1. Creating your 1st module

Writing a module isn’t hard. In this workshop, we write a module, which is able to create and remove a file from a specified location. If the location is not specified the module puts the file into the /tmp directory.

You can write the module with your favorite editor. It will be a module written in python which can be used at Unix and linux platforms.

This document consists of code and explanation. Code is surrounded by 1. #box and Font Consolas.

Explanation is normal text written in Arial

Take into account that python uses 4 spaces to align the code.

So let’s start Coding!

In the directory where you will create the playbook, create a directory: library and cd into it.

Start editing a file called: demo.py This will be your module

First start with the hashbang and import of the AnsibleModule

1. #!/usr/bin/python
2. # Import necessary libraries
3. **from** ansible.module\_utils.basic **import** AnsibleModule
5. # end import modules

Next add the structure for the module

We know we add a function for creating a file, create\_file() and removing a file, remove\_file(). Of course we have to add the main() function and call it

1. #!/usr/bin/python
2. # Import necessary libraries
3. **from** ansible.module\_utils.basic **import** AnsibleModule
5. # end import modules
6. # start defining the functions
8. **def** create\_file():

11. **def** remove\_file():
13. **def** main():
15. **if** \_\_name\_\_ == '\_\_main\_\_':
16. main()

Now we start to define the module arguments this is the start of the main() function code sits is between the line 14: def main() and the line 16 if \_\_name\_\_ == ‘\_\_main\_\_’:

We create an instance of the AnsibleModule library and we name it module

As we can see we define 3 arguments:

* name: The name of the file
* location: The location of the file, if nothing is given, the default is /tmp
* state: The state, should the file be present or absent, default is present

1. **def** main():
2. module = AnsibleModule(
3. argument\_spec=dict(
4. name=dict(type='str', required=True),
5. location=dict(type='str', default='/tmp'),
6. state=dict(type='str', choices=['absent', 'present'],
7. default='present'),
8. ),
9. )

Because at the end of the module we have to give results back to ansible let’s define a result dictionary

1. result = {
2. 'msg': "",
3. 'changed': False
4. }

Now let’s create a parameter, full\_path\_name with the full path name of the file which you are going to create or remove.

To retrieve values from the arguments, you use the function module.params[‘<argument>’]

2. full\_path\_name = module.params['location'] + "/" + module.params['name']

as there are 2 action in the module, create or remove a file, the logic has to be written. The result of the action is put in the result parameter.

So we add the module.exit\_json(\*\*result) at the end of the code.

This will exit the module and send the result as JSON to ansible.

1. **if** module.params['state'] == 'present':
2. result = create\_file(module, full\_path\_name)
3. **else**:
4. result = remove\_file(module, full\_path\_name)
6. module.exit\_json(\*\*result)

This end the main() function of the module. Now the create\_file can be coded. In the main() function the create\_file function is called with 2 parameters, module and full\_path\_name, so the definition of this function has to be adjusted to:

create\_file(module, full\_path\_name)

First we define a empty results dictionary to put the results in and to return it.

Then we find out where the touch command can be found, and we run the command using the module.run\_command

The results are checked, and is the returncode is 0 success results are put into result, otherwise the module wil call back with the module.fail\_json. This will let ansible know that the module has failed with the correct JSON

1. **def** create\_file(module, full\_path\_name):
2. # create file
3. # uses touch to create a file
4. result = {}
5. touch = module.get\_bin\_path("touch")
6. (rc, out, err) = module.run\_command([touch, full\_path\_name])
7. **if** rc == 0:
8. result['changed'] = True
9. result['msg'] = "file: " + full\_path\_name + " created"
10. **else**:
11. module.fail\_json(
12. msg="could not create " + full\_path\_name, rc=rc, err=err)
13. **return** result

For the removal of a file, similar code will do the trick.

Do not forget to adjust the definition of the remove\_file function

2. **def** remove\_file(module, full\_path\_name):
3. # remove file
4. # uses rm to remove file
5. result = {}
6. rm = module.get\_bin\_path("rm")
7. (rc, out, err) = module.run\_command([rm, full\_path\_name])
8. **if** rc == 0:
9. result['changed'] = True
10. result['msg'] = "file: " + full\_path\_name + " removed"
11. **else**:
12. module.fail\_json(
13. msg="could not remove " + full\_path\_name, rc=rc, err=err)
14. **return** result

Here is the complete code:

1. #!/usr/bin/python
2. # Import necessary libraries
3. **from** ansible.module\_utils.basic **import** AnsibleModule
5. # end import modules
6. # start defining the functions
8. **def** create\_file(module, full\_path\_name):
9. # create file
10. # uses touch to create a file
11. result = {}
12. touch = module.get\_bin\_path("touch")
13. (rc, out, err) = module.run\_command([touch, full\_path\_name])
14. **if** rc == 0:
15. result['changed'] = True
16. result['msg'] = "file: " + full\_path\_name + " created"
17. **else**:
18. module.fail\_json(
19. msg="could not create " + full\_path\_name, rc=rc, err=err)
20. **return** result

23. **def** remove\_file(module, full\_path\_name):
24. # remove file
25. # uses rm to remove file
26. result = {}
27. rm = module.get\_bin\_path("rm")
28. (rc, out, err) = module.run\_command([rm, full\_path\_name])
29. **if** rc == 0:
30. result['changed'] = True
31. result['msg'] = "file: " + full\_path\_name + " removed"
32. **else**:
33. module.fail\_json(
34. msg="could not remove " + full\_path\_name, rc=rc, err=err)
35. **return** result

38. **def** main():
39. module = AnsibleModule(
40. argument\_spec=dict(
41. name=dict(type='str', required=True),
42. location=dict(type='str', default='/tmp'),
43. state=dict(type='str', choices=['absent', 'present'],
44. default='present'),
45. ),
46. )
48. result = {
49. 'msg': "",
50. 'changed': False
51. }
53. full\_path\_name = module.params['location'] + "/" + module.params['name']
55. **if** module.params['state'] == 'present':
56. result = create\_file(module, full\_path\_name)
57. **else**:
58. result = remove\_file(module, full\_path\_name)
60. module.exit\_json(\*\*result)

63. **if** \_\_name\_\_ == '\_\_main\_\_':
64. main()

Now this module can be tested

Go back to the directory where your playbooks reside, and create a small playbook to use your new module to create a file name it create\_file.yml

1. ---
2. - name: create\_file
3. hosts: all
4. gather\_facts: False
6. tasks:
7. - name: Add a file
8. demo:
9. name: "hello\_world"
10. location: /tmp
11. ...

And run this playbook

# ansible-playbook -i inventory create\_file.yml

PLAY [create\_file] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

TASK [Add a file] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

changed: [192.168.10.128]

PLAY RECAP \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

192.168.10.128 : ok=1 changed=1 unreachable=0 failed=0 skipped=0

You will get similar output as above

When you run this again, output keeps the same.

Now create a similar playbook, remove\_file.yml for removing the file

1. ---
2. - name: remove\_file
3. hosts: all
4. gather\_facts: False
6. tasks:
7. - name: remove a file
8. demo:
9. name: "hello\_world"
10. location: /tmp
11. state: absent
12. ...

And run the playbook. Output will be similar to:

# ansible-playbook -i inventory remove\_file.yml

PLAY [remove\_file] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

TASK [remove a file] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

changed: [192.168.10.128]

PLAY RECAP \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

192.168.10.128 : ok=1 changed=1 unreachable=0 failed=0 skipped=0

However when you run this again, you will get output as below:

# ansible-playbook -i inventory remove\_file.yml

PLAY [remove\_file] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

TASK [remove a file] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

fatal: [192.168.10.128]: FAILED! => {"changed": false, "err": "/usr/bin/rm: cannot remove ‘/tmp/hello\_world’: No such file or directory\n", "msg": "could not remove /tmp/hello\_world", "rc": 1}

to retry, use: --limit @/root/devel/modules\_workshop/remove\_file.retry

PLAY RECAP \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

192.168.10.128 : ok=0 changed=0 unreachable=0 failed=1 skipped=0

1. Add Idempotency

As seen above the current module is not idempotent, creating the file 2 times results in the message that the playbook has a changed result, and removing the file the second time, results in a playbook failure. So let us add a function to make the module idempotent

Open the module in your favorite editor and create a function to check if the file is present or not. As with the create and remove functions it’s pretty simple code add this new function just before the create\_file(module, full\_path\_name) function

1. **def** check\_file(module, full\_path\_name):
2. # check if file exists
3. # uses ls to check if file exists
4. # returns a boolean if the file exists True, else False
5. exists = False
6. ls = module.get\_bin\_path('ls')
7. (rc, out, err) = module.run\_command([ls, full\_path\_name])
8. **if** rc == 0:
9. exists = True
10. **return** exists

Now we add a bit of more logic to the module

replace this code

1. **if** module.params['state'] == 'present':
2. result = create\_file(module, full\_path\_name)
3. **else**:
4. result = remove\_file(module, full\_path\_name)

with:

1. # check if file exists
2. file\_exists = check\_file(module, full\_path\_name)
4. # if state is present and file does not exist create file
5. # if state is present and file does exist do nothing
6. # if state is absent and file does not exist do nothing
7. # if state is absent and file does exist remove file
9. **if** module.params['state'] == 'present':
10. **if** (**not** file\_exists):
11. result = create\_file(module, full\_path\_name)
12. **else**:
13. **if** (file\_exists):
14. result = remove\_file(module, full\_path\_name)

The complete code is now:

1. #!/usr/bin/python
3. # Import necessary libraries
4. **from** ansible.module\_utils.basic **import** AnsibleModule
6. # start defining the functions

9. **def** check\_file(module, full\_path\_name):
10. # check if file exists
11. # uses ls to check if file exists
12. # returns a boolean if the file exists True, else False
13. exists = False
14. ls = module.get\_bin\_path('ls')
15. (rc, out, err) = module.run\_command([ls, full\_path\_name])
16. **if** rc == 0:
17. exists = True
18. **return** exists

21. **def** create\_file(module, full\_path\_name):
22. # create file
23. # uses touch to create a file
24. result = {}
25. touch = module.get\_bin\_path("touch")
26. (rc, out, err) = module.run\_command([touch, full\_path\_name])
27. **if** rc == 0:
28. result['changed'] = True
29. result['msg'] = "file: " + full\_path\_name + " created"
30. **else**:
31. module.fail\_json(
32. msg="could not create " + full\_path\_name, rc=rc, err=err)
33. **return** result

36. **def** remove\_file(module, full\_path\_name):
37. # remove file
38. # uses rm to remove file
39. result = {}
40. rm = module.get\_bin\_path("rm")
41. (rc, out, err) = module.run\_command([rm, full\_path\_name])
42. **if** rc == 0:
43. result['changed'] = True
44. result['msg'] = "file: " + full\_path\_name + " removed"
45. **else**:
46. module.fail\_json(
47. msg="could not remove " + full\_path\_name, rc=rc, err=err)
48. **return** result

51. **def** main():
52. module = AnsibleModule(
53. argument\_spec=dict(
54. name=dict(type='str', required=True),
55. location=dict(type='str', default='/tmp'),
56. state=dict(type='str', choices=['absent', 'present'],
57. default='present'),
58. ),
59. )
61. result = {
62. 'msg': "",
63. 'changed': False
64. }
66. full\_path\_name = module.params['location'] + "/" + module.params['name']
67. # check if file exists
68. file\_exists = check\_file(module, full\_path\_name)
70. # if state is present and file does not exist create file
71. # if state is present and file does exist do nothing
72. # if state is absent and file does not exist do nothing
73. # if state is absent and file does exist remove file
75. **if** module.params['state'] == 'present':
76. **if** (**not** file\_exists):
77. result = create\_file(module, full\_path\_name)
78. **else**:
79. **if** (file\_exists):
80. result = remove\_file(module, full\_path\_name)
82. module.exit\_json(\*\*result)

85. **if** \_\_name\_\_ == '\_\_main\_\_':
86. main()

Now if your file is not present at the ansible client machine and you run the playbook to create the file you will see:

# ansible-playbook -i inventory create\_file.yml

PLAY [create\_file] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

TASK [Add a file] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

changed: [192.168.10.128]

PLAY RECAP \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

192.168.10.128 : ok=1 changed=1 unreachable=0 failed=0 skipped=0

And when you run this the second time:

# ansible-playbook -i inventory create\_file.yml

PLAY [create\_file] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

TASK [Add a file] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

ok: [192.168.10.128]

PLAY RECAP \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

192.168.10.128 : ok=1 changed=0 unreachable=0 failed=0 skipped=0

When you remove the file when the file is present you will get:

# ansible-playbook -i inventory remove\_file.yml

PLAY [remove\_file] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

TASK [remove a file] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

changed: [192.168.10.128]

PLAY RECAP \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

192.168.10.128 : ok=1 changed=1 unreachable=0 failed=0 skipped=0

When run a second time, when the file is already absent you will get:

# ansible-playbook -i inventory remove\_file.yml

PLAY [remove\_file] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

TASK [remove a file] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

ok: [192.168.10.128]

PLAY RECAP \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

192.168.10.128 : ok=1 changed=0 unreachable=0 failed=0 skipped=0

The module is now Idempotent !

1. Add Checkmode

When you run your playbook with the checkmode parameter “ -C” , you will see the task running this module will be skipped. This is not the wanted behavior

# ansible-playbook -i inventory create\_file.yml -C

PLAY [create\_file] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

TASK [Add a file] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

skipping: [192.168.10.128]

PLAY RECAP \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

192.168.10.128 : ok=0 changed=0 unreachable=0 failed=0

This part of the workshop adds checkmode to the module

In the argument\_spec add the line “supports\_check\_mode=True,” do not miss the comma!

1. module = AnsibleModule(
2. argument\_spec=dict(
3. name=dict(type='str', required=True),
4. location=dict(type='str', default='/tmp'),
5. state=dict(type='str', choices=['absent', 'present'],
6. default='present'),
7. ),
8. supports\_check\_mode=True,
9. )

Now the checkmode has to be added in the create\_file and remove\_file function

Add the line:

“If not module.check\_mode:” just in front of the run \_command line

And add the 4 spaces in front of the run\_command line see lines: 27/28 and 44/45

Because the run\_command line isn’t run when check\_mode is True, we have to define the rc variable. Add line 24 and 41

1. def create\_file(module, full\_path\_name):
2. # create file
3. # uses touch to create a file
4. rc = 0
5. result = {}
6. touch = module.get\_bin\_path("touch")
7. if not module.check\_mode:
8. (rc, out, err) = module.run\_command([touch, full\_path\_name])
9. if rc == 0:
10. result['changed'] = True
11. result['msg'] = "file: " + full\_path\_name + " created"
12. else:
13. module.fail\_json(
14. msg="could not create " + full\_path\_name, rc=rc, err=err)
15. return result

18. def remove\_file(module, full\_path\_name):
19. # remove file
20. # uses rm to remove file
21. rc = 0
22. result = {}
23. rm = module.get\_bin\_path("rm")
24. if not module.check\_mode:
25. (rc, out, err) = module.run\_command([rm, full\_path\_name])
26. if rc == 0:
27. result['changed'] = True
28. result['msg'] = "file: " + full\_path\_name + " removed"
29. else:
30. module.fail\_json(
31. msg="could not remove " + full\_path\_name, rc=rc, err=err)
32. return result

When the playbook is run in testmode, you can see the file is nor created or removed.

# ansible-playbook -i inventory create\_file.yml -C

PLAY [create\_file] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

TASK [Add a file] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

changed: [192.168.10.128]

PLAY RECAP \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

192.168.10.128 : ok=1 changed=1 unreachable=0 failed=0

And run the second time

# ansible-playbook -i inventory create\_file.yml -C

PLAY [create\_file] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

TASK [Add a file] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

changed: [192.168.10.128]

PLAY RECAP \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

192.168.10.128 : ok=1 changed=1 unreachable=0 failed=0

And when you want to remove the file

# ansible-playbook -i inventory remove\_file.yml -C

PLAY [remove\_file] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

TASK [remove a file] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

ok: [192.168.10.128]

PLAY RECAP \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

192.168.10.128 : ok=1 changed=0 unreachable=0 failed=0

1. Documenting the module

This chapter adds the documentation to the module and shows how to retrieve the module documentation

Start adding the license

1. # (c) 2017, Joris Weijters <joris.weijters@gmail.com>
2. # GNU General Public License v3.0+ (see COPYING or https://www.gnu.org/licenses/gpl-3.0.txt)

And add the ANSIBLE\_METADATA

2. ANSIBLE\_METADATA = {'metadata\_version': '1.1',
3. 'status': ['preview'],
4. 'supported\_by': 'community'}

Now the documentation must be added each documentation block starts and ends with '''

In the documentation block substitute the AUTHOR with your name and optionally add the (@GITHUBUSER) name. This is necessary when you want to add your module to Ansible.

1. DOCUMENTATION = '''
2. ---
3. author:
4. - AUTHOR (@GITHUBUSER)
5. module: demo
6. short\_description: demo module
7. description:
8. - demo module for the Ansible meetup benelux
9. version\_added: "2.7"
10. options:
11. name:
12. description:
13. - Name of the file
14. required: yes
15. location:
16. description:
17. - Location of the file
18. default: /tmp
19. state:
20. description:
21. - whether the file should be present or absent
22. choices: [ absent, present ]
23. default: present
24. notes:
25. - puts a file at a specific location
26. - this is just to demo the writing of modules
27. '''
29. EXAMPLES = '''
30. # Add a file
31. - name: Add a file
32. demo:
33. name: "hello\_world"
34. location: /tmp
36. # remove a file
37. - name: remove file
38. demo:
39. name: "hello\_world"
40. location: /tmp
41. state: absent
42. '''
44. RETURN = '''
45. msg:
46. description: return message
47. returned: always
48. type: string
49. sample: file /tmp/hello\_world created
50. changed:
51. description: whether the file add or removal has been changed
52. returned: always
53. type: boolean
54. sample: true
55. '''

You can retrieve the documentation of the module with the ansible-doc command

# ansible-doc -M library demo

> DEMO (/root/devel/modules\_workshop/library/demo.py)

demo module for the Ansible meetup benelux

OPTIONS (= is mandatory):

- location

Location of the file

[Default: /tmp]

= name

Name of the file

- state

whether the file should be present or absent

(Choices: absent, present)[Default: present]

This is an example of the documentation

1. Testing Modules

Make sure git is installed at your ansible server

And clone the ansible code from github ansible code can be found at

<https://github.com/ansible/ansible>

[root@ansiblehost devel]# git clone https://github.com/ansible/ansible.git

Cloning into 'ansible'...

remote: Enumerating objects: 149, done.

remote: Counting objects: 100% (149/149), done.

remote: Compressing objects: 100% (130/130), done.

remote: Total 362326 (delta 107), reused 21 (delta 19), pack-reused 362177

Receiving objects: 100% (362326/362326), 131.28 MiB | 6.06 MiB/s, done.

Resolving deltas: 100% (229993/229993), done.

Checking out files: 100% (11166/11166), done.

Now load the testing environment with the command:

source ansible/hacking/env-setup

[root@ansiblehost devel]# source ansible/hacking/env-setup

running egg\_info

creating lib/ansible.egg-info

writing requirements to lib/ansible.egg-info/requires.txt

writing lib/ansible.egg-info/PKG-INFO

writing top-level names to lib/ansible.egg-info/top\_level.txt

writing dependency\_links to lib/ansible.egg-info/dependency\_links.txt

writing manifest file 'lib/ansible.egg-info/SOURCES.txt'

reading manifest file 'lib/ansible.egg-info/SOURCES.txt'

reading manifest template 'MANIFEST.in'

no previously-included directories found matching 'hacking'

warning: no files found matching 'SYMLINK\_CACHE.json'

writing manifest file 'lib/ansible.egg-info/SOURCES.txt'

Setting up Ansible to run out of checkout...

PATH=/root/devel/ansible/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/root/bin

PYTHONPATH=/root/devel/ansible/lib

MANPATH=/root/devel/ansible/docs/man:/usr/local/share/man:/usr/share/man

Remember, you may wish to specify your host file with -i

Done!

Normally you can test the modules with the ansible-test command. This only tests modules found in the ansible/lib/ansible/module/\* directory. But because we didn’t put the test module there, we need to find the underlying test commands

**First do a Compile test**

You have to find-out what command is used for the compile test. This can be done using the -e parameter

# ansible-test sanity --test compile --python 2.7 lineinfile -e

Sanity check using compile with Python 2.7

Run command: /usr/bin/python test/sanity/compile/compile.py

From this you can build your own test command.

# /usr/bin/python ansible/test/sanity/compile/compile.py modules\_workshop/library/demo.py

The directory module\_workshop/library is the location of your demo.py module

If nothing is returned, compilation succeeds.

**Pep8 code style check**

First findout what the command is:

# ansible-test sanity --test pep8 -e lineinfile

Sanity check using pep8

Run command: /usr/bin/python -m pycodestyle --max-line-length 160 --config /dev/null --ignore E402,E722,E741,W503,W504 lib/ansible/modules/files/lineinfile.py

Now test your module

# /usr/bin/python -m pycodestyle --max-line-length 160 --config /dev/null --ignore E402,E722,E741,W503,W504 modules\_workshop/library/demo.py

modules\_workshop/library/demo.py:99:17: E126 continuation line over-indented for hanging indent

modules\_workshop/library/demo.py:116:17: E126 continuation line over-indented for hanging indent

As you can see at line 99 and 116 continuation is over-ident

Let’s fix this

1. **else**:
2. module.fail\_json(
3. msg="could not create " + full\_path\_name, rc=rc, err=err)

it seems there are 8 spaces, these should be only 4 so adjust line 99 to

1. **else**:
2. module.fail\_json(
3. msg="could not create " + full\_path\_name, rc=rc, err=err)

This also accounts for line 116

**Validate the module**

If the module runs through the validation, you might have created a valid module, which if this module add’s something to Ansible, might be included in Ansible.

The command to use to find out how to validate is:

# ansible-test sanity --test validate-modules -e lineinfile

Sanity check using validate-modules

WARNING: Cannot perform module comparison against the base branch. Base branch not detected when running locally.

Run command: /usr/bin/python test/sanity/validate-modules/validate-modules --format json --arg-spec lib/ansible/modules/files/lineinfile.py

WARNING: Reviewing previous 1 warning(s):

WARNING: Cannot perform module comparison against the base branch. Base branch not detected when running locally.

Now run the command to your module:

Cd to the ansible directory and run the command

# cd ansible

[ansible]# /usr/bin/python test/sanity/validate-modules/validate-modules --format json --arg-spec ../modules\_workshop/library/demo.py

{

"../modules\_workshop/library": {

"warning\_traces": [],

"errors": [

{

"msg": "Ansible module subdirectories must contain an \_\_init\_\_.py",

"column": 0,

"line": 0,

"code": 502

}

],

"traces": [],

"warnings": []

}

}

As you can see the library directory where your mode resides needs a file called: \_\_init\_\_.py

This file must be empty so create it with touch

And run the test again.

[ansible]# /usr/bin/python test/sanity/validate-modules/validate-modules --format json --arg-spec ../modules\_workshop/library/demo.py

{}

If only the {} returns, all is oke

Test the module local at the Ansible server

This can be done by the test-module tool located in your ansible development environment

[ansible]# hacking/test-module -m ../modules\_workshop/library/demo.py -a 'name=test\_file'

\* including generated source, if any, saving to: /root/.ansible\_module\_generated

\* ansiballz module detected; extracted module source to: /root/debug\_dir

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

RAW OUTPUT

{"msg": "file: /tmp/test\_file created", "invocation": {"module\_args": {"state": "present", "location": "/tmp", "name": "test\_file"}}, "changed": true}

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

PARSED OUTPUT

{

"changed": true,

"invocation": {

"module\_args": {

"location": "/tmp",

"name": "test\_file",

"state": "present"

}

},

"msg": "file: /tmp/test\_file created"

}

If all is oke, you should have a “test\_file” in the /tmp directory at your ansible server

Complete code of the demo.py module

1. #!/usr/bin/python
3. # (c) 2017, Joris Weijters <joris.weijters@gmail.com>
4. # GNU General Public License v3.0+ (see COPYING or https://www.gnu.org/licenses/gpl-3.0.txt)
6. ANSIBLE\_METADATA = {'metadata\_version': '1.1',
7. 'status': ['preview'],
8. 'supported\_by': 'community'}
10. DOCUMENTATION = '''''
11. ---
12. author:
13. - Joris Weijters (@molekuul)
14. module: demo
15. short\_description: demo module
16. description:
17. - demo module for the Ansible meetup benelux
18. version\_added: "2.7"
19. options:
20. name:
21. description:
22. - Name of the file
23. required: yes
24. location:
25. description:
26. - Location of the file
27. default: /tmp
28. state:
29. description:
30. - whether the file should be present or absent
31. choices: [ absent, present ]
32. default: present
33. notes:
34. - puts a file at a specific location
35. - this is just to demo the writing of modules
36. '''
38. EXAMPLES = '''''
39. # Add a file
40. - name: Add a file
41. demo:
42. name: "hello\_world"
43. location: /tmp
45. # remove a file
46. - name: remove file
47. demo:
48. name: "hello\_world"
49. location: /tmp
50. state: absent
51. '''
53. RETURN = '''''
54. msg:
55. description: return message
56. returned: always
57. type: string
58. sample: file /tmp/hello\_world created
59. changed:
60. description: whether the file add or removal has been changed
61. returned: always
62. type: boolean
63. sample: true
64. '''
66. # Import necessary libraries
67. from ansible.module\_utils.basic import AnsibleModule
69. # end import modules
71. # start defining the functions

74. def check\_file(module, full\_path\_name):
75. # check if file exists
76. # uses ls to check if file exists
77. # returns dictionary content is boolean exists:[ True, False ]
78. exists = False
79. ls = module.get\_bin\_path('ls')
80. (rc, out, err) = module.run\_command([ls, full\_path\_name])
81. if rc == 0:
82. exists = True
83. return exists

86. def create\_file(module, full\_path\_name):
87. # create file
88. # uses touch to create a file
89. rc = 0
90. result = {}
91. touch = module.get\_bin\_path("touch")
92. if not module.check\_mode:
93. (rc, out, err) = module.run\_command([touch, full\_path\_name])
94. if rc == 0:
95. result['changed'] = True
96. result['msg'] = "file: " + full\_path\_name + " created"
97. else:
98. module.fail\_json(
99. msg="could not create " + full\_path\_name, rc=rc, err=err)
100. return result

103. def remove\_file(module, full\_path\_name):
104. # remove file
105. # uses rm to remove file
106. rc = 0
107. result = {}
108. rm = module.get\_bin\_path("rm")
109. if not module.check\_mode:
110. (rc, out, err) = module.run\_command([rm, full\_path\_name])
111. if rc == 0:
112. result['changed'] = True
113. result['msg'] = "file: " + full\_path\_name + " removed"
114. else:
115. module.fail\_json(
116. msg="could not remove " + full\_path\_name, rc=rc, err=err)
117. return result

120. def main():
121. module = AnsibleModule(
122. argument\_spec=dict(
123. name=dict(type='str', required=True),
124. location=dict(type='str', default='/tmp'),
125. state=dict(type='str', choices=['absent', 'present'],
126. default='present'),
127. ),
128. supports\_check\_mode=True,
129. )
131. result = {
132. 'msg': "",
133. 'changed': False
134. }
136. # check if file exists
137. full\_path\_name = module.params['location'] + "/" + module.params['name']
138. file\_exists = check\_file(module, full\_path\_name)
140. # if state is present and file does not exist create file
141. # if state is present and file does exist do nothing
142. # if state is absent and file does not exist do nothing
143. # if state is absent and file does exist remove file
145. if module.params['state'] == 'present':
146. if (not file\_exists):
147. result = create\_file(module, full\_path\_name)
148. else:
149. if (file\_exists):
150. result = remove\_file(module, full\_path\_name)
152. module.exit\_json(\*\*result)

155. if \_\_name\_\_ == '\_\_main\_\_':
156. main()