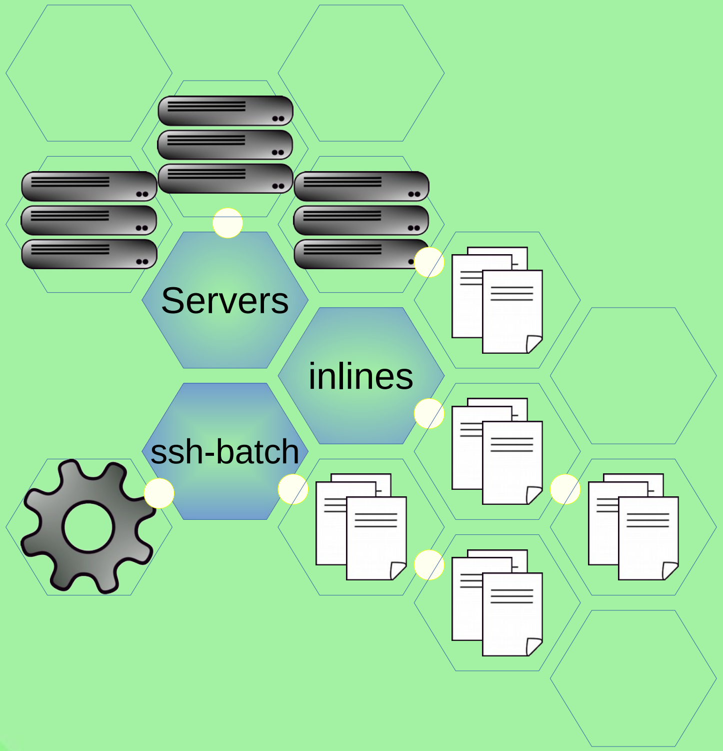
Orchestrating ssh-batch runs with Evidencer

EVIDENCER

SSH-Batch

So you have ssh-batch inline scripts



And it’s getting hard to maintain which scripts can be run on which servers.

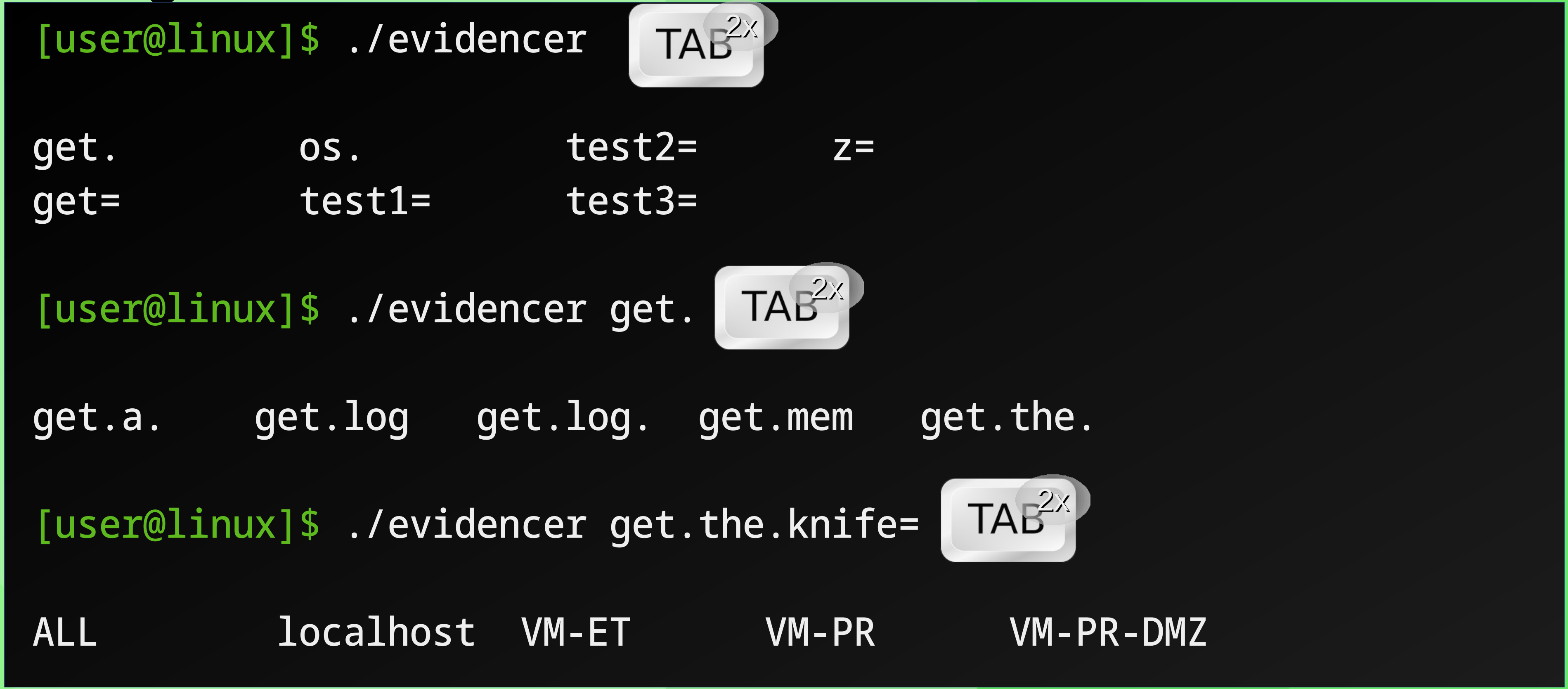
Let Evidencer do the orchestration

- It provides **Structure**
  - ./scripts/ for your scripts.
  - ./servers/ for your servers files.
  - ./results/ for execution logs and server output.
- It provides **Isolation**
  - Suits have their own configuration and directory structure and accommodate different projects.
- It provides **Guided Execution**
  - Enforced through strict naming convention, tab expansion, dryrun and more.
- It provides **Flexibility**
  - pre- and post- scripts to prepare and report the output, optional per-script configuration files and enough modifiers to deviate or extend on-the-fly.

Suit – a group of things forming a unit

Unix tab expansion

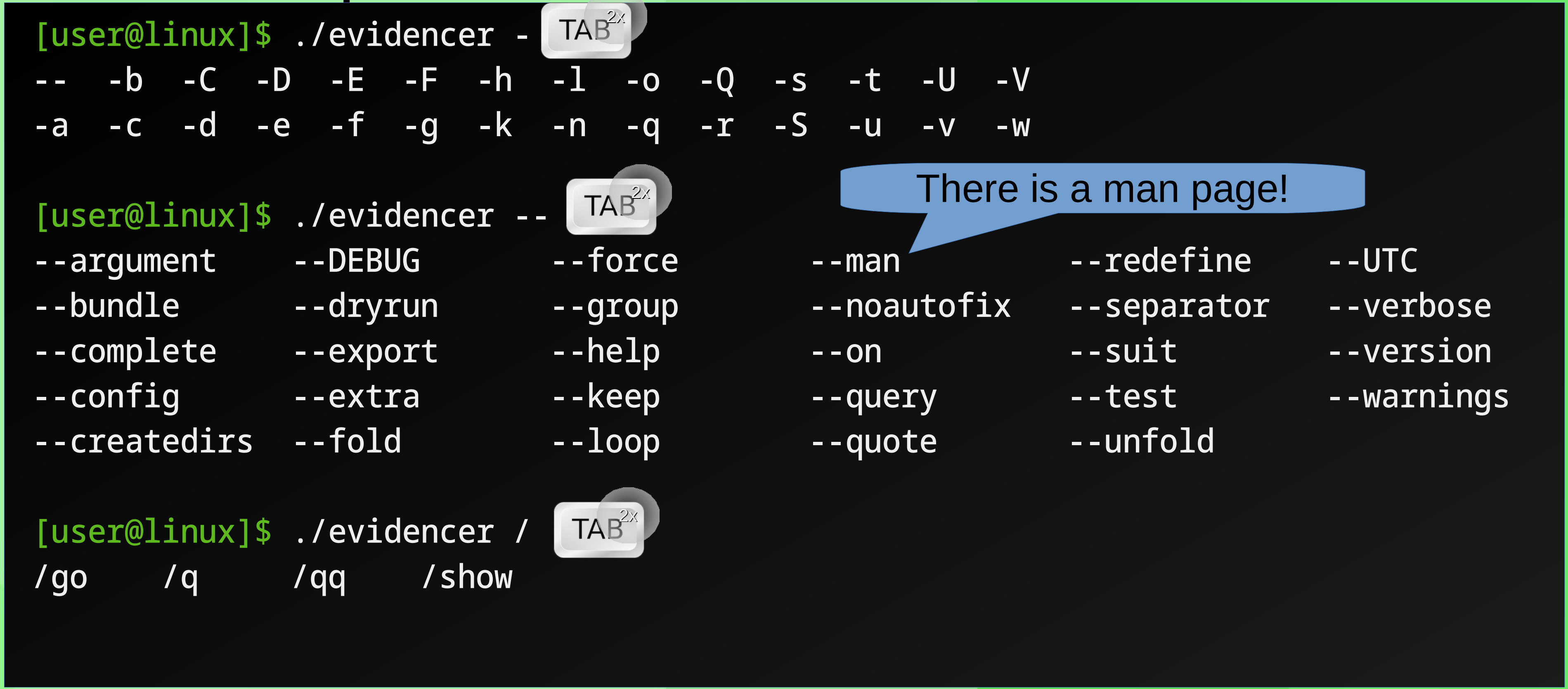
- Single and Double tab



```
[user@linux]$ ./evidencer

get.          os.           test2=        z=
get=          test1=        test3=

[user@linux]$ ./evidencer get.

get.a.     get.log    get.log.   get.mem    get.the.

[user@linux]$ ./evidencer get.the.knife=

ALL          localhost  VM-ET        VM-PR        VM-PR-DMZ
```

Unix tab expansion

- Also for options and aliases



```
[user@linux]$ ./evidencer -
--  -b  -C  -D  -E  -F  -h  -l  -o  -Q  -s  -t  -U  -V
-a  -c  -d  -e  -f  -g  -k  -n  -q  -r  -S  -u  -v  -w

[user@linux]$ ./evidencer --
--argument    --DEBUG       --force       --man         --redefine    --UTC
--bundle      --dryrun      --group       --noautofix   --separator   --verbose
--complete    --export      --help        --on          --suit        --version
--config      --extra       --keep        --query       --test        --warnings
--createdirs  --fold        --loop        --quote       --unfold

[user@linux]$ ./evidencer /
/go    /q     /qq    /show
```

Included help/search

- --help [--verbose]

```
[user@linux]$ ./evidencer get.the.knife= -h

=== get.the.knife ===

 This script fetches the knife from the cupboard.
 Warning: You might need to replenish your cupboard when empty.

[user@linux]$ ./evidencer knife -hv

[user@linux]$ ./evidencer knife+  TAB 1x
```

Help search for “knife” in all scripts

suggest scripts with “knife” in name

Included help/search

- Scripts contain the help:

```
[user@linux]$ cat scripts/get.the.knife\=+
#!/usr/bin/bash
# A normal comment (will not be shown but it will be word-searched)
echo 'I am getting the knife'

#: This script fetches the knife from the <B>cupboard.
#: <R>Warning: You might need to replenish your cupboard when empty.
#+: <Y>Usage: <L>./<.> <L><0>=#
#+:
```

help needs -h to show

Extended help needs -hv

Create suits to separate duties

- Easy to add: --createdirs --suit

```
[user@linux]$ ./evidencer -Cs Project_A

mkdir /home/user/evidencer/suits/Project_A
mkdir /home/user/evidencer/suits/Project_A/servers
mkdir /home/user/evidencer/suits/Project_A/scripts
mkdir /home/user/evidencer/suits/Project_A/results
4 directories created
symlinked /home/user/evidencer/suits/Project_A/evidencer
cfg copied /home/user/evidencer/suits/Project_A/evidencer.cfg
```

These last 2 require SUIT_LINK=1 and SUIT_CFG=1 in evidencer.cfg

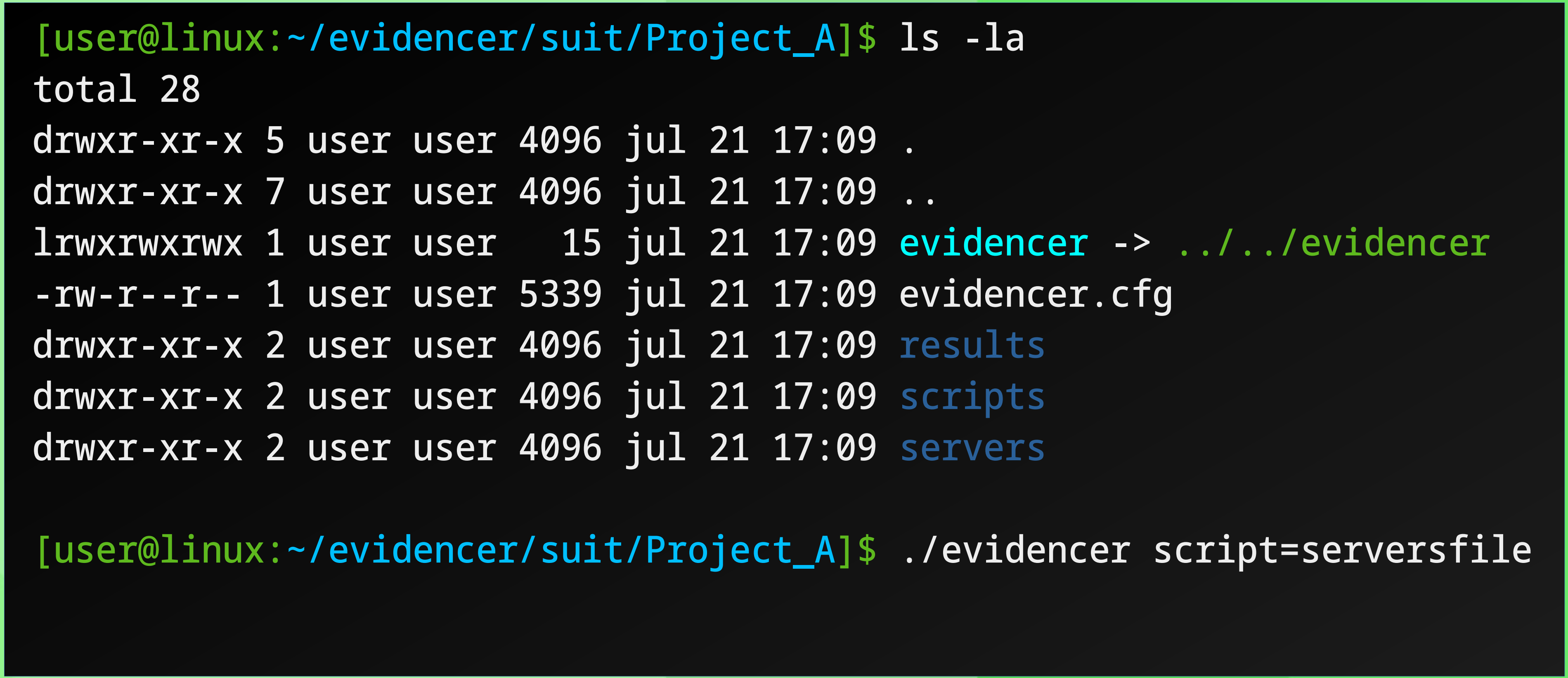
Reference suits from the toplevel

- Address the suit with --suit or with a colon

```
[user@linux]$ ./evidencer -s Project_A script=serversfile


[user@linux]$ ./evidencer Project_A:script=serversfile
```

Or go into that suit directory

- And work without the clutter



```
[user@linux:~/evidencer/suit/Project_A]$ ls -la
total 28
drwxr-xr-x 5 user user 4096 jul 21 17:09 .
drwxr-xr-x 7 user user 4096 jul 21 17:09 ..
lrwxrwxrwx 1 user user   15 jul 21 17:09 evidencer -> ../../evidencer
-rw-r--r-- 1 user user 5339 jul 21 17:09 evidencer.cfg
drwxr-xr-x 2 user user 4096 jul 21 17:09 results
drwxr-xr-x 2 user user 4096 jul 21 17:09 scripts
drwxr-xr-x 2 user user 4096 jul 21 17:09 servers

[user@linux:~/evidencer/suit/Project_A]$ ./evidencer script=serversfile
```

There is more than one way to do it!

- Define ALIASes to chain multiple actions
- Accommodates almost all ssh-batch features
  - Running remote scripts with arguments
  - Connect through jumphosts
  - Run sequentially or/and in parallel
- Skip RUN and only run RUN_POST report but not refetch results
  - By using --redefine
  - By using different configuration files with --config
- Change configuration files for different results.
- Define actions in the configuration, or define scripts and only reference them to run
- Everything is free sourcecode. Read, use, transform.

Welcome aboard!