ipdb: How to use it with Nornir

Python comes with a debuger called pdb. The python debugger is very powerful and really facilitates troubleshooting code. You can find the official documentation here. ipdb builds on top of ipython and pdb to give users a more interactive experience.

This how to isn't very extensive, it's just a very quick and dirty demo to show some of it's capabilities and how to use it with nornir. It doesn't assume previous knowledge but doesn't spend too much on explanations either so it expects users to give enough material for further investigation.

Installing ipdb

First, you will need to install ipython, follow the official guide to do so, then you will need to install `ipdb`. You can install the latter via pip:

```
pip install ipdb
```

Finding and fixing a bug with ipdb

For the sake of the demo I have written some code that contains a bug. The code is not important but what it does is:

1. Read a yaml file containing a bunch of users we want to have configured on our devices
2. Connect to a couple of network devices and get the users configured
3. Check with users are already configured, which ones we want and compute two lists; a list with the users we need to configure and another list with the ones we need to remove.
4. Finally, we pass those lists to a template and we generate some configuration.

First, let's run the script:

```
dbarroso@rlyeh-dev: ~/workspace/dravetech/nornir-workshop/notebooks
(nornir-workshop) ➜  notebooks git:(master) ✗ python demo.py
manage_users*********************************************************************
* spine00.bma ** changed : False ************************************************
vvvv manage_users ** changed : False vvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvv INFO
---- template_file ** changed : False ---------------------------------------- INFO
no username b
no username o
no username b

username joe privilege 15 role network-admin secret sha512 $6$KxgYSk1jaMjZxxBs$1Qmn8UKx7rj3tREiXrdbhXKCC2Mw0w7Lw
ysWCY/xjAX3QDOe0whVdSbn6O50pCsAhNAqB3pk9kY0Nw2OB/Uhz1
username jane privilege 15 role network-admin secret sha512 $6$KxgYSk1jaMjZxxBs$1Qmn8UKx7rj3tREiXrdbhXKCC2Mw0w7L
wysWCY/xjAX3QDOe0whVdSbn6O50pCsAhNAqB3pk9kY0Nw2OB/Uhz1
username admin privilege 15 role network-admin secret sha512 $6$KxgYSk1jaMjZxxBs$1Qmn8UKx7rj3tREiXrdbhXKCC2Mw0w7
LwysWCY/xjAX3QDOe0whVdSbn6O50pCsAhNAqB3pk9kY0Nw2OB/Uhz1

^^^^ END manage_users ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
* spine01.bma ** changed : False ************************************************
vvvv manage_users ** changed : False vvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvv INFO
---- template_file ** changed : False ---------------------------------------- INFO
delete system login user b
delete system login user o
delete system login user b

set system login user joe uid 2000
set system login user joe class super-user
set system login user jane uid 2000
set system login user jane class super-user
set system login user admin uid 2000
set system login user admin class super-user

^^^^ END manage_users ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
(nornir-workshop) ➜  notebooks git:(master) ✗
```

Looks like the code is trying to remove 3 users; “b”, “o” and “b”. You probably figured that’s not correct. In our devices we have a user “bob” that is not listed in our yaml file describing the users we want so we should be removing “bob”. However, something went wrong and we are doing something different.

You can insert a tracepoint in several ways, for the sake of the demo we are going to do it directly in the so let’s open up the code and insert a break point:

```
nvim demo.py
1 demo.py
 1 import logging
 2
 3 import ruamel.yaml
 4
 5 from nornir import InitNornir
 6 from nornir.plugins.functions.text import print_result
 7 from nornir.plugins.tasks.networking import napalm_get
 8 from nornir.plugins.tasks.text import template_file
 9
10
11 def manage_users(task, desired_users):
12     # get users from device
13     state_users = task.run(
14         task=napalm_get, getters=["users"], severity_level=logging.DEBUG
15     )
16
17     # let's verify if the users we got are in desired_users
18     # if they are not we have to remove them
19     users_to_remove = []
20     import ipdb; ipdb.set_trace()
21     for user in state_users.result["users"]:
22         if user not in desired_users:
23             users_to_remove = user
24
25     # we render the template for the platform passing
26     # desired_users and users_to_remove
27     task.run(
28         task=template_file,
29         path=f"5_manage_users/templates/{task.host.platform}",
30         template="users.j2",
31         desired_users=desired_users,
32         remove_users=users_to_remove,
33         #  severity_level=logging.DEBUG,
34     )
35
36
37 def main():
38     nr = InitNornir(
39         core={"num_workers": 1},
40         inventory={
41             "options": {
42                 "host_file": "5_manage_users/inventory/hosts.yaml",
43                 "group_file": "5_manage_users/inventory/groups.yaml",
master> demo.py                                                    33/31/62
```

Notice the following lines:

- line 20: that's where we introduced the tracepoint.
- line 39: the python debugger doesn't interact well with threads so we need to disable parallelization by setting 1 worker

Now that we have the break point let's run the code again:

```
IPython: nornir-workshop/notebooks
(nornir-workshop) ➜  notebooks git:(master) ✗ python demo.py
manage_users*********************************************************************
* spine00.bma ** changed : False ***********************************************
vvvv manage_users ** changed : False vvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvv INFO
---- template_file ** changed : False ---------------------------------------- INFO
no username b
no username o
no username b

username joe privilege 15 role network-admin secret sha512 $6$KxgYSk1jaMjZxxBs$1Qmn8UKx7rj3tREiXrdbhXKCC2Mw0w7Lw
ysWCY/xjAX3QDOe0whVdSbn6O50pCsAhNAqB3pk9kY0Nw2OB/Uhz1
username jane privilege 15 role network-admin secret sha512 $6$KxgYSk1jaMjZxxBs$1Qmn8UKx7rj3tREiXrdbhXKCC2Mw0w7L
wysWCY/xjAX3QDOe0whVdSbn6O50pCsAhNAqB3pk9kY0Nw2OB/Uhz1
username admin privilege 15 role network-admin secret sha512 $6$KxgYSk1jaMjZxxBs$1Qmn8UKx7rj3tREiXrdbhXKCC2Mw0w7
LwysWCY/xjAX3QDOe0whVdSbn6O50pCsAhNAqB3pk9kY0Nw2OB/Uhz1

^^^^ END manage_users ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
* spine01.bma ** changed : False ***********************************************
vvvv manage_users ** changed : False vvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvv INFO
---- template_file ** changed : False ---------------------------------------- INFO
delete system login user b
delete system login user o
delete system login user b

set system login user joe uid 2000
set system login user joe class super-user
set system login user jane uid 2000
set system login user jane class super-user
set system login user admin uid 2000
set system login user admin class super-user

^^^^ END manage_users ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
(nornir-workshop) ➜  notebooks git:(master) ✗ nvim demo.py
(nornir-workshop) ➜  notebooks git:(master) ✗ python demo.py
> /home/dbarroso/workspace/dravetech/nornir-workshop/notebooks/demo.py(21)manage_users()
     20     import ipdb; ipdb.set_trace()
---> 21     for user in state_users.result["users"]:
     22         if user not in desired_users:

ipdb>
```

After running the script again we get a shell. We can also see a few lines of code to give us context and know where we are. In this shell you can execute python code and also pdb commands. Some of the pdb commands we are going to be using are:

- `l` (list) - prints the code around the current break point
- `n` (next) - executes the code in the current line
- `c` (continue) - executes normally until it finds another break point

```
IPython: nornir-workshop/notebooks
LwysWCY/xjAX3QDOe0whVdSbn6O50pCsAhNAqB3pk9kY0Nw2OB/Uhz1

^^^^ END manage_users ^^^^
```

```
IPython: nornir-workshop/notebooks
     26     # desired_users and users_to_remove

ipdb> users_to_remove
[]
ipdb> desired_users
ordereddict([('joe', None), ('jane', None), ('admin', None)])
ipdb> state_users.result["users"]
{'admin': {'password': '$1$waUQAEpO$kSmdRL84F5sCoWKHZ8/kU1', 'sshkeys': [], 'level': 15}, 'bob': {'password': '$
1$waUQAEpO$kSmdRL84F5sCoWKHZ8/kU1', 'sshkeys': [], 'level': 15}}
ipdb> n
> /home/dbarroso/workspace/dravetech/nornir-workshop/notebooks/demo.py(22)manage_users()
     21     for user in state_users.result["users"]:
---> 22         if user not in desired_users:
     23             users_to_remove = user

ipdb> user
'admin'
ipdb> n
> /home/dbarroso/workspace/dravetech/nornir-workshop/notebooks/demo.py(21)manage_users()
     20     import ipdb; ipdb.set_trace()
---> 21     for user in state_users.result["users"]:
     22         if user not in desired_users:

ipdb>
> /home/dbarroso/workspace/dravetech/nornir-workshop/notebooks/demo.py(22)manage_users()
     21     for user in state_users.result["users"]:
---> 22         if user not in desired_users:
     23             users_to_remove = user

ipdb> user
'bob'
ipdb> n
> /home/dbarroso/workspace/dravetech/nornir-workshop/notebooks/demo.py(23)manage_users()
     22         if user not in desired_users:
---> 23             users_to_remove = user
     24

ipdb>
> /home/dbarroso/workspace/dravetech/nornir-workshop/notebooks/demo.py(21)manage_users()
     20     import ipdb; ipdb.set_trace()
---> 21     for user in state_users.result["users"]:
     22         if user not in desired_users:

ipdb> users_to_remove
'bob'
ipdb>
```

In the capture above you can see we used `l` to show the code around current break point and then we printed some of the variables in it.

Now we can keep executing the code line by line with `n` and print the variables to see what's going on. A few things to note:

- we can see how the execution loops around in the for loop
- we can also see how the user "admin" doesn't meet the criteria in line 22 causing the code to loop back to line 21
- when we run the loop for user "bob", the condition in line 22 is met so we go to line 23

At this point we can print `users_to_remove` and realize we assigned a string even though we started by assigning an empty list to it. This would explain why "bob" turns into "b", "o", "b" (strings are iterables like lists). Let's exit `ipdb`:

```
dbarroso@rlyeh-dev: ~/workspace/dravetech/nornir-workshop/notebooks

ipdb> users_to_remove
'bob'
ipdb> q
> /home/dbarroso/workspace/dravetech/nornir-workshop/notebooks/demo.py(21)manage_users()
     20     import ipdb; ipdb.set_trace()
---> 21     for user in state_users.result["users"]:
     22         if user not in desired_users:

ipdb> q
manage_users*********************************************************************
* spine00.bma ** changed : False ************************************************
vvvv manage_users ** changed : False vvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvv ERROR
Traceback (most recent call last):
  File "/home/dbarroso/.virtualenvs/nornir-workshop/lib/python3.6/site-packages/nornir/core/task.py", line 67, i
n start
    r = self.task(self, **self.params)
  File "demo.py", line 21, in manage_users
    for user in state_users.result["users"]:
  File "demo.py", line 21, in manage_users
    for user in state_users.result["users"]:
  File "/usr/lib/python3.6/bdb.py", line 51, in trace_dispatch
    return self.dispatch_line(frame)
  File "/usr/lib/python3.6/bdb.py", line 70, in dispatch_line
    if self.quitting: raise BdbQuit
bdb.BdbQuit

^^^^ END manage_users ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
* spine01.bma ** changed : False ************************************************
vvvv manage_users ** changed : False vvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvv ERROR
Traceback (most recent call last):
  File "/home/dbarroso/.virtualenvs/nornir-workshop/lib/python3.6/site-packages/nornir/core/task.py", line 67, i
n start
    r = self.task(self, **self.params)
  File "demo.py", line 21, in manage_users
    for user in state_users.result["users"]:
  File "demo.py", line 21, in manage_users
    for user in state_users.result["users"]:
  File "/usr/lib/python3.6/bdb.py", line 51, in trace_dispatch
    return self.dispatch_line(frame)
  File "/usr/lib/python3.6/bdb.py", line 70, in dispatch_line
    if self.quitting: raise BdbQuit
bdb.BdbQuit

^^^^ END manage_users ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
(nornir-workshop) ➜  notebooks git:(master) ✗
```

Now let's finally fix line 23, appending the user instead of assigning it to the variable should do the trick:

nvim demo.py

```
1 demo.py
import logging

import ruamel.yaml

from nornir import InitNornir
from nornir.plugins.functions.text import print_result
from nornir.plugins.tasks.networking import napalm_get
from nornir.plugins.tasks.text import template_file


def manage_users(task, desired_users):
    # get users from device
    state_users = task.run(
        task=napalm_get, getters=["users"], severity_level=logging.DEBUG
    )

    # let's verify if the users we got are in desired_users
    # if they are not we have to remove them
    users_to_remove = []
    for user in state_users.result["users"]:
        if user not in desired_users:
            users_to_remove.append(user)

    # we render the template for the platform passing
    # desired_users and users_to_remove
    task.run(
        task=template_file,
        path=f"5_manage_users/templates/{task.host.platform}",
        template="users.j2",
        desired_users=desired_users,
        remove_users=users_to_remove,
        #  severity_level=logging.DEBUG,
    )


def main():
    nr = InitNornir(
        inventory={
            "options": {
                "host_file": "5_manage_users/inventory/hosts.yaml",
                "group_file": "5_manage_users/inventory/groups.yaml",
                "defaults_file": "5_manage_users/inventory/defaults.yaml",
            }
master> demo.py[+]                                                  41/22/60
```

And now let's run the code again:

```
dbarroso@rlyeh-dev: ~/workspace/dravetech/nornir-workshop/notebooks
vvvv manage_users ** changed : False vvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvv ERROR
Traceback (most recent call last):
  File "/home/dbarroso/.virtualenvs/nornir-workshop/lib/python3.6/site-packages/nornir/core/task.py", line 67, i
n start
    r = self.task(self, **self.params)
  File "demo.py", line 21, in manage_users
    for user in state_users.result["users"]:
  File "demo.py", line 21, in manage_users
    for user in state_users.result["users"]:
  File "/usr/lib/python3.6/bdb.py", line 51, in trace_dispatch
    return self.dispatch_line(frame)
  File "/usr/lib/python3.6/bdb.py", line 70, in dispatch_line
    if self.quitting: raise BdbQuit
bdb.BdbQuit

^^^^ END manage_users ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
(nornir-workshop) → notebooks git:(master) ✗ nvim demo.py
(nornir-workshop) → notebooks git:(master) ✗ python demo.py
manage_users*********************************************************************
* spine00.bma ** changed : False ************************************************
vvvv manage_users ** changed : False vvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvv INFO
---- template_file ** changed : False ------------------------------------------ INFO
no username bob

username joe privilege 15 role network-admin secret sha512 $6$KxgYSk1jaMjZxxBs$1Qmn8UKx7rj3tREiXrdbhXKCC2Mw0w7Lw
ysWCY/xjAX3QDOe0whVdSbn6O50pCsAhNAqB3pk9kY0Nw2OB/Uhz1
username jane privilege 15 role network-admin secret sha512 $6$KxgYSk1jaMjZxxBs$1Qmn8UKx7rj3tREiXrdbhXKCC2Mw0w7L
wysWCY/xjAX3QDOe0whVdSbn6O50pCsAhNAqB3pk9kY0Nw2OB/Uhz1
username admin privilege 15 role network-admin secret sha512 $6$KxgYSk1jaMjZxxBs$1Qmn8UKx7rj3tREiXrdbhXKCC2Mw0w7
LwysWCY/xjAX3QDOe0whVdSbn6O50pCsAhNAqB3pk9kY0Nw2OB/Uhz1

^^^^ END manage_users ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
* spine01.bma ** changed : False ************************************************
vvvv manage_users ** changed : False vvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvv INFO
---- template_file ** changed : False ------------------------------------------ INFO
delete system login user bob

set system login user joe uid 2000
set system login user joe class super-user
set system login user jane uid 2000
set system login user jane class super-user
set system login user admin uid 2000
set system login user admin class super-user

^^^^ END manage_users ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
(nornir-workshop) → notebooks git:(master) ✗
```