



Registered Variables in Ansible

By: Er. Vikas Nehra (M. Tech, B. Tech), Experience: 15 + Years

Session - 12 Agenda:

1. Registered Variables in Ansible.
-

Registering variables:

You can create variables from the output of an Ansible task with the task keyword register. You can use registered variables in any later tasks in your play(s).

Registered variables are stored in memory. You cannot cache registered variables for use in future playbook runs. Registered variables are only valid on the host for the rest of the current playbook run, including subsequent plays within the same playbook run.

Registered variables are host-level variables. When you register a variable in a task with a loop, the registered variable contains a value for each item in the loop. The data structure placed in the variable during the loop will contain a results attribute, that is a list of all responses from the module.

Ansible by default doesn't keep or carry the output of any command (task) executed on the managed nodes. It leaves the output value while executing the command on the managed nodes. But we manually make ansible to keep the output of the command(s) executed on the managed nodes.

Let's create a playbook file to register the variable to check the hostname of all the node machines.

```
$ vim register_variable.yml
---
- name: Registering the variable in Ansible.
  hosts: all
  tasks:
    - name: Check the machine hostname.
      command: hostname
...

```

Check the playbook for syntax errors.

```
$ ansible-playbook register_variable.yml --syntax-check
```

Now, execute the playbook.

```
$ ansible-playbook register_variable.yml
```

Here you can see that playbook has been successfully executed, ansible has



Registered Variables in Ansible

By: Er. Vikas Nehra (M. Tech, B. Tech), Experience: 15 + Years

checked the hostnames by executing the hostname command on the managed nodes. But the output is not saved anywhere so that we can use it further. In case if we want to use this output information again, we need to save this output in a variable and here we have to register it using register keyword in the task of the playbook.

Let's register the variable (output of the command) now.

```
$ vim register_variable.yml
```

```
- name: Registering the variable in Ansible.
```

```
hosts: all
```

```
tasks:
```

```
  - name: Check the machine hostname.
```

```
    command: hostname
```

```
    register: abc
```

```
  - debug:
```

```
    var: abc
```

...



Now, execute the playbook.

```
$ ansible-playbook register_variable.yml
```

Now, you can see the output on the screen because you have used the debug option in the playbook. The stdout value shows the output of hostname command which has been stored in the abc variable now.

In case if we need to see the stdout value only in the output, we can mention it in the debug field of the playbook as shown below.

```
$ vim register_variable.yml
```

```
- name: Registering the variable in Ansible.
```

```
hosts: all
```

```
tasks:
```

```
  - name: Check the machine hostname.
```

```
    command: hostname
```

```
    register: abc
```

```
  - debug:
```

```
    var: abc.stdout
```

...

Now, execute the playbook.



Registered Variables in Ansible

By: Er. Vikas Nehra (M. Tech, B. Tech), Experience: 15 + Years

```
$ ansible-playbook register_variable.yml
```

Now, we will only see the stdout value of abc variable in the output.

Let's create another playbook file to check the hardware mode (architecture information) using registered variables.

```
$ vim register_variables.yml
```

```
---
```

```
- name: Registering the variable in Ansible.
```

```
hosts: all
```

```
tasks:
```

```
  - name: Check the hardware mode.
```

```
    command: uname -m
```

```
...
```

Check the playbook for syntax errors.

```
$ ansible-playbook register_variable.yml --syntax-check
```

Now, execute the playbook.

```
$ ansible-playbook register_variables.yml
```

Here you can see that playbook has been successfully executed, ansible has checked the hardware mode by executing the uname -m command on the managed nodes. But the output is not saved anywhere so that we can use it further. In case if we want to use this output information again, we need to save this output in a variable and here we have to register it using register keyword in the task of the playbook.

Let's register the variable (output of the command) now.

```
$ vim register_variables.yml
```

```
---
```

```
- name: Registering the variable in Ansible.
```

```
hosts: all
```

```
tasks:
```

```
  - name: Check the hardware mode.
```

```
    command: uname -m
```

```
    register: abc
```

```
  - debug:
```

```
    var: abc
```

```
...
```

Now, execute the playbook.



Registered Variables in Ansible

By: Er. Vikas Nehra (M. Tech, B. Tech), Experience: 15 + Years

\$ ansible-playbook register_variables.yml

Now, you can see the output on the screen because you use debug option in the playbook. The stdout value shows the output of uname -m which has been stored in the abc variable.

In case if we need to see the stdout value only in the output, we can mention it in the debug field of the playbook as shown below.

\$ vim register_variables.yml

```
---
- name: Registering the variable in Ansible.
  hosts: all
  tasks:
    - name: Check the hardware mode.
      command: uname -m
      register: abc

    - debug:
        var: abc.stdout
...

```

Now, execute the playbook.

\$ ansible-playbook register_variables.yml

Let's save the value in a file on all the managed nodes.

\$ vim register_variable2.yml

```
---
- name: Registering the variable in Ansible.
  hosts: all
  become: true
  tasks:
    - name: Check the hardware mode.
      command: uname -m
      register: abc
    - debug:
        var: abc.stdout
    - name: Save the information over all the nodes.
      copy:
        content: "This is the hardware mode {{ abc.stdout }}"
        dest: /tmp/hwmode.txt
...

```

Now, execute the playbook.



Registered Variables in Ansible

By: Er. Vikas Nehra (M. Tech, B. Tech), Experience: 15 + Years

\$ ansible-playbook register_variable2.yml

Verify the result using ansible ad-hoc command.

\$ ansible all -m command -a 'cat /tmp/hemode.txt'

Let's take one more example, let's check the version of Apache using registered variable.

\$ vim register_variable3.yml

```
- name: Registering the variable in Ansible.
  hosts: node1
  become: true
  tasks:
    - name: Installing Apache HTTPD Package.
      yum:
        name: httpd
        state: latest
    - name: Checking Apache Version & Storing In ver Variable.
      command: httpd -v
      register: ver
    - debug:
        var: ver
  ...
```

Now, execute the playbook.

\$ ansible-playbook register_variable3.yml

We can also mention ver.stdout instead of ver in the debug section of the playbook to print only the Apache version information.

Thanks